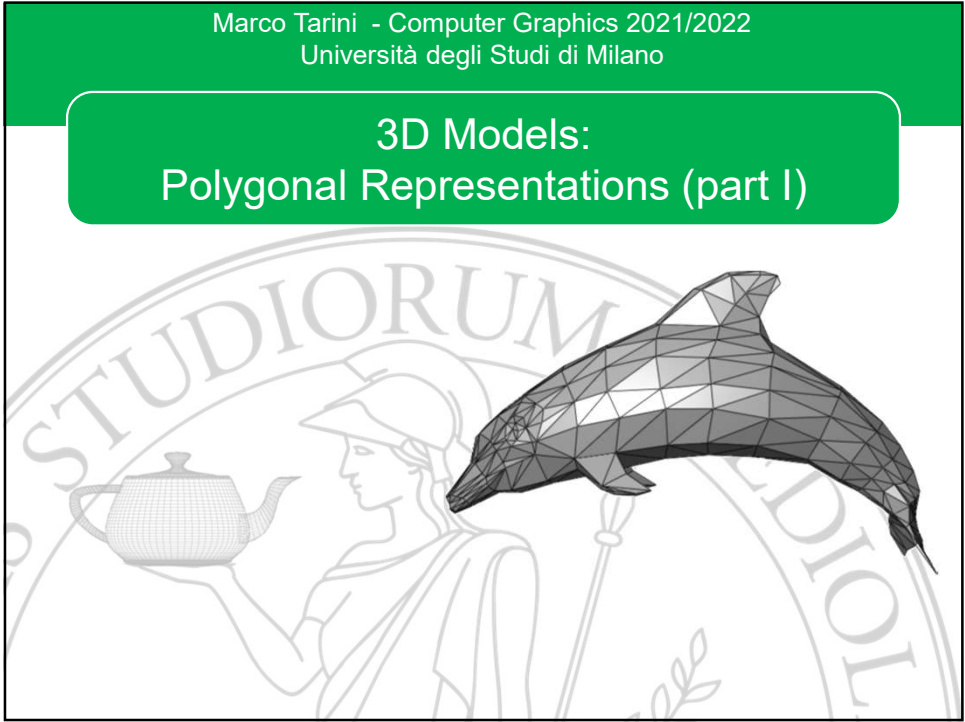




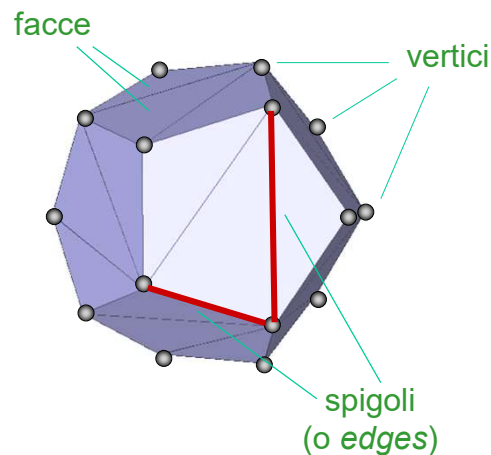
1

| Una (imperfetta) categorizzazione dei tipi di modelli digitali 3D |                                                           |                                                         |                            |                                                           |                                                                      |
|-------------------------------------------------------------------|-----------------------------------------------------------|---------------------------------------------------------|----------------------------|-----------------------------------------------------------|----------------------------------------------------------------------|
|                                                                   |                                                           | ELEMENTI DISCRETI                                       |                            |                                                           | CONTINUI                                                             |
|                                                                   |                                                           | regolari<br>«a griglia»                                 | semi-regolari o irregolari |                                                           |                                                                      |
|                                                                   |                                                           |                                                         | elementi<br>simpliciali    | elementi non<br>simpliciali                               |                                                                      |
| SUPERFICIALI                                                      | 2-manifold<br><br>«rappresenta<br>una vera<br>superficie» | Height Field<br><br>Range Scan<br><br>(Geometry Images) | Triangle Mesh              | Polygonal Mesh<br><br>Quad-Mesh<br><br>Quad dominant Mesh | Subdivision surface<br><br>Parametric Surface<br><br>(es. B-splines) |
|                                                                   | non-manifold<br><br>«non<br>rappresenta<br>una sup»       | Set di Range Scan                                       | Point Cloud                |                                                           |                                                                      |
|                                                                   | VOLUMETRICI                                               | (3-manifold)                                            | Voxels<br>Solid Textures   | Tetra Mesh                                                | Hexa Mesh                                                            |

3

### Mesh triangolare (o mesh simpliciale)

- ✓ Un insieme di poligoni adiacenti



4

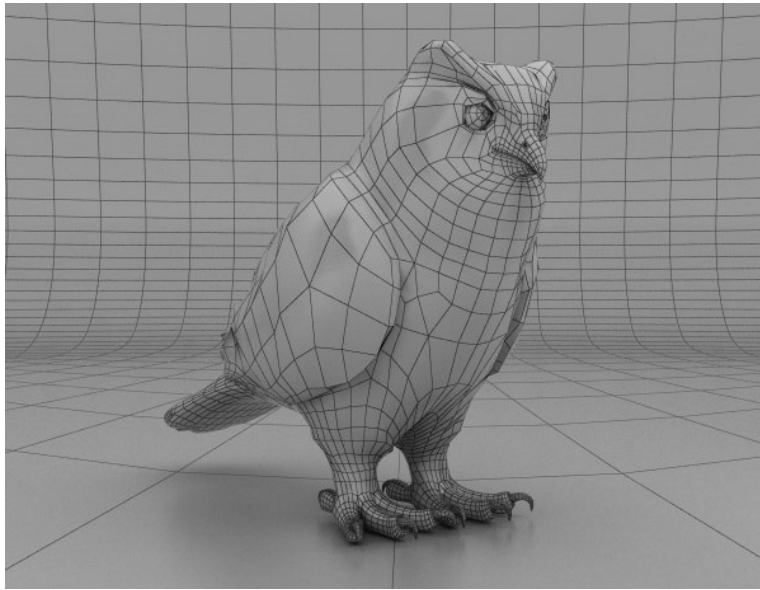
### Polygonal Mesh (mesh poligonale)

