

Grafica al calcolatore - Computer Graphics



5 - Modellazione

Modellazione

- Definiamo ora possibili strutture dati per modellare gli oggetti nello spazio.
- Poi vedremo come modellare anche la formazione delle immagini attraverso il “rendering”

mondo reale,
modello matematico,
artista 3D ...

acquisizione 3D (scansione)
simulazione (elementi finiti)
modellazione (3DStudioMax, Maya)

Geometria

rendering

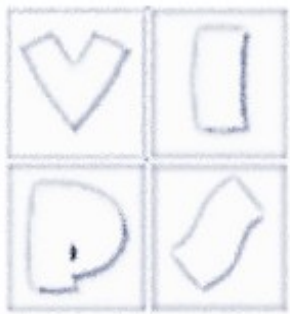
Immagine/i

preprocessing
(*modelling*)

Grafica 2015

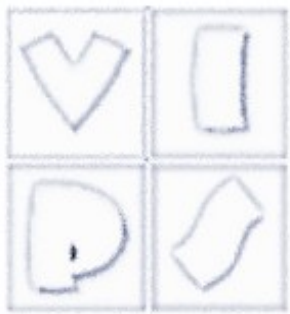
applicazione interattiva

Geometria analitica



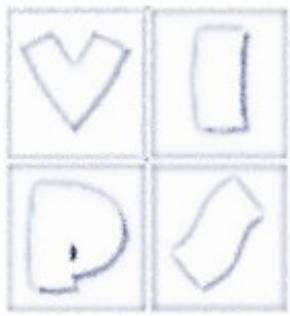
- Premessa: prima di vedere come si modellano gli oggetti, ricordiamo come si definiscono nello spazio Euclideo 3D le figure geometriche importanti dal punto di vista della modellazione grafica e del rendering
 - Rette: sono identificabili da un punto qualsiasi Q che giaccia sulla retta e da una direzione data da un vettore u . È facile vedere che sono il luogo dei punti dati da $P = Q + tu$ $t \in \mathbb{R}$
 - In termini di componenti si vede facilmente che vale la seguente equazione
$$\frac{x - x_Q}{u_x} = \frac{y - y_Q}{u_y} = \frac{z - z_Q}{u_z}$$
 - Se si vuole specificare una retta dati due punti R e Q , basta usare le formule date qui sopra tenendo conto che il vettore che identifica la retta è dato da $u = (R - Q) / |R - Q|$

Geometria analitica



- Semiretta: basta aggiungere il vincolo $t \geq 0$
- Segmenti: dati i punti iniziale e finale P e Q possiamo scriverli come $P = Q + t(R-Q)$ $t \in [0; 1]$
- Sfere: dato centro O e raggio r , i punti della superficie sferica sono dati dall'equazione $P = O + r\mathbf{u}$ con \mathbf{u} versore generico.
- Si dimostra facilmente che, in termini delle coordinate, la superficie sferica è data dall'equazione
$$(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 = r^2$$
- Piani: dati 3 punti non allineati P , Q ed R il luogo dei punti che descrive il piano che li comprende è la combinazione affine
$$S = \alpha P + \beta Q + \gamma R \quad \alpha, \beta, \gamma \in \mathbb{R} \quad \alpha + \beta + \gamma = 1$$
- Alternativamente si può definire un piano a partire da un punto Q che vi appartiene e da un vettore \mathbf{u} che ne identifica la normale come il luogo dei punti P tali che $(P - Q) \cdot \mathbf{u} = 0$

Geometria analitica



In termini di coordinate abbiamo

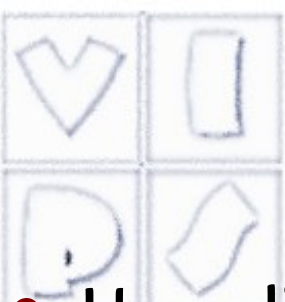
$$(x-x')u_x + (y-y')u_y + (z-z')u_z = 0$$

Per passare dalla prima alla seconda rappresentazione basta prendere come punto Q e come vettore $\mathbf{u} = (P-Q) \times (R-Q)$

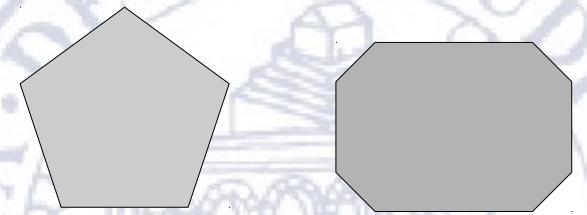
- Semispazi: il piano di cui sopra identifica due semispazi, uno positivo ed uno negativo:

$$(P-Q) \cdot \mathbf{u} > 0$$

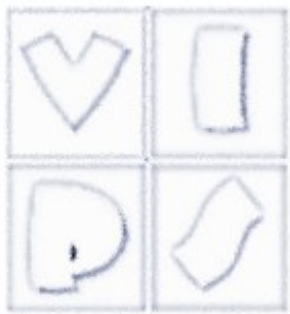
$$(P-Q) \cdot \mathbf{u} < 0$$



Poligoni

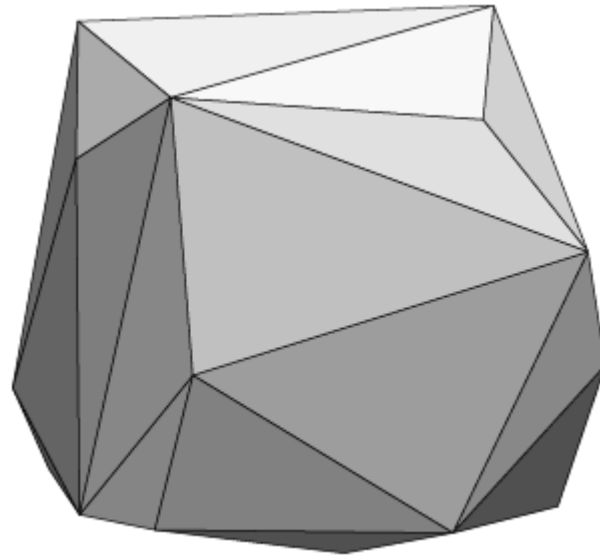


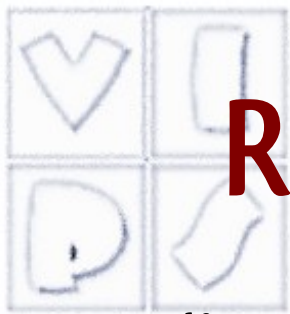
- Un poligono P è un insieme finito di segmenti (spigoli) di \mathbb{R}^2 , in cui ogni estremo (vertice) è comune a esattamente due segmenti, che si dicono adiacenti.
- Un poligono è detto semplice se ogni coppia di spigoli non adiacenti ha intersezione vuota.
- Teorema di Jordan: Un poligono semplice P divide il piano in due regioni o facce, una limitata (detta interno di P) ed una illimitata (detta esterno di P).
- Per convenzione, un poligono viene rappresentato dalla sequenza dei suoi vertici $P_1 \dots P_n$ ordinati in modo che l'interno del poligono giaccia alla sinistra della retta orientata da P_i a P_{i+1} , ovvero i vertici sono ordinati in senso antiorario.



Poliedri

In \mathbb{R}^3 un poliedro semplice è definito da un insieme finito di poligoni (facce) tali che ciascuno spigolo di una faccia è condiviso da esattamente un'altra faccia e le facce non si intersecano che negli spigoli.





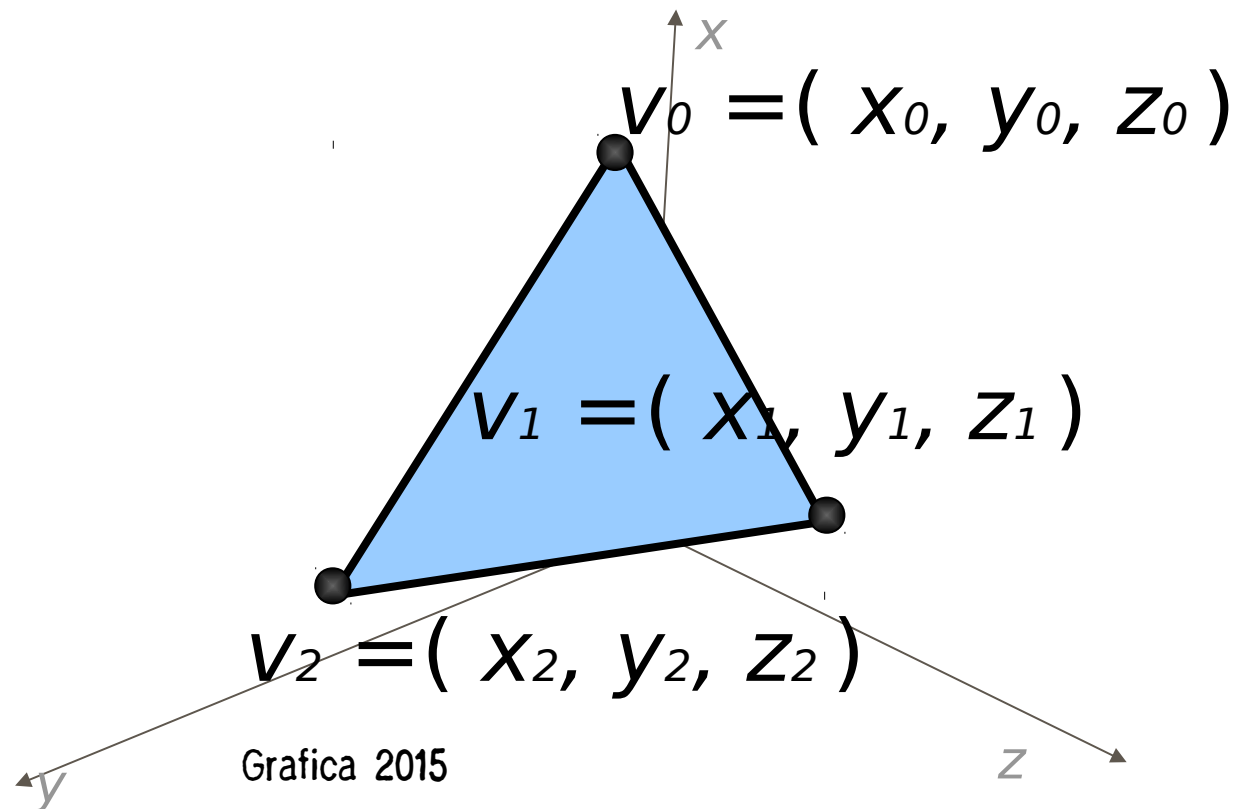
Rappresentazione degli oggetti

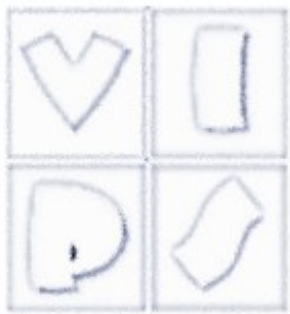
- Gli oggetti che si vogliono rappresentare in una applicazione grafica hanno di solito caratteristiche particolari
 - Sono finiti
 - Sono chiusi (non sempre)
 - Sono continui
- Le rappresentazioni di oggetti (regioni dello spazio, in generale) si suddividono in
 - basate sul contorno (boundary): descrivono una regione in termini della superficie che la delimita (boundary representation, o b-rep).
 - basate sullo spazio occupato (o volumetriche).



