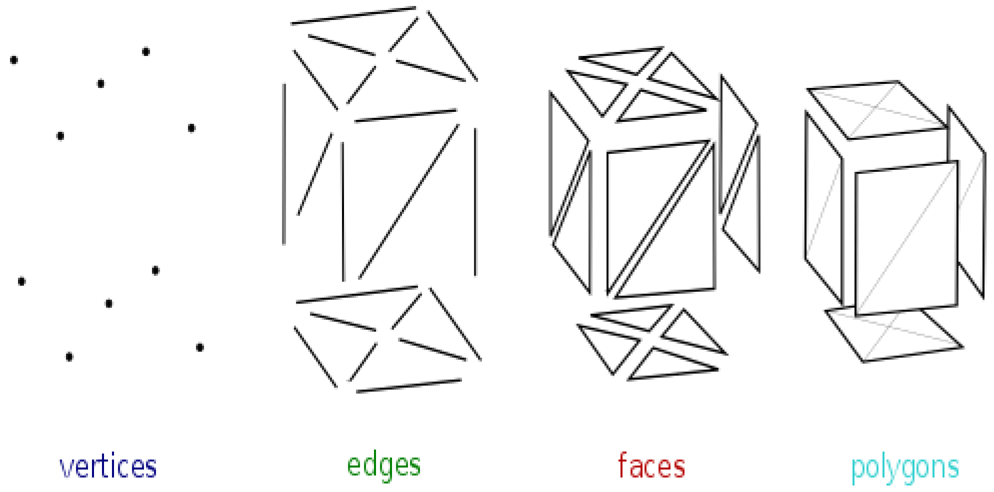
Mesh Poligonali

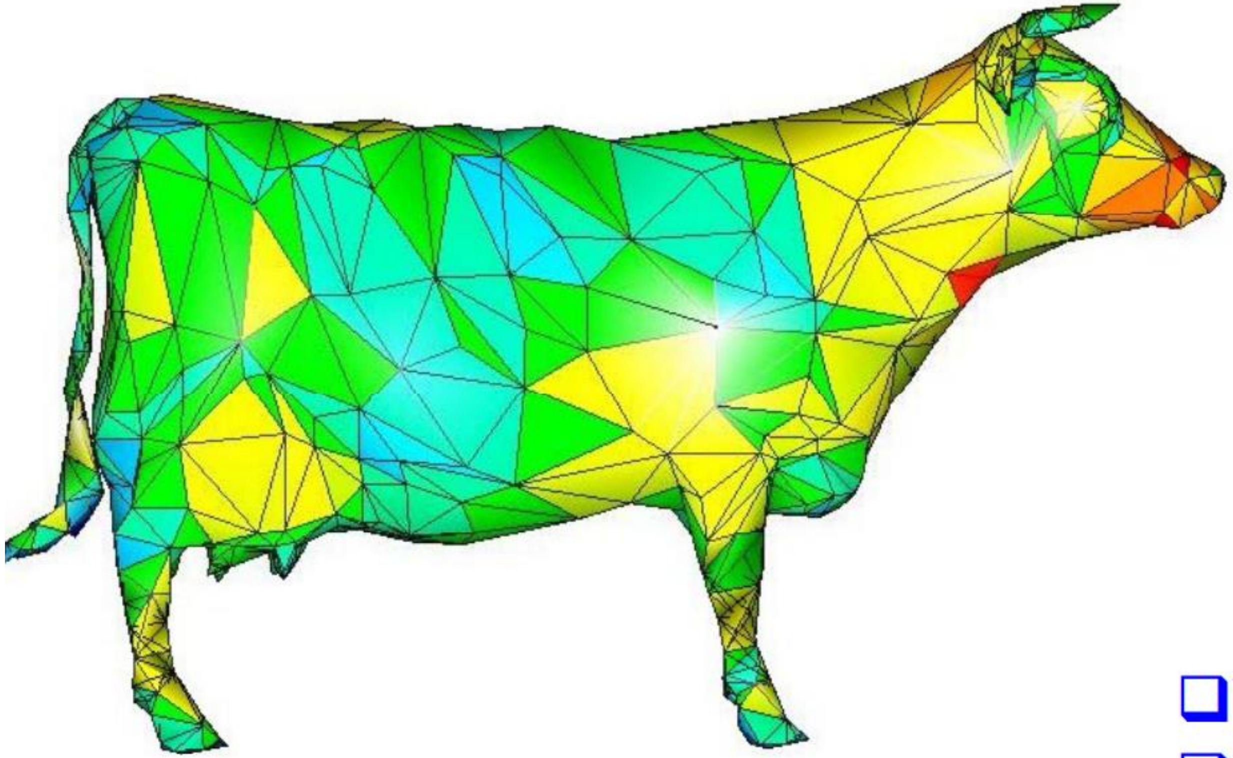
Una mesh poligonale consiste di tre tipi di elementi:

vertice – è una posizione nello spazio e ha informazioni relative al colore e al vettore normale;

lato (edge) – è una connessione tra due vertici;

faccia - un insieme chiuso di lati, ad esempio una faccia triangolare avrà tre lati e una **faccia quadrangolare** avrà quattro lati.
Un **poligono** è un insieme complanare di facce.





- **Vertices: 703**
- **Edges: 2106**
- **Faces: 1401**

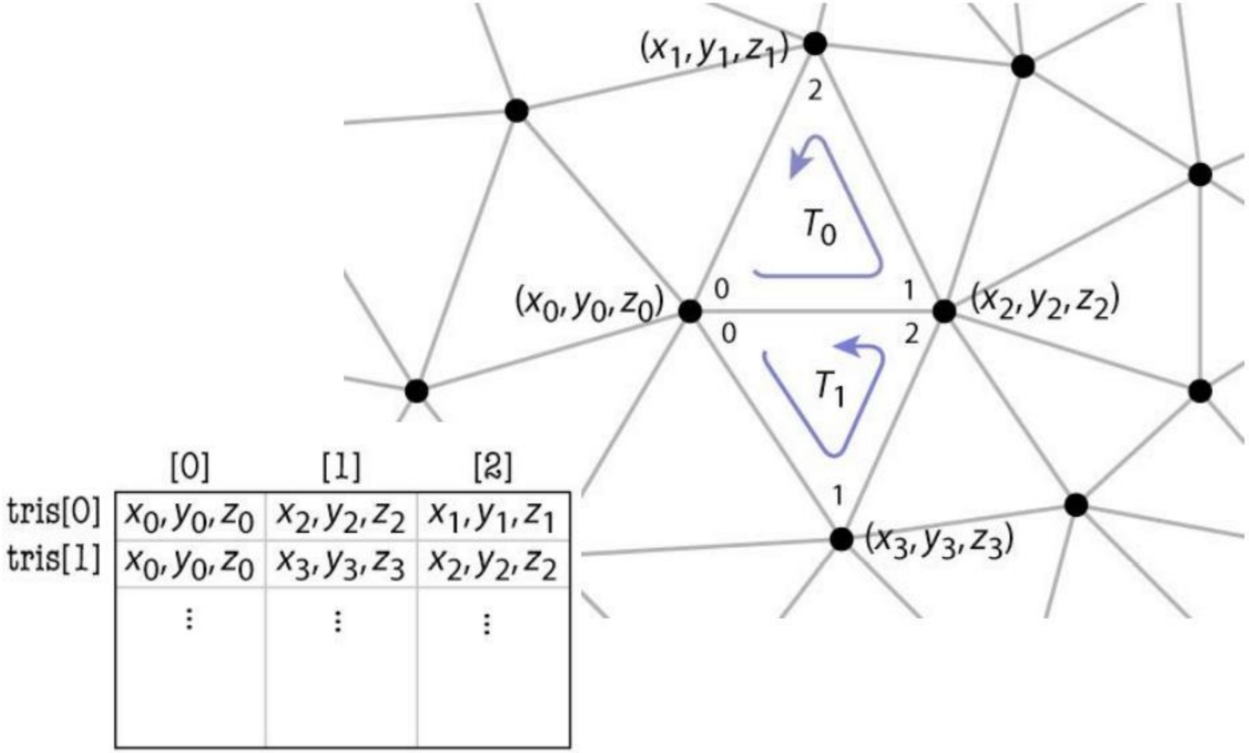
- Una mesh poligonale è discretizzazione lineare a tratti di una superficie continua (un “2 manifold”) immersa in R^3

Connettività e Geometria sono le informazioni che descrivono la mesh;

La **Connettività della mesh, o topologia**, descrive la relazione di incidenza tra gli elementi della mesh.

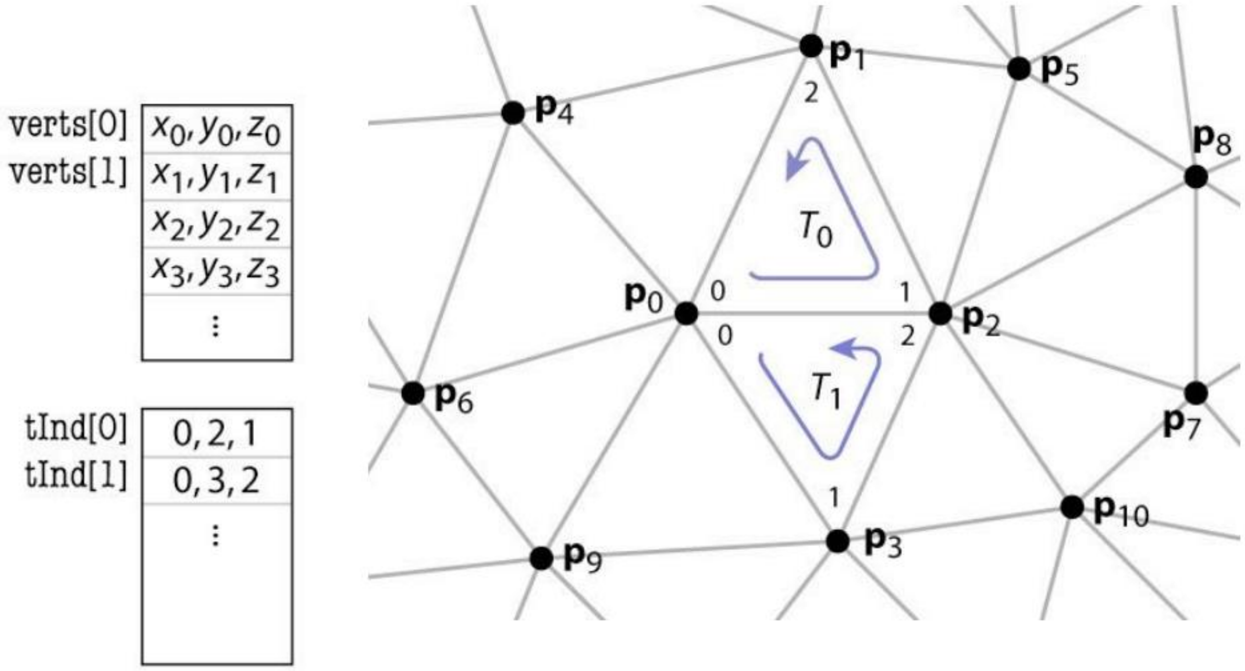
La **Geometria della mesh** specifica la posizione e altra caratteristiche geometriche di ogni vertice.

Rappresentazione di una mesh come lista di triangoli:



	[0]	[1]	[2]
tris[0]	x_0, y_0, z_0	x_2, y_2, z_2	x_1, y_1, z_1
tris[1]	x_0, y_0, z_0	x_3, y_3, z_3	x_2, y_2, z_2
	⋮	⋮	⋮

Rappresentazione di una mesh come lista di vertici e triangoli indicizzati



Confronto tra i due tipi di rappresentazione

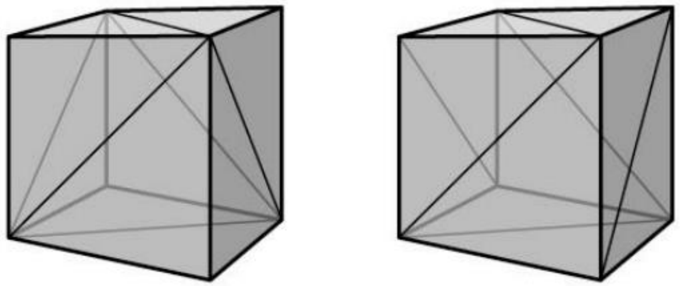
Lista di Triangoli

- Più Semplice
- Informazioni ridondanti

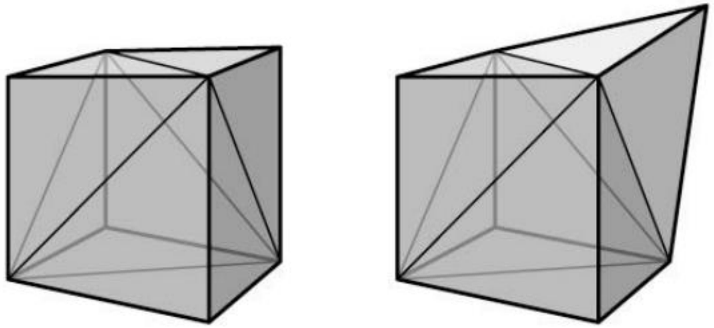
Lista di vertici e Triangoli indicizzati

- La condivisione dei vertici riduce l'utilizzo della memoria
- Garantisce l'integrità della mesh (spostare un vertice, sposta quel vertice in tutti i poligoni che lo condividono.