- ✓ Superficie approssimata da “facce” poligonali
  - ⇒ adiacenti («incollate») lato a lato
- ✓ Rappresentazione “lineare a tratti” delle superfici
- ✓ Il modello 3D digitale più diffuso!
  - ⇒ spesso (per es, nei games) è un *sinonimo* di modello 3D
  - ⇒ E' GPU friendly: le schede video sono pensate per renderizzare le mesh (attraverso uno specifico algoritmo, che vedremo nella 2° metà del corso: rasterization-based)



5

## Mesh poligonale



7

## Mesh di poligonale: struttura dati

### ✓ Componenti:

#### ⇒ geometria

- i vertici, ciascuno con pos (x,y,z)
- un campionamento della superficie!

#### ⇒ connettività (detta anche: "topologia")

- come sono connessi i vertici
- ogni poligono connette alcuni vertici

#### ⇒ attributi

- Definiti sulla superficie
- es: colore, normali, UV, (indice di) materiali, ...



9

## Mesh: geometria

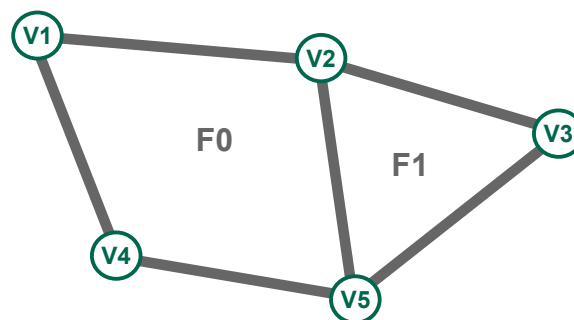
- ✓ Insieme di posizioni dei vertici
  - ⇒ Un vettore posizione  $(x,y,z)$  per ogni vertice



10

## Mesh: connettività (o topologia)

- ✓ Facce
  - ⇒ poligoni che connettono fra loro i vertici
  - ⇒ simile a: nodi connessi da archi, in un grafo



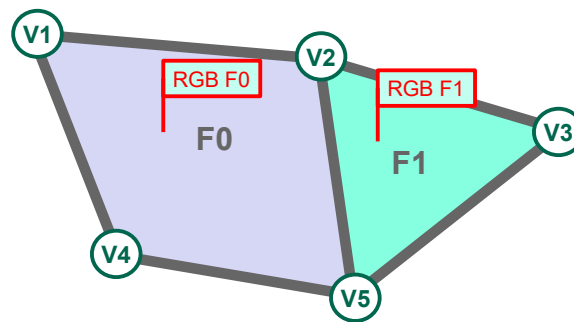
11

### Mesh: attributi

✓ Quantità che variano sulla superficie

⇒ es: colore RGB

⇒ definiti per faccia, costanti su ogni faccia?