Rappresentazione comune: maglie (mesh) di triangoli

- E' la rappresentazione più usata nella grafica interattiva per motivi di efficienza
 - La pipeline delle schede grafiche è ottimizzata per il rendering dei triangoli





Esistono però alternative

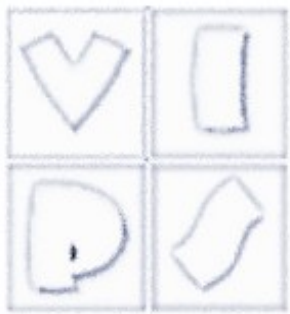
- **Boundary**
 - Superfici parametriche (lisce, non hanno problemi di “tessellazione” cioè visibilità degli spigoli tra le facce)
- **Volumetriche**
 - Rappresentazione “voxellizzata” (a cubetti)
 - Geometria costruttiva solida
- **Image based rendering:**
 - Non si modella effettivamente la scena, ma si memorizzano campionamenti della luce, rendendo però possibile una visualizzazione da più punti di vista, interattiva
 - Light fields



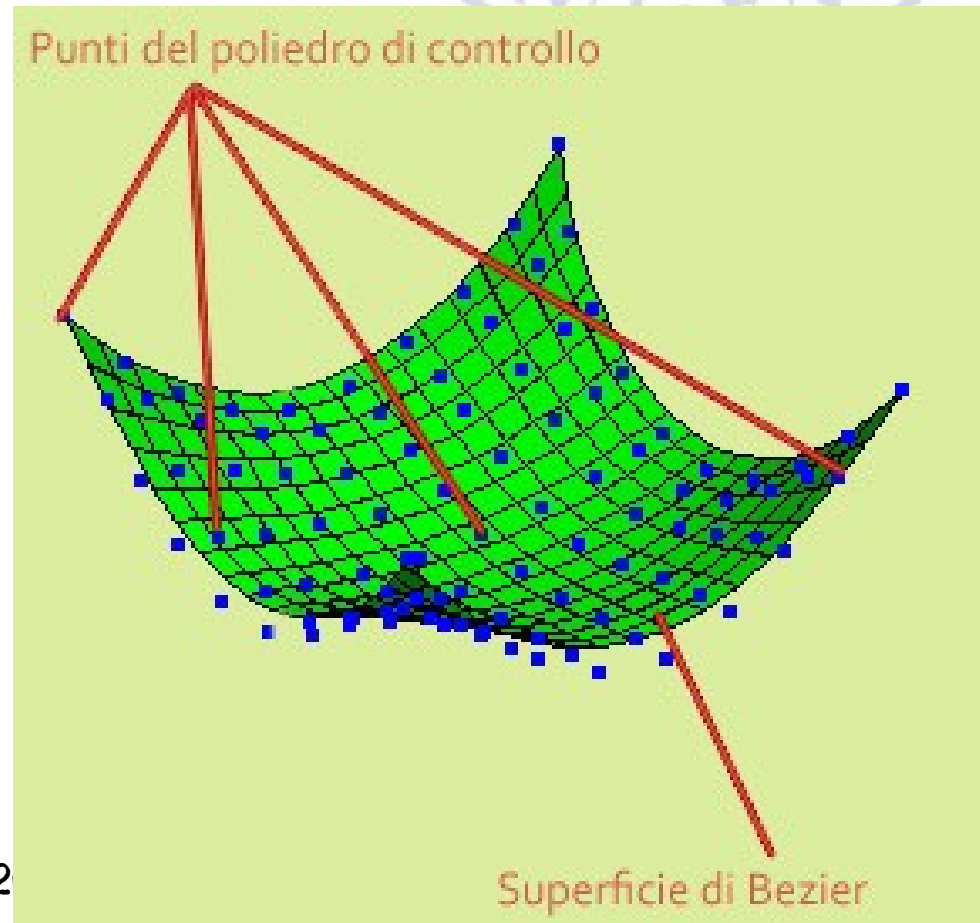
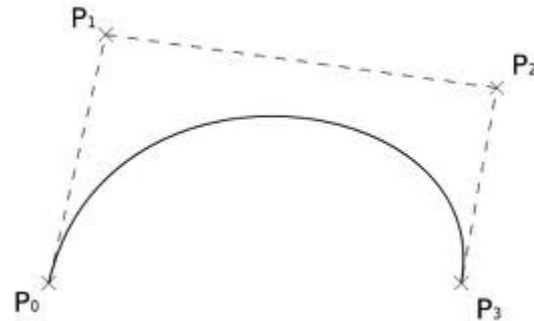
Curve e superfici lisce

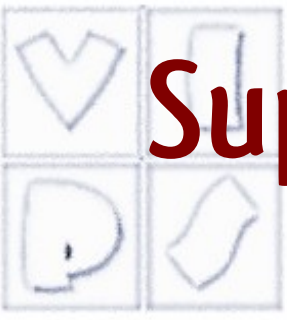
- Il vantaggio principale nell'uso di superfici per modellare un oggetto sarebbe l'assenza del problema della tessellazione visibile (cioè approssimo una superficie liscia coi triangoli, ma vedo poi i triangoli evidenti, effetto ridotto di solito con trucchi opportuni nel rendering)
- Uso di superfici parametriche pesante per applicazioni in tempo reale; per lo più utilizzate in fase di modellazione o per rendering non interattivo
- Negli ultimi tempi le cose sono cambiate ed oggi cominciano ad apparire applicazioni di grafica avanzata che usano superfici curve anche in tempo reale

Curve e superfici lisce

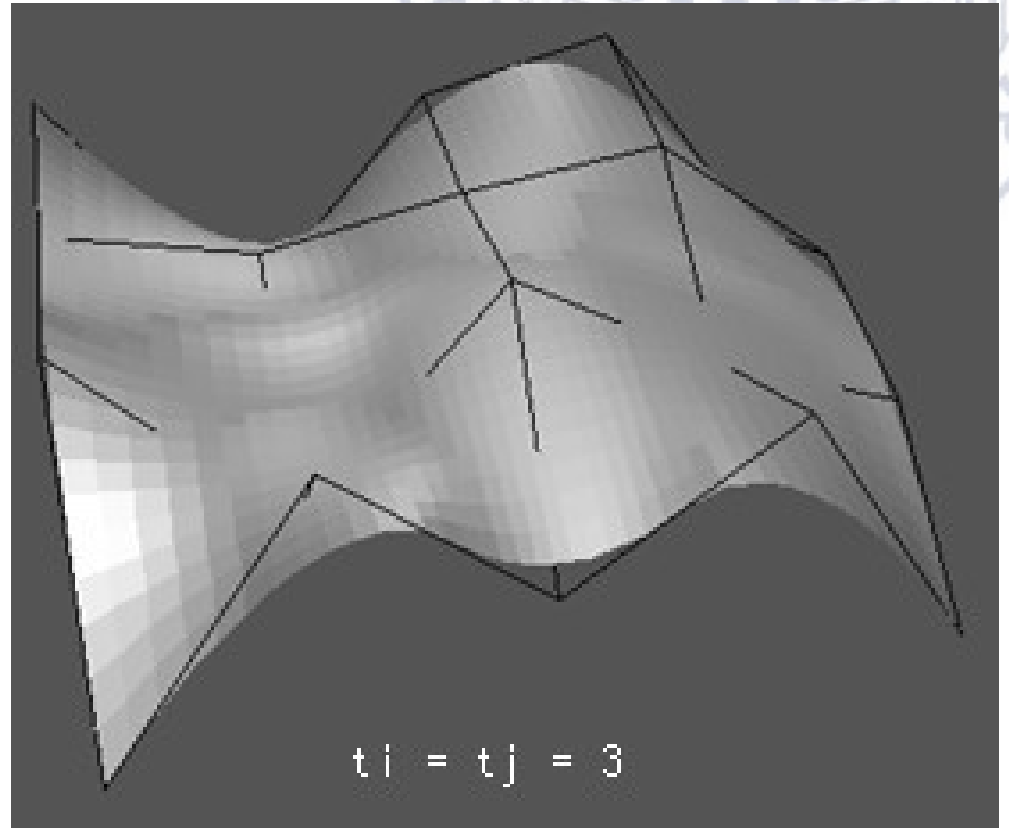
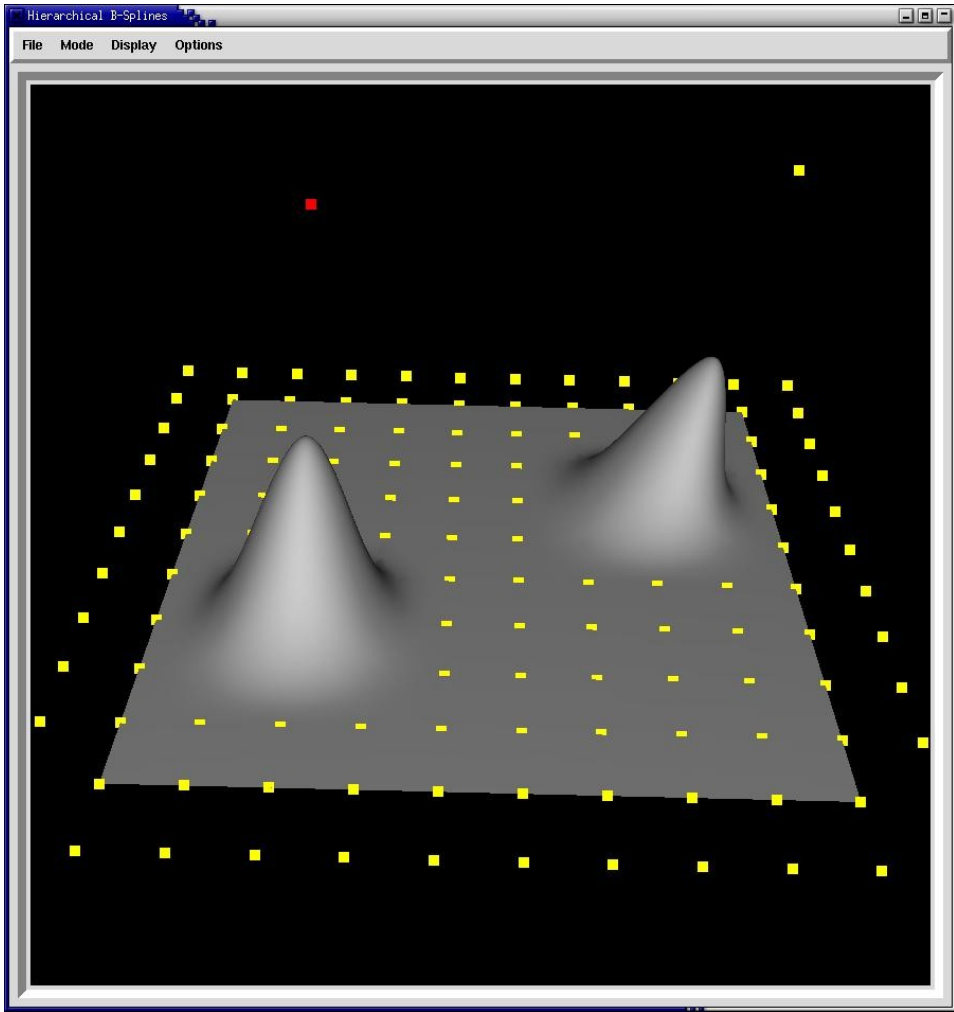


- Esempio: curve/superfici di Bezier
 - Dati N punti di controllo la curva passa per il primo e l'ultimo e approssima gli altri con una funzione da essi dipendente
 - Con una griglia si generano superfici





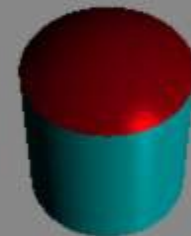
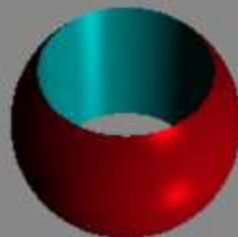
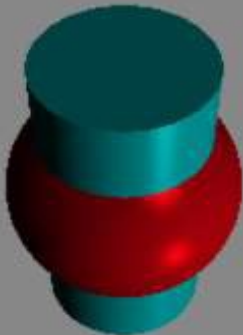
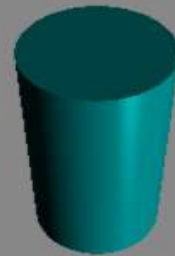
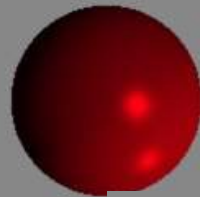
Superfici parametriche - Esempi

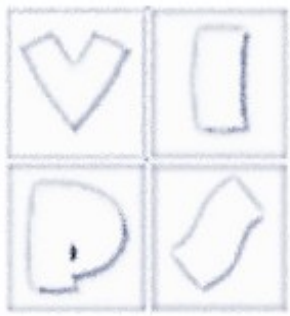




Geometria costruttiva solida (CSG)

- Altra rappresentazione particolarmente adatta per il modeling (diffusa nel settore CAD), ma poco efficiente per il rendering.
- Si tratta, essenzialmente, di costruire degli oggetti geometrici complessi a partire da modelli base con operazioni booleane