Stessa geometria – Differente Topologia della Mesh



Stessa Topologia della Mesh – Differente geometria



Informazioni sulla Topologia delle mesh

- Accesso costante nel tempo ai vicini

(utile per esempio nel calcolo della normale alla superficie

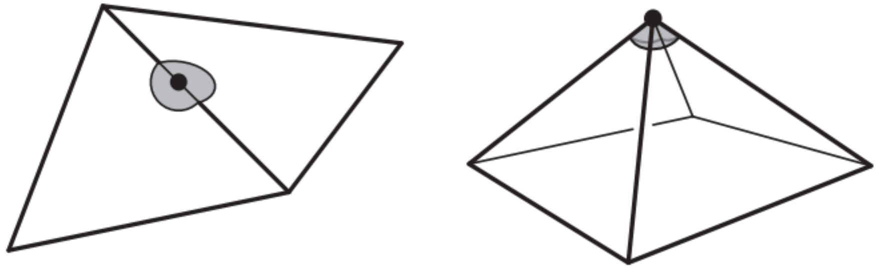
- Modifica della geometria per esempio. aggiunta / rimozione di vertici, facce, bordi, ecc.

E’ necessario utlizzare strutture dati topologiche

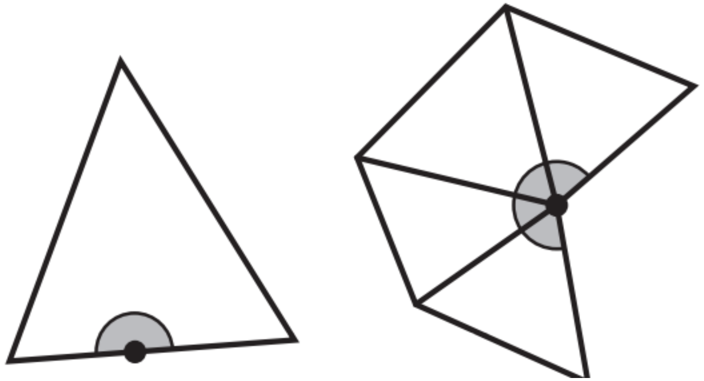
Definizione:

un 2D – manifold è una superficie che, se tagliata con una piccola sfera, produce sempre un disco.

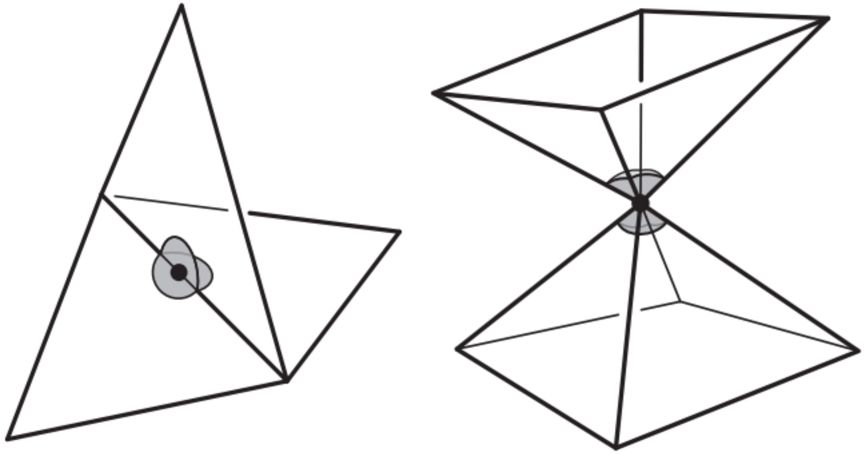
MANIFOLD



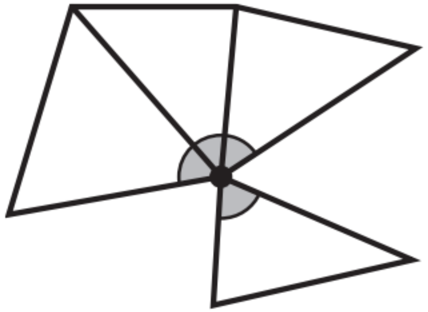
bordo



NON-MANIFOLD



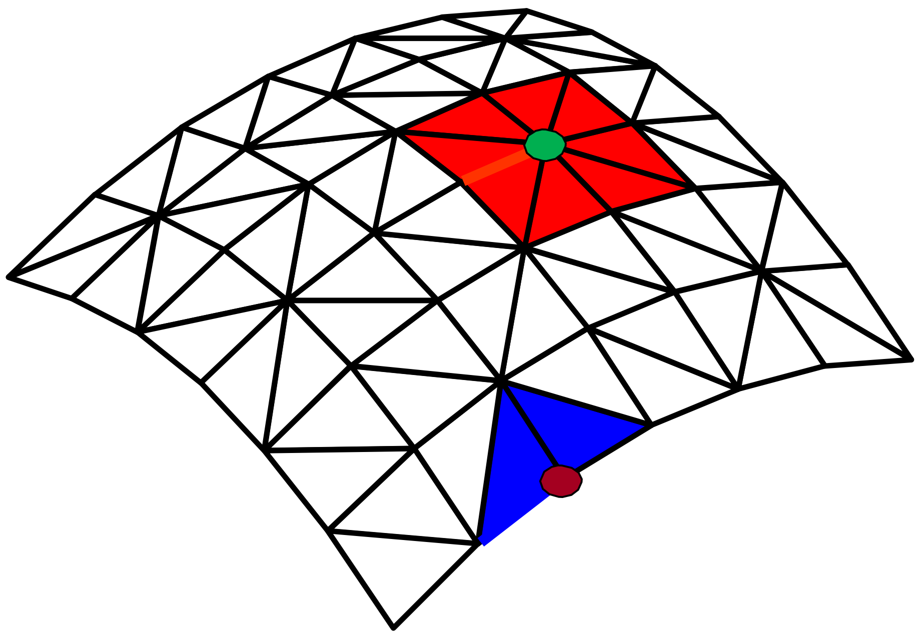
bordo

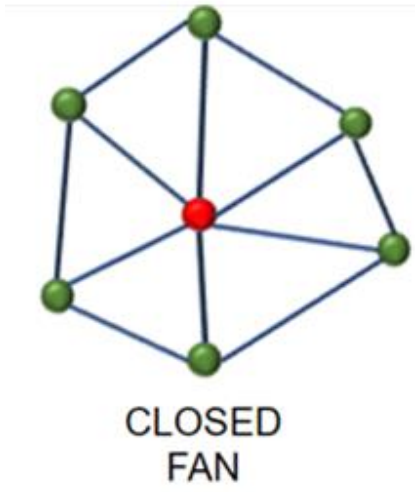


Manifold Meshes:

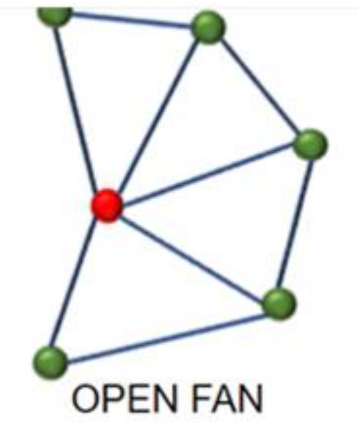
Se una mesh è un 2-**manifold** valgono le seguenti proprietà:

- Ogni edge nella mesh è incidente con solo una faccia (se è di bordo) o due facce (se è interno).
- Le facce incidenti in ogni vertice formano un **OPEN FAN** (se il vertice è di bordo) oppure un **CLOSE FAN** (se il vertice è interno)





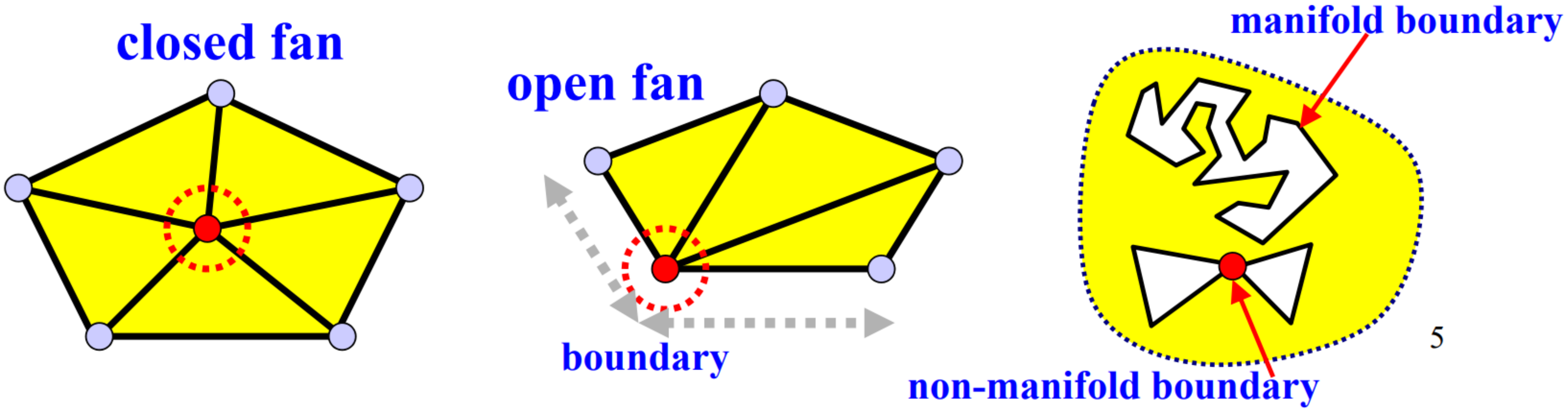
Vertice interno

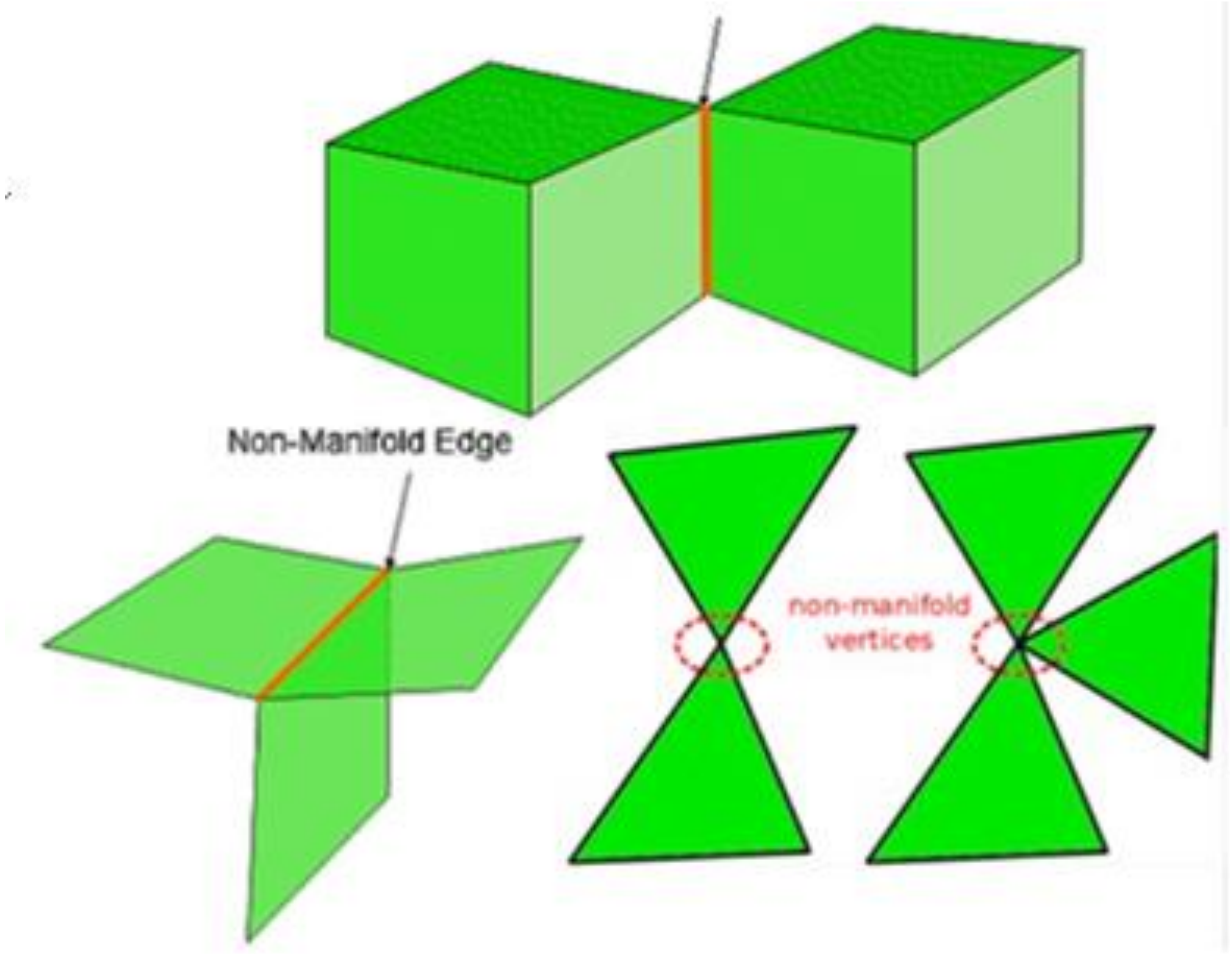


Vertice di bordo o di boundary

Manifold con Boundary o senza Boundary

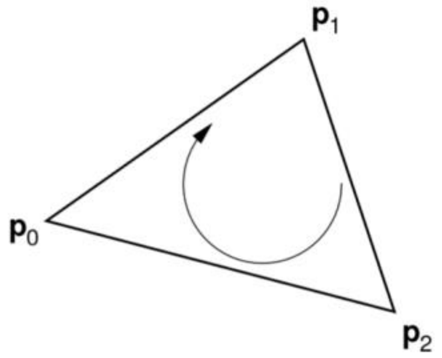
- Se ogni vertice ha un closed fan, allora il manifold **non ha boundary**.
- Gli edge che sono incidenti solo ad una faccia formano il boundary del manifold.
- **Il boundary è l'unione di poligoni semplici**



