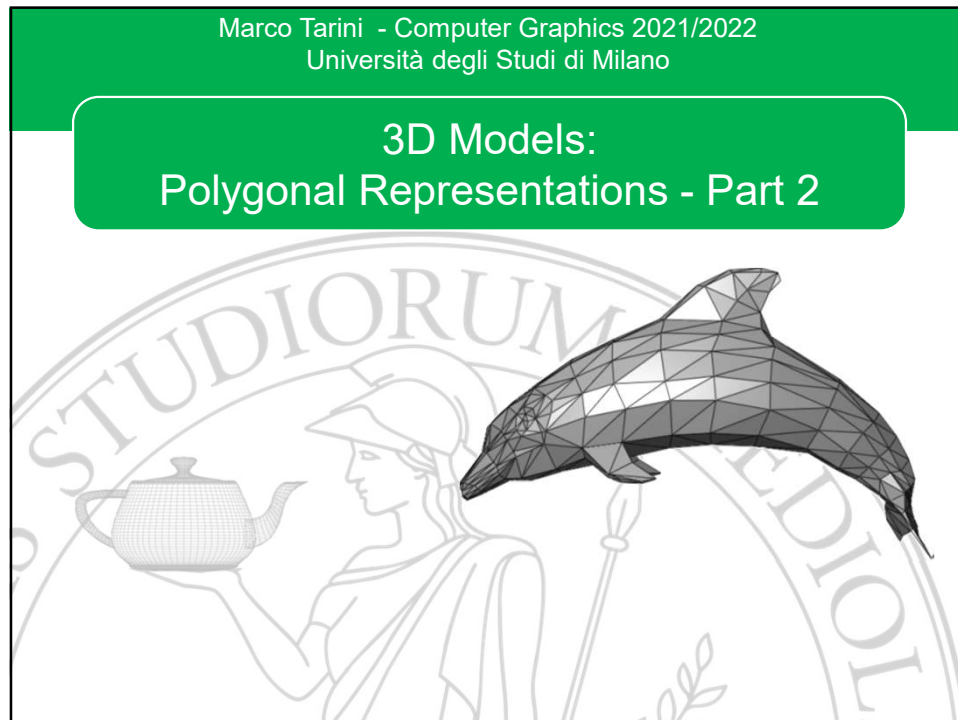
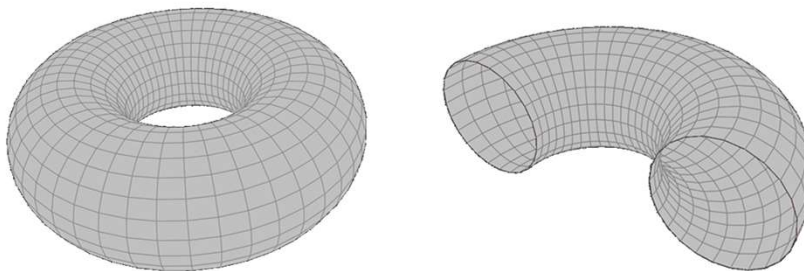




58

Mesh chiuse e aperte

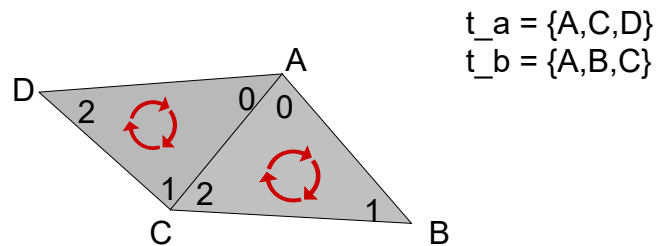
- ✓ un edge condiviso da solo 1 faccia è di bordo;
- ✓ un edge condiviso da 2 faccie è interno;
- ✓ se una mesh non ha edge di bordo, è chiusa (altrimenti, è aperta)



59

Orientamento delle facce

- ✓ Una mesh two manifold può essere ben orientata oppure no



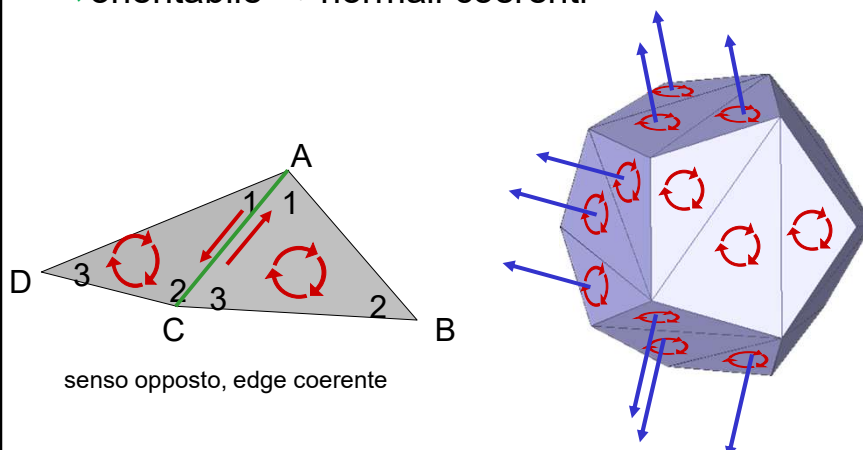
61

Mesh

- ✓ Orientabile, non orientabile

⇒ è possibile assegnare un orientamento ad ogni faccia coerentemente?