Operazioni

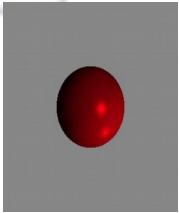
- Unione: l'unione $A \cup B$ è l'insieme dei punti che appartengono ad almeno uno dei due solidi (or non esclusivo)
- Differenza: la differenza $A - B$ è l'insieme dei punti che appartengono ad A , ma non a B
- Intersezione: l'intersezione $A \cap B$ è l'insieme dei punti che appartengono ad A ed a B (and)
- Le operazioni CSG possono essere descritte tramite un albero (gerarchia)
- Ciascun nodo di un albero che non sia una foglia contiene una delle tre operazioni elementari \cup , \cap o $-$
- Ciascuna foglia contiene una primitiva



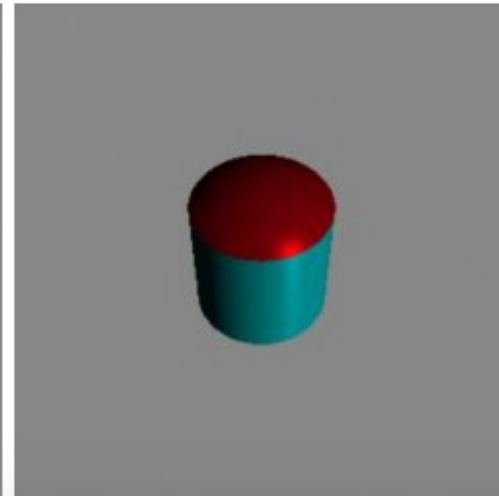
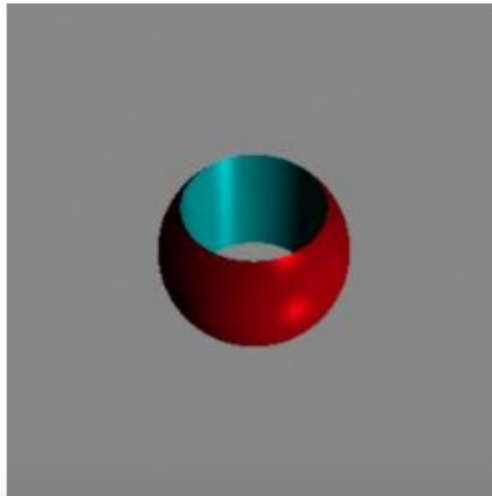
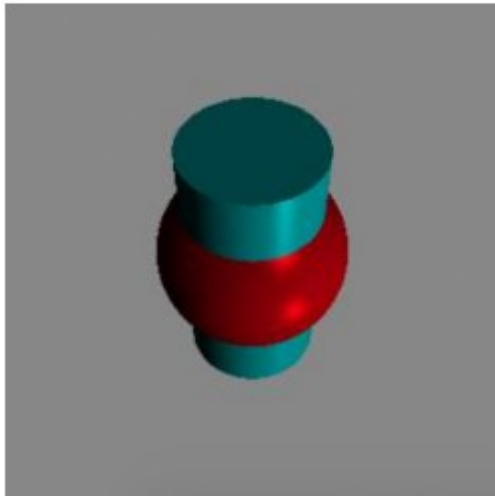
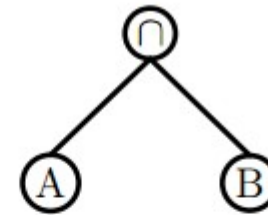
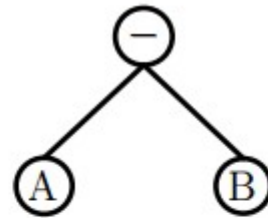
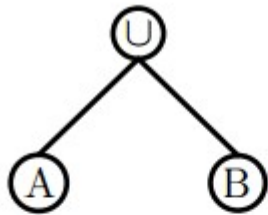
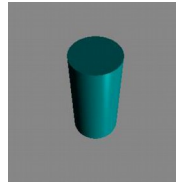
Esempio

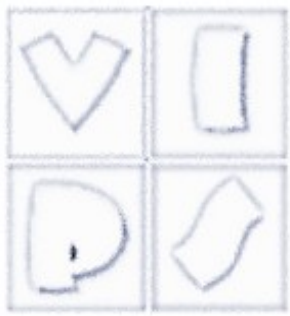


• A



B





Geometria costruttiva solida

- Riassumendo: operazioni Unione, Intersezione, Differenza
- Forme base (primitive) tipicamente cuboidi, sfere, cilindri, piramidi, prismi (dipende dal software)
- Possibile creare / modificare in modo procedurale i modelli
- I software possono in genere esportare poi modelli triangolati per il rendering interattivo
- Garanzia sul fatto che la superficie risultante rappresenti effettivamente il bordo di un oggetto 3D (sia watertight)
- Supportata da molte tipologie di software (CAD, modellazione per giochi, rendering non interattivo, ecc.)

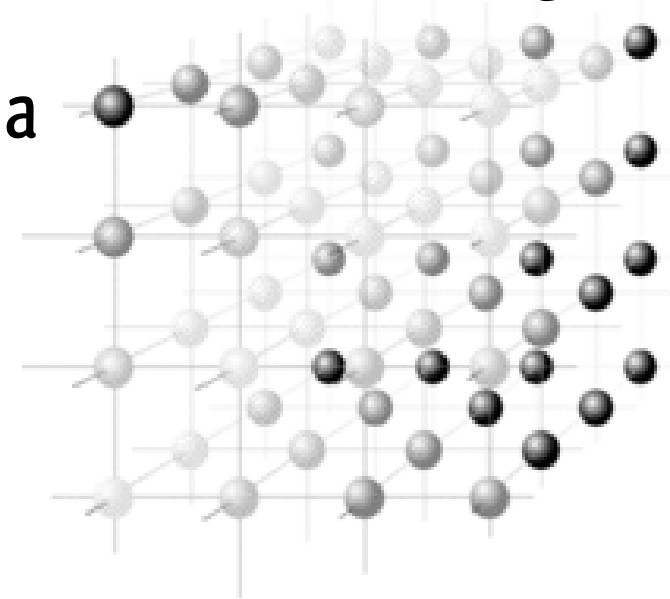


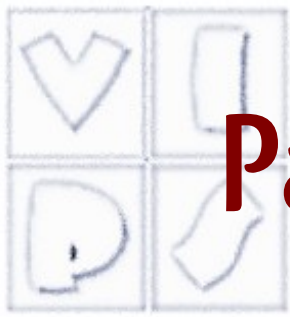
Acquisizione dal vero: point clouds

- Un modo per generare modelli 3D per mondi virtuali è acquisire dal vero
- Scansione 3D, oggi tecnologia matura con diverse tecnologie
 - Laser
 - Luce strutturata
 - Visione computazionale
- Gli scanner (esattamente come le macchine fotografiche in 2D) di fatto non acquisiscono un modello del mondo, ma campionano la geometria (con eventuali attributi, es. colore) in una serie di punti discreti: si parla di **nuvole di punti (point clouds)**

Partizionamento spaziale (voxel)

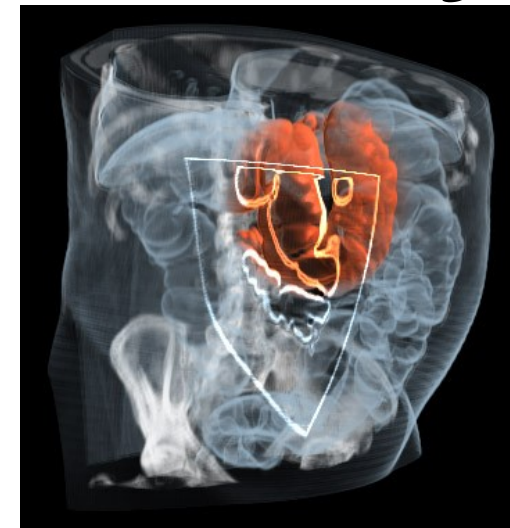
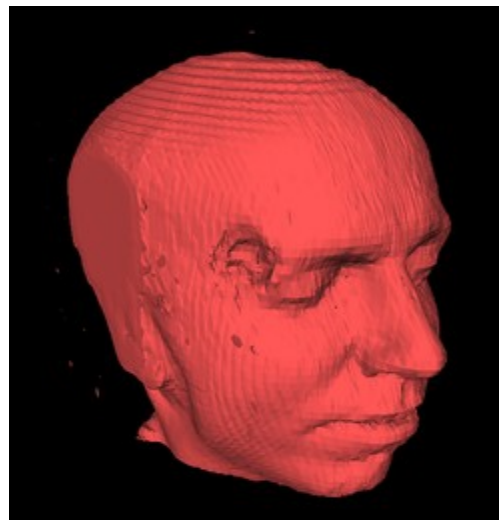
- Lo spazio viene suddiviso in celle adiacenti dette in 3D voxel (equivalente dei pixel delle immagini): una cella è “piena” se ha intersezione non vuota con la regione, è detta vuota in caso contrario. Oppure contiene un valore di densità (tipico dei dati diagnostici es. TAC)
- Una rappresentazione di una scena complessa ad alta risoluzione richiederebbe l’impiego di un numero enorme numero di voxel, per cui questa rappresentazione è in genere limitata a singoli oggetti.
- Ma le cose stanno cambiando grazie a progressi nell’HW e nel SW

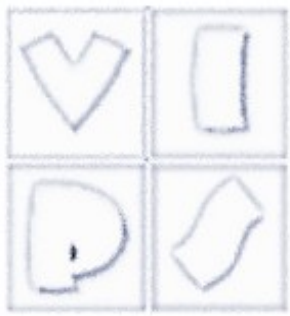




Partizionamento spaziale (voxel)

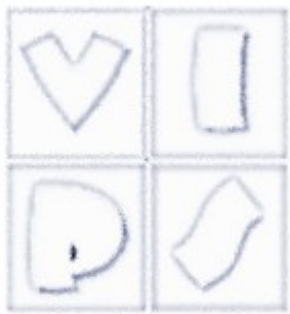
- Rendering di modelli voxel-based:
 - algoritmi ad hoc (tecniche direct volume rendering)
 - conversione da voxel a rappresentazione per superfici (triangle-based)
- Da una rappresentazione volumetrica voxelizzata si può passare efficientemente a una rappresentazione poligonale della superficie mediante l'algoritmo detto marching cubes.



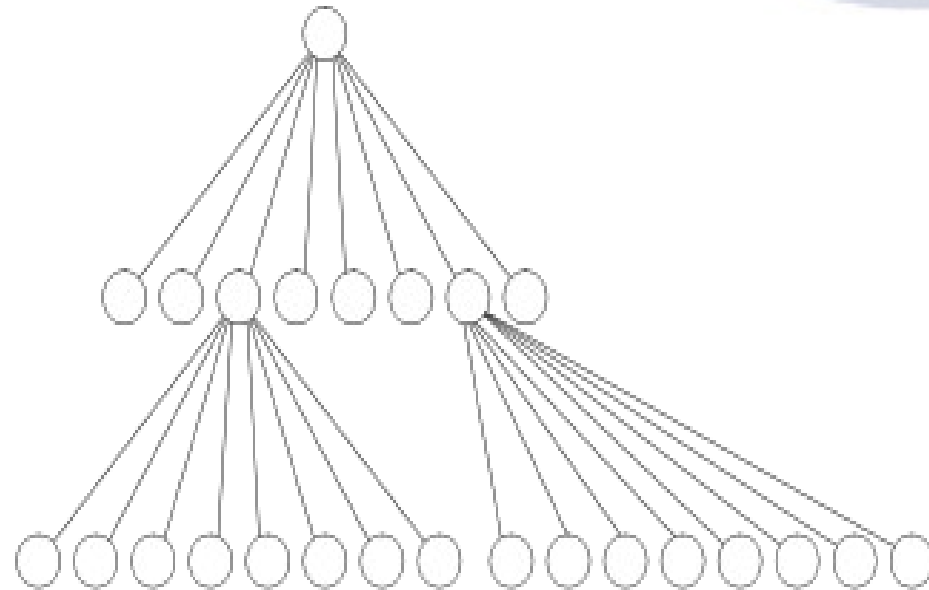
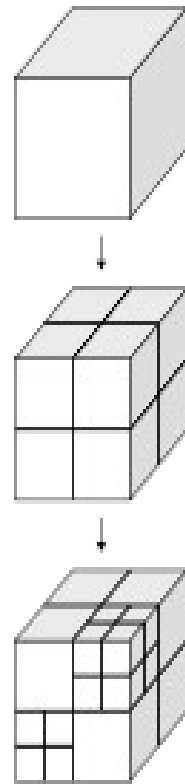