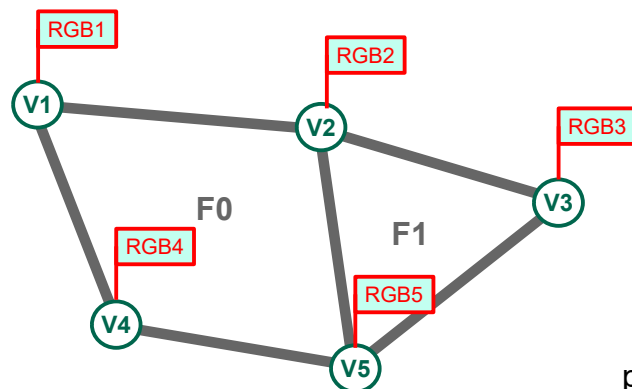


12

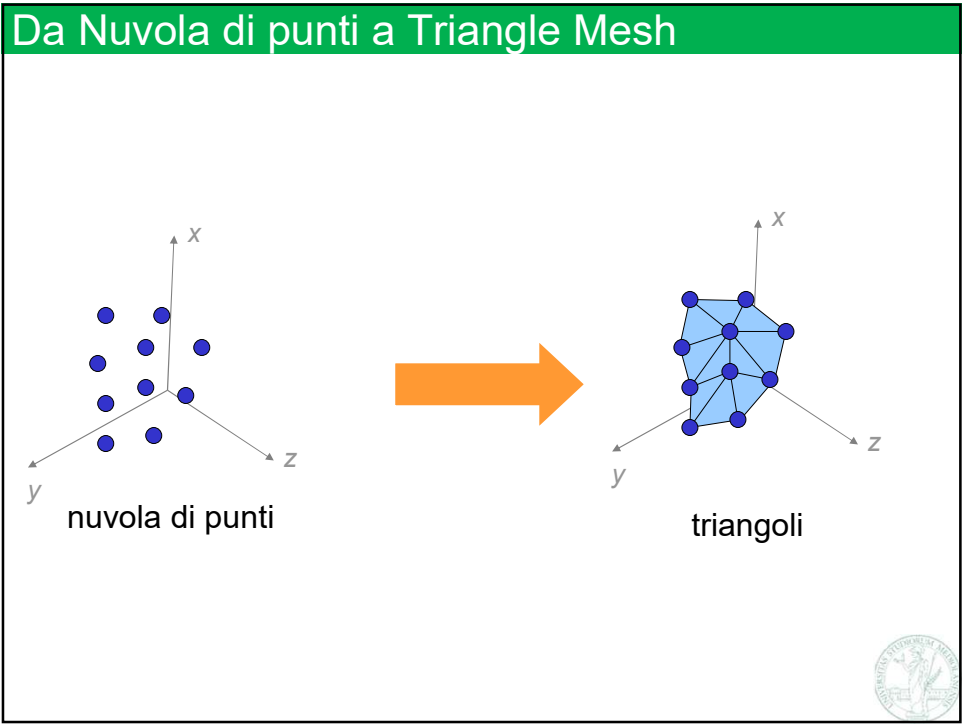
### Mesh: attributi

✓ Quantità che variano sulla superficie

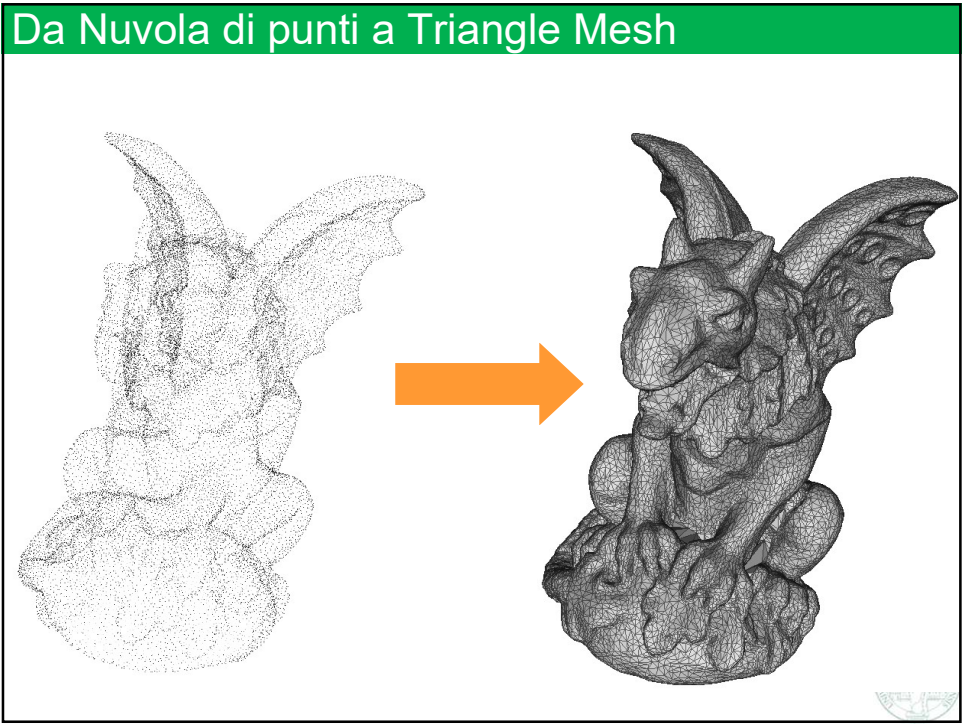
⇒ Campionati per vertice, interpolati nelle facce



13



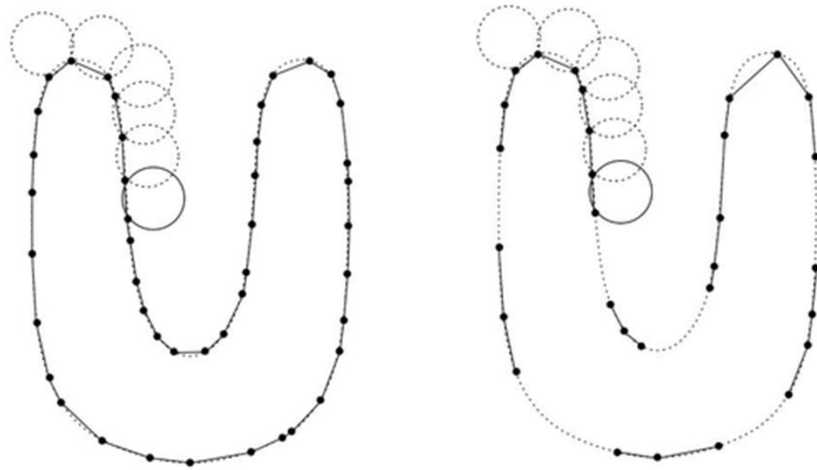
14



15

## Da Nuvola di punti a Triangle Mesh

- ✓ Una classe di algoritmi: Front Advancing:
  - ⇒ Come per es «ball pivoting»

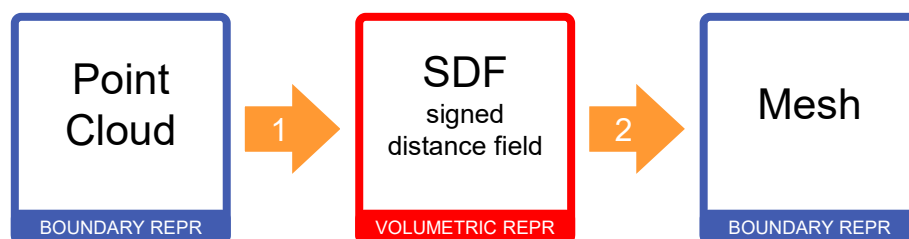


[Bernardini, Mittleman, Rushmeier, Silva TVCG 99]

16

## Da Nuvola di punti a Triangle Mesh

- ✓ Modo più diffuso:  
passa attraverso una  
*rappresentazione intermedia volumetrica*
  - ⇒ Vedi lezione su dati volumetrici



17

## Da Nuvola di punti a Triangle Mesh... in 2D

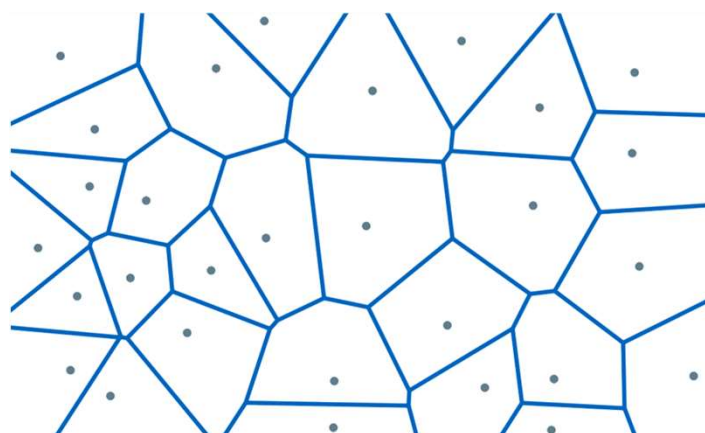
- ✓ In 2D: il problema ha un'ottima soluzione:  
la triangolazione di Delaunay
  - ⇒ Ben nota (in geom. computazionale) dagli anni '70