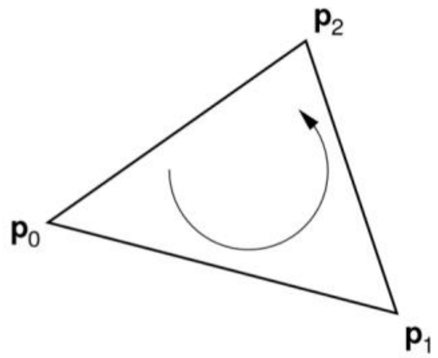

Non-Manifold Edge
non-manifold
vertices

L'orientamento di una faccia è un ciclo ordinato di vertici incidenti;

Dall'orientamento dipende la direzione delle normali.

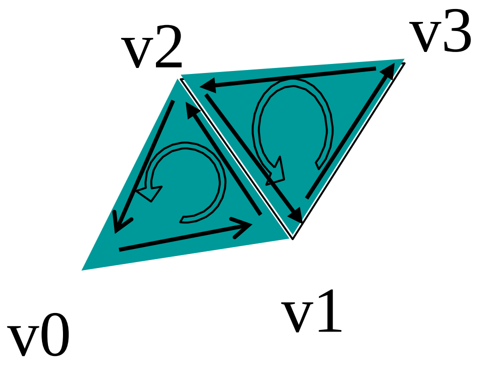


clockwise



counter-clockwise

Si considera per i vertici di un triangolo un ordinamento in senso antiorario



F v0 v1 v2

F v1 v3 v2

L'orientamento di ***due facce adiacenti*** è **compatibile** se i due vertici del lato comune sono in ordine opposto;

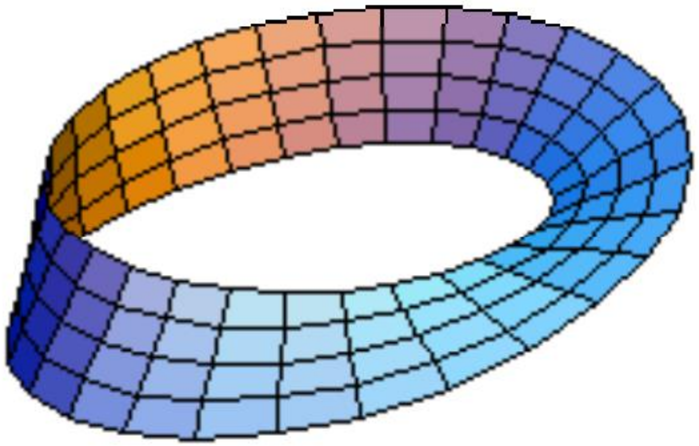
Un mesh manifold è **orientabile** se ogni coppia di **facce adiacenti** ha un orientamento compatibile;

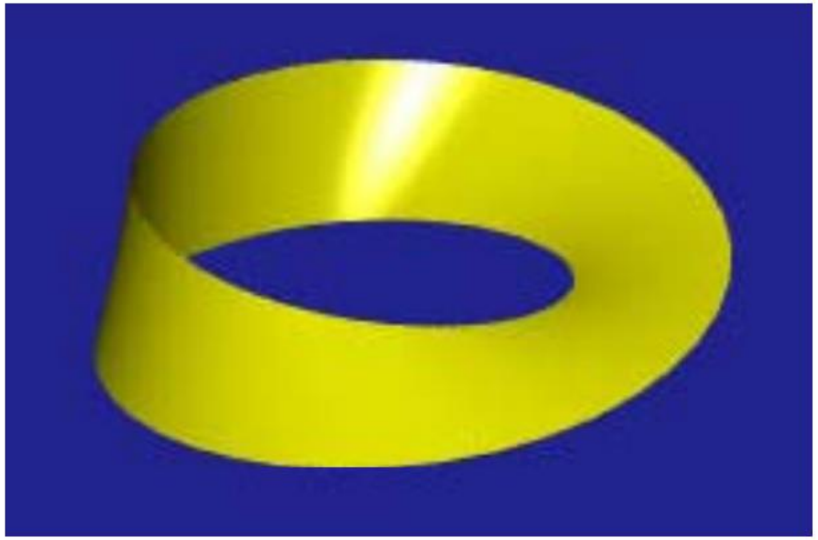
Importante per la visibilità delle facce: in questo modo le normali sono coerenti hanno lo stesso verso.

Ci permette di distinguere tra «davanti» e «dietro di una faccia»

Superfici non orientabili

- Nastro di Moebius
- si passa da una superficie a quella "dietro" senza attraversare il nastro e senza saltare il bordo ma semplicemente camminando a lungo.





Mesh poligonali

I poligoni possono essere:

tutti triangoli: ⇒Triangular mesh, o tri-mesh, o “mesh simpliciale”
tutti quadrilateri (“quads”) ⇒Quad-qesh (a volte pure-quad mesh)

quasi tutti quadrilateri (ma alcuni triangoli, pentagoni, etc) ⇒ “quad-dominant” mesh

poligoni generici (triangoli, quadrilateri, etc)

Mesh triangolari o Tri-Mesh

Vantaggio: facce sempre planari
⇒ tre punti nello spazio sono sempre co-planari
Vantaggio: semplice modo di interpolare attributi

Una mesh poligonale è discretizzazione lineare a tratti di una superficie continua (un “2 manifold”) immersa in R^3 .

Le mesh triangolari rappresentano l’unico tipo di mesh che può essere renderizzato direttamente dalla GPU

⇒ altri poligoni vengono scomposti in triangoli!

Da quad-mesh a triangle-mesh

Da polygonal -mesh a triangle-mesh

RISOLUZIONE DI UNA MESH

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

High Resolution: maggiore accuratezza

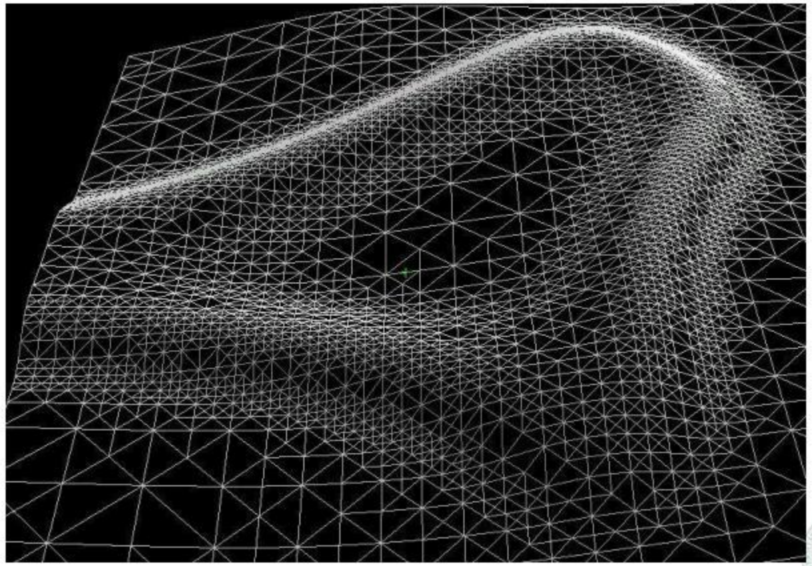
low Resolution: maggiore efficienza

Le mesh low-Resolution sono anche dette detta anche **low-poly mesh**

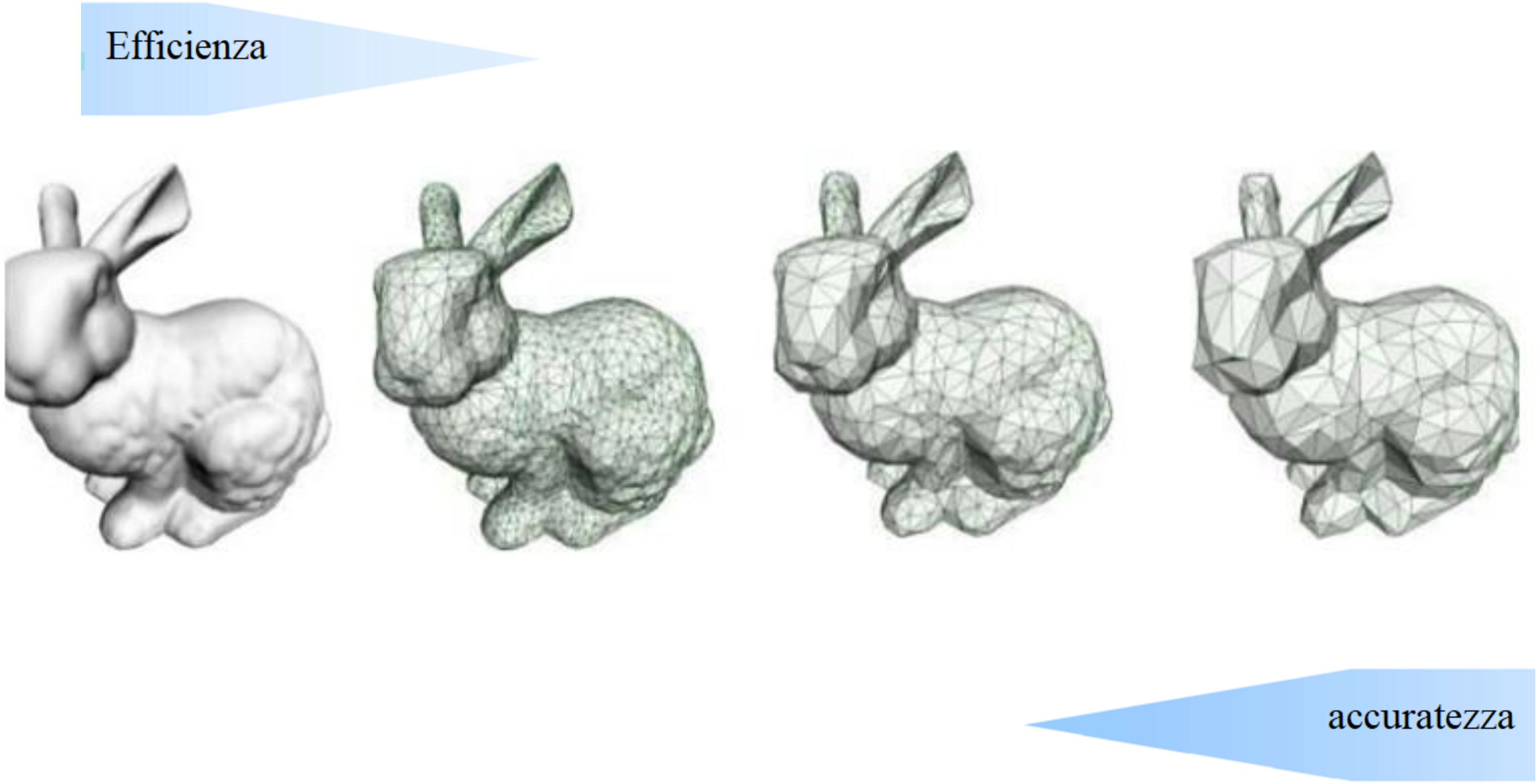
La risoluzione di una mesh può essere adattiva:

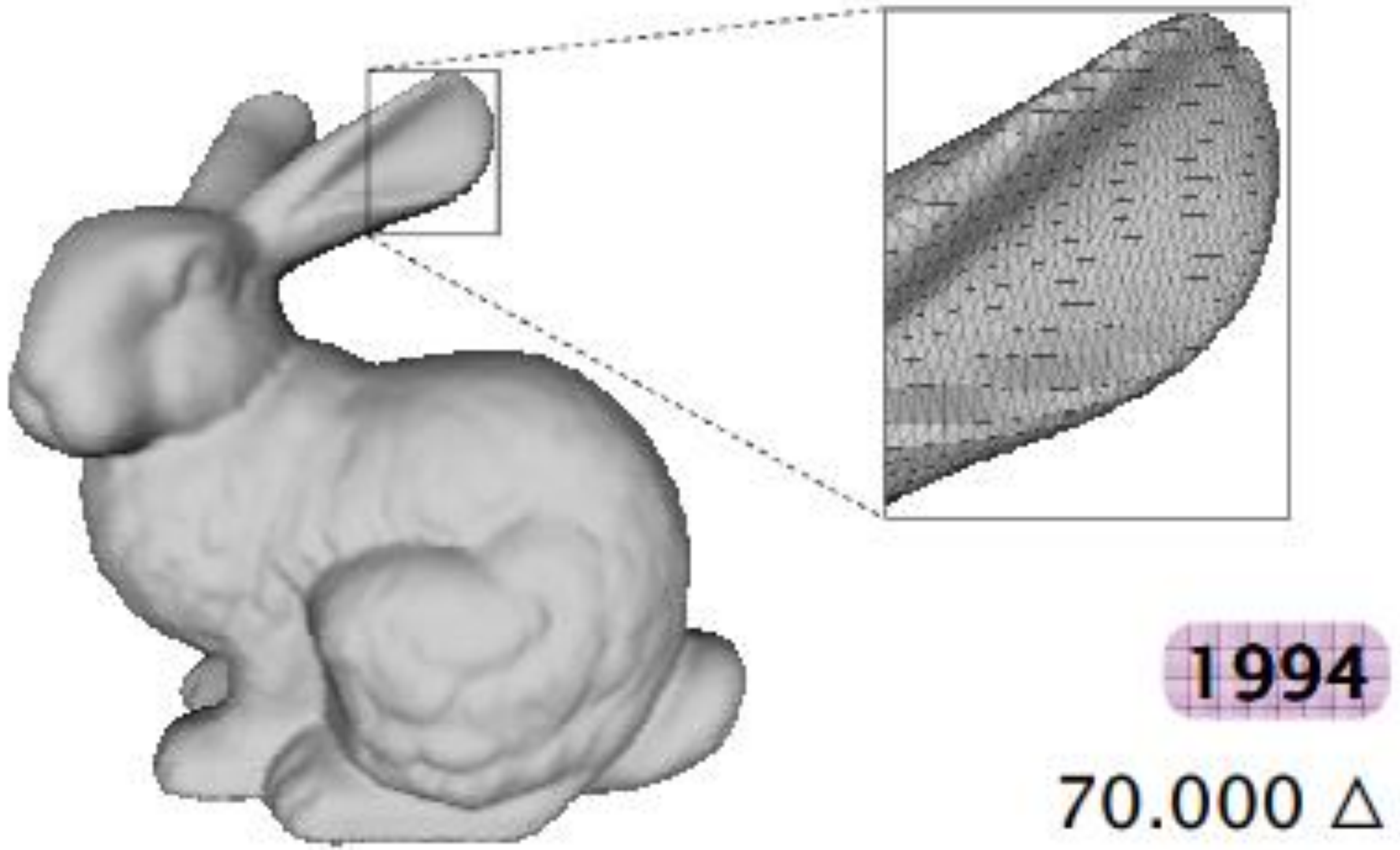
- tassellamento più fine (campionamento più fitto), dove necessario, per es, dove la curvatura della mesh è alta
- dove la mesh è piatta, bastano meno triangoli