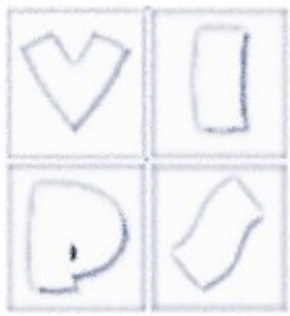
Rappresentazioni compatte

- Se ho solo i valori pieno/vuoto, posso rappresentare in modo compatto il volume con una struttura octree
- Si parte con un cubo contenente la regione e si suddivide ricorsivamente. Ci si ferma ogni volta che un ottante contiene tutte celle piene o tutte celle vuote.
- Più economica rispetto alla enumerazione delle singole celle, poiché grandi aree uniformi (piene o vuote) vengono rappresentate con una sola foglia (anche se nel caso peggiore il numero delle foglie è pari a quello delle celle)



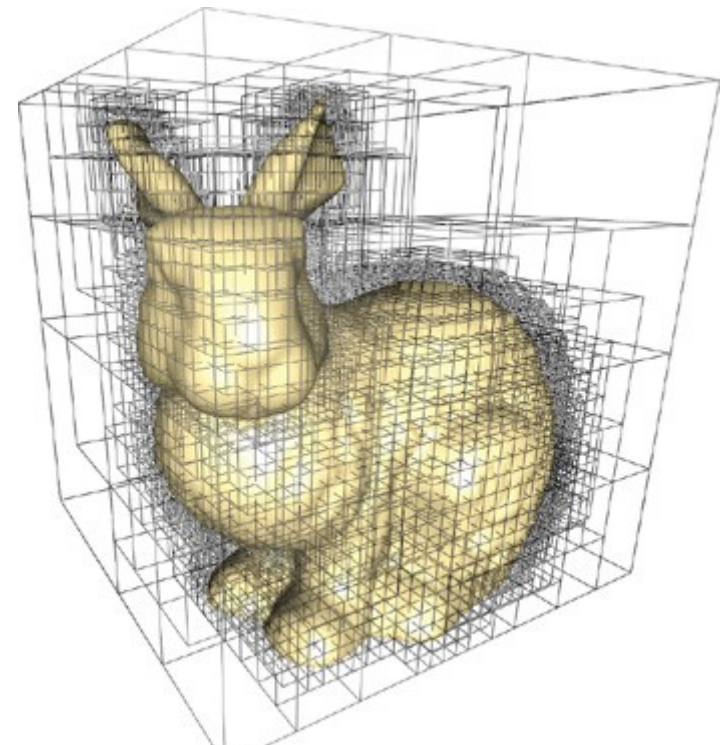
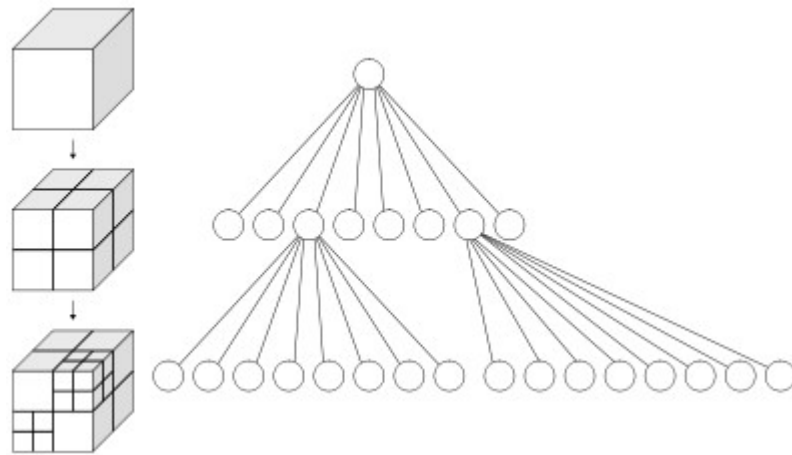
Octree

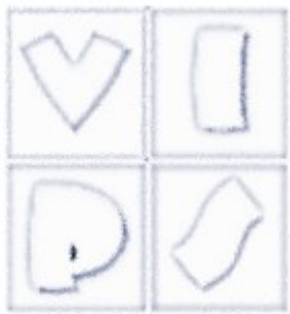




Octree

- Le strutture di partizionamento spaziale sono anche utili per “contenere” le geometrie poligonali: come vedremo consente di rendere più semplice la ricerca di intersezioni tra oggetti e con i raggi ottici. Vedremo



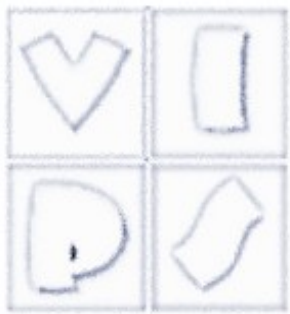


Maglie poligonali (mesh)



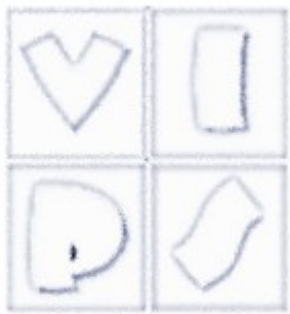
Poligoni e triangoli

- Nella grafica 3D interattiva si usa quasi sempre la modellazione basata su approssimazione poligonale degli oggetti (del loro contorno: è una *boundary representation*).
- Si tratta di approssimare una superficie 2D con un insieme di poligoni convessi opportunamente connessi gli uni agli altri.
- Nella pipeline di rendering si lavora in genere con i soli triangoli
 - tutte le altre rappresentazioni eventualmente usate nel programma sono convertite prima del rendering in triangoli
- Possiamo usare la geometria per definire rigorosamente le proprietà dei modelli triangolati



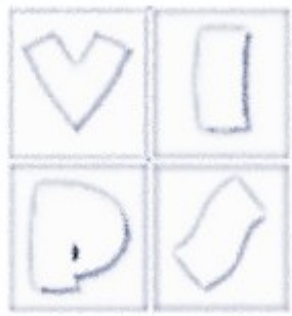
Varietà

- Una varietà k -dimensionale X è un sottoinsieme di \mathbb{R}^d in cui ogni punto ha un intorno omeomorfo alla sfera aperta di \mathbb{R}^k .
- In generale le superfici degli oggetti solidi (sfere, poliedri, ecc.) sono varietà bidimensionali.
- Omeomorfismo: applicazione biiettiva, continua, con inversa continua. Intuizione: trasformazione senza "strappi".
 - In una varietà k -dimensionale **con bordo** ogni punto ha un intorno omeomorfo alla sfera aperta o alla semisfera aperta di \mathbb{R}^k .
 - Il bordo di X è l'insieme dei punti che hanno un intorno omeomorfo alla semisfera aperta.
 - Una varietà è sempre una varietà con bordo, eventualmente vuoto.
 - Il bordo, se non è vuoto, è a sua volta una varietà $k-1$ dimensionale senza bordo.

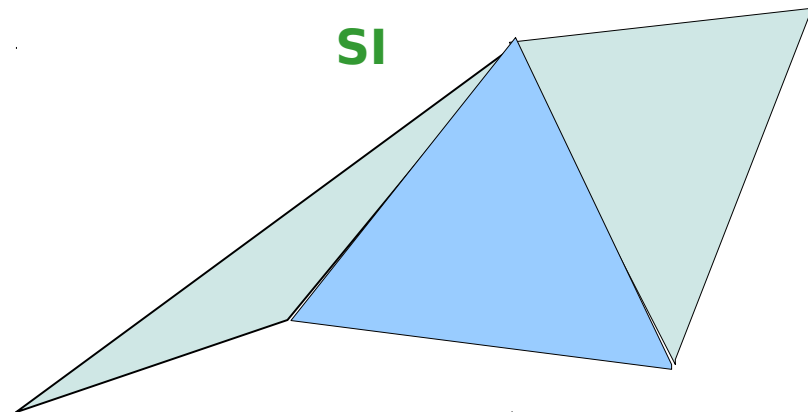
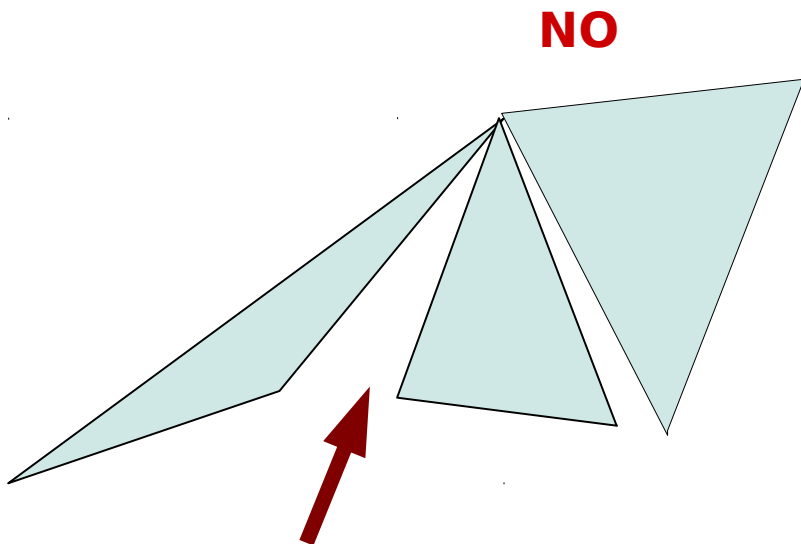
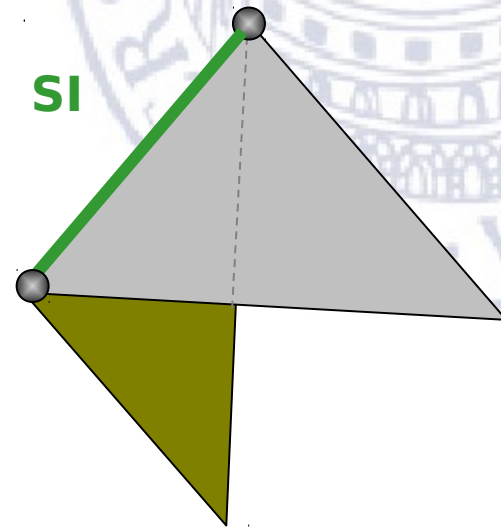
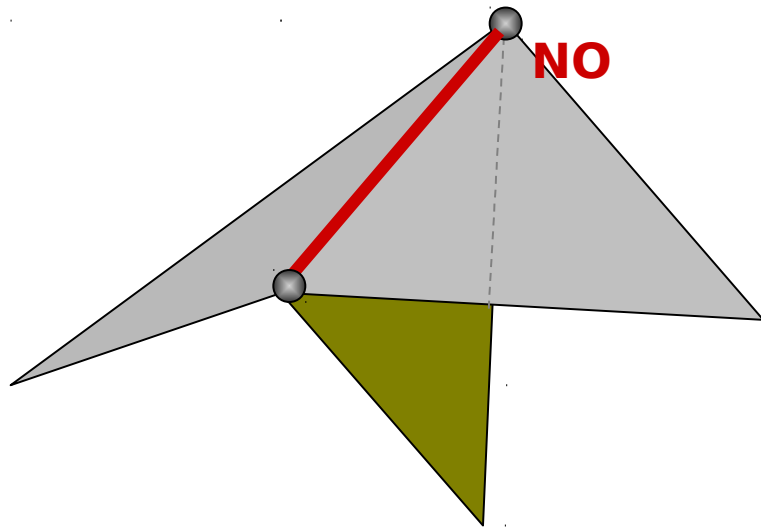
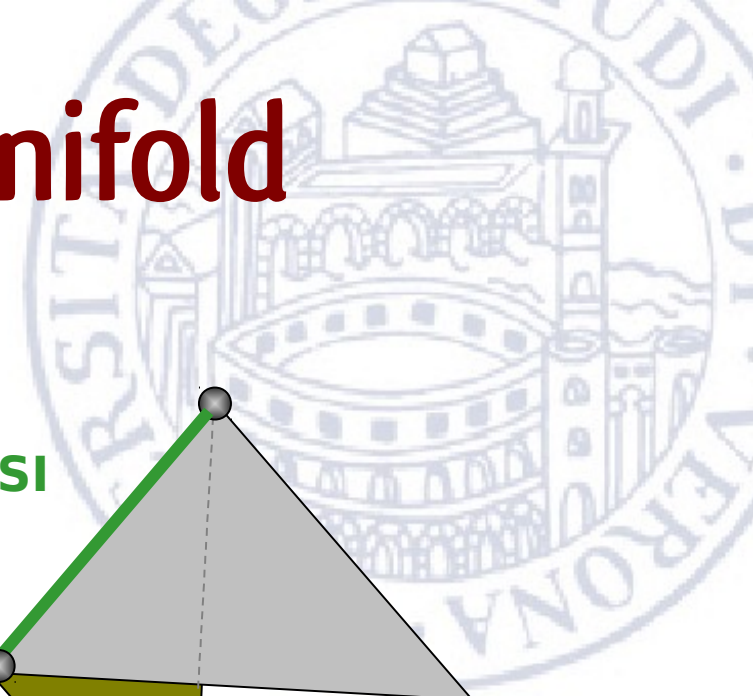


