Attributi per vertice con discontinuità

- Gli attributi definiti per vertice (interpolati dentro le facce) sono per costruzione continui (C0) fra faccia e la faccia adiacente
 - ⇒ i valori dell'attributo sull'edge infatti è interpolato solo fra gli attributi definiti nei vertici agli estremi di quell'edge, e sono quindi condivisi dalle due facce adiacienti
- √ Volendo, posso introdurre delle discontinuità (C0) di attributo introducendo duplicazioni di vertici
 - ⇒ La mesh avrà due vertici geometricamente coincidenti (stessa posizione) con attributi associati differenti
 - ⇒ Facce adiacienti useranno una oppure l'altra di questa coppia di vertice
 - ⇒ Questa configurazione si chiama un «seam» (letteralmente: cucitura) o «vertex seam»



50

Come rappresento internamente una mesh?

Modo diretto: (anche detto: zuppa di triangoli)

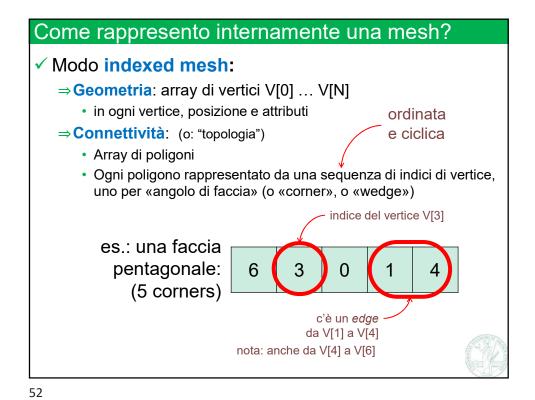
- ✓ un lungo vettore di poligoni
 - \Rightarrow per ogni poligono (di n lati):

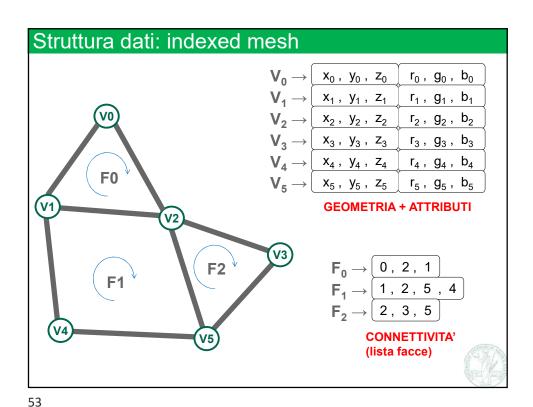
è un vettore di *n* vertici:

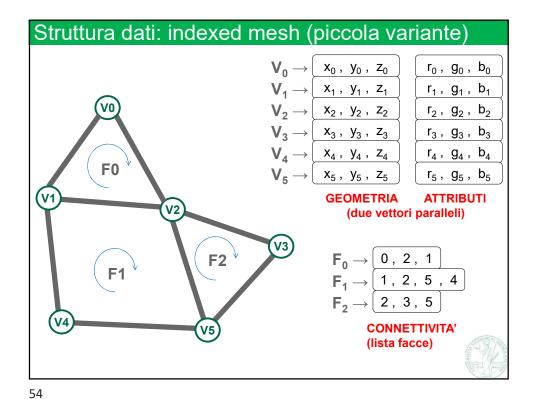
- posizione (x,y,z)
- attributi (colore r,g,b, e/o altro)
- ✓ Grosso difetto: replicazione dei vertici!
 - ⇒poco efficiente in spazio
 - ⇒complicato fare updates
 - (bisogna mantenere le copie uguali durante il processing)
- ✓ Metodo non molto usato