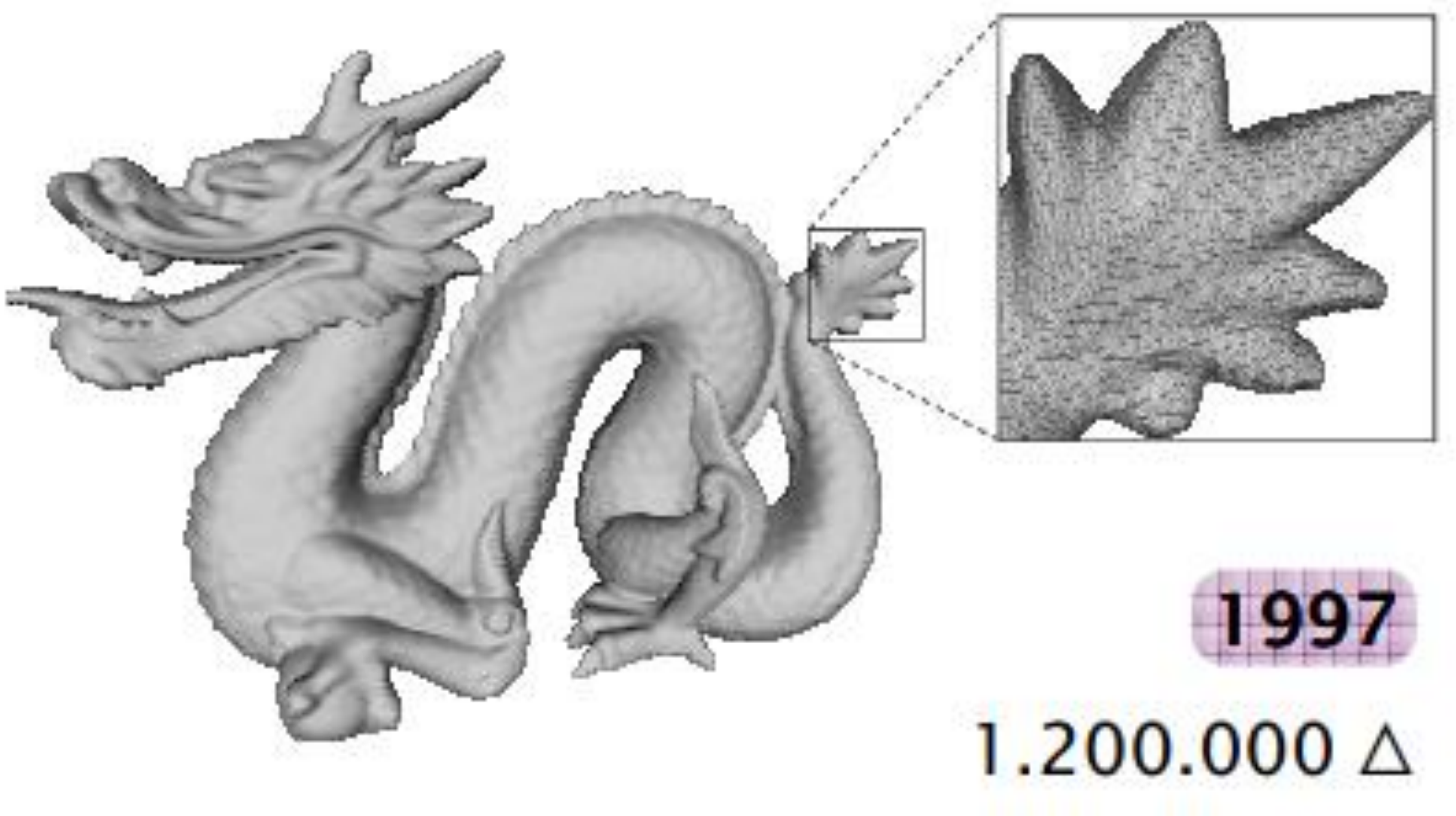
ESEMPIO DI RISOLUZIONE ADATTIVA

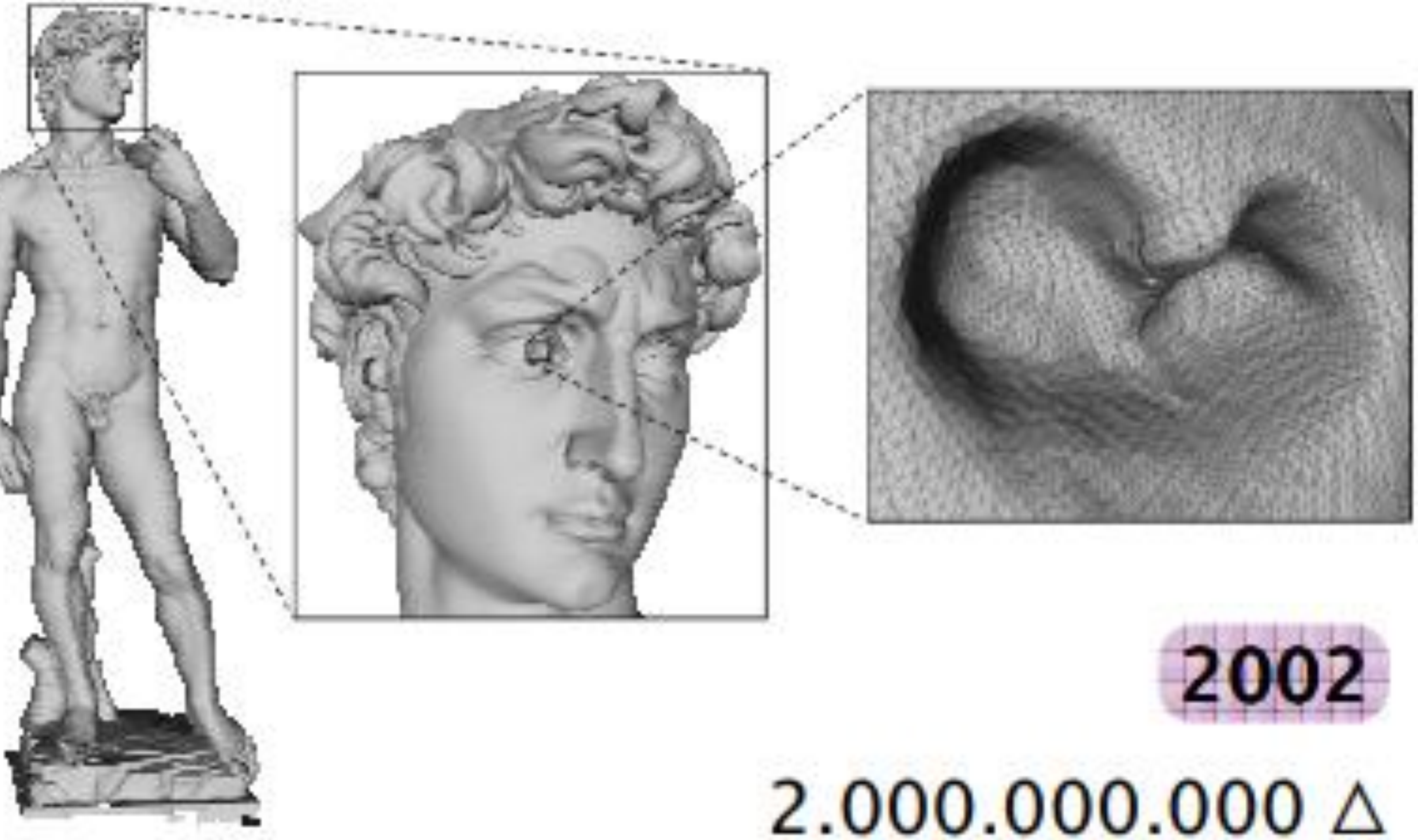


Accuratezza ed efficienza sono inversamente proporzionali




1994
70.000 △


1997
1.200.000 △



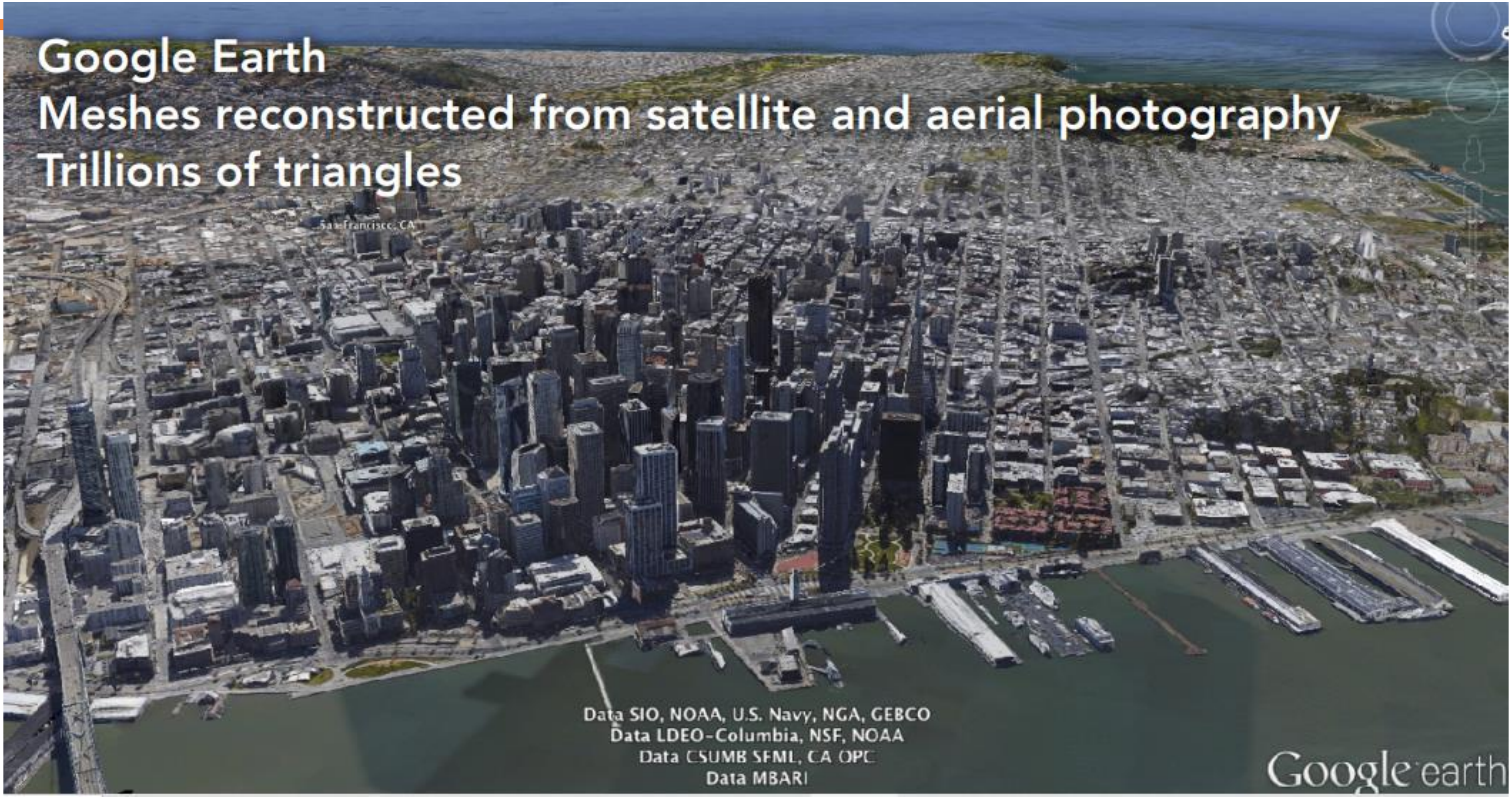
2002

2.000.000.000 △

Google Earth

Meshes reconstructed from satellite and aerial photography

Trillions of triangles



Mesh regolare

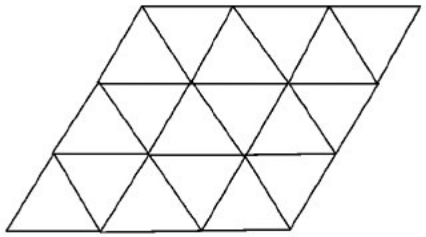
Si definisce «Valenza» di un vertice: **il numero di facce (o di edge) adiacenti ad quel vertice**

Un Vertice si dice regolare (per vertici interni): se il vertice ha **valenza 4** nel caso di quad mesh
valenza 6 nel caso di tri mesh

Una mesh si dice regolare o strutturata se tutti i suoi vertici interni sono regolari

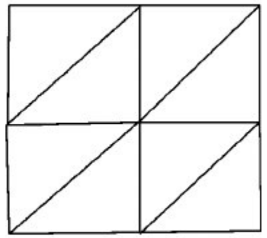
Mesh irregolare , se pochi dei suoi vertici sono regolari, (2/3 o la metà)

Mesh semi-regolare: ottenuta mediante suddivisione regolare di una mesh irregolare: tutti i vertici sono regolari, eccetto che per un piccolo numero di vertici straordinari



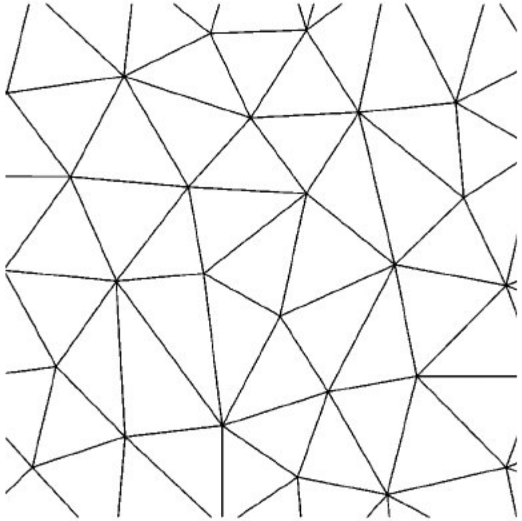
(a)

regolare



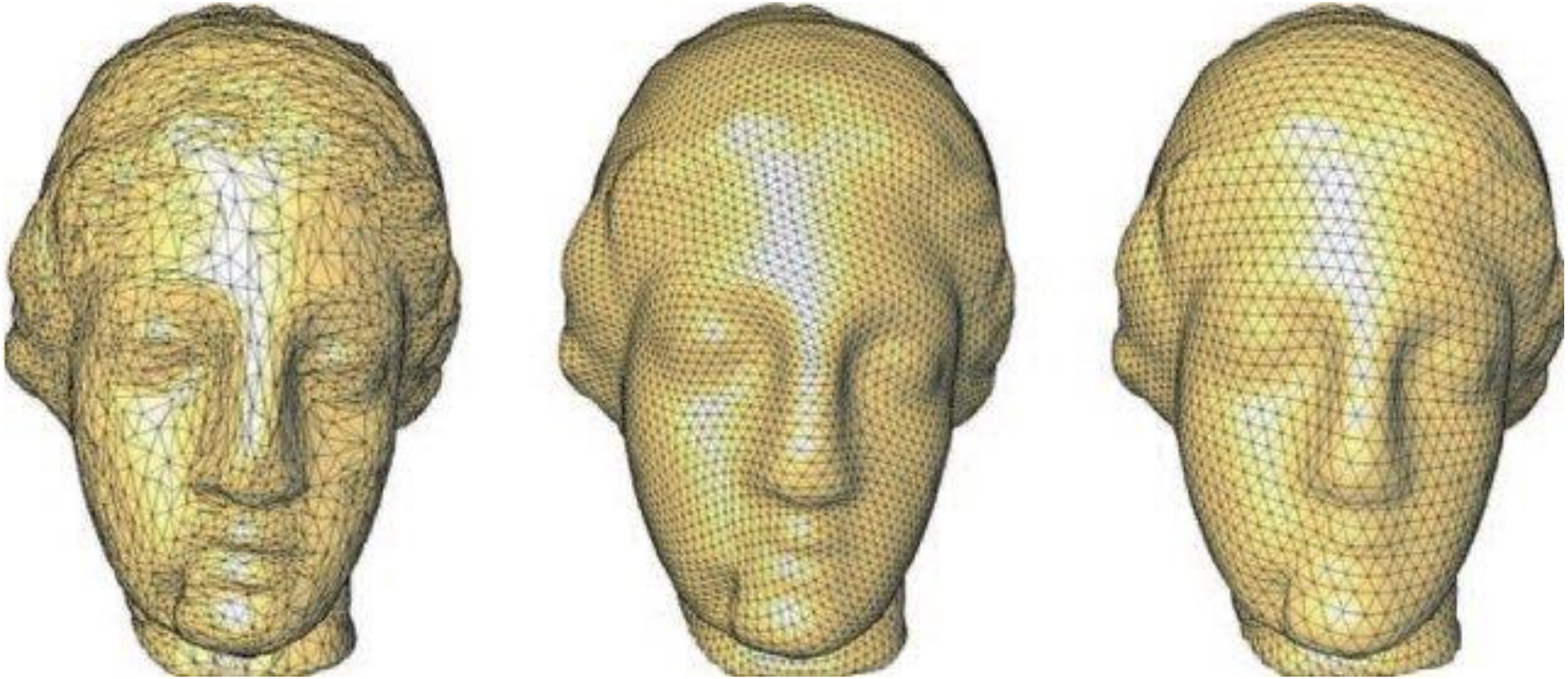
(b)

regolare



(c)

irregolare

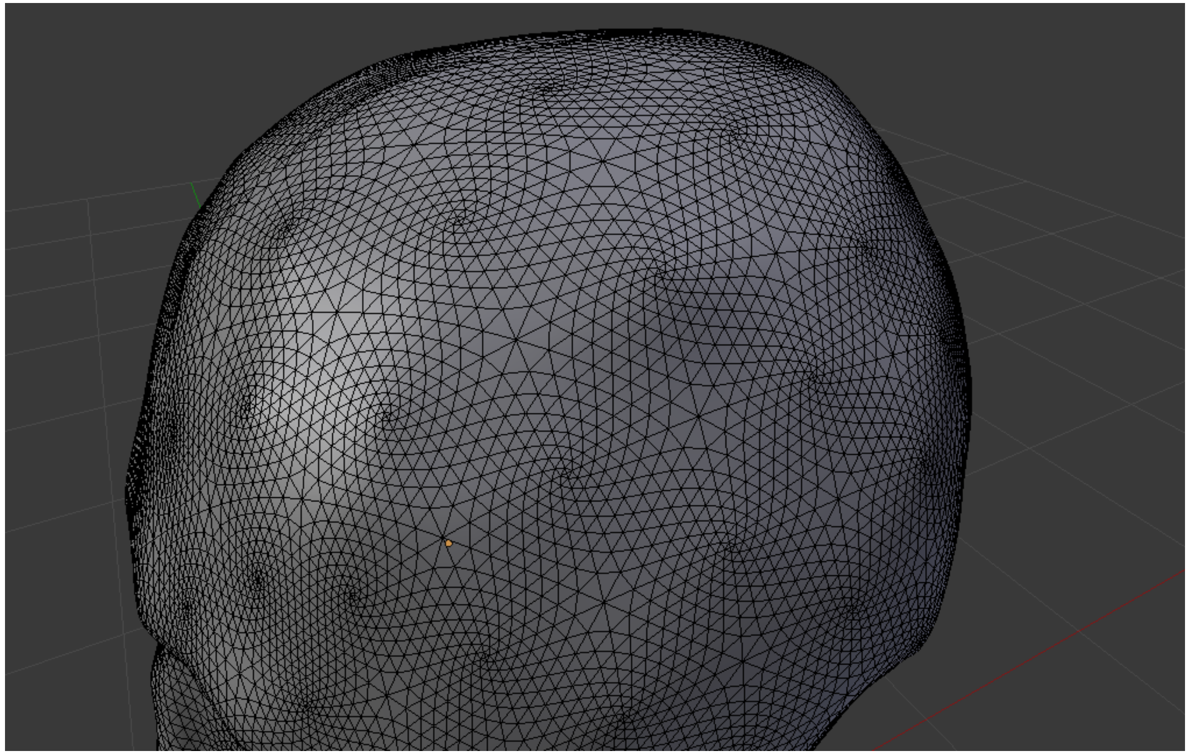


Irregolare Semiregolare Regolare

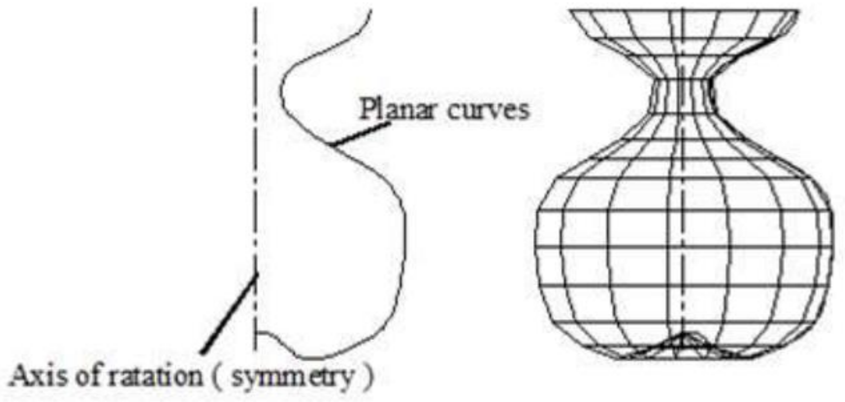
Mesh regolari e mesh irregolari :

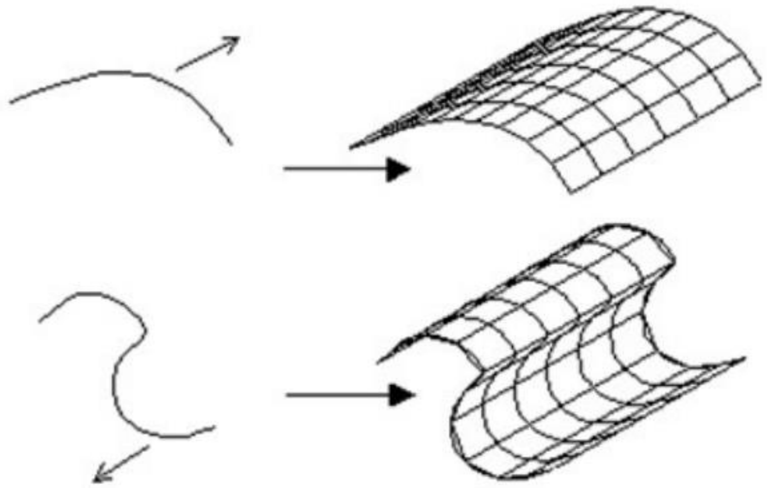
Le mesh irregolari permettono una maggiore adattività della risoluzione

Alcuni metodi per generare mesh producono tipicamente mesh di triangoli irregolari (La maggior parte dei metodi di acquisizione 3D (scanning), Mesh sculpting , Direct “low poly editing”)