⇒ orientabile → normali coerenti



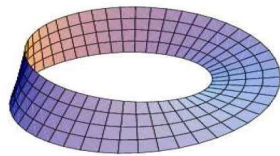
62

Circa la connettività di una mesh

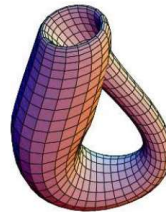
✓ **Orientabile**, non **orientabile**

⇒ esempi di mesh non orientabili:

- mesh non two-manifold
- e...



Nastro di Moebius
(non orientabile, aperta)



Bottiglia di Klein
(non orientabile, chiusa)



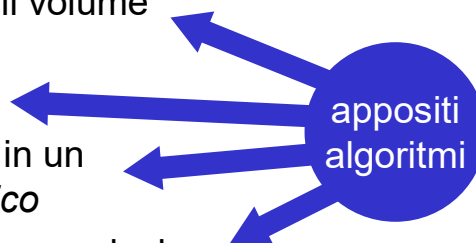
63

Osservazione

✓ Una mesh two-manifold, ben-orientata e chiusa separa uno spazio *interno* da quello *esterno*

✓ Quindi:

- ⇒ rappresenta un oggetto **solido**!
- ⇒ possiamo calcolarne il volume (oltre che l'area)
- ⇒ e il suo baricentro
- ⇒ possiamo convertirla in un modello 3D *volumetrico*
- ⇒ dato un punto, possiamo calcolare se è esterno o interno
- ⇒ possiamo stamparla in 3D!



64

File formats per indexed mesh	
<pre> 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 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 </pre>	header geometria connettività 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 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? 

65

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 condividono 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.
- 

67

Mesh: attributi

- ✓ Modellano quantità che variano sulla superficie
- ✓ Esempi:
 - ⇒ Colore («diffusivo»)
 - tipicamente espresso come RGB
 - Parte della descrizione del materiale
 - ⇒ Vettore «Normale»
 - quale orientamento ha la superficie in quel punto?
 - ⇒ Le cose più varie, dipendenti dall'applicazione:
 - per es: temperatura e pressione alla superficie (in una simulazione fisica)
 - per es: «coefficiente di vulnerabilità» (in un gioco)
 - per es: qualità della ricostruzione, in un modello di un oggetto reale (quanto è accurata la superficie qui?)



68

Mesh: attributi

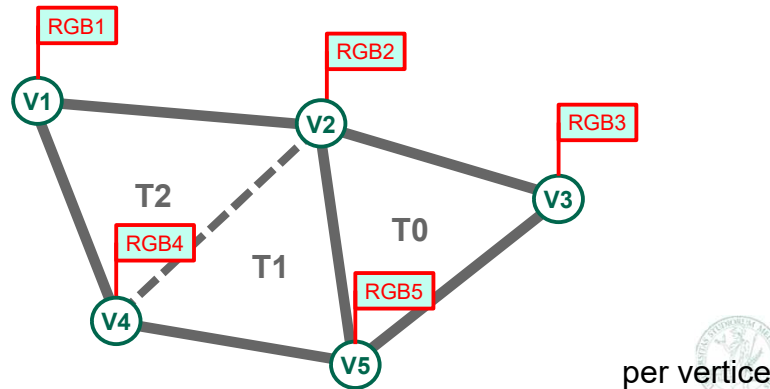
- ✓ Possono essere di qualsiasi tipo:
 - ⇒ scalari (es: temperatura, qualità...),
 - ⇒ vettori (es: colore, normale...)
- ✓ E possono essere memorizzati
 - ⇒ per ogni vertice
 - ed estesi dentro le faccie
 - è questo caso più comune
 - ⇒ oppure per ogni faccia
 - vengono considerati costanti su quella faccia
 - quindi discontinuità C0 fra le facce in corrispondenza, cioè, degli edge



69

Mesh: attributi per vertice (mesh simpliciale)

- ✓ L'interpolazione degli attributi definiti sui vertici è definita dentro a facce *triangolari*
 - ⇒ Vedremo come, nella track matematica del corso
- ✓ Quindi, per prima cosa, gli altri poligoni devono essere suddivisi in triangoli