18

## Da Nuvola di punti a Triangle Mesh... in 2D

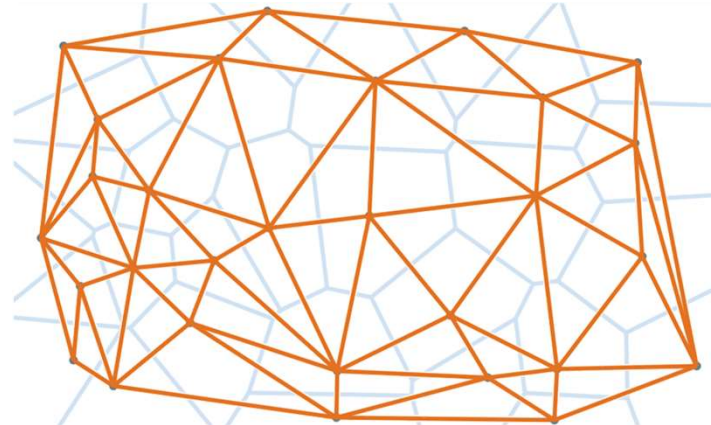
- ✓ In 2D: il problema ha un'ottima soluzione:  
la triangolazione di Delaunay
  - ⇒ Ben nota (in geom. computazionale) dagli anni '70



19

## Da Nuvola di punti a Triangle Mesh... in 2D

- ✓ In 2D: il problema ha un'ottima soluzione:  
la triangolazione di Delaunay
  - ⇒ Ben nota (in geom. computazionale) dagli anni '70



Triangolazione di Delaunay



20

## Triangolazione di Delaunay +

- ✓ Diagramma di Voronoi
  - ⇒ Una scomposizione in regioni di piano indotta da  $n$  punti 2D (detti "seed")  $s_0, s_1, s_2, \dots$
  - ⇒ Ogni seed produce una regione
  - ⇒ Una regione di un seed  $s_i$ : insieme di punti  $p$  del piano che sono più vicini a  $s_i$  che ad ogni altro seed
- ✓ Triangolazione di Delaunay
  - ⇒ Il "duale" di un diagramma di Voronoi cioè:
  - ⇒ La connettività ottenuta connettendo con un edge ogni coppia di seed di regioni confinanti
  - ⇒ Si ottiene una triangolazione con ottime proprietà, come ad esempio: il circocentro che inscrive ogni triangolo non include nessun altro seed



21

### Da Nuvola di punti a Triangle Mesh: sommario

- ✓ Alcuni metodi aggiungono solo la connettività
  - ⇒ la nuvola di punti costituisce già la geometria della mesh
  - ⇒ [Front Advancing](#), come [Ball Pivoting](#)
  - ⇒ In 2D: [Delaunay triangulation](#) (da [Voronoi diagram](#))
- ✓ Altri metodi ricampionano anche i vertici
  - ⇒ Uso di rappresentazioni volumetriche intermedie
  - ⇒ vedi lez. dati volumetrici: "[Poisson Reconstruction](#)"
  - ⇒ Utile, quando la una nuvola di punti è affetta da errori e outlier
  - ⇒ In pratica, questi metodi consentono di ottenere denoising e outlier removal come effetto collaterale



22

### Mesh poligonali

I poligoni possono essere:

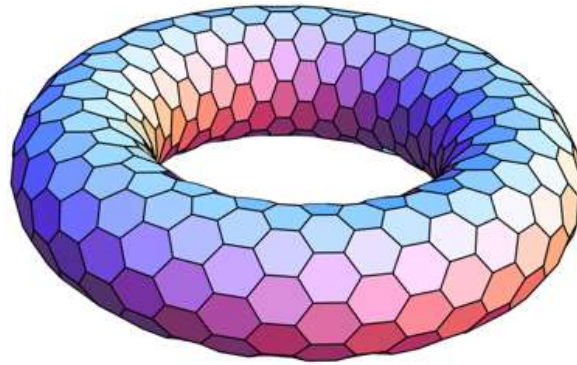
- ✓ triangoli:
  - ⇒ Triangular mesh, o tri-mesh, o "mesh simpliciale"
- ✓ quadrilateri (nello slang della CG: "quads")
  - ⇒ Quad-qesh (a volte pure-quad mesh)
- ✓ *quasi* tutti quadrilateri (ma alcuni triangoli, pentagoni, etc)
  - ⇒ ho una "quad-dominant" mesh
- ✓ poligoni generici (triangoli, quadrilateri, pentagoni, esagoni, etc)
  - ⇒ mesh poligonale (generica)



23

## Mesh poligonali

- ✓ volendo, esistono anche le hexa-mesh (e hexa-dominant mesh) (ma assai poco usate)



24

## Mesh Polionali: risoluzione

- ✓ Risoluzione:  
il numero di facce (o di vertici) che compongono la mesh

⇒ Hi-res: più accuratezza

⇒ Low-res: più efficienza

⇒ Mesh low res:  
detta anche low-poly mesh

⇒ La risoluzione di una mesh può essere **adattiva**:  
tassellamento più fine (campionamento più fitto)  
dove necessario

- Per es, dove la curvatura della mesh è alta  
Dove la mesh è piatta, bastano meno triangoli
- Per paragone: la risoluzione di una immagine rasterizzata non è adattiva (rate costante di num pixel per unità di superficie)

num facce lineare con num vertici.