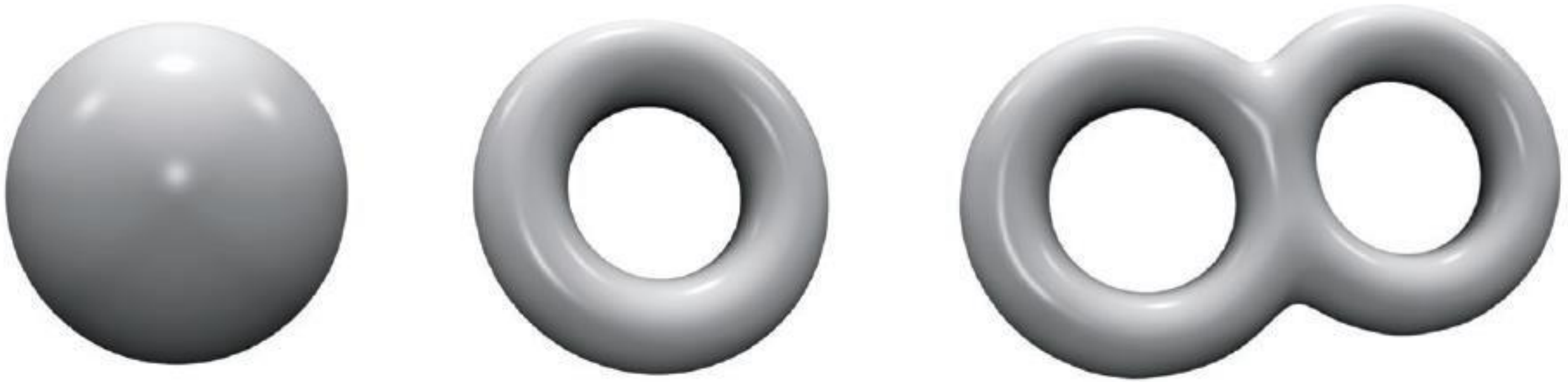
Altri metodi per generare mesh producono tipicamente quad-mesh regolari: Mesh ottenute per rivoluzione, o mesh spazzate





Genus di una superficie

- Per una superficie orientabile, si definisce genus (o genere) il numero dei buchi che attraversano la superficie.



sphere of genus 0

torus of genus 1

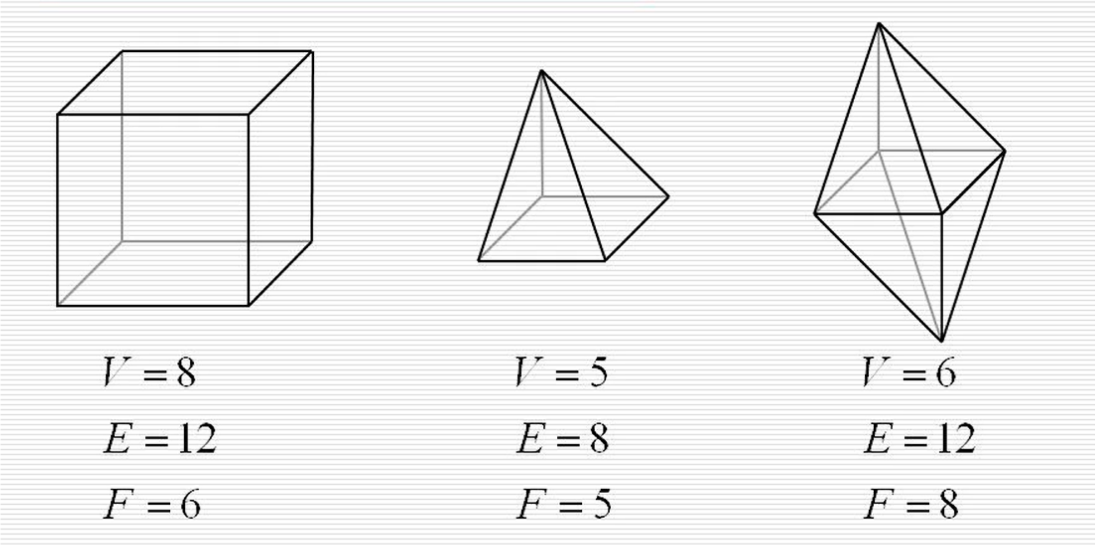
double torus of genus 2

Formula di Eulero per una mesh senza boundaries

La formula di Eulero fornisce una relazione fondamentale tra il numero di facce, bordi e vertici per poliedri in una mesh chiusa e connessa

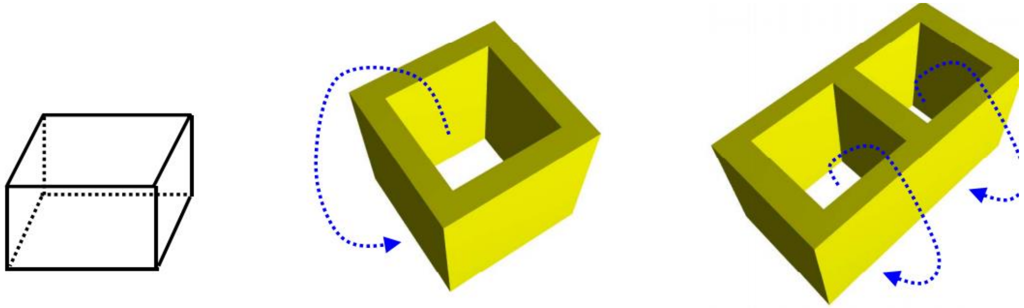
Per una mesh connessa, semplice e solida, la caratteristica di Eulero χ :

$$\chi = |V|-|E|+|F| = 2$$



Per una mesh che non è semplice:

$$\chi = |V|-|E|+|F| = 2(1-g)$$



$g=0 \Rightarrow \chi(\mathbf{M}) = 2(1-0) = 2$ $g=1 \Rightarrow \chi(\mathbf{M}) = 2(1-1) = 0$ $g=2 \Rightarrow \chi(\mathbf{M}) = 2(1-2) = -2$

V=16, E=24, F=8

Formula di Eulero per una mesh con boundary

Il Boundary di un 2-manifold orientabile è l'unione di un insieme di poligoni semplici.

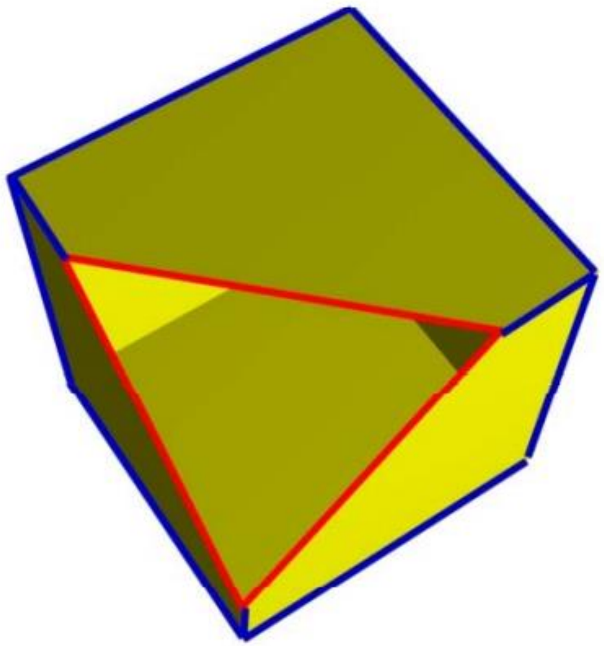
Poiché ogni poligono limita una faccia , queste facce di boundary possono essere aggiunte di nuovo per calcolare la formula caratteristica di Eulero

La caratteristica di Eulero di un 2-manifold orientabile con boundary è:

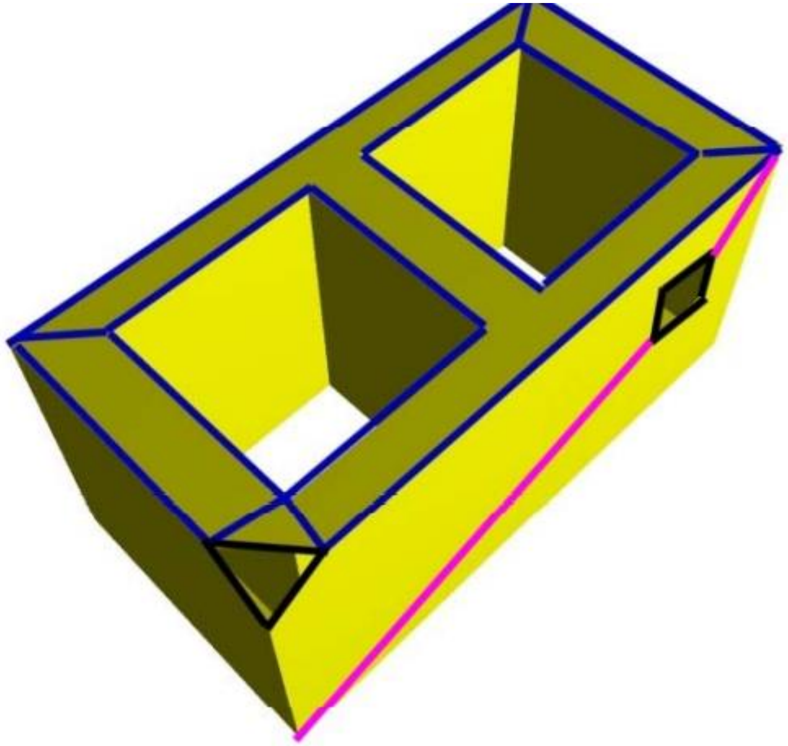
$$\chi = |V|-|E|+|F| +|B| = 2(1-g)$$

$$\chi = |V|-|E|+|F| = 2(1-g) -|B|$$

dove $|B|$ è il numero dei poligoni di boundary.



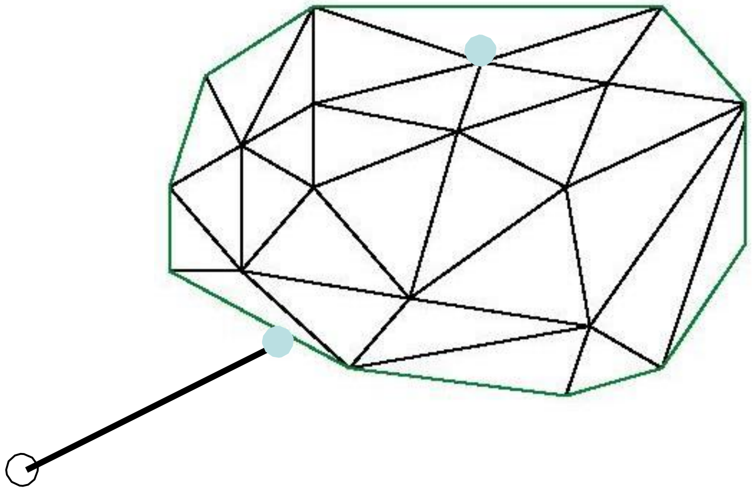
$V = 10, E = 15, F = 6$
$g = 0, \partial = 1$
$\chi(\mathbf{M}) = V-E+F = 1$
$\chi(\mathbf{M}) = 2(1-g)-\partial = 1$



$V = 30, E = 54, F = 20$
$g = 2, \partial = 2$
$\chi(\mathbf{M}) = V-E+F = -4$
$\chi(\mathbf{M}) = 2(1-g)-\partial = -4$

Mesh di triangoli

Una mesh di triangoli M consiste di una componente geometrica ed una componente topologica: la componente topologica può essere rappresentata mediante una struttura a grafo della forma (V,E,F) formata da



Vertici

$$V = \{1, .., N_v\}$$

Edges (spigoli)

$$E = \{(i, j) \in V \times V : X_j \in N(X_i)\}$$

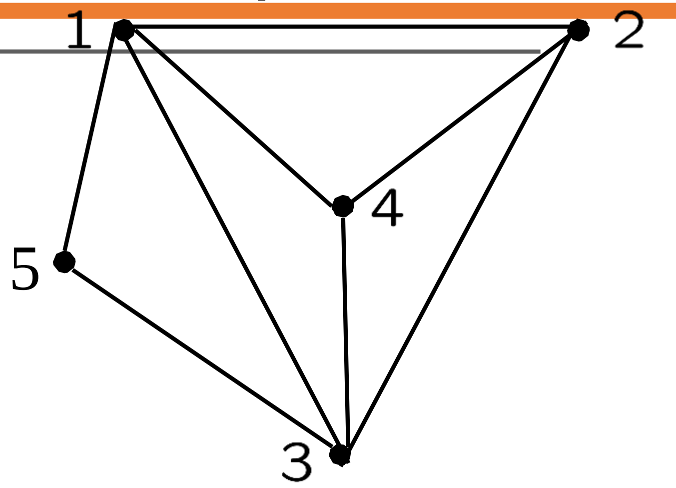
Tra due vertici

Facce

$$F = \{(i, j, k) \in V \times V \times V : (i, j), (i, k), (k, j) \in E\}$$

Tra edges

Matrice di connettività
(Matrice Laplaciano discretizzata)



Il grafo della connettività può essere rappresentato come una matrice L di dimensioni N_v x N_v
(dove N_v è il numero di vertici della mesh)

$$L_{ij} = \begin{cases} -1 & i = j \\ \lambda_{ij} & (i,j) \in E \quad \text{(vicini)} \\ 0 & \text{altrimenti} \end{cases}$$