70

Interpolazione attributi dentro un triangolo

- ✓ Ad ogni punto q in un triangolo T viene implicitamente attribuita un'interpolazione di dei tre attributi definiti sui vertici di T , usando, come peso dell'interpolazione, le coordinate baricentriche di q in T
 - ⇒ Gli attributi, salvati sui vertici, sono così estesi implicitamente su tutta la superficie
- ✓ GPU support:
 - ⇒ La GPU è specializzata nei casi quali il computo delle coordinate baricentriche di punti dentro ai triangoli, e la conseguente l'interpolazione degli attributi definiti sui vertici
 - ⇒ Durante il rendering, questo procedimento viene fatto per ogni pixel che compone l'immagine un triangolo
 - ⇒ (nota: i pixel corrispondono a punti dentro le facce di una mesh, non a vertici di una mesh)

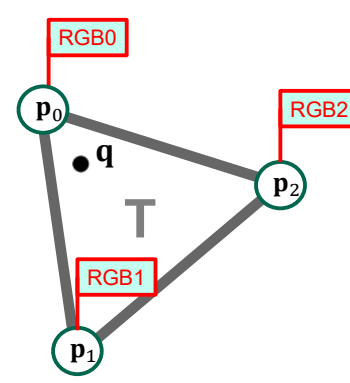
71

Interpolazione attributi dentro un triangolo

- ✓ Un triangolo T ha tre attributi colore assegnati ai suoi vertici p_0 , p_1 , p_2
- ✓ Prendiamo un punto q su T di coordinate baricentriche k_0, k_1, k_2 in T , cioè con

$$q = k_0 p_0 + k_1 p_1 + k_2 p_2$$
- ✓ Allora a q assegno il colore

$$k_0 \text{RGB0} + k_1 \text{RGB1} + k_2 \text{RGB2}$$

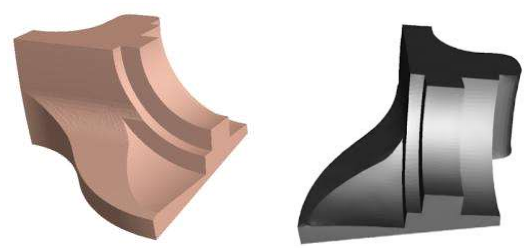


per vertice 

72

Normali di una superficie: osservazioni

- ✓ Normali costanti => superficie piatta
- ✓ Normali che variano con continuità => superficie curva
- ✓ Normali che variano con discontinuità => superficie con un *crease*
 - => il *crease* è costituito dai punti dove la normale è discontinua





73

Attributi per faccia: normali

Nel caso in cui l'attributo è la normale

✓ per faccia:

- ⇒ normale costante per faccia
- ⇒ facce (dall'aspetto) piatto
- ⇒ discontinuità C0 sugli edge:
edge sono «spigoli taglienti»,
detti edge di crease, o «hard» edges

✓ per vertice

- ⇒ normale che varia sulla faccia
- ⇒ facce (dall'aspetto) curvo
- ⇒ continuità C0: → aspetto «smooth» (liscio),
⇒ (ma non continuità C1)



74

Normali come attributi per vertice di una mesh

✓ Normali definite per faccia:

- ⇒ Le normali sono costanti sulle facce:
- ⇒ Le facce appaiono piatte
- ⇒ coerentemente col modello digitale, ma a differenza (spesso)
dell'oggetto 3D che si intendeva rappresentare!
- ⇒ Appaiono crease su ogni edge della mesh

✓ Normali definite per vertice:

- ⇒ Le normali variano sulle facce
(vengono interpolate dentro le facce, come qualsiasi altro attributo)
- ⇒ Le facce appaiono in generale curve
- ⇒ Come ottengo una faccia che appaia piatta?
Uso una stessa normale come attributo dei tre vertici di un triangolo
- ⇒ Come ottengo un crease (una discontinuità di normale)? (vedi next)

✓ Nota: le normali risultano visibili all'osservatore attraverso
l'illuminazione del modello digitale

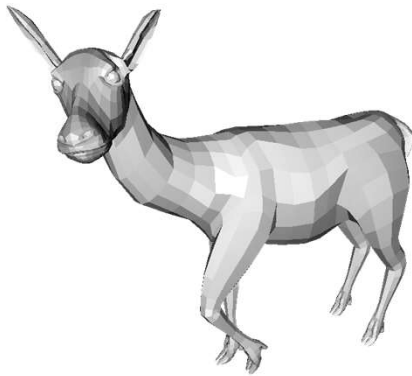
- ⇒ Il calcolo dell'illuminazione è un task del rendering (2nda metà del corso)