Statisticamente:

per tri-mesh:

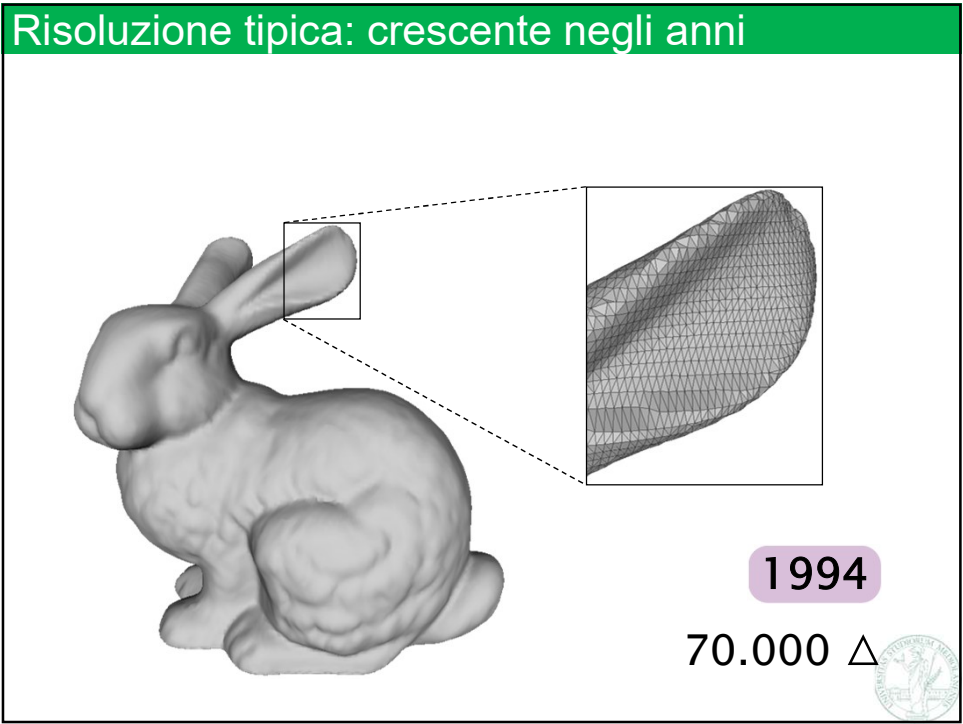
$\text{num facce} \cong 2 \times \text{num vertici}$

per quad-mesh:

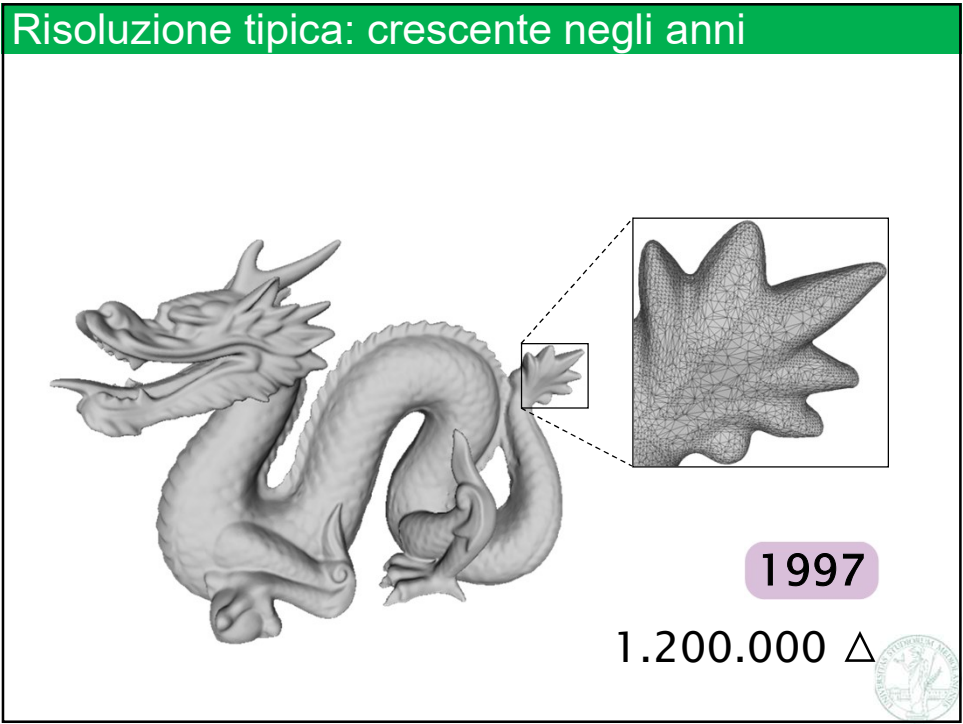
$\text{num facce} \cong \text{num vertici}$



25



26



27

Risoluzione tipica: crescente negli anni

2002

2.000.000.000  $\Delta$

28

3D Meshes: resolution

performance

quality

29

### Mesh Polionali: risoluzione

✓ La risoluzione di una mesh può essere adattiva: tassellamento più fine (campionamento più fitto) dove necessario

⇒ Per es:

dove la curvatura della mesh è alta: più triangoli;  
dove invece la mesh è piatta, meno triangoli

