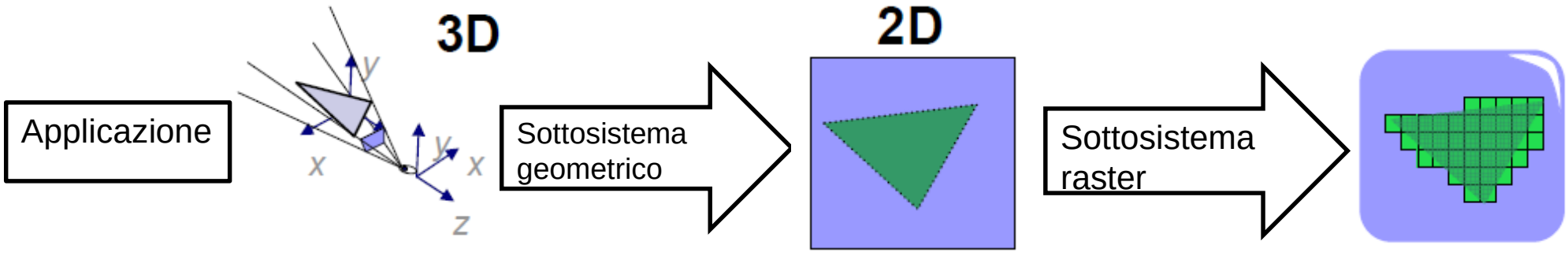
Introduzione alla pipeline grafica
e
Modellazione Geometrica

Introduzione alla pipeline di rendering

- Pipeline: La serie di passi che trasformano la descrizione geometrica di una scena in un'immagine sullo schermo
- Nelle GPU moderne la pipeline di rendering è interamente a carico della scheda grafica
- La CPU (o meglio le applicazioni in esecuzione) si limita a descrivere la scena
- L'accesso all'hardware grafico viene gestito in modo trasparente dalle librerie grafiche (OpenGL, DirectX)

Introduzione alla pipeline di rendering

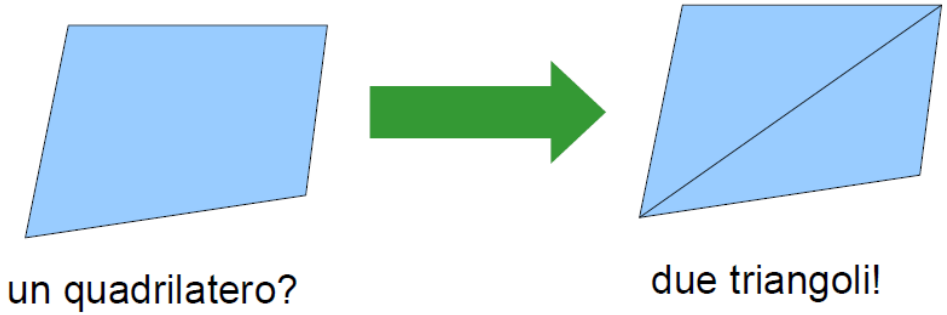
- La pipeline di rendering si suddivide in due parti
  - **Sottosistema geometrico:** porta la geometria del modello (3D) nelle coordinate dello schermo (2D)
  - **Sottosistema raster:** trasforma gli elementi della scena proiettati sullo schermo (2D) nell'insieme di pixel da "accendere", in funzione dei parametri come l'illuminazione, le texture, il colore degli oggetti, la geometria.

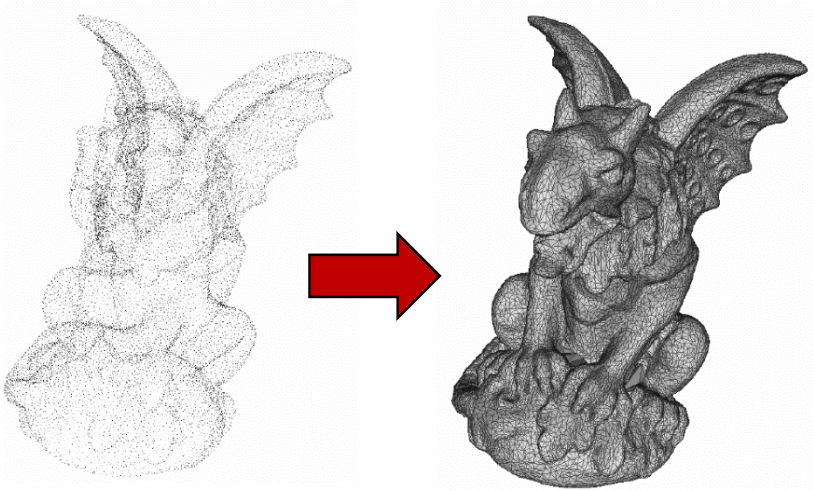


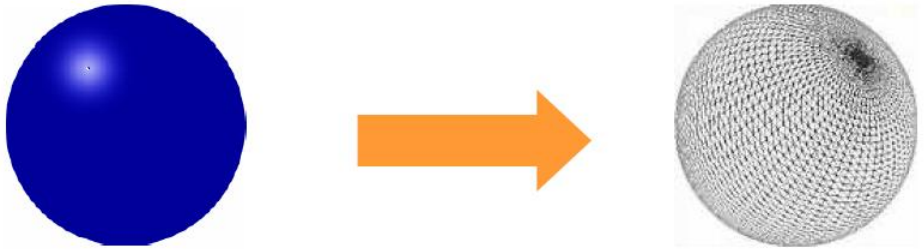
Vedremo una versione più dettagliata nelle prossime lezioni

Introduzione alla pipeline di rendering

- L'implementazione hardware della pipeline (sulla scheda grafica) si basa su un'assunzione: "**Il mondo è fatto di triangoli (poligoni), segmenti e punti**"