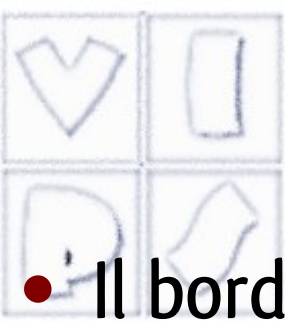
Mesh poligonali

- Una maglia (mesh) triangolare è 2-complesso simpliciale puro che è anche una varietà bidimensionale con bordo.
- I triangoli della maglia si chiamano anche facce.
- La condizione di essere varietà si traduce nei seguenti vincoli ulteriori sulla struttura del complesso simpliciale:
 - uno spigolo appartiene al massimo a due triangoli (quelli eventuali che appartengono ad uno solo formano il bordo della maglia)
 - se due triangoli incidono sullo stesso vertice allora devono appartenere alla chiusura transitiva della relazione di 1-adiacenza, ovvero devono formare un ventaglio o un ombrello.
- Si usa il termine condizione 2-manifold (varietà)



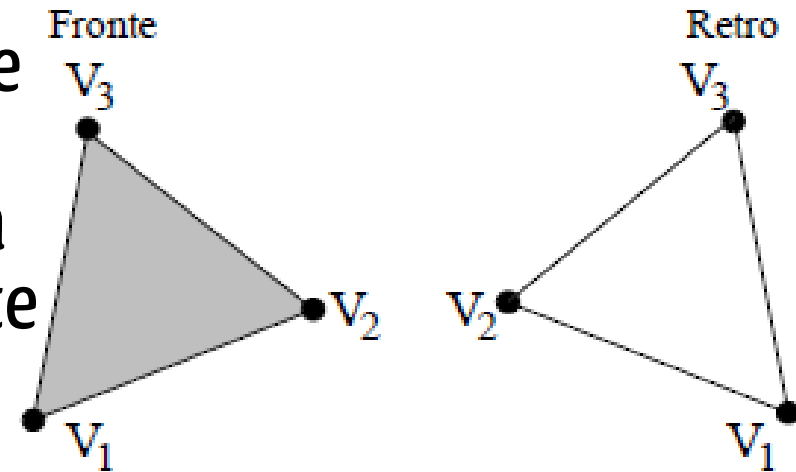
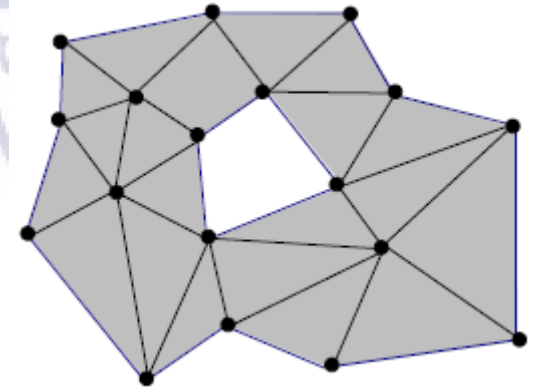
Condizione 2-manifold

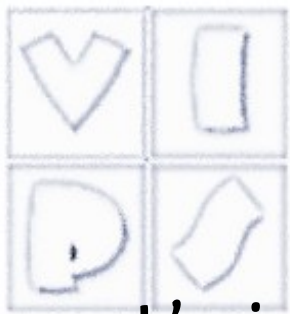




Orientazione

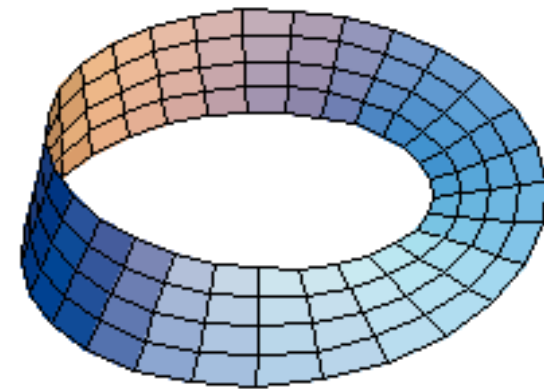
- Il bordo della maglia consiste di uno o più anelli (sequenza chiusa di spigoli) o loop.
- Se non esistono spigoli di bordo la maglia è chiusa (come quelle che rappresentano la superficie di una sfera).
- L'orientazione di una faccia è data dall'ordine ciclico (orario o antiorario) dei suoi vertici incidenti. L'orientazione determina il fronte ed il retro della faccia. La convenzione (usata anche da OpenGL) è che la faccia mostra il fronte

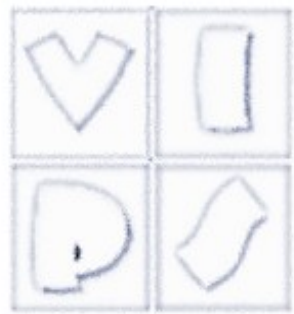




Mesh orientabili

- L'orientazione di due facce adiacenti è compatibile se i due vertici del loro spigolo in comune sono in ordine inverso. Vuol dire che l'orientazione non cambia attraversando lo spigolo in comune.
- La maglia si dice orientabile se esiste una scelta dell'orientazione delle facce che rende compatibili tutte le coppie di facce adiacenti.
 - Non tutte le mesh 2-manifold sono orientabili (es. anello di Moebius)

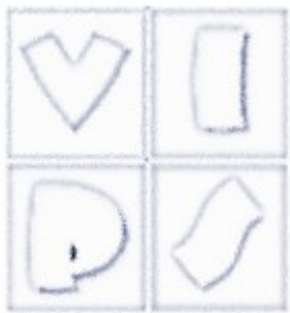




Maglie/Mesh generiche

- Abbiamo definito la maglia triangolare. In maniera analoga si può estendere la definizione a maglie poligonali generiche
 - Maglie poligonali generiche: i poligoni possono avere qualsiasi numero di spigoli e non è detto che ci sia un solo tipo di poligono. Sono raramente utilizzate in grafica al calcolatore
 - Quadrangolari (quad meshes): gli elementi poligonali sono tutti quadrilateri. Sono alle volte usate, per esempio se si vuole fare il rendering di un terreno descritto da un array di altezze. In una maglia quadrangolare bisogna imporre un vincolo aggiuntivo di planarità per ogni quadrilatero che la compone.
- OpenGL consente di descrivere maglie poligonali generiche, ma per disegnarle li suddivide usualmente in triangoli.

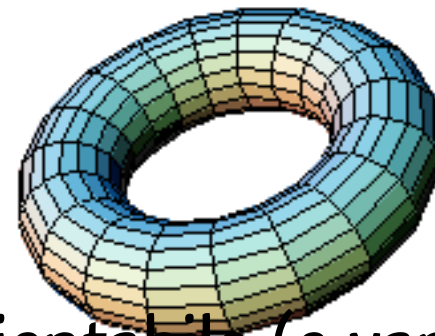
Equazione di Eulero



Se V è il numero di vertici, L il numero di spigoli ed F il numero di facce della maglia poligonale orientabile chiusa di genere G , allora vale la Formula di Eulero $V - L + F = 2 - 2G$

Una superficie ha genere G se può essere tagliata lungo G linee semplici chiuse senza disconnetterla (intuitivamente, ma non rigorosamente “numero di buchi”)

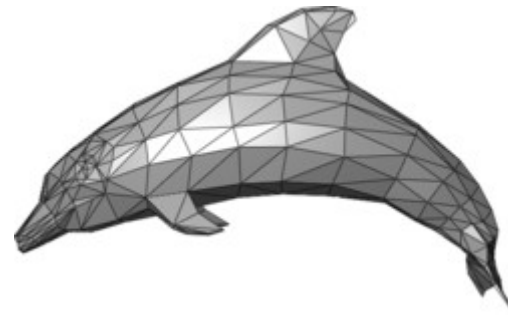
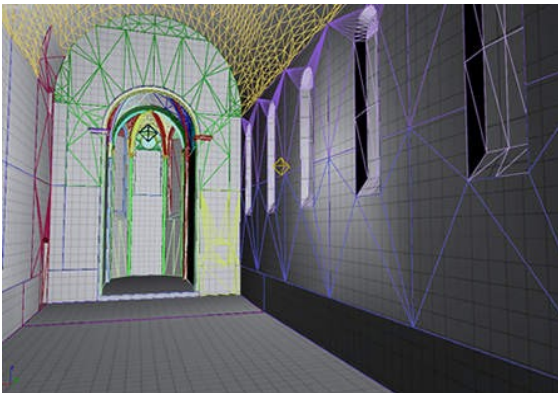
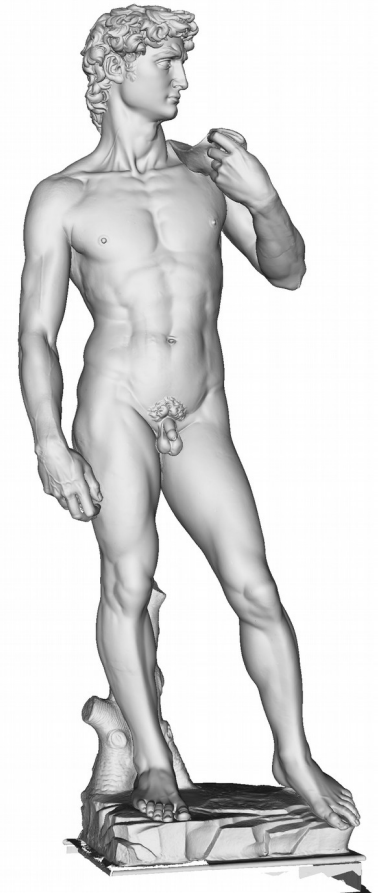
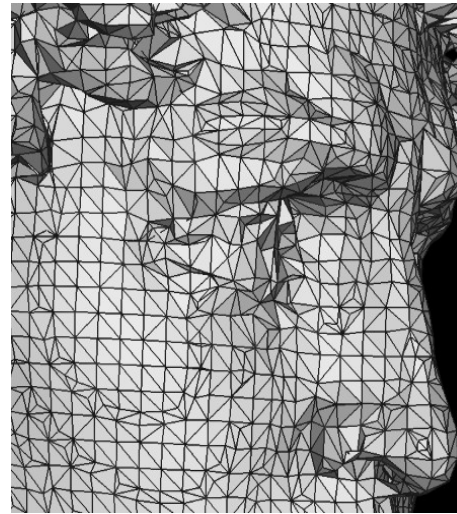
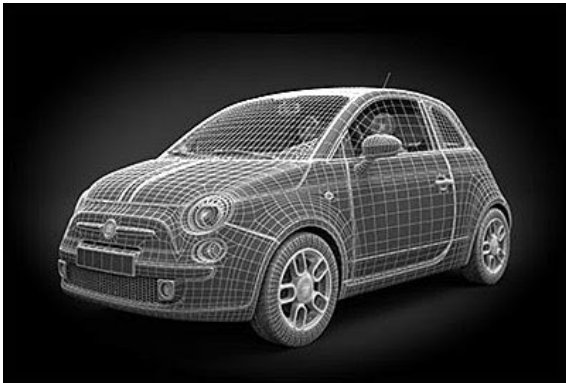
- Il genere di una superficie determina la sua topologia; per una sfera, per esempio, $G = 0$, mentre per un toro (una ciambella) $G = 1$.