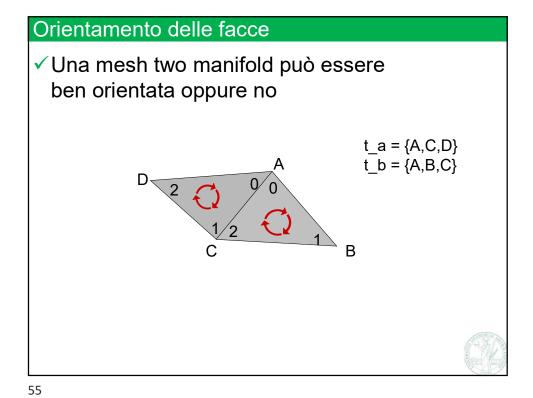


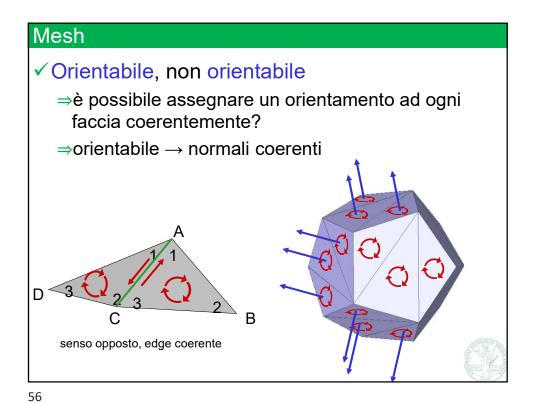
51

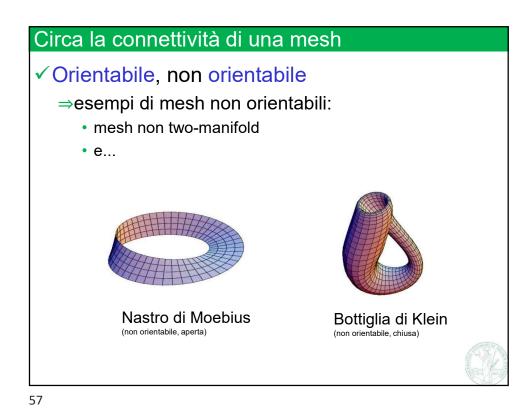


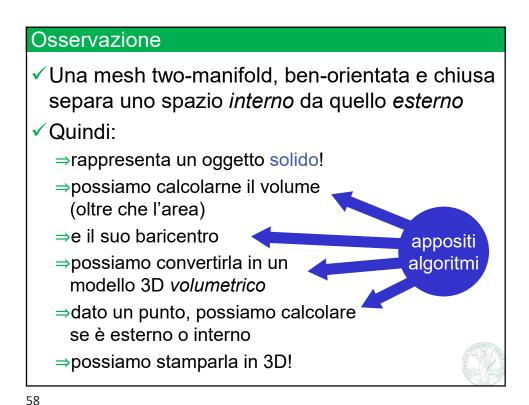


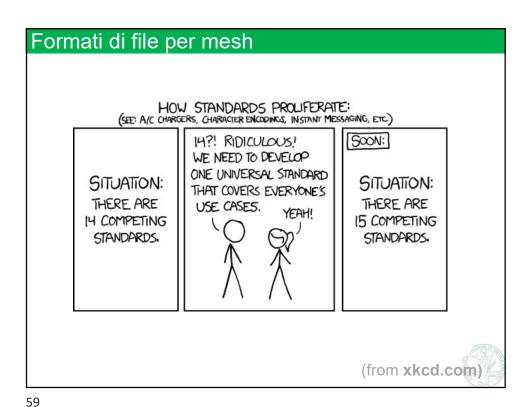














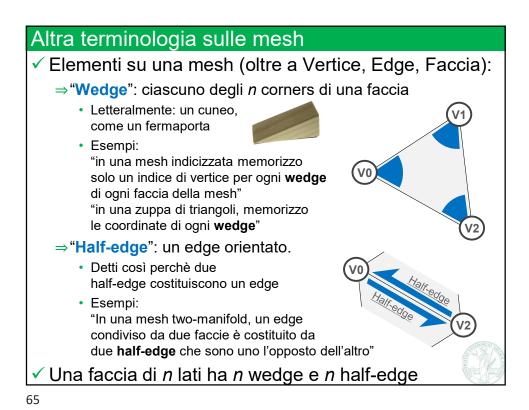
File formats per indexed mesh: Per esempio: OFF ("Object File Format") # facce # edges LetteraL.off 1 5 1 OFF 12 10 40 # vertici -0 5 1 prima faccia: 0 0 4 3 2 1 0 - indice 0 4 vertici: 3 0 0 5 4 indice 1 3 0 X, Y, Z con indici 3 1 0 4 6 7 8 9 2ndo — indice 2 3, 2, 1 e 0vert 1 1 0 4 6 9 10 11 indice 3 4 0 1 7 6 0 5 0 4 1 2 8 7 0 0 1 4 2 3 9 8 3 4 10 9 3 0 1 3 1 1 4 4 5 11 10 1 1 1 4 5 0 6 11