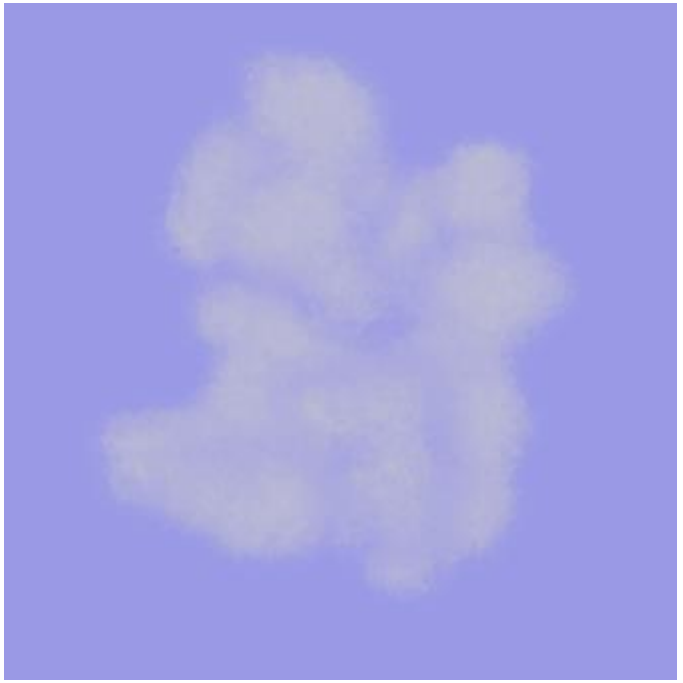


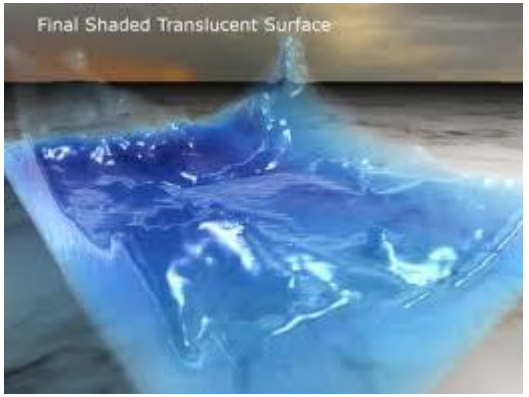




Introduzione alla pipeline di rendering

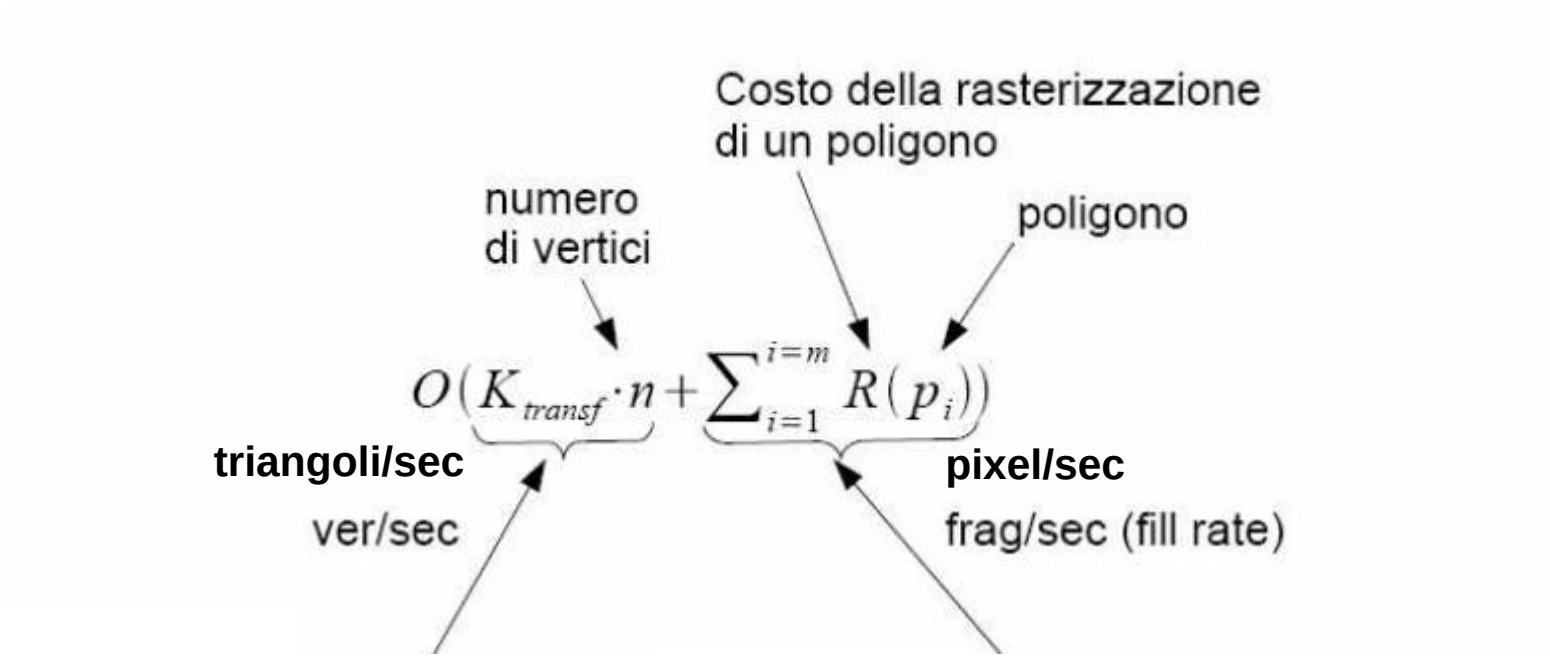
- Limitazioni: come modellare elementi complessi?

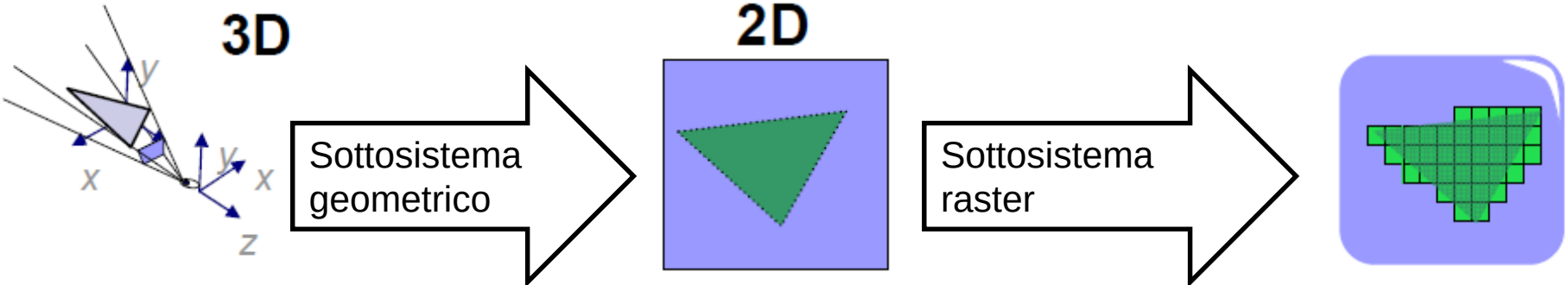




Introduzione alla pipeline di rendering

- Complessità computazionale (la stima della fluidità di una applicazione grafica si misura in termini di frames/secondo)





Introduzione alla pipeline di rendering

Esercizio

State effettuando il rendering di un modello composto da un milione di triangoli su un display con risoluzione 1280 x1024 pixel. Il rendering finale occupa il 64% del display. Se la scheda che state utilizzando ha una *performance* di 20 milioni di triangoli al secondo ed un fill-rate di 20 Mpixel al secondo, qual'è il numero massimo di fotogrammi al secondo che potete ottenere?

Soluzione:

- 1280x1024x0.64= 838.860,8 numero di pixels da accendere per un rendering
- 20*2^20= 20971520 numero di pixels che la scheda può rasterizzare al secondo

- Il sottosistema raster può produrre:

  20971520 / 838.860,8 = 25 fotogrammi al secondo

- Il sottositema geometrico può produrre

  20 * 10^6 / 10^6 = 20

Quindi la risposta è 20

Gli stadi della pipeline lavorano in parallelo su dati diversi.
La velocita' della pipeline e' limitata dallo stadio piu' lento.