- Più in generale, per una maglia poligonale orientabile (e varietà bidimensionale) vale la formula $V - L + F = 2(S - G) - B$
 - S numero di componenti connesse, B è il numero di anelli di bordo

Mesh di triangoli

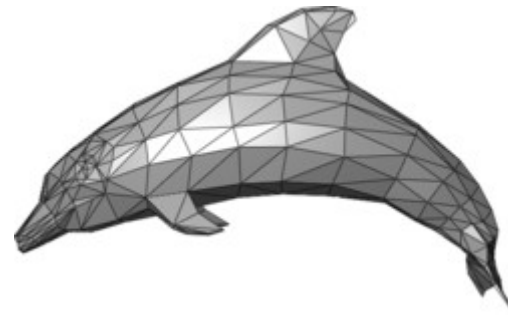
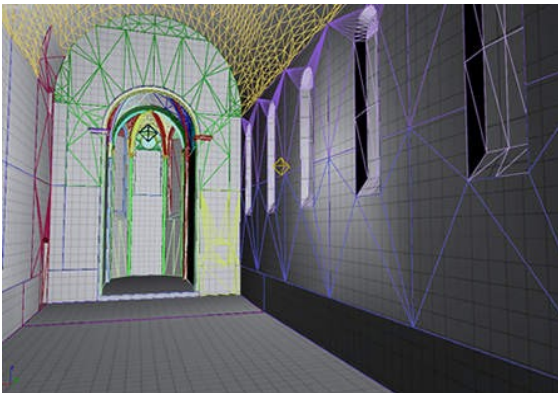
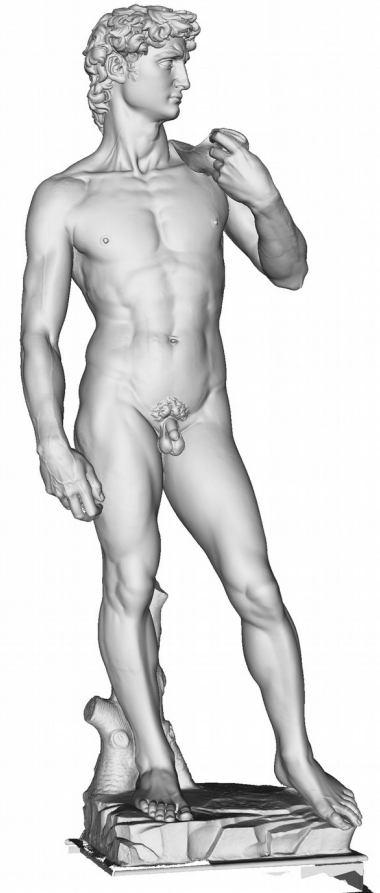
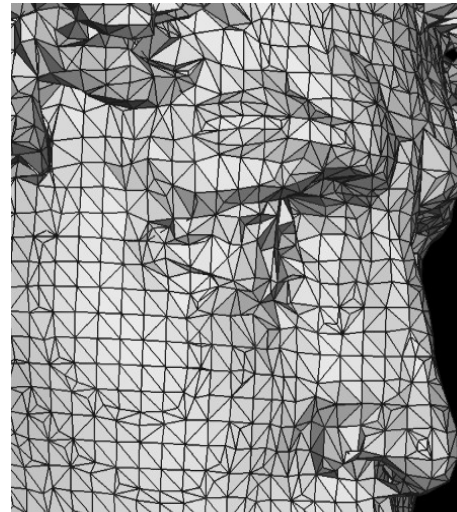
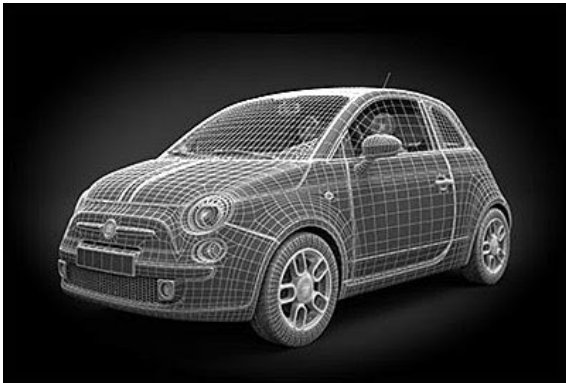
- Nella pratica sono il tipo di modello dominante
- Usato nella gran parte delle applicazioni interattive
 - Dato che il rendering è ottimizzato in hardware



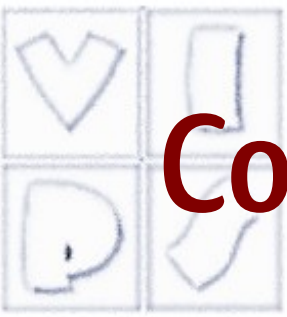
Gratica 2015

Mesh di triangoli

- Generate da modellazione CAD, acquisizione con scanner, ricostruzione da immagini (Computer Vision)
- Il numero di poligoni determina il dettaglio, ma il costo in memoria può essere notevole



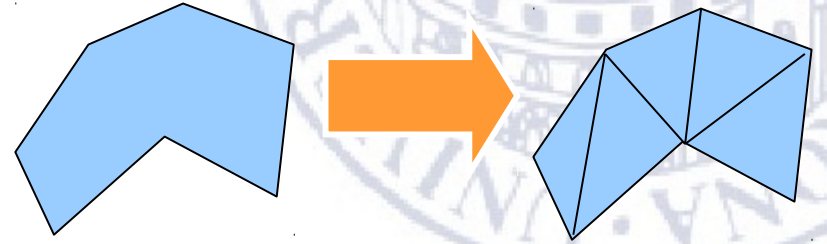
Gratica 2015



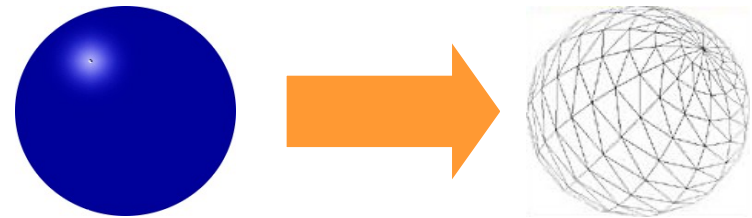
Costruzione della mesh triangolare

- Conversione da altri formati:

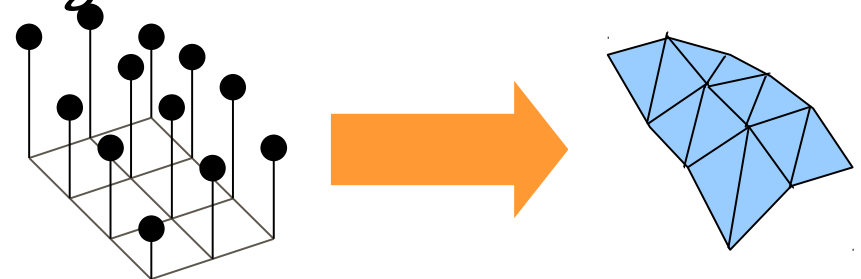
- Poligoni → Triangoli

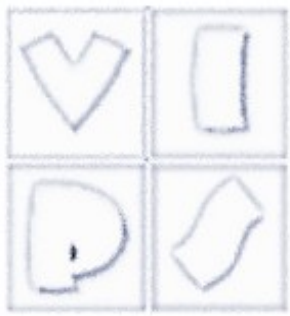


- Superf. Quadriche → Triangoli



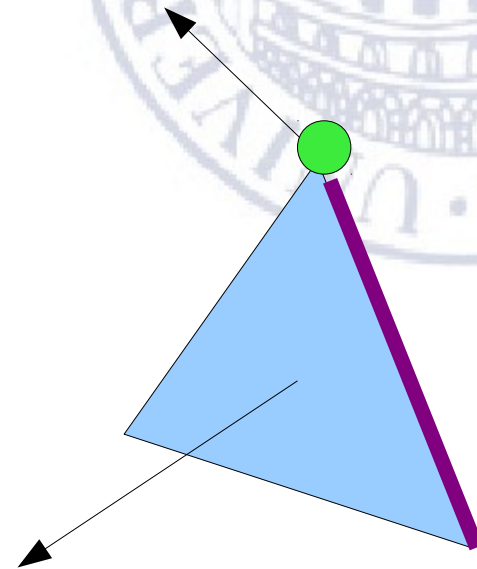
- Campi di altezze o Punti → Triangoli



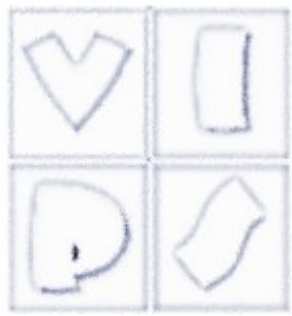


Mesh triangolare - Attributi

- Posso definirli:
 - per vertice
 - esplicito un attributo per ogni vertice
 - per faccia
 - esplicito un attributo per ogni faccia
 - per wedge (vertice di faccia)
 - esplicito tre attributi per ogni faccia
- Attributi più comuni:
 - colore
 - coordinate texture
 - normali



Mesh triangolare - Limiti



- Non è sempre semplice modellare le entità da rappresentare con triangoli...
 - Esempi:
 - Nuvole
 - Fiamme
 - Capelli, pelliccia



by Niniane Wang
18/02/16 (non real time)



by N. Adabala Florida Uni
(non real time) Grafica 2015

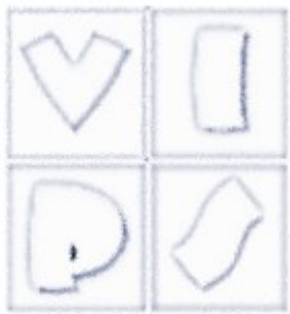


by M. Turitzin and J. Jacobs
Stanford Uni (real time!)

Evitare ridondanze

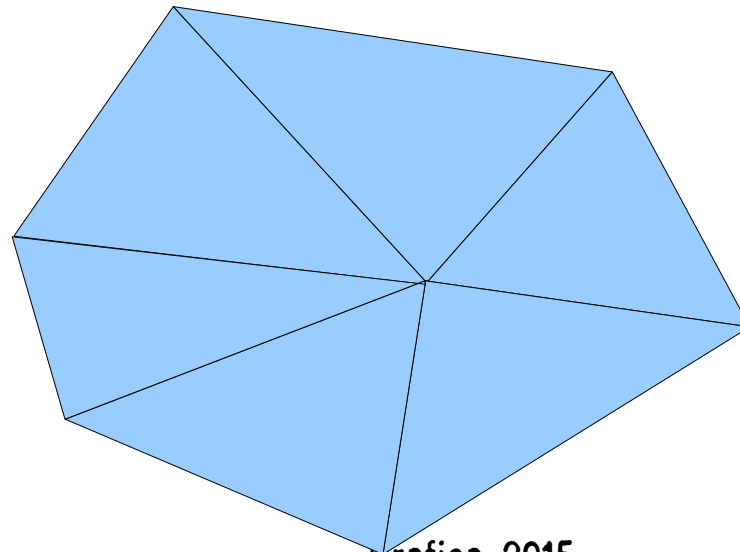
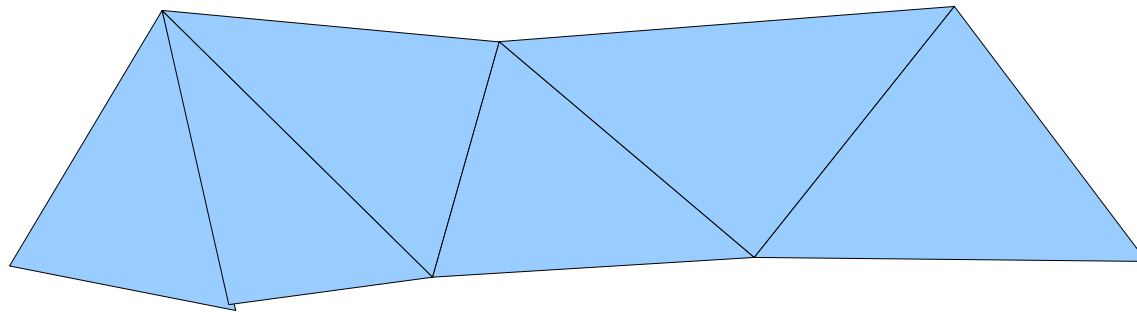