Diverse possibili scelte per i pesi λ_{ij}

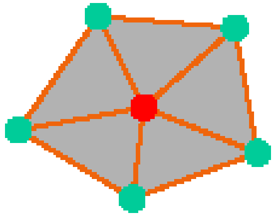
$$L = \begin{bmatrix} -1 & \lambda_{12} & \lambda_{13} & \lambda_{14} & \lambda_{15} \\ \lambda_{21} & -1 & \lambda_{23} & \lambda_{24} & 0 \\ \lambda_{31} & \lambda_{32} & -1 & \lambda_{34} & \lambda_{35} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & -1 & 0 \\ \lambda_{51} & 0 & \lambda_{53} & 0 & -1 \end{bmatrix}$$

Vettori Normali

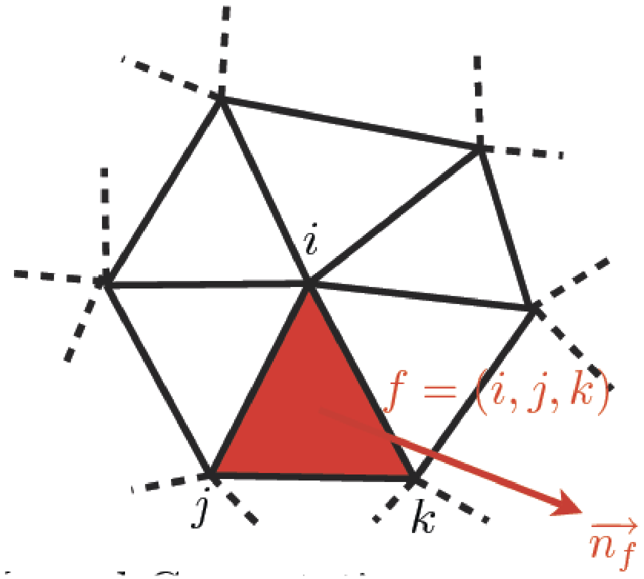
Vertex 1-ring $N(i) = \{ j \in V : (i, j) \in E \} \subset V$

Face 1-ring $N(f) = \{ (i, j, k) \in F : j, k \in V \} \subset T$

Facce/vertici adiacenti al vertice i



- Calcolo delle normali



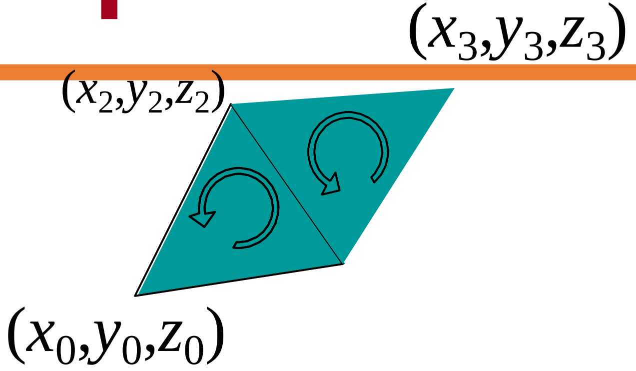
Normale ad una faccia

$$\forall f = (i, j, k) \in T, \quad n_f = \frac{(X_j - X_i) \times (X_k - X_i)}{\left\| (X_j - X_i) \times (X_k - X_i) \right\|}$$

Normale ad un vertice

$$\forall i \in V, \quad n_i = \frac{\overline{n_i}}{\left\| \overline{n_i} \right\|} \quad media \quad \overline{n_i} = \frac{1}{|N(f)|} \sum_{f \in N(f)} n_f$$

.obj : Formato di File per mesh



- Formato di file Popolare
  - Lista ordinata di vertici
  - preceduti da "v" (Wavefront)
  - Coordinate spaziali x,y,z
  - L'indice è dato dall'ordine
- Lista di poligoni
  - preceduti da "f" (Wavefront)
  - Lista ordinata di indici di vertici
  - Lunghezza: numero dei lati
  - Orientazione data dall'ordine

v x_0 y_0 z_0

v x_1 y_1 z_1

v x_2 y_2 z_2

v x_3 y_3 z_3

f 0 1 2

f 1 3 2

Altri attributi

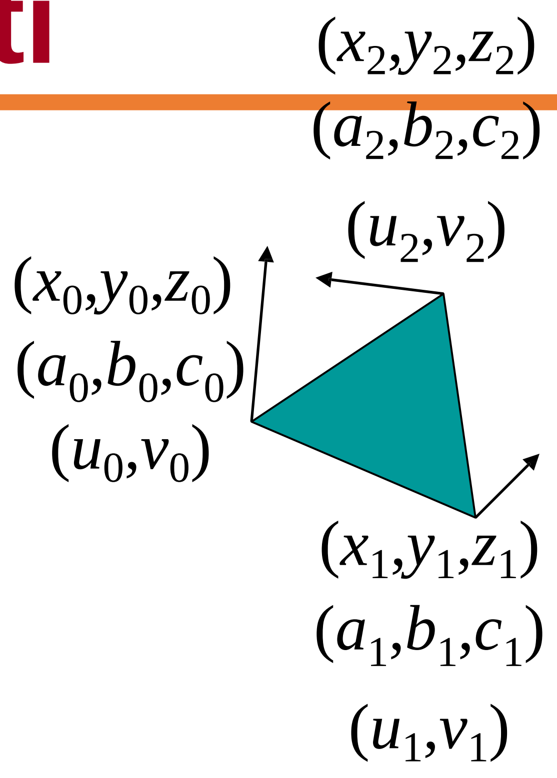
- Normali nei vertici

precedute da w/ “vn” (Wavefront)

  - Contiene x,y,z delle normali
  - Non necessariamenente di lunghezza unitaria
  - Indicizzate come i vertici

- Coordinate di texure
  - Precedute da“vt” (Wavefront)
  - Contiene le coordinate uv
- Facce
  - Usa “/” per separare gli indici
  - Vertex “/” normal “/” texture

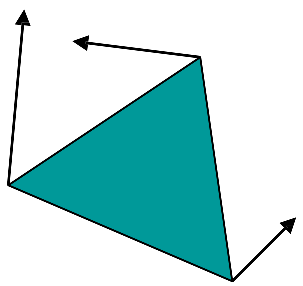
- Normali e texture sono opzionali
- È possibile eliminare le normali “//”



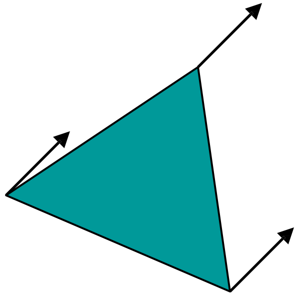
v x_0 y_0 z_0
v x_1 y_1 z_1
v x_2 y_2 z_2

vn a_0 b_0 c_0
vn a_1 b_1 c_1
vn a_2 b_2 c_2

vt u_0 v_0
vt u_1 v_1
vt u_2 v_2



f 0/0/0 1/1/1 2/2/2



f 0/0/0 1/0/1 2/0/2

Formato file .obj (Wavefront)

.obj formato file: tetraedral mesh

v 1.0 0.0 0.0

v 0.0 1.0 0.0

v 0.0 0.0 1.0

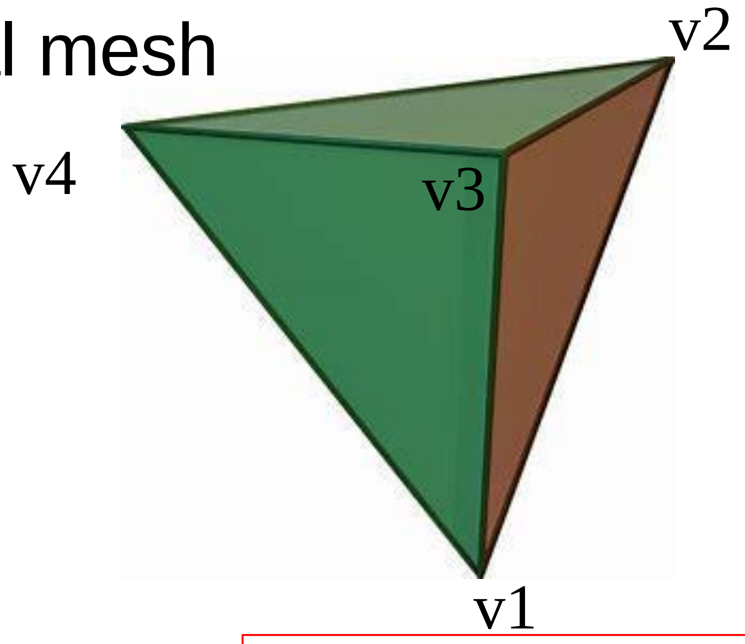
v 0.0 0.0 0.0

f 2 4 3

f 4 2 1

f 1 2 3

f 1 3 4



TRI/QUAD Face

v x,y,z vertex

f v1 v2 v3 face

comment

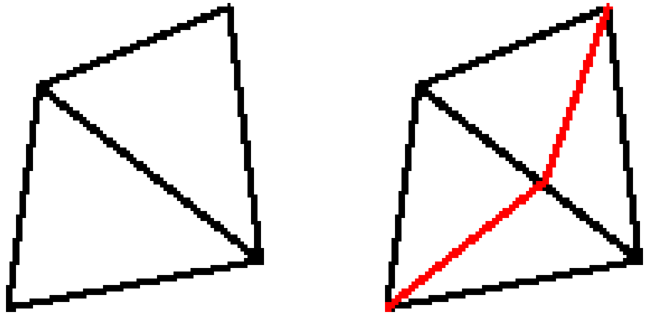
Operazioni comuni sulle Mesh

- Accesso diretto agli elementi
- Accesso ordinato agli elementi
  - Da un elemento iniziale, camminare attraverso gli elementi della mesh per elementi adiacenti (cioè lungo i bordi, le facce, ...)

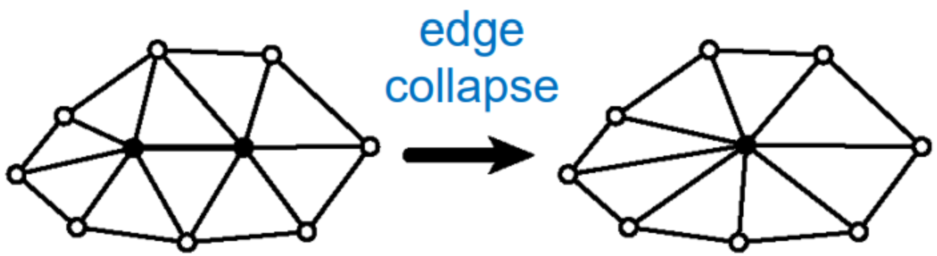
- Relazioni topologiche: data una faccia, scoprire i suoi edge ed i suoi vertici. Dato un vertice scoprire l'insieme degli elementi incidenti