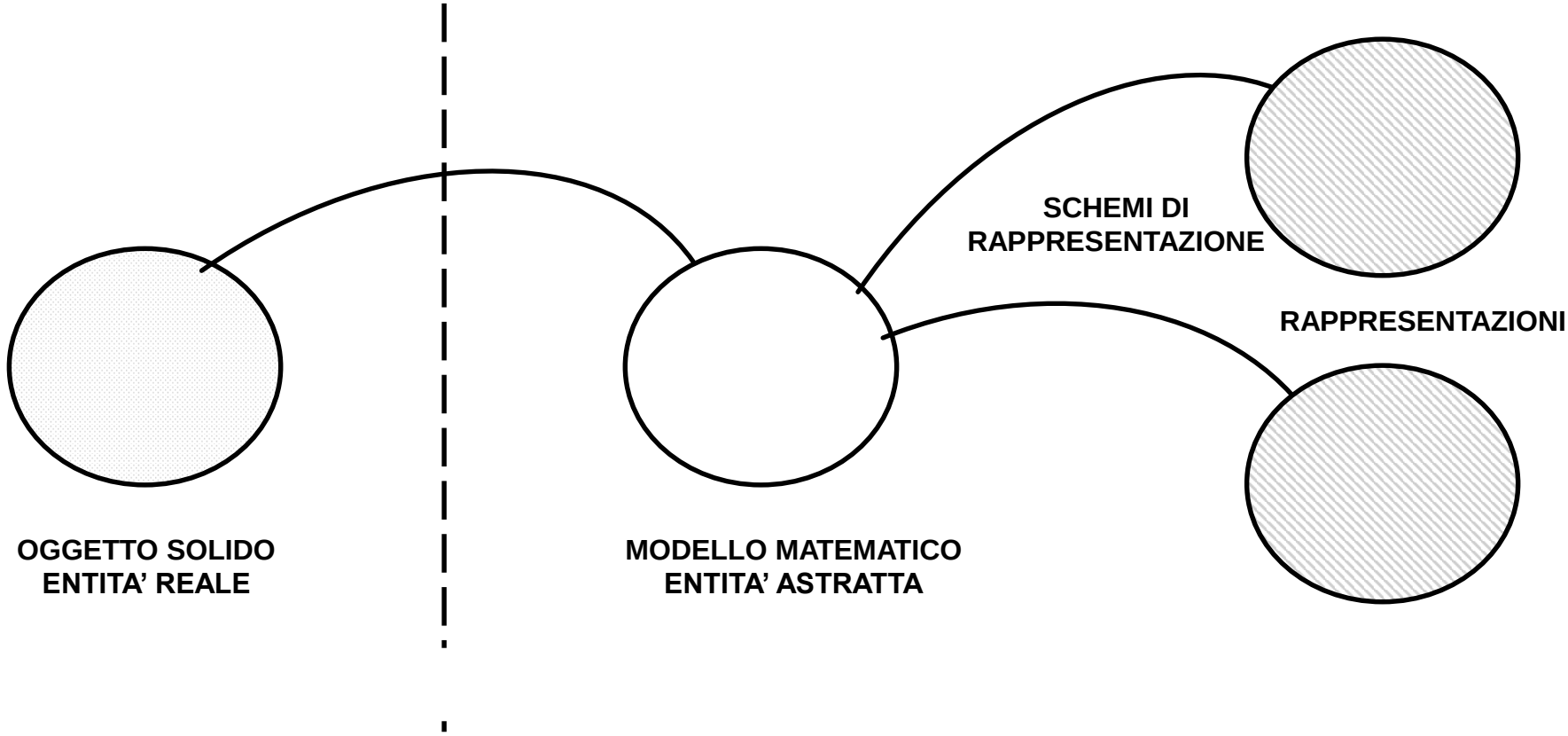
… alcuni concetti di base

- **Modellazione Geometrica**
  - è l'insieme dei metodi e delle tecniche che permettono di rappresentare modelli matematici astratti corrispondenti ad oggetti reali tridimensionali, può essere:

    - Automatica (3D scanner)
    - Manuale (Blender, Maya, Rhino3D, 3DStudio Max)
    - Procedurale (frattali, grammatiche, semi-automatica con impostazione di parametri es: modellazione di tessuti, capelli etc…)

Modello e Rappresentazioni

- Rappresentazione: insieme di regole utilizzate per la descrizione del modello
- Modello può avere più rappresentazioni



Schemi di Rappresentazione

- *Proprietà formali*
  - **dominio**
    - potere descrittivo. Piu' ampio è il dominio, maggiore è la varieta' degli oggetti solidi rappresentabili con uno schema di rappresentazione
  - **validità**
    - ogni rappresentazione (immagine di un elemento del dominio) è detta "valida"
  - **completezza (non ambiguità)**
    - una rappresentazione è completa o non ambigua se modella esattamente un solo solido
  - **unicità**
    - schema di rappresentazione è unico se tutti i solidi hanno una sola possibile rappresentazione

Schemi di Rappresentazione

- Rappresentazioni del contorno (B-rep="boundary representation")
  - mesh poligonali, rapp. implicite e parametriche (curve e superfici)
- Rappresentazioni volumetriche

  Rappresentazioni per decomposizione
  - Decomposizione in celle
  - Enumerazione per occupazione spaziale (voxel)
  - Partizione spaziale binaria (octree)

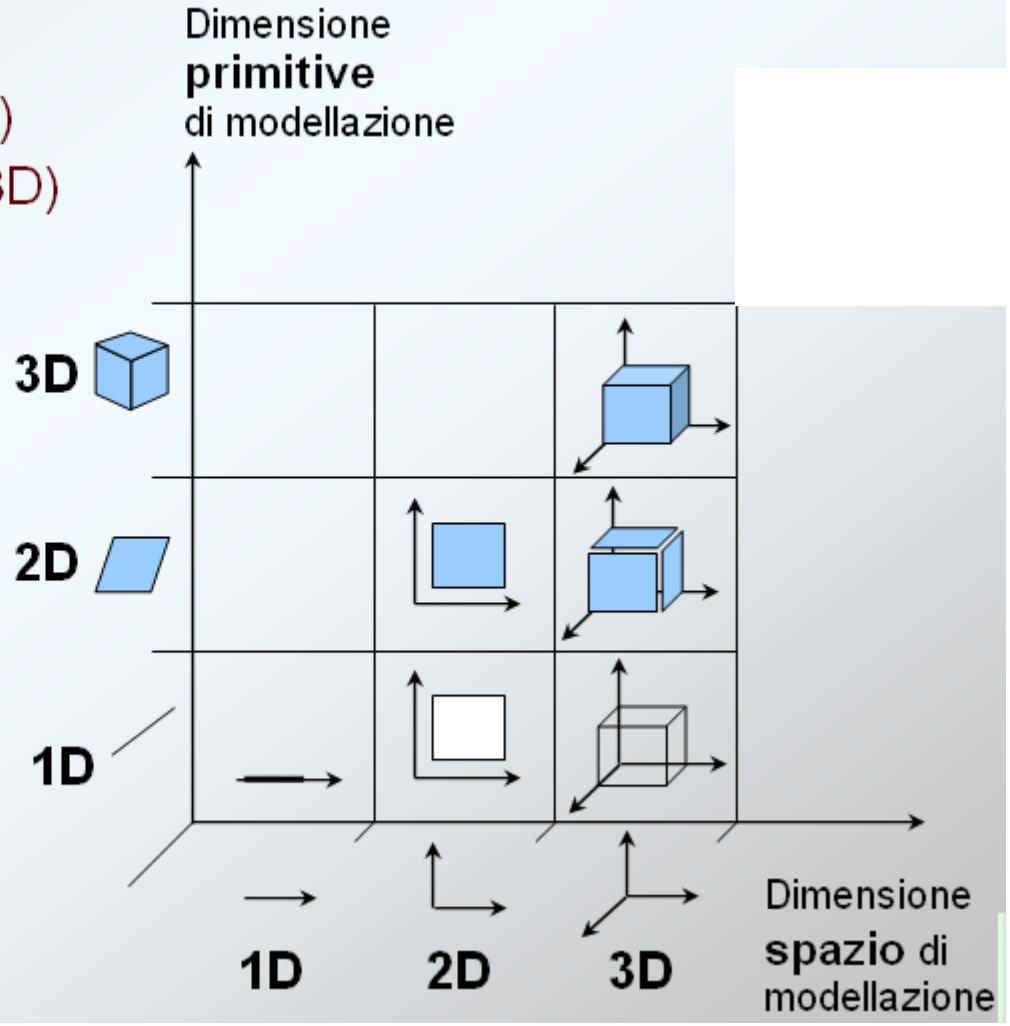
  Rappresentazioni costruttive
  - Constructive Solid Geometry – CSG
  - Istanza di primitive
- Rappresentazioni di tipo sweep