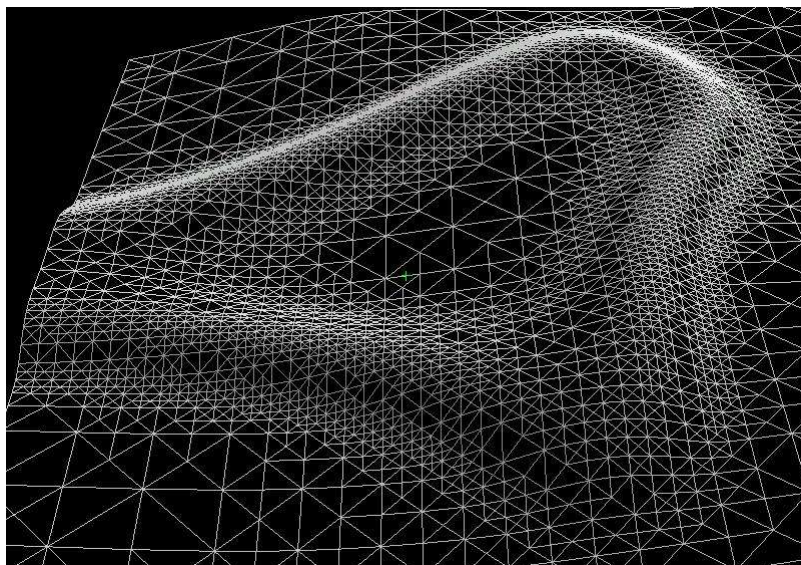
⇒ Per es:

dove la mesh è semanticamente importante  
(es sul volto di un personaggio): più triangoli



30

### 3D Meshes: *risoluzione adattiva*



31

## Mesh Triangolare o Tri-meshes

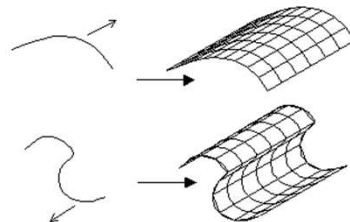
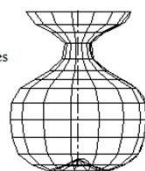
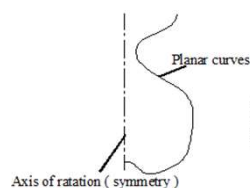
- ✓ Vantaggio: le facce sono sempre planari
  - ⇒ Perché tre punti nello spazio sono sempre co-planari
  - ⇒ matematicamente: la mesh è una approssimazione “lineare a tratti” della superficie che stiamo rappresentando
  - ⇒ Come una linea curva può essere approssimata da *segmenti dritti*, una superficie curva può essere approssimata da *triangoli piatti*
- ✓ Per questo, detta anche mesh simpliciale
- ✓ Di gran lunga il tipo di mesh più diffuso nel rendering
  - ⇒ Vantaggio: GPU hardware support (come vedremo)
  - ⇒ Cioè, questo tipo di mesh può essere renderizzato dalla GPU
  - ⇒ altri poligoni vengono scomposti in triangoli!



32

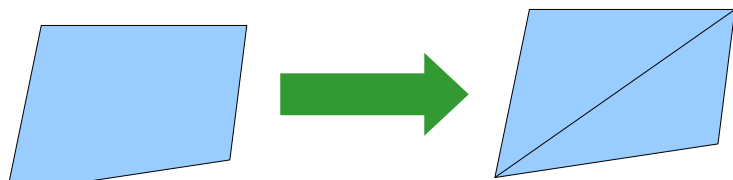
## Pure quad-mesh

- ✓ Generalmente, più difficili da generare automaticamente
  - ⇒ Ma molto usate da artisti e modellatori (ad esempio, nel CAD)
  - ⇒ Per esempio, superfici «di spazzata»



33

Da Quad meshes a Triangle meshes



“un quad”

due triangoli!

“diagonal split”



35

Da Polygonal meshes a triangle meshes



un poligono a  $n$  lati?

$(n-2)$  triangoli!

triangolarizzazione di poligono:  
(in 3D, un problema non del tutto banale...)



36

## Come rappresento internamente una mesh?

Modo diretto: (anche detto: zuppa di triangoli)

✓ un lungo vettore di poligoni

⇒ 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



38

## Come rappresento internamente una mesh?

✓ Modo **indexed mesh**:

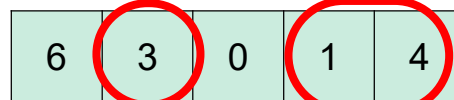
⇒ **Geometria**: array di vertici  $V[0] \dots V[N]$

- in ogni vertice, posizione e attributi

⇒ **Connettività**: (o: "topologia")

- Array di poligoni
- Ogni poligono rappresentato da una sequenza di indici di vertice, uno per «angolo di faccia» (o «corner», o «wedge»)

es.: una faccia  
pentagonale:  
(5 corners)