OFF 12 10 0 0.0 0.0 0.0 3.0 0.0 0.0 3.0 1.0 0.0 1.0 1.0 0.0 1.0 5.0 0.0 0.0 5.0 0.0 0.0 0.0 1.0 3.0 0.0 1.0 3.0 1.0 1.0 1.0 1.0 1.0 1.0 5.0 1.0 0.0 5.0 1.0	geometria header	File formats per indexed mesh Mesh in formato OFF ("Object File Format") Es 0: costruisci un file di testo con ←questo contenuto (cut and paste), rinominalo «test.off», e visualizzalo con meshlab
4 3 2 1 0 4 5 4 3 0 4 6 7 8 9 4 6 9 10 11 4 0 1 7 6 4 1 2 8 7 4 2 3 9 8 4 3 4 10 9 4 4 5 11 10 4 5 0 6 11	connettività	Es 1: puoi dire, guardando il testo del file ←qui accanto, se si tratti una mesh: • quad-dominant, pure-quad, o tri? • two-manifold o no? • chiusa o aperta? • ben orientata o no? • low-poly o hi-res?

Note sull'Esercizio 1

- ✓ Ho un («half») edge per ogni coppia di indici consecutivi in ogni faccia (l'ordine CONTA)
 - ⇒ compreso: da ultimo vertice a primo vertice
- ✓ Edge two manifold se: compare in max due facce,
- Due facce che condivisono un edge sono orientate consistentemente se:
 - l'edge appare flipped nei due casi (es: 1-5 vs 5-1)
- ✓ Edge di bordo se: appare in esattamente una faccia
- ✓ Mesh two-manifold richiede: tutti gli edge sono two-manifold
- ✓ Mesh chiusa se: non ci sono edge di bordo
- ✓ Mesh ben orientata se: tutti gli edge interni (= non di bordo) appaiono flipped nelle due facce in cui appaiono
- ✓ NOTA: dipende solo dalla connettività. Non dalla geometria.

64



```
Indexed mesh: as a class (here: in C++)
```

```
class Vertex {
  vec3 pos;
  rgb color;  /* attribute 1 */
  vec3 normal; /* attribute 2 */
};

class Face{
  vector<int> vertexIndex;
};

class Mesh{
  vector<Vertex> vert; /* geom + attr */
  vector<Face> face; /* connettivita' */
};
```

Indexed mesh: as a class (here: in C++)

```
class Vertex {
  vec3 pos;
  rgb color;  /* attribute 1 */
};

class Face{
  int vertexIndex[4];
  vec3 normal; /* attribute 2 */
};

class Mesh{
  vector<Vertex> vert; /* geom + attr */
  vector<Face> face; /* connettivita' */
};
```

Esempio di variante: normali memorizzate per faccia, mesh pure quad (4 indici di vertice per faccia)

67

Regolarità di una mesh (definizioni)