!!! modellatori software avanzati consentono di convertire i modelli da uno schema di rappresentazione all'altro

Schemi di Rappresentazione

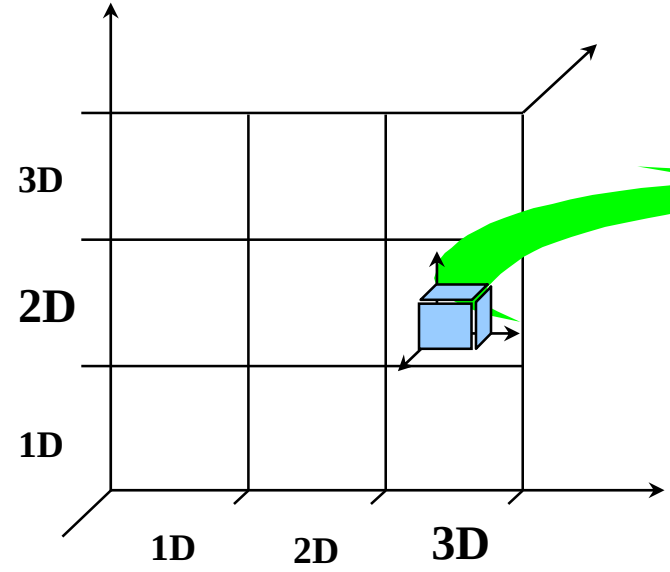
Classificazione topologica

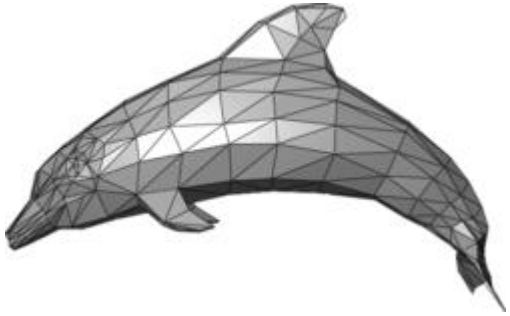
- Modelli a linee
  - Modelli 1D in 2D (disegno in 2D)
  - Modelli 1D in 3D (wireframe in 3D)
- Modelli di figure
  - Modelli 2D in 2D (per aree)
- Modelli di solidi
  - Modelli 2D in 3D (per contorni)
  - Modelli 3D in 3D (per volumi)



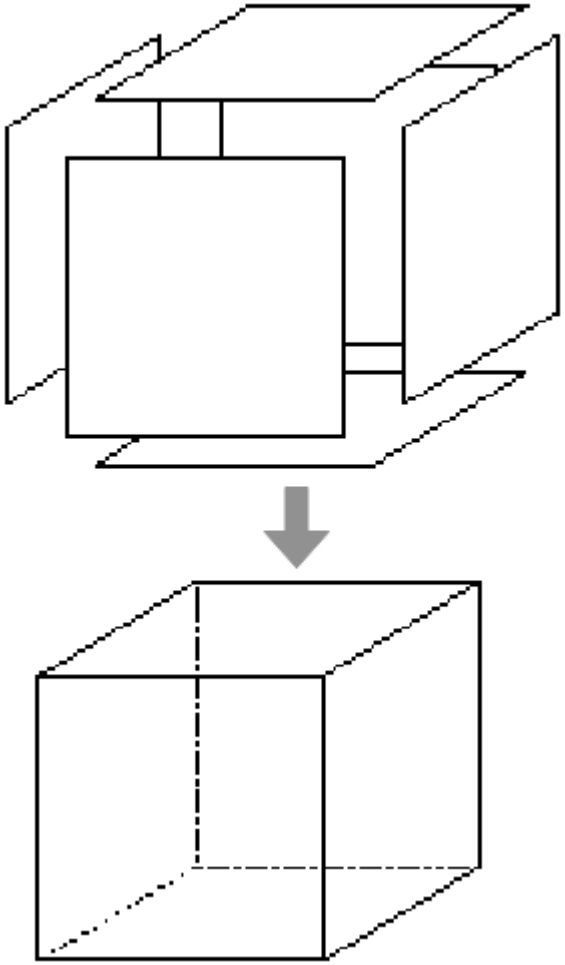
BRep - Mesh poligonali

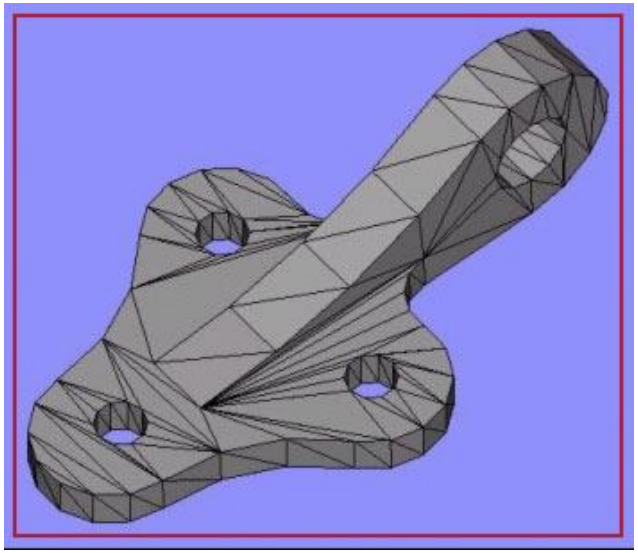
- **Oggetto definito dalle facce (triangoli) che lo delimitano e relazioni esistenti tra:**
  - facce
  - spigoli
  - vertici