indice del vertice  $V[3]$

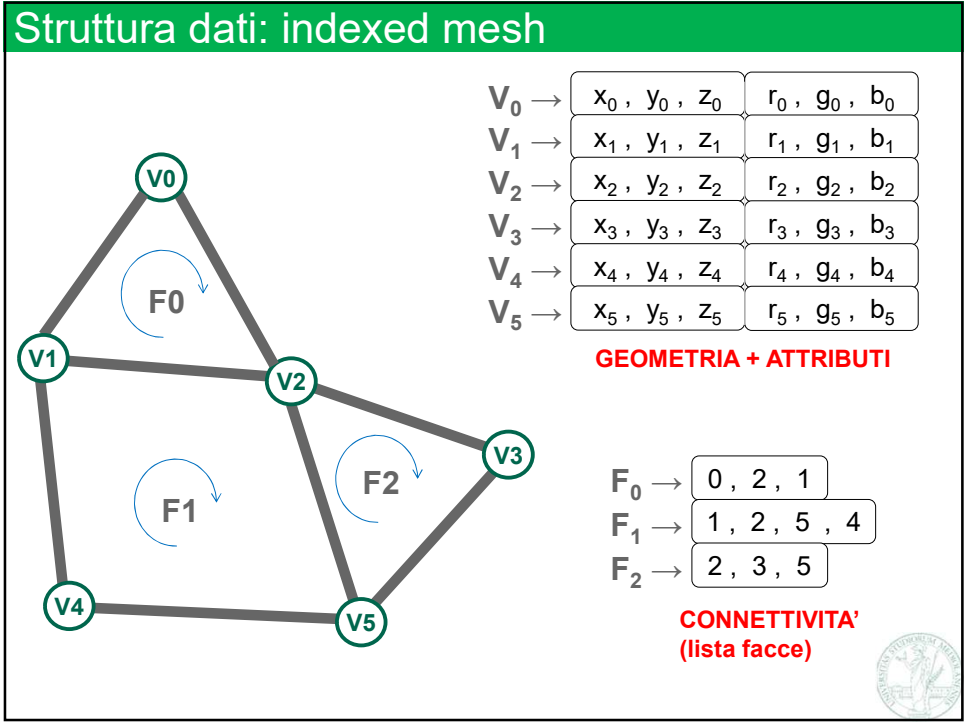
ordinata  
e ciclica

c'è un edge  
da  $V[1]$  a  $V[4]$

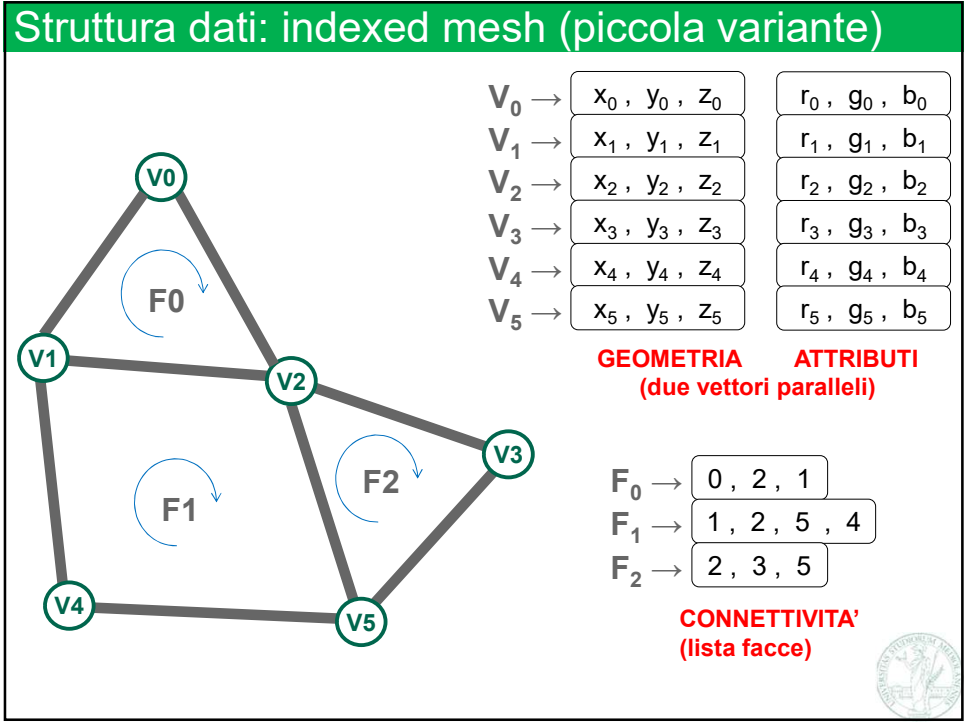
nota: anche da  $V[4]$  a  $V[6]$



39



40



41

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

```
class Vertex {  
    vec3 pos;  
};  
  
class Face{  
    vector<int> vertexIndex;  
    rgb color; /* attribute 1 */  
    vec3 normal; /* attribute 2 */  
};  
  
class Mesh{  
    vector<Vertex> vert; /* geom + attr */  
    vector<Face> face; /* connettivita' */  
};
```



42

## 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 una variante: normali memorizzate per faccia,  
e la mesh è pure-quad (4 indici di vertice per faccia)



43

Alcuni dei formati utilizzati  
I primi nella lista sono i più diffusi

- 3DS - 3D Studio Max
- OBJ - Uno dei più diffuse. Molto semplice
- MA, MB - di Maya
- OFF - ancora più semplice
- 3DX - Rhinoceros file format
- BLEND - Blender file format
- DAE - COLLADA (by Khornos, "universale")
- FBX - Autodesk interchange file format
- X - Direct X object
- SMD - (by Valve) Mesh animate
- BVH - Behavioral file (mesh animate)
- MD3 - quake 3 vertex animations
- DEM - Digital Elevation Models
- DXF - (Autodesk's AutoCAD, for CAD)
- FIG - Used by REND386/AVRIL
- FLT - MultGen Inc.'s OpenFlight format

- IGES - Graphics Exchange Specification
- IV - Open Inventor File Format Info
- LWO, LWB & LWS - Lightwave 3D file formats
- MAZ - Used by Division's dVS/dVISE
- MGF - Materials and Geometry Format
- MSDL - Manchester Scene Description Language
- 3DML - by Flatland inc.
- C4D - Cinema 4D file format
- SLDPTR - SolidWork "part"
- WINGS - Wings3D object
- NFF - Used by Sense8's WorldToolKit
- SKP - Google sketch up
- KMZ - Google Earth model
- OOG - Object Oriented Graphics Library
- PLG - Used by REND386/AVRIL

- POV - "persistence of vision" per ray-tracer
- QD3D - by apple, pple's QuickDraw 3D Metafile format
- TDDD - for Imagine & Turbo Silver r
- NFF & ENFF - (Extended) Neutral File Format
- VIZ - Used by Division's dVS/dVISE
- VRML, VRML97 Virtual Reality Modeling Language (RIP)
- X3D - tentato successore di VRML
- PLY - introdotto by Cyberware - tipic. dati range scan
- DICOM - Dalla casa omonima - tipic. dati CAT scan
- Renderman - per l'omonimo renderer
- RWX - RenderWare Object
- Z3D - ZModeler File format
- HDF - Hierarchical Data Format
- etc

45

File formats per indexed mesh:  
Per esempio: OFF ("Object File Format")

# vertici

# facce

# edge

LetteraL.off

|           |     |     |     |            |
|-----------|-----|-----|-----|------------|
| OFF       | 12  | 10  | 0   |            |
|           | 0.0 | 0.0 | 0.0 | ← indice 0 |
| x,y,z     | 3.0 | 0.0 | 0.0 | ← indice 1 |
| 2ndo vert | 3.0 | 1.0 | 0.0 | ← indice 2 |
|           | 1.0 | 1.0 | 0.0 | ← indice 3 |
|           | 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 |            |

prima faccia:  
4 vertici:  
con indici  
3, 2, 1 e 0

46

File formats per indexed mesh

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

header

geometria

connettività

Mesh in formato OFF  
("Object File Format")

Quale oggetto rappresenta?