- Quando si devono disegnare due triangoli con uno spigolo in comune, questo viene disegnato due volte. Questo introduce un certo grado di ridondanza che può incidere sulle prestazioni
- Si preferisce quindi raggruppare i triangoli di una maglia in opportuni gruppi che possono essere elaborati in maniere più efficienti. Si possono ad esempio utilizzare
 - **Fan di triangoli:** è un gruppo di triangoli che hanno in comune un vertice. Il primo viene specificato completamente, per i successivi basta dare il nuovo vertice. Efficiente, ma i triangoli che incidono su un vertice sono in genere pochi
 - **Strip di triangoli:** gruppo di triangoli che posseggono a due a due uno spigolo in comune. Di nuovo il primo triangolo viene specificato normalmente, per i successivi basta specificare il nuovo vertice. Meno efficiente, ma le strip in genere contengono più triangoli delle fan



Strip e fan

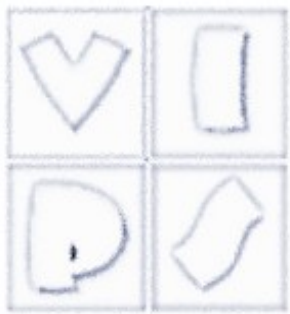
- Esistono algoritmi per creare queste rappresentazioni dalle mesh





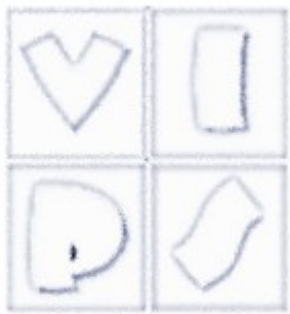
Mesh e rendering

- Per determinare l'effetto di una qualsiasi trasformazione affine su un oggetto (traslazione, rotazione, scalatura, composizioni varie di queste), basta applicare la trasformazione ai vertici (che sono punti); le informazioni connettive date dagli spigoli non cambiano in questo tipo di trasformazioni
 - Questo rende piuttosto semplice il rendering di oggetti descritti in termini di maglie poligonali. qualsiasi trasformazione viene eseguita sui vertici, cioè si tratta di applicare trasformazioni affini su punti
 - L'affermazione precedente è vera anche per la proiezione; per vedere come si proietta la forma di una maglia su un piano (l'immagine), basta seguire la proiezione dei vertici.



Memorizzazione

- Alla creazione dei modelli vengono in genere generati nodi, connettività e attributi. Gli algoritmi di processing e rendering devono accedere in vario modo a tale informazione
- Diversi modi di rappresentare questa informazione
- Nei programmi applicativi sono quindi necessarie delle procedure per convertire una rappresentazione in un'altra
- Progettare ed implementare tali procedure è un ottimo modo per capire nel dettaglio le varie rappresentazioni utilizzate per descrivere maglie poligonali
- Nei disegni e negli esempi ci concentreremo sul caso di maglia triangolare, ma il discorso è valido in generale per tutti i poligoni convessi



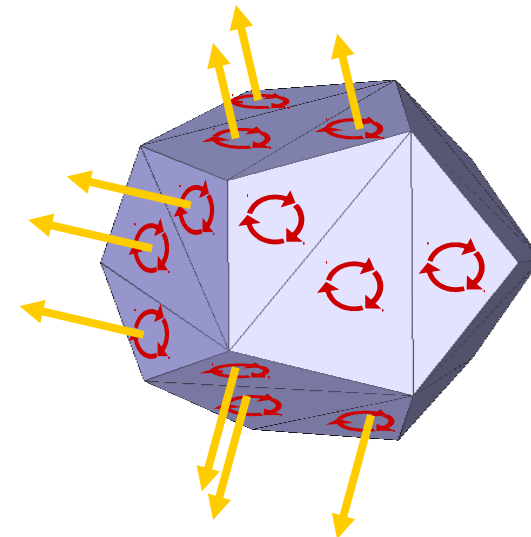
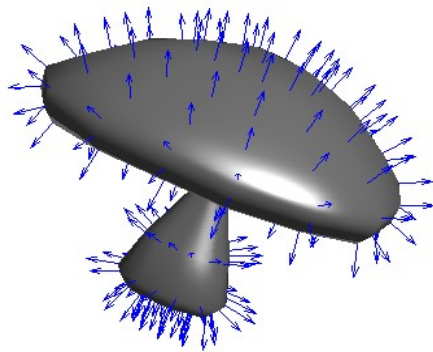
Elementi base

- Vertici: sono gli elementi 0 dimensionali e sono identificabili con punti nello spazio 3D (essenzialmente tre coordinate); alle volte può essere utile associare ai vertici altre caratteristiche oltre alla posizione (tipo il colore)
- Spigoli: sono elementi 1 dimensionali e rappresentano un segmento che unisce due vertici. Di solito non contengono altre informazioni.
- Facce: sono i poligoni bidimensionali, formati da un certo numero di spigoli e di vertici (dimostrare che sono in numero uguale). I vertici o gli spigoli si usano per identificare la faccia; possono contenere altre informazioni (tipo il colore)



Elementi base

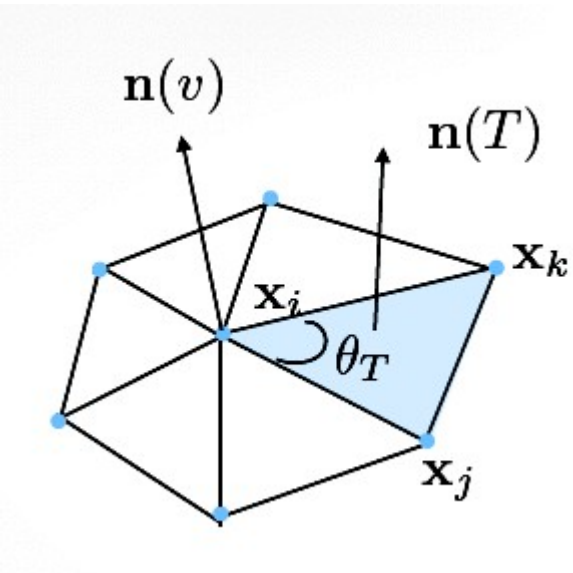
- Normali: è fondamentale sapere quale è l'esterno della superficie e quale l'interno, e qual è l'orientazione locale della superficie; a tal scopo si associa spesso ad una maglia poligonale anche l'informazione sulla normale uscente.
- La normale \mathbf{n} ad una faccia è data dal prodotto vettore di due suoi spigoli consecutivi non collineari
 - attenti al verso: la normale è uscente dal fronte della faccia
 - Per un triangolo (V_1, V_2, V_3) si ha: $\mathbf{n} = (V_3 - V_2) \times (V_1 - V_2)$.



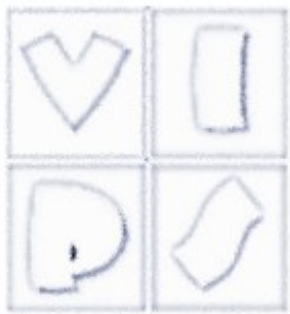
Normali sui vertici

- Posso fare una media (pesata) delle normali delle facce. Pesì?
 - Uniformi? (1)
 - Area? $|T|$
 - Angolo?

$$\mathbf{n}(T) = \frac{(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)}{\|(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)\|}$$

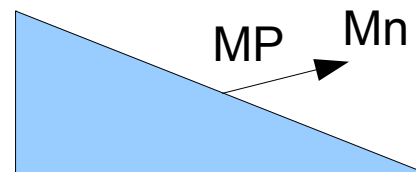
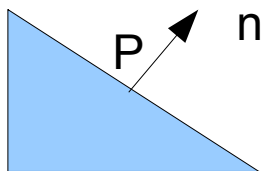


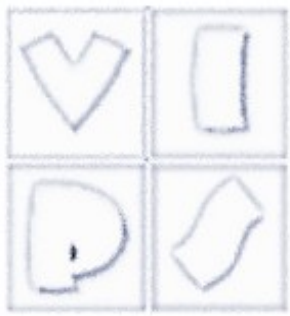
$$\mathbf{n}(v) = \frac{\sum_{T \in \mathcal{N}_1(v)} \alpha_T \mathbf{n}(T)}{\left\| \sum_{T \in \mathcal{N}_1(v)} \alpha_T \mathbf{n}(T) \right\|}$$



Normali e trasformazioni

- Attenzione (vedremo in lab): ci servono le normali per calcolare il modello di Phong. Le possiamo calcolare per i modelli, cui poi sono applicate le trasformazioni geometriche e di proiezione
- Ma se applichiamo trasformazioni generiche ai modelli queste trasformazioni non preservano necessariamente l'ortogonalità di normale e superficie.





Dovremo...

- Per trasformare senza questo problema le normali possiamo trasformarle con una matrice diversa

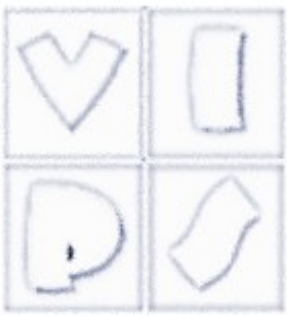
- Consideriamo un vettore tangente t alla superficie

$$\mathbf{n}t^T = 0$$

$$\mathbf{n}M^{-1}Mt = 0$$

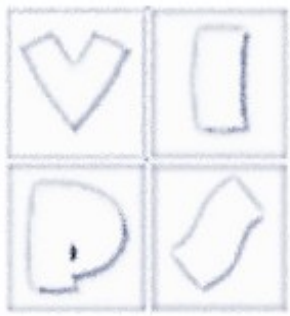
$$(M^{-1})^T \mathbf{n}Mt = 0$$

- Quindi la normale trasformato da $(M^{-1})^T \mathbf{n}$ è perpendicolare alla superficie trasformata. Cioè per evitare problemi si trasformeranno le normali con l'inversa trasposta della matrice di trasformazione del modello



Struttura della mesh

- I vertici danno informazioni di tipo posizionale, gli spigoli informazioni di tipo connettivo (non c'è informazione spaziale)
- Gli spigoli connettono i vertici, permettendo di introdurre un concetto di “vicinanza” tra vertici e dando le informazioni di tipo topologico (ovvero definiscono un grafo)
- Le facce sono determinate una volta dati i vertici e gli spigoli, quindi non introducono nulla
 - Al più possono avere associati attributi, anche se è raro



Strutture dati per le mesh

- Ci sono vari modi di conservare le informazioni sui modelli. Questi possono differire per
 - Memoria occupata
 - Complessità di implementazione di operazioni di ricerca (es. ricerche di adiacenza)
 - Quali sono i vertici vicini a uno dato?
 - Quali sono gli spigoli che contengono un vertice dato?
 - Quali sono le facce che contengono uno spigolo?
 - ...
 - Possibilità di controllo sulla qualità delle superfici (es. manifoldness)

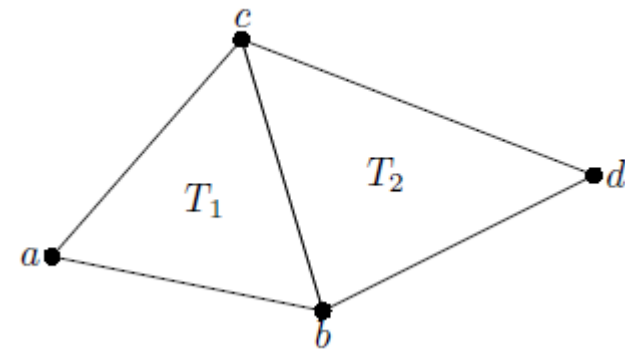


Strutture semplici: lista triangoli e indexed

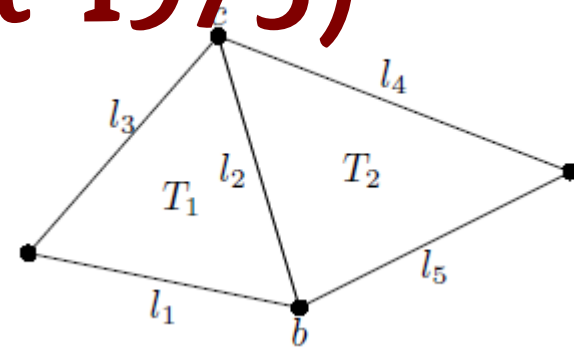
- Immediata: specificare tutte le facce della maglia come terne di triplette di coordinate cartesiane
- Spreco di memoria: duplico le coordinate. Meglio usare una struttura indicizzata, con la lista dei vertici e la lista delle facce con i puntatori ai vertici (indici)
- Normalmente usate in OpenGL per il rendering
- Non ottimali per le ricerche

```
typedef struct {  
    float v1[3];  
    float v2[3];  
    float v3[3];  
} faccia;
```

```
typedef  
struct {  
    float  
    x, y, z;  
} vertice;  
typedef  
struct {  
    vertice*  
    v1, v2, v3;  
} faccia;
```