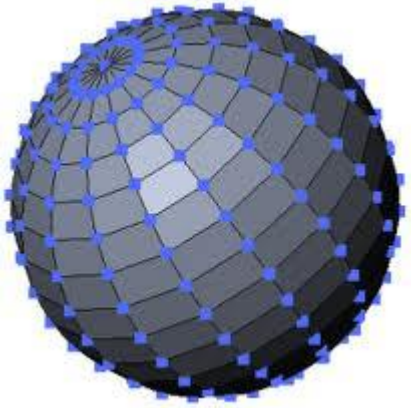




BRep - Mesh poligonali

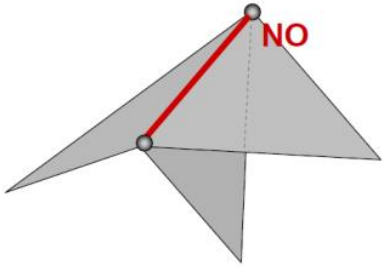


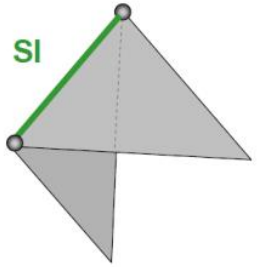




BRep - Mesh poligonali

Poliedro: solido la cui superficie è costituita da un insieme di poligoni tali che due e solo due di essi si intersecano in uno spigolo e sia possibile percorrerne la superficie spostandosi da una faccia all'altra attraversando i suoi spigoli finché tutti i poligoni siano stati percorsi da questo cammino continuo.





Poliedri semplici: possono sempre essere deformati in modo che ne sia rispettata la continuitá in una sfera. Ovvero senza fori.

Per i poliedri semplici vale la regola di Eulero

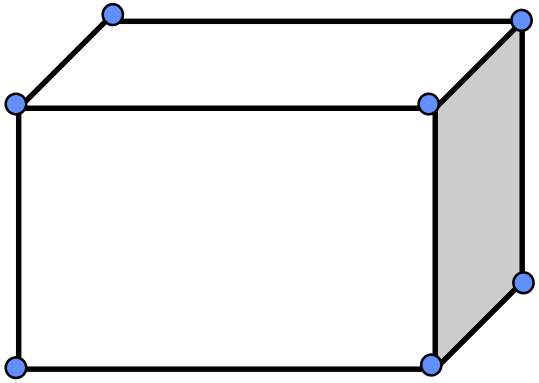
$$V - E + F = 2$$

Numero di vertici — Numero di spigoli — Numero di facce

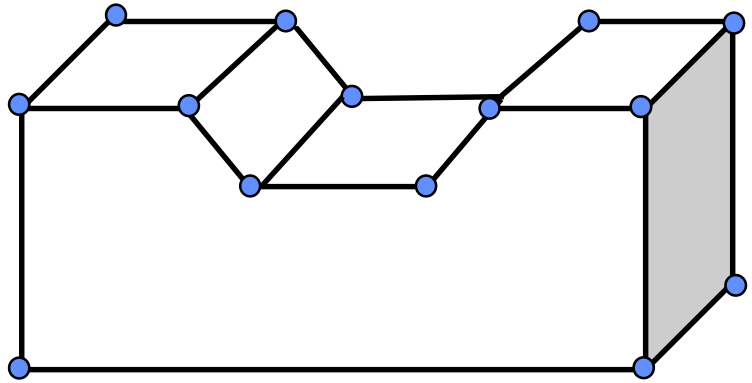
BRep - Mesh poligonali

Operatori di Eulero

Esempi



V - E + F = 2
8 - 12 + 6 = 2
2 = 2

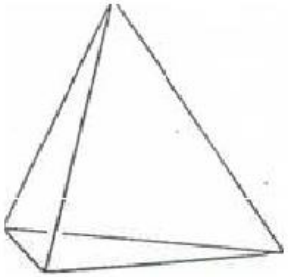


V - E + F = 2
16 - 24 + 10 = 2
2 = 2

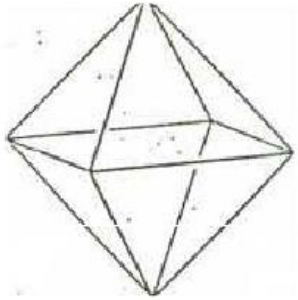
BRep - Mesh poligonali

Poliedri regolari: poliedri semplici convessi. Tutte le facce hanno lo stesso numero *h* di spigoli, ogni vertice ha lo stesso numero *k* di spigoli incidenti e tutti gli spigoli sono della stessa lunghezza. Si può dimostrare che esistono Solo cinque poliedri regolari (tetraedro, cubo, ottaedro, dodecaedro, icosaedro).

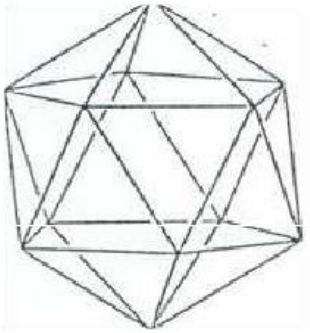
Tetraedro



Ottaedro

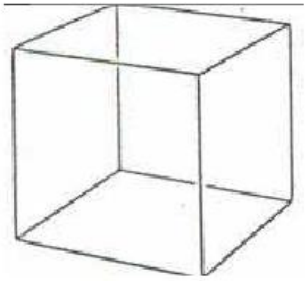


Icosaedro

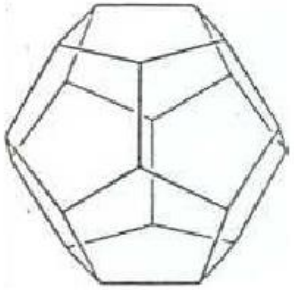


20 facce

Cubo

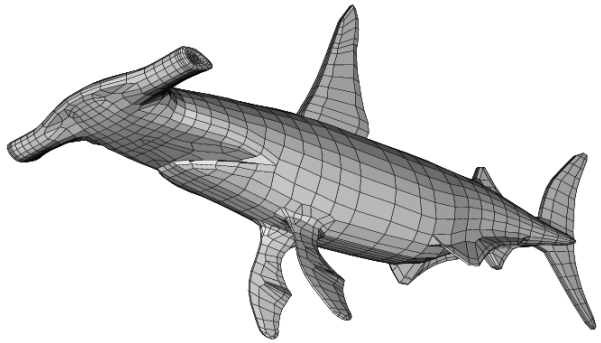


Dodecaedro

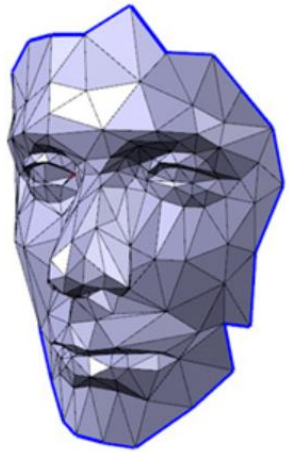


BRep - Mesh poligonali

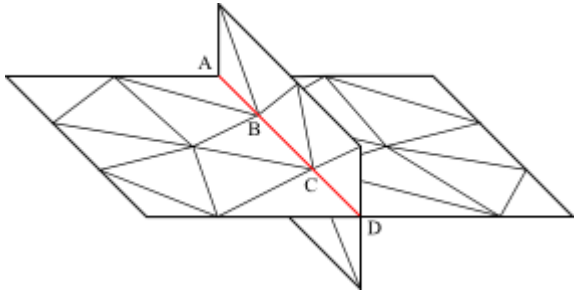
- La modellazione mediante mesh poligonali consente di rappresentare poliedri