47

Mesh: strutture dati

✓ indexed mesh di triangoli

V1

V2

V3

V4

V5

T1

T2

T3

| vert | X  | Y  | Z  | R  | G  | B  |
|------|----|----|----|----|----|----|
| V1   | x1 | y1 | z1 | r1 | g1 | b1 |
| V2   | x2 | y2 | z2 | r2 | g2 | b2 |
| V3   | x3 | y3 | z3 | r3 | g3 | b3 |
| V4   | x4 | y4 | z4 | r4 | g4 | b4 |
| V5   | x5 | y5 | z5 | r5 | g5 | b5 |

GEOMETRIA + ATTRIBUTI

| Tri: | Wedge 1: | Wedge 2: | Wedge 3: |
|------|----------|----------|----------|
| T1   | V4       | V1       | V2       |
| T2   | V4       | V2       | V5       |
| T3   | V5       | V2       | V3       |

CONNETTIVITA'

48

## Mesh two-manifold e non

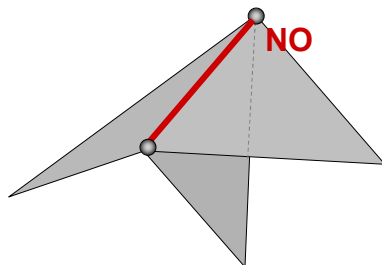
- ✓ una mesh è detta two-manifold (una "varietà due")  
*se rappresenta in effetti una superficie*
  - ⇒ molti algoritmi di geometry processing necessitano che questo sia il caso!
- ✓ Non tutte le mesh (= insiemi di poligoni che condividono dei vertici e degli edge) lo sono!
  - ⇒ le **facce** di una mesh rappresentano sempre (pezzi di) superficie – tutto ok
  - ⇒ su **edge** e **vertici** le cose possono andare storte
  - ⇒ quali condizioni devono verificare?



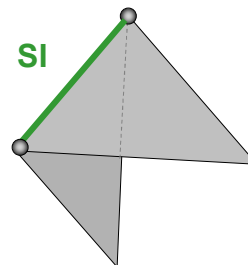
50

## Edge two manifold

- ✓ un edge interno è two-manifold se è condiviso da al massimo due facce



NO

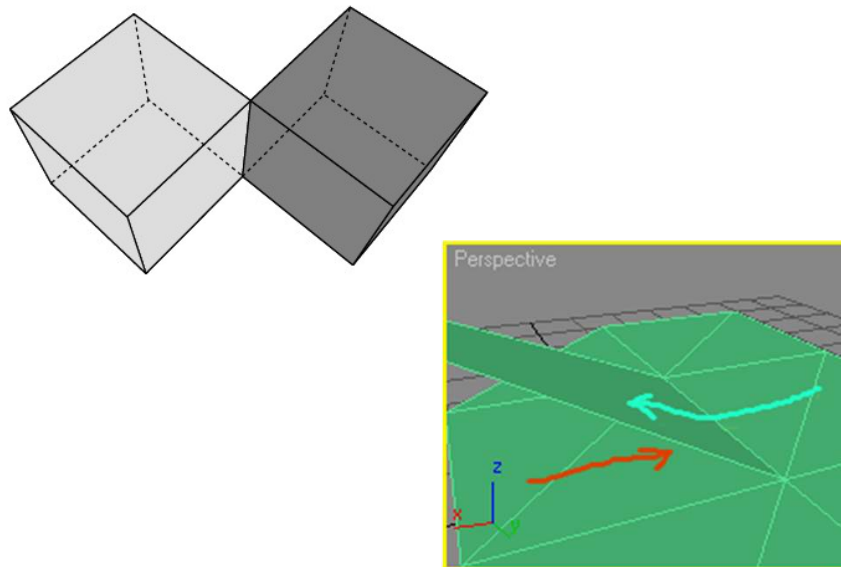


SI



51

## Esempi di edge non 2-manifolds

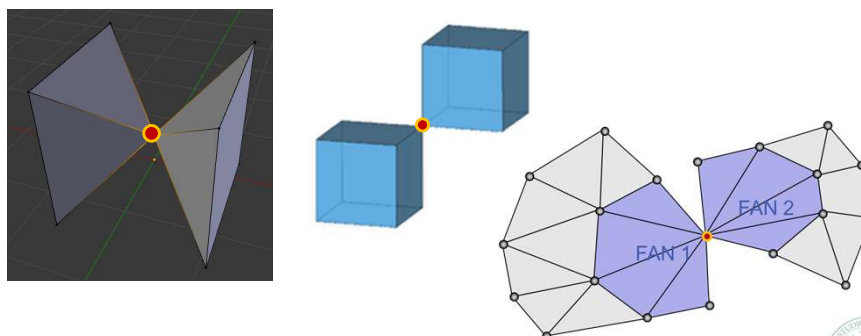


52

## Vertice two-manifold

- ✓ due facce sono adiacenti se condividono un edge
- ✓ L'insieme di facce che condividono un vertice tutte adiacenti a coppie = un fan fan (lett.: un ventaglio)
- ✓ un **vertice** è two manifold se ha un solo fan

Esempi di vertici non 2-Manifold:



55