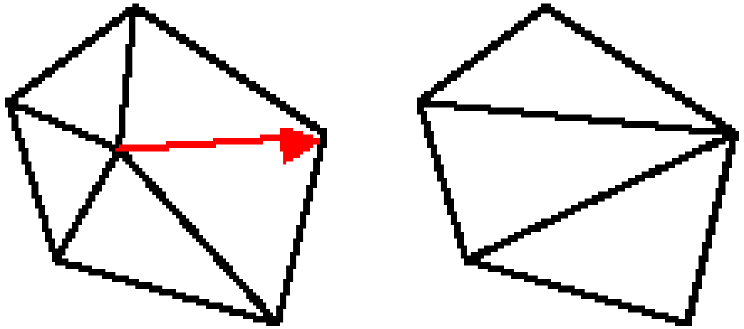
Altre operazioni

Edge Split
Aggiungere un vertice per ottenere 4 triangoli

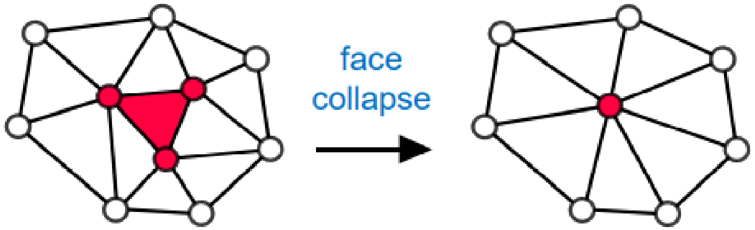


Edge collapse
Rimuovere un edge
Rimuovere un vertice





Face collapse:
Rimuovere una faccia



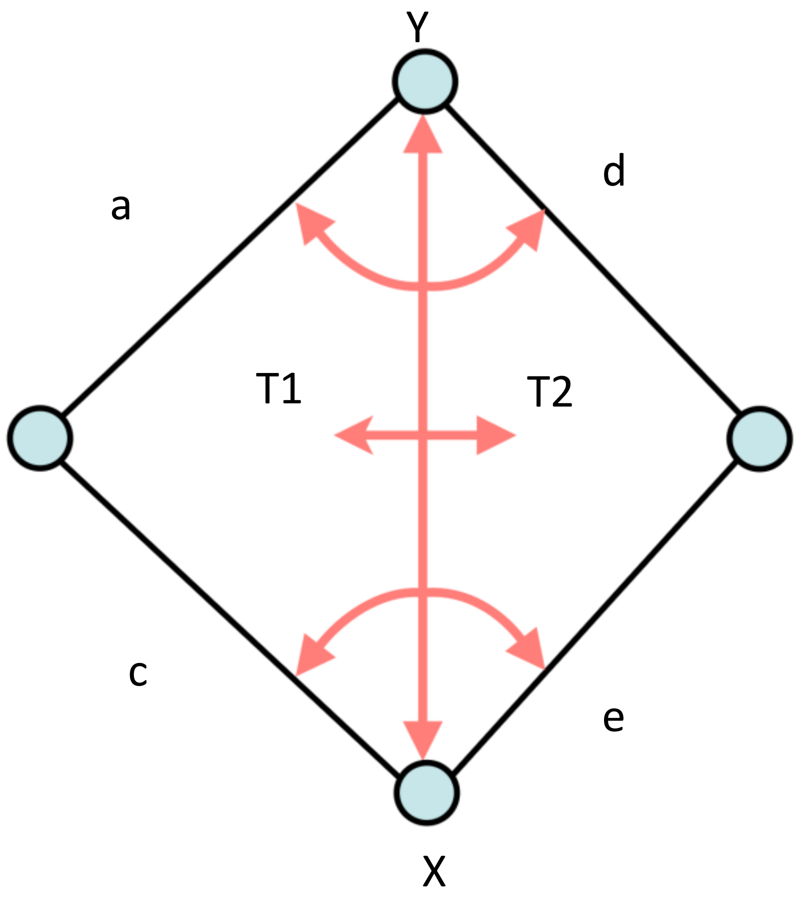
Quasi tutti gli algoritmi che elaborano mesh necessitano di operazioni base sulla connettività come per esempio:
(operazioni di navigazione su mesh)
Data una faccia, contare tutte le facce adiacenti (separate da un edge)
Dato una faccia ed un edge, trova la faccia (se esiste) che sta dall’altra faccia di quell’edge
Dato un vertice, elencare tutte le facce che includono quel vertice (detto ring o ring-1 del vertice)
Dato un vertice, elencare tutti i vertici che sono connessi a quel vertice da un Edge
Dato un edge, scoprire se è un edge di bordo.
Dato un edge di bordo, elencare tutti gli altri edge che fanno parte di quel bordo
Data una faccia, elencare tutti i vertici che fanno parte di quella faccia

E’ necessario che queste operazioni siano effettuate efficientemente, idealmente in tempo costante.

La struttura «lista facce» della mesh indexed, usata per memorizzare la connettività, consente di effettuare queste operazioni attraverso una scansione dell'intero vettore delle facce, che richiede ovviamente un tempo lineare col numero di facce) ⇒ (eccetto: «data una faccia, elenca tutti i vertici che fanno parte di quella faccia»)

Per effettuare mesh processing, sono necessarie strutture più adatte per memorizzare la connettività di una mesh che permettano una navigazione più efficiente

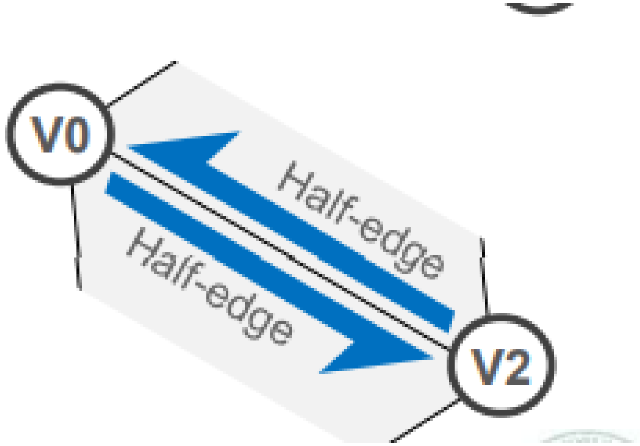
⇒ svantaggio: più prolisse, onerose da mantenere durante le modifiche ⇒ ma consentono di navigare sulla mesh molto più agevolmente



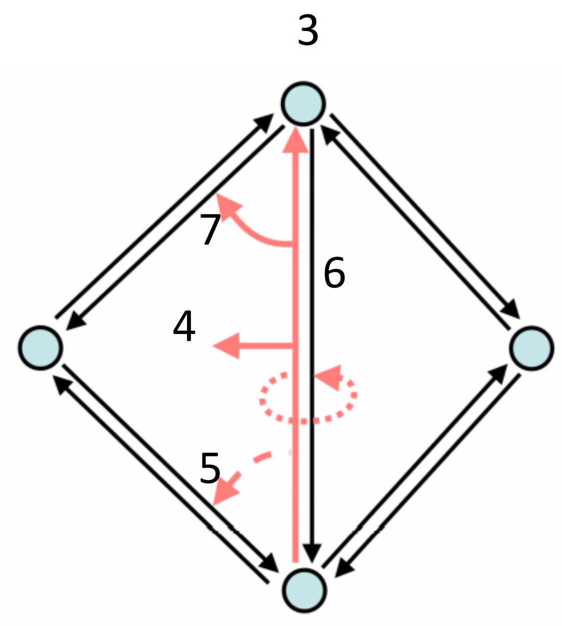
Dato un edge b della mesh che collega i vertici X ed Y, i due triangoli T1, T2 adiacenti su b sono detti le sue wings (ali).
Supponendo che le facce siano orientate in senso orario, ad ogni lato corrispondono 8 informazioni.
Consideriamo il lato b, abbiamo:
:: i 2 vertici X, Y ;
:: le 2 facce incidenti 1, 2;
:: il lato precedente a e il lato successivo c, rispetto alla faccia 1;
:: il lato precedente e e il lato successivo d, rispetto alla faccia 2;

“**Half-edge**”: un edge orientato.

• In una mesh 2--manifold, un edge condiviso da due facce è costituito da
due half-edge che sono uno l’opposto dell’altro



Half-edge data structure

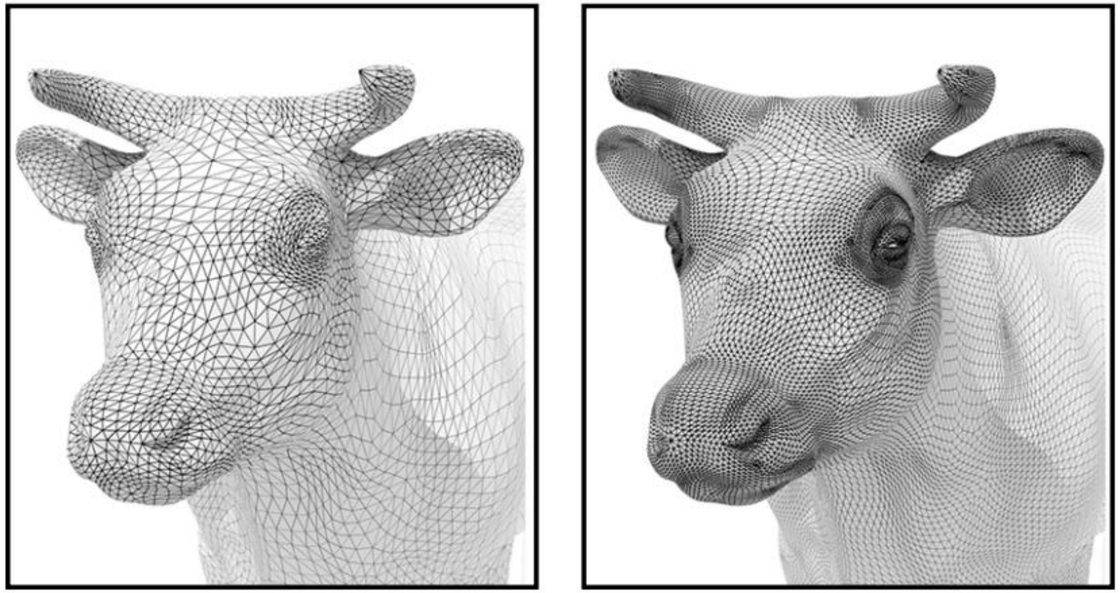


Ogni vertice mantiene referenza ad un half-edge uscente

ogni faccia memorizza la referenza a un half-edge che ne compone il perimetro

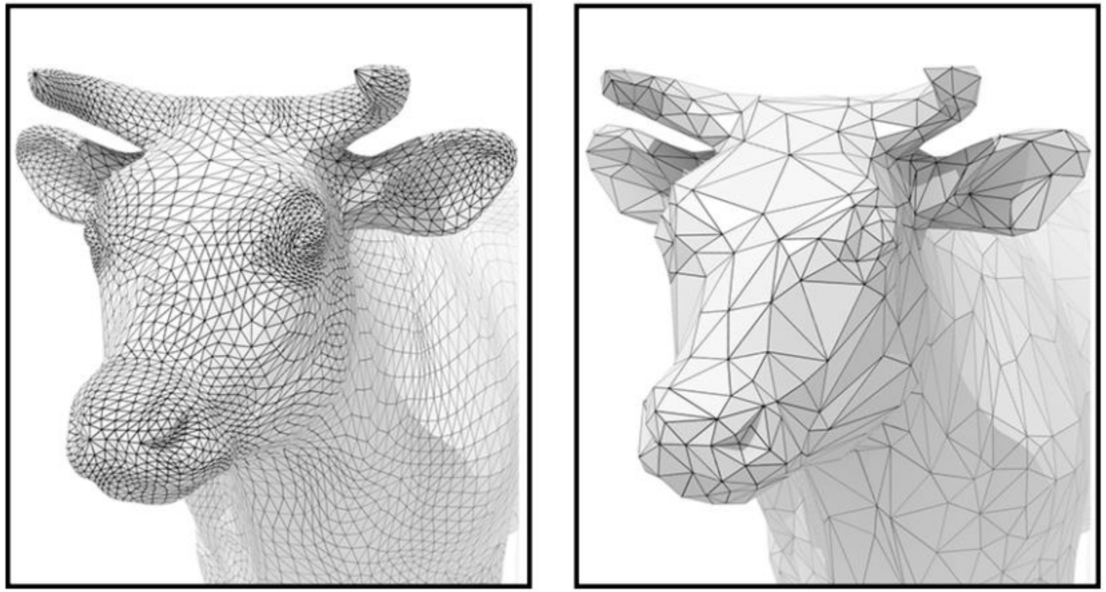
ogni half-edge mantiene la referenza a:
:: il vertice a cui punta (3);
:: la faccia a cui appartiene (4);
:: l'half-edge successivo (all'interno della faccia) (5);
:: l'half-edge opposto (6);
:: l'half-edge precedente (all'interno della faccia) (7)

MESH PROCESSING – UPSAMPLING-SUBDIVISION



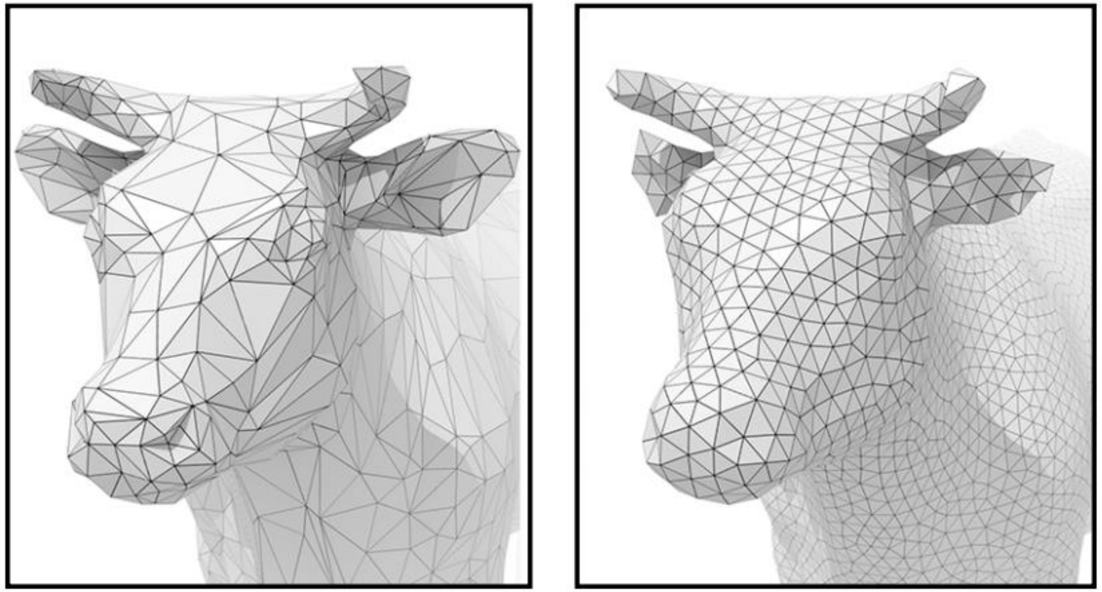
Aumentare la risoluzione tramite interpolazione

Mesh Downsampling – Simplification



Diminuire la risoluzione, mantenendo la forma e l’apparenza.

Mesh Regularization



Modificare la distribuzione dei campioni per migliorare la qualità.