- è possibile utilizzare mesh di poligoni per rappresentare anche superfici aperte



- è possibile rappresentare anche superfici non poliedrali (non-manifold)

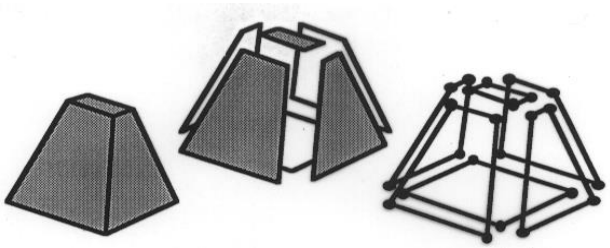


BRep - Mesh poligonali

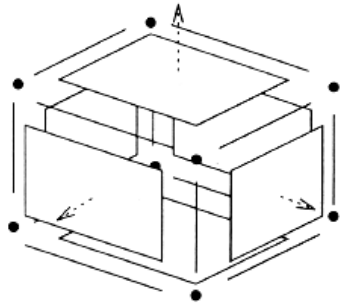
Riassumendo, mesh poligonali contengono:

- **informazioni geometriche**: componenti geometriche del mesh (vertici, lati, facce)
- **informazioni topologiche:** specificano le relazioni di connessione tra le componenti geometriche

Tronco di piramide

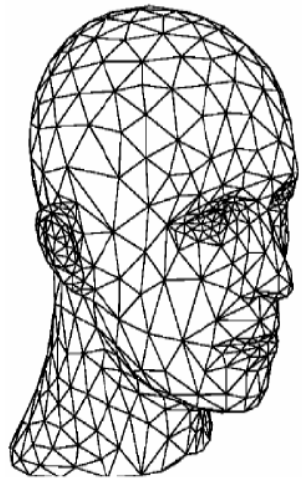


Parallelepipedo



- mesh poligonali sono un’approssimazione discreta di una superficie

- tipicamente vengono utilizzate mesh triangolari



BRep - Mesh poligonali

Alcuni tipi di rappresentazioni:

- Rappresentazione semplice
- Rappresentazione con lista di vertici (Indexed faces)
- Rappresentazione con lista di spigoli
- Rappresentazione Winged-Edge
- Rappresentazione Half-Edge

Spesso nella grafica e nella geometria computazionale è necessario effettuare delle query (richieste) su mesh.

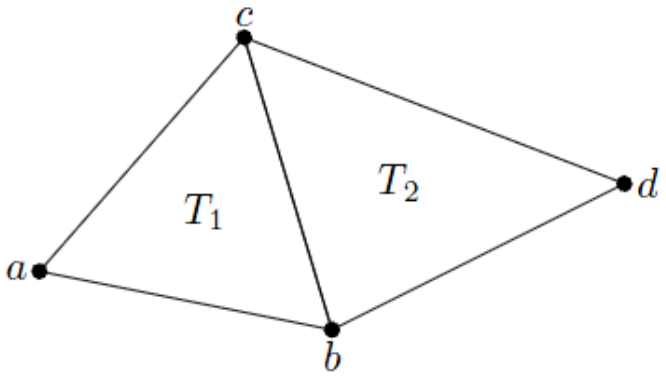
- Quali facce incidono su un certo vertice?
- Quali facce incidono su un certo lato?
- Quali vertici sono collegati con un dato vertice?

Rappresentazioni con strutture dati più sofisticate consentono di eseguire query più efficienti

BRep - Mesh poligonali

Rappresentazione semplice

specificare tutte le facce del mesh
come terne di triplette di coordinate cartesiane



• Ad esempio i triangoli T_1 e T_2 si possono definire come
$T_1 = \{(a_x, a_y, a_z), (b_x, b_y, b_z), (c_x, c_y, c_z)\}$.
$T_2 = \{(b_x, b_y, b_z), (d_x, d_y, d_z), (c_x, c_y, c_z)\}$.

• rappresentazione semplice e compatta, ma non efficiente, poiché i vertici vengono ripetuti nella lista dei poligoni.

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

• ricerche di incidenza sono onerose

• la rappresentazione di numeri in virgola mobile all'interno di un elaboratore elettronico può subire lievi variazioni a seguito delle approssimazioni nei calcoli, per cui lo stesso vertice puo spostarsi lievemente ed il test di uguaglianza tra due vertici può fallire !!!!!!!
(es: *x=0.99999999 x=1.0000000*)

BRep - Mesh poligonali

Rappresentazione con lista di vertici (Indexed faces)

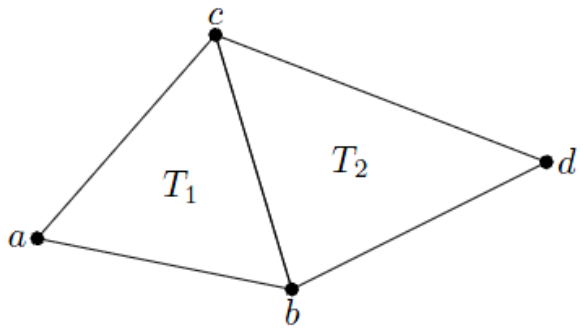
si elimina la ridondanza dei vertici

- si memorizza una lista dei vertici V (senza ripetizioni) e una lista di facce, ciascuna descritta dai puntatori ai vertici che la compongono (in verso antiorario, vedi es. OpenGL)

- Ad esempio la faccia T1 punta ai tre vertici a, b e c

- Le ricerche di incidenza continuano ad essere complesse

$V = \{(a, b, c, d)\}=\{(a_x, a_y, a_z), ..., (d_x, d_y, d_z)\}$
$T_1=(a,b,c)$
$T_2=(b,d,c)$

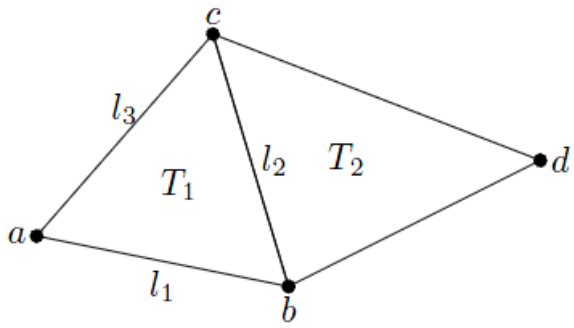



```
typedef struct {
   float x,y,z;
} vertice;
typedef struct {
   vertice* v1,v2,v3;
} faccia;
```