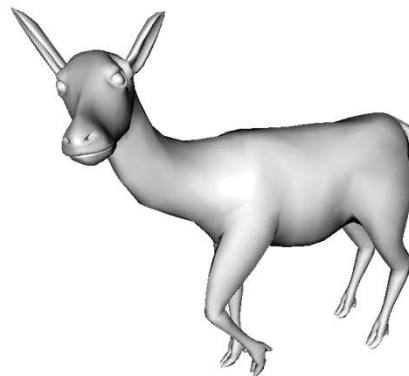
75

Attributi per faccia: normali

Normali per faccia



Normali per vertice



Nota: le normali sono rivelate all'occhio dall'*illuminazione*
(computata durante del rendering)



76

Attributi per faccia: normali

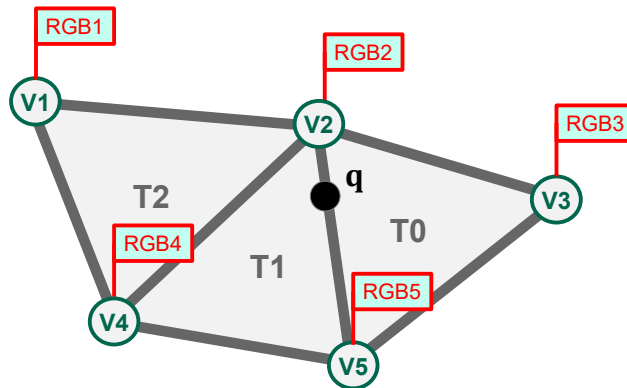
- ✓ Come abbiamo visto, interpolando vettori unitari si ottengono vettori non unitari
 - ⇒ Che vanno quindi ri-normalizzati
- ✓ Questo significa che gli attributi di tipo *vettore unitario*, come le normali, devono essere rinormalizzati dopo ogni l'interpolazione
- ✓ Esercizio: determinare con una stima se questo comporta un onere di calcolo eccessivo, per una GPU, durante un rendering in real-time
- ✓ Traccia
 - ⇒ stimare quante operazioni in virgola mobile (Floating Point Operation) sono necessarie per normalizzare un vettore (soluzione: circa 10)
 - ⇒ stimare quante volte sarà necessario ri-normalizzare un attributo «normale» in un rendering (stimiamo, a braccio: 1 volta in ogni pixel, quindi circa $1000 \times 1000 = 10^6$)
 - ⇒ frame per secondo, in un rendering in tempo reale: stimiamo 60
 - ⇒ totale **F**loating Point **O**peration al **S**econdo (FLOPS) necessari a questo passaggio: $10 \times 10^6 \times 60 = 6 \times 10^8 = 0.6$ Giga-FLOPS
 - ⇒ Una GPU in commercio è capace di erogare... Tera-FLOPS = migliaia di Giga-FLOPS



77

Attributi per vertice sono continui

- ✓ Gli attributi definiti per vertice (interpolati dentro le facce) sono per costruzione continui (C^∞) dentro alle facce
- ✓ Anche fra coppie di facce adiacenti, abbiamo una continuità di attributo (C^0)

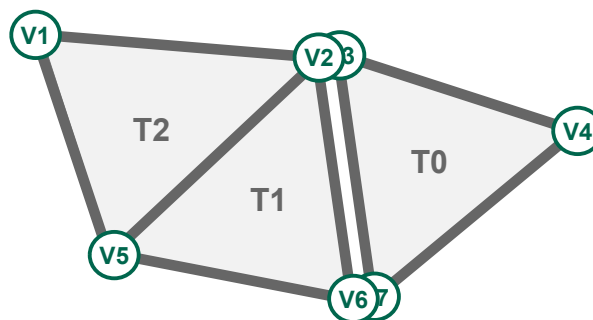


- ✓ Infatti: si prenda un punto q a cavallo fra T_0 e T_1 .
- ✓ Quale attributo è associato a q , se lo si considera parte di T_0 ?
- ✓ Quale attributo è associato a q , se lo si considera parte di T_1 ? (lo stesso!)

78

Attributi per vertice con discontinuità

- ✓ Se è necessario, è possibile modellare delle discontinuità (C^0) di attributo lungo un edge, ma solo duplicando i vertici ai suoi estremi
 - ⇒ La mesh avrà due vertici geometricamente coincidenti (stessa posizione) ma con attributi associati differenti
 - ⇒ Due edge tecnicamente aperti saranno quindi perfettamente sovrapposti, causando una superficie continua (senza «spiragli»)
 - ⇒ Facce adiacenti useranno una oppure l'altra di questa coppia di vertice



- ⇒ Questa configurazione si chiama un «seam» (letteralmente: cucitura) o «vertex seam»

79