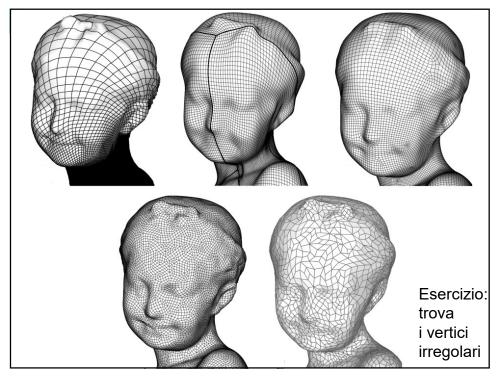
- ✓ «Valenza» di un vertice
 - ⇒numero di facce (o di edge) adiacenti ad quel vertice
- ✓ «Vertice regolare» (per vertici interni): un vertice di valenza:
 - ⇒4, per quad mesh
 - ⇒6, per tri mesh
 - ⇒(3 per hexagonal mesh)
- ✓ Le tri-mesh e le quad-mesh (pure o dominant) hanno un grado maggiore o minore di «regolarità»
 - ⇒ Quanti dei loro vertici sono «regolari»?
 Tutti = mesh (perfettamente) regolare. O «structured»
 Quasi tutti = mesh semi-regolare.

Pochi (per es 2/3 o la metà) = mesh irregolare

✓ Quad mesh regolare = è un grigliato

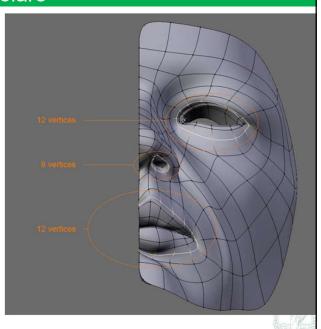


70

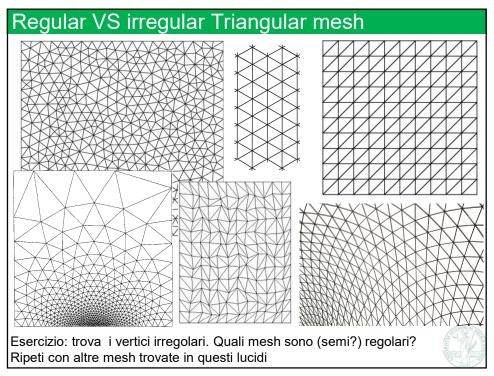


Mesh (Semi) regolare

- ✓ E' fornita di "edge-loops" e "edge-flows"
- ✓ rendono la mesh molto più semplice da editare
- ✓ (per un modellatore umano)



73



Alcuni vantaggi delle mesh (semi) regolari

- ✓ Mesh quad-dominant semi-regolari:
 - ⇒flessibilità

(rispetto alle mesh pure-quad: facile ottenere la forma voluta)

⇒editabilità

(rispetto alle mesh irregolari: presenza di «edge loops» o «edge flows»)

- ⇒grigliati regolari: sono intuitivi e utili
- ⇒sono comprimibili (con algoritmi di compressione)
- ⇒minore errore geometrico a parità di risoluzione (di solito)
- ⇒soprattutto per le quad mesh: gli edge sono allinea

75

Mesh regolari o irregolari - note

- ✓ Mesh irregolari : maggiore adattività della risoluzione
 - ⇒ Confronta l'esempio di mesh con risoluzione adattiva
- ✓ Alcuni metodi per gererare mesh producono tipicamente mesh di triangoli irregolari, per es
 - ⇒ Front advancing methods
 - ⇒ La maggior parte dei metodi di acuisizione 3D (scanning)
 - ⇒ Mesh sculpting
 - ⇒ Direct "low poly editing"
- ✓ Altri metodi per generare mesh producono tipicamente mesh regolari, spesso di quad
 - ⇒ Mesh che sono superfici di risoluzione o "spazzate"
 - ⇒ CAD modelling
- ✓ Nota: diagonal split su una mesh di quad (semi-)regolare ottiene una tri-mesh (semi-)regolare (provare)
 - ⇒ A condizione che... (provare)

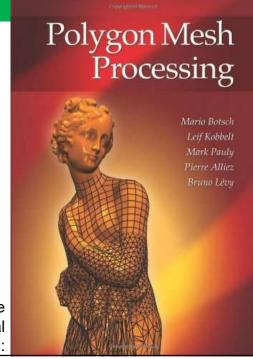


76

Next lecture:

Mesh processing (geometry processing eseguito su mesh poligonali)

Un buon manuale per le basi al mesh processing:



77