BRep - Mesh poligonali

Rappresentazione con lista di spigoli

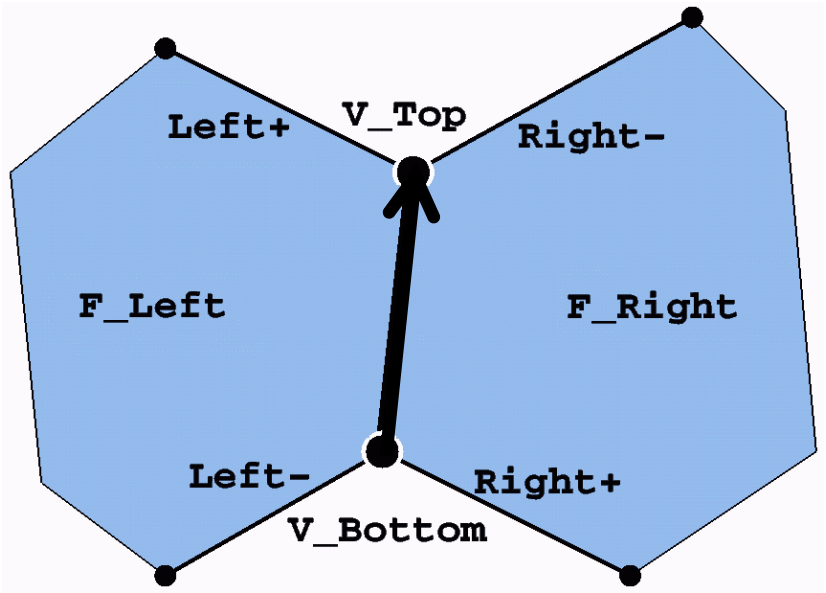
- si costruisce una lista dei vertici (senza ripetizioni)
- e una lista degli spigoli, composti dai due puntatori ai vertici incidenti sullo spigolo e i due puntatori alle facce incidenti sullo spigolo; ciascuna faccia viene descritta dai puntatori degli spigoli che la compongono
- Ad esempio lo spigolo l_2 punta ai vertici *b* e *c* e alle facce T_1 e T_2
- In questo modo si semplificano di molto le ricerche di incidenza spigolo-faccia



V = {(a, b, c, d)}={(a_x, a_y, a_z), ..., (d_x, d_y, d_z)}
l_1=(b,a,T_1, NULL)
l_2=(b,c, T_1, T_2)
l_3=(a,c, T_1 , NULL)
T_1 =(l_3, l_2, l_1)

BRep - Mesh poligonali

Rappresentazione Winged-Edge



- contiene ancora più dati nella rappresentazione
- ogni spigolo è orientato e contiene, oltre ai vertici, due puntatori alle facce adiacenti più i puntatori ai 4 lati che toccano le facce.
- ogni vertice contiene un puntatore ad uno dei lati che incide su di esso e le sue coordinate.
- una faccia è definita da un puntatore ad uno dei lati che vi incide.
- vantaggio: ancora più efficiente la ricerca (es: trovare tutte le facce incidenti un vertice)

BRep - Mesh poligonali

Rappresentazione Half-Edge

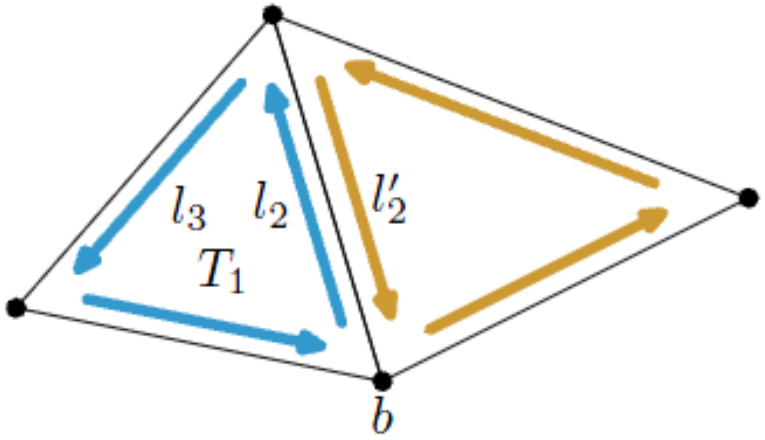
- Ogni spigolo viene diviso in due mezzi-spigoli orientati in modo opposto (da cui il nome half-edge).

- Ciascun half-edge contiene un puntatore al vertice iniziale, alla faccia a cui appartiene, al mezzo spigolo gemello ed al mezzo-spigolo successivo.

- Ogni vertice contiene un puntatore ad uno qualsiasi dei mezzi-spigoli uscenti e le coordinate.

- Ogni faccia contiene uno dei suoi mezzi-spigoli.

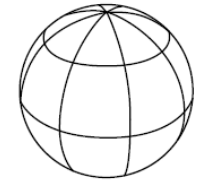
- vantaggi: Più efficiente rispetto a winged-edge.



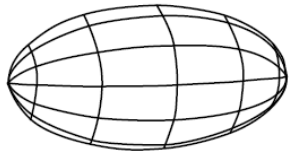
Brep – rappresentazioni implicite

- Rappresentazione di una curva/superficie (chiusa o aperta) come luogo dei punti dove una funzione implicita si annulla