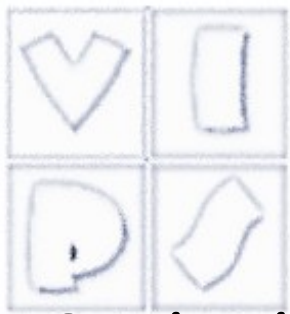


Winged edge (Baugmart 1975)

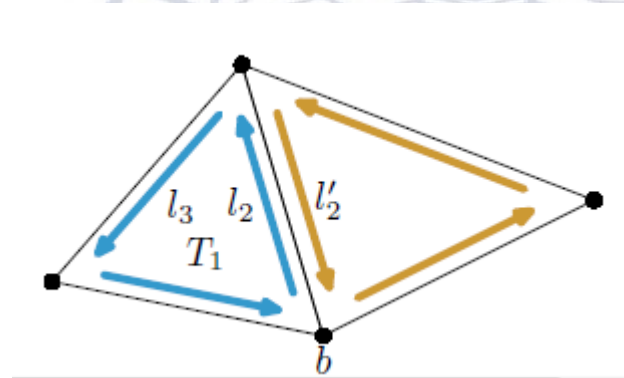


```
typedef struct {  
    we_vertice* v_ini,  
    v_fin;  
    we_spigolo* vi_sin,  
    vi_dstr;  
    we_spigolo* vf_sin,  
    vf_dstr;  
    we_faccia* f_sin,  
    f_dstr;  
} we_spigolo;  
typedef struct {  
    float x, y, z;  
    we_spigolo* spigolo;  
} we_vertice;  
typedef struct {  
    we_spigolo* spigolo;  
} we_faccia;
```

- Si aggiungono dei puntatori allo spigolo per rendere più semplice l'analisi delle incidenze.
- L'elemento base è lo spigolo (edge) con le sue due facce incidenti (wings)
 - Lo spigolo l_2 contiene un puntatore ai due vertici su cui incide ($b; c$), alle due facce su cui incide (T_1, T_2) ed ai due spigoli uscenti da ciascun vertice
 - Un vertice contiene un puntatore ad uno degli spigoli che incide su di esso, più le coordinate (ed altro)
 - La faccia contiene un puntatore ad uno degli spigoli che vi incide (ed altro).
- La struttura assume che ogni spigolo non di bordo abbia due facce incidenti (manifold)



Half edge

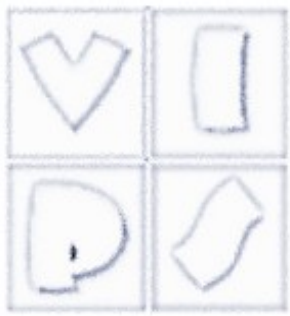


```

typedef struct {
    he_vertice*
    origine;
    he_spigolo*
    gemello;
    he_faccia* faccia;
    he_spigolo*
    successivo;
} he_spigolo;
typedef struct {
    float x, y, z;
    he_spigolo*
    spigolo;
} he_vertice;
typedef struct {
    he_spigolo*
    spigolo;
} he_faccia;

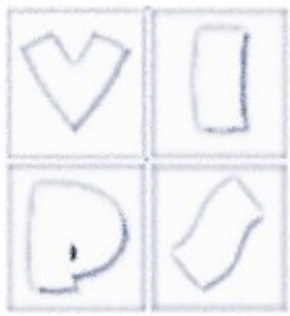
```

- Ogni spigolo viene diviso in due spigoli orientati in modo opposto
 - Ciascun mezzo spigolo contiene un puntatore al vertice iniziale, alla faccia a cui “appartiene”, al mezzo spigolo successivo (seguendo l’ordinamento) ed al mezzo spigolo gemello
 - Ogni vertice, oltre alle coordinate (e attributi) contiene un puntatore ad uno qualsiasi dei mezzi spigoli che esce da tale vertice
 - Ogni faccia contiene uno dei suoi mezzi spigoli (oltre ad altre caratteristiche quali, ad esempio, la normale)
- Efficiente per le ricerche ed elegante



Note

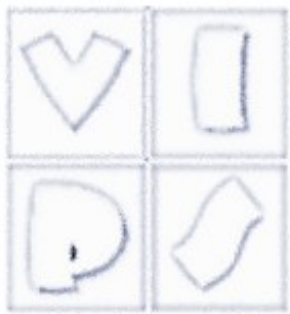
- La stessa applicazione grafica può far uso di più di una struttura dati
- La rappresentazione con la lista di vertici essendo semplice e leggera è tipicamente usata come formato per i file contenenti la geometria di oggetti
- Le applicazioni grafiche in genere caricano tali file ed usano l'informazione contenuta in essi per riempire una struttura dati più utile ai fini algoritmici (per esempio la half-edge)



Esercizi

- descrivere una procedura che, data una rappresentazione winged-edge di una maglia triangolare, “stampi” tutti le facce incidenti su un dato vertice v , assumendo data una procedura stampaFaccia(we faccia $*f$)
- descrivere una procedura che, data una rappresentazione half-edge di una maglia triangolare, “stampi” tutti i mezzi-spigoli uscenti dal vertice v con una procedura stampaSpigolo(he spigolo $*l$)

Formati per mesh

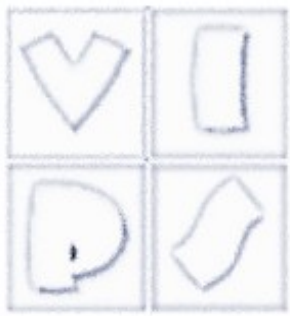


- Esempio .off, obj, .ply, .wrl, x3d, .mesh
- Possono supportare anche diversi tipi di primitive, scene graph
- Esempio .off

```
OFF
4 4 0
-1 -1 -1
1 1 -1
1 -1 1
-1 1 1
3 1 2 3
3 1 0 2
3 3 2 0
3 0 1 3
```

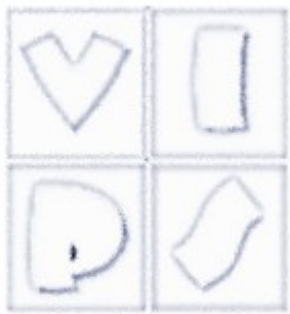


Qualità delle mesh



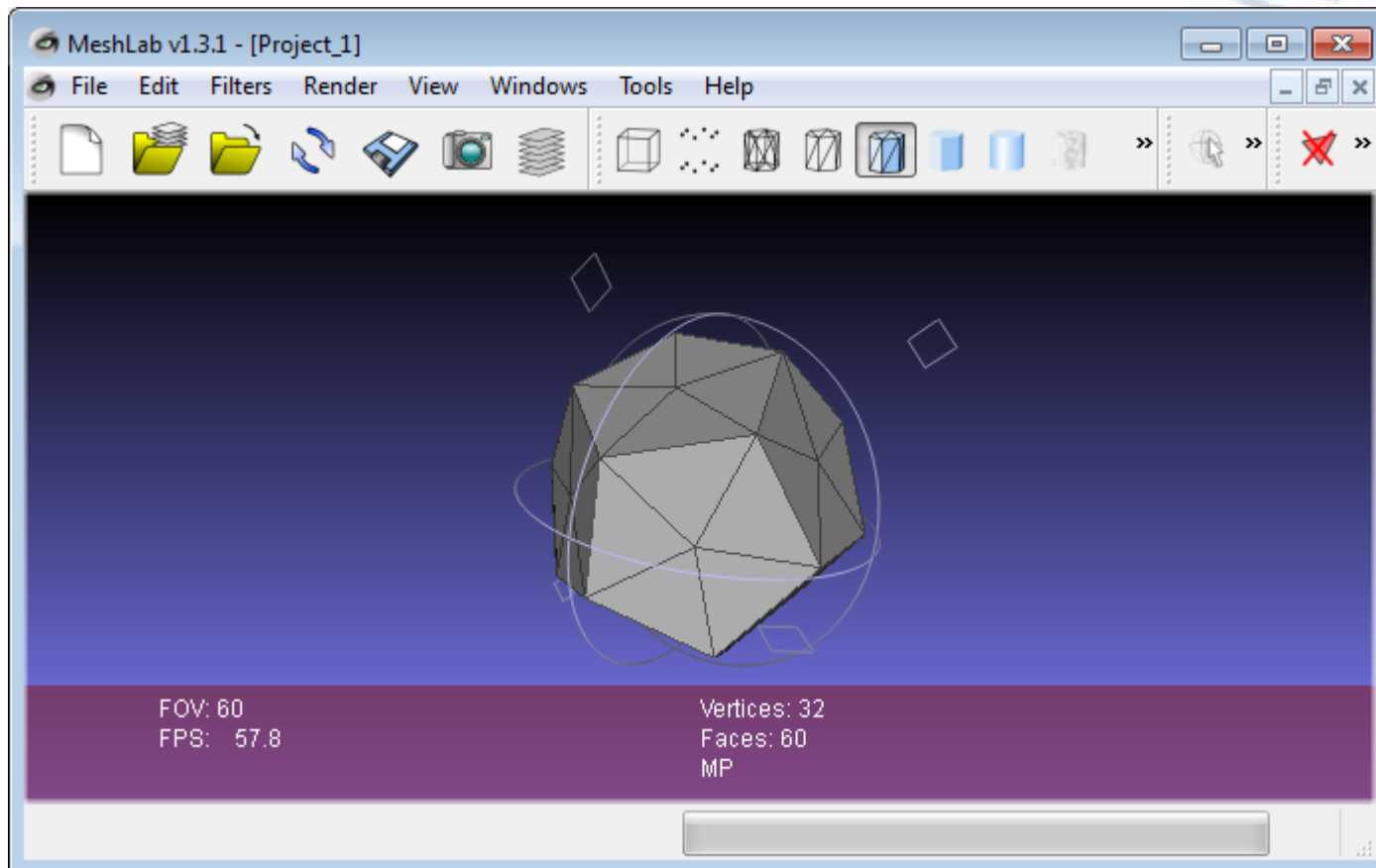
- Manifoldness
- Risoluzione. Numero di triangoli/poligoni
- Accuratezza nell'approssimazione
- Attenzione: se la geometria sottostante è semplice, i triangoli possono essere inutili
 - Algoritmi di semplificazione
- Curvature e normali
 - La loro qualità influirà poi sugli algoritmi di illuminazione

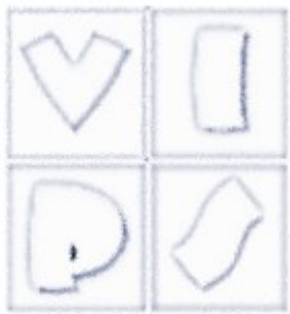




Meshlab

- Strumento open source ottimo per il processing di mesh con algoritmi di analisi e processing moderni
 - <http://meshlab.sourceforge.net/>



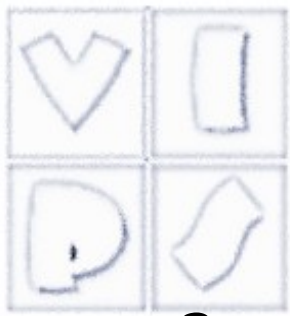


Riferimenti

- Wikipedia voce polygon mesh
- <http://meshlab.sourceforge.net/>
- Angel cap 2.3



Domande di verifica



- Cosa si intende per Constructive Solid Geometry?
- Cos'è una mesh di poligoni?
- Cosa si intende per mesh orientabile?
- Cosa si intende per varietà (2-manifold)
- Come si può ridurre la complessità di una mesh?