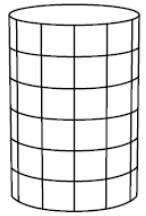
  Es: quadriche $F(x,y,z) = 0$



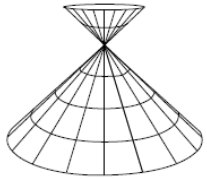
sphere
$x^2 + y^2 + z^2 - 1 = 0$



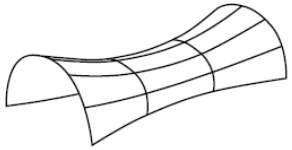
ellipsoid
$ax^2 + by^2 + cz^2 - 1 = 0$



cylinder
$x^2 + z^2 - 1 = 0$

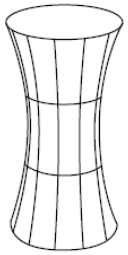


cone
$x^2 + z^2 - y^2 = 0$

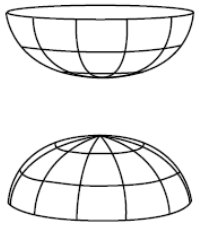


hyperbolic paraboloid
(saddle)
$x^2 - z^2 - y = 0$

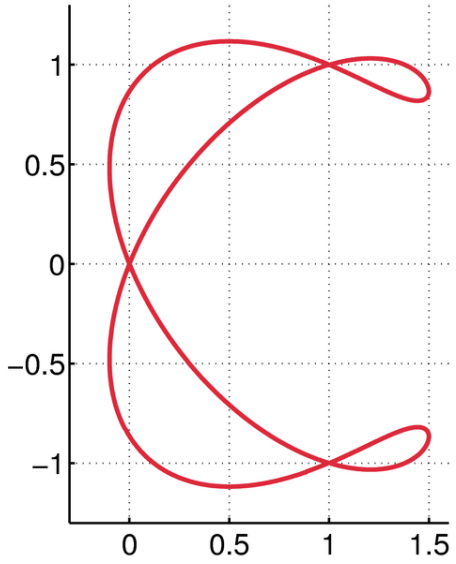
paraboloid
$x^2 + z^2 - y = 0$



hyperboloid of one sheet
$x^2 + z^2 - y^2 - 1 = 0$



hyperboloid of two sheets
$x^2 + z^2 - y^2 + 1 = 0$



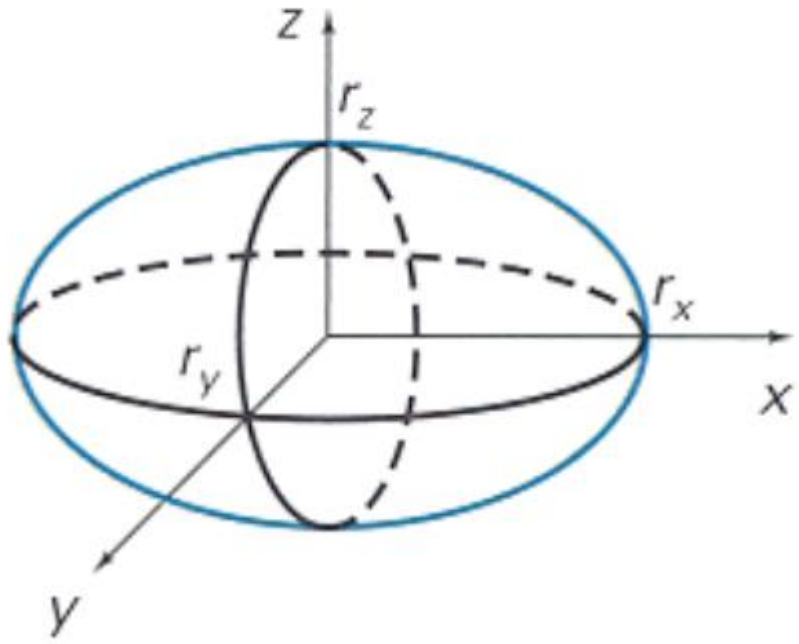
$$(y^2 - x^2)(x - 1)(2x - 3) = 4(x^2 + y^2 - 2x)^2$$

Brep – rappresentazioni parametriche

- Rappresentazione di una curva/superficie in forma parametrica
- Le rivedremo in una lezione specifica.....

$$x = f_x(u,v)$$
$$y = f_y(u,v)$$
$$z = f_z(u,v)$$

$$x = r_x \, sin\theta \, cos\varphi \qquad 0 \le \theta \le \pi$$
$$y = r_y \, sin\theta \, sin\varphi \qquad -\pi \le \varphi \le \pi$$
$$z = r_z \, cos\theta$$



- sono flessibili, consentono una elaborazione analitica
- possono essere convertite “facilmente” in mesh di poligoni
- E’ una rappresentazione efficiente dal punto di vista della memoria

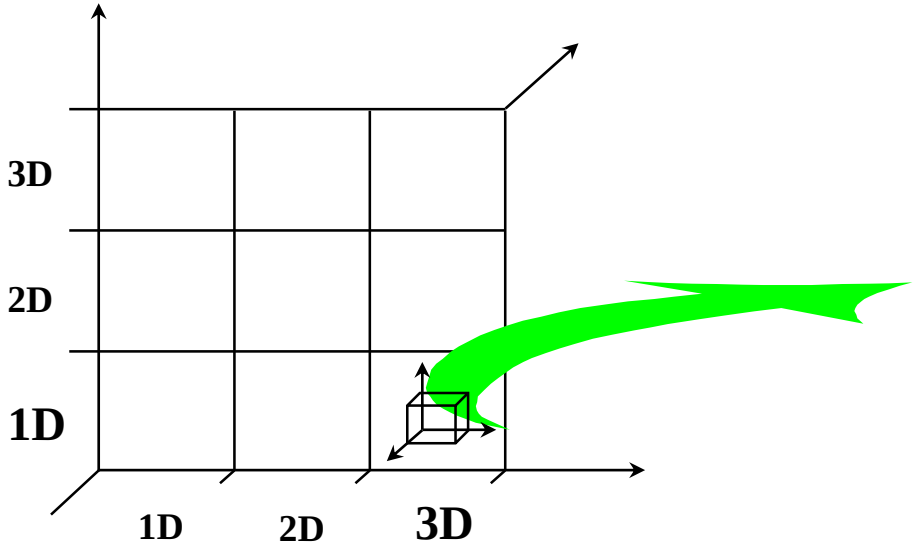
B-rep: proprietà

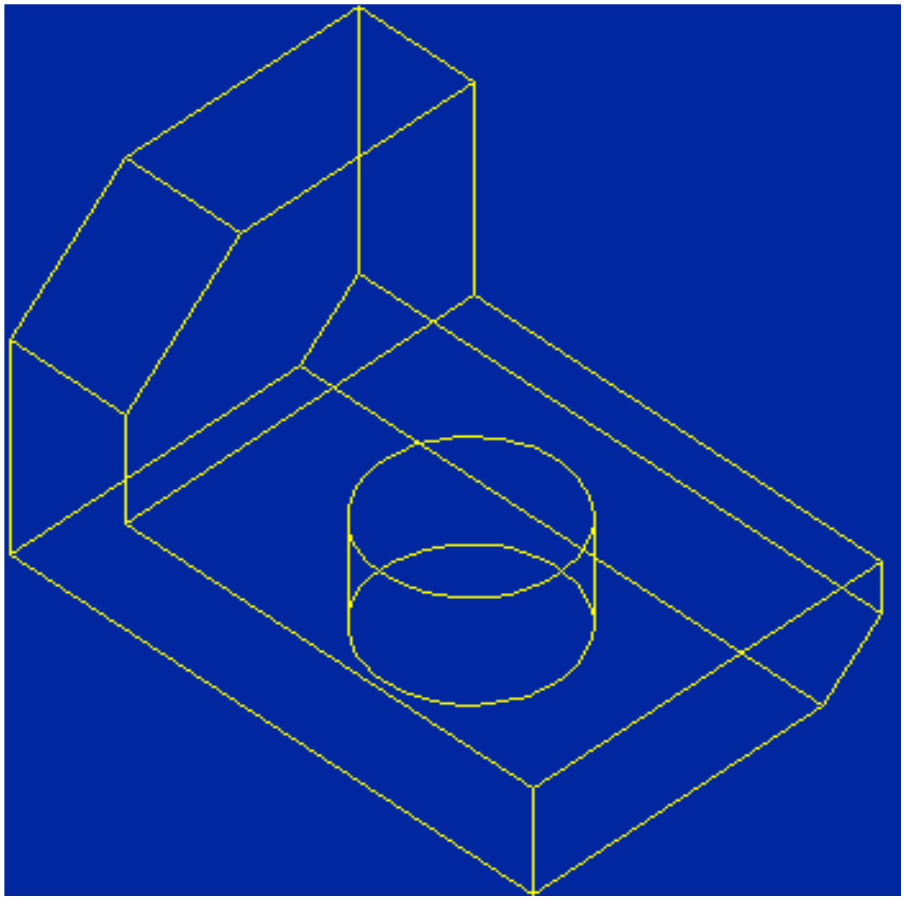
	B-rep
Dominio	buono
Validità	difficile da validare (regola di eulero)
Completezza	si
Unicità	no
Concisione	no
Facilità di creazione	difficile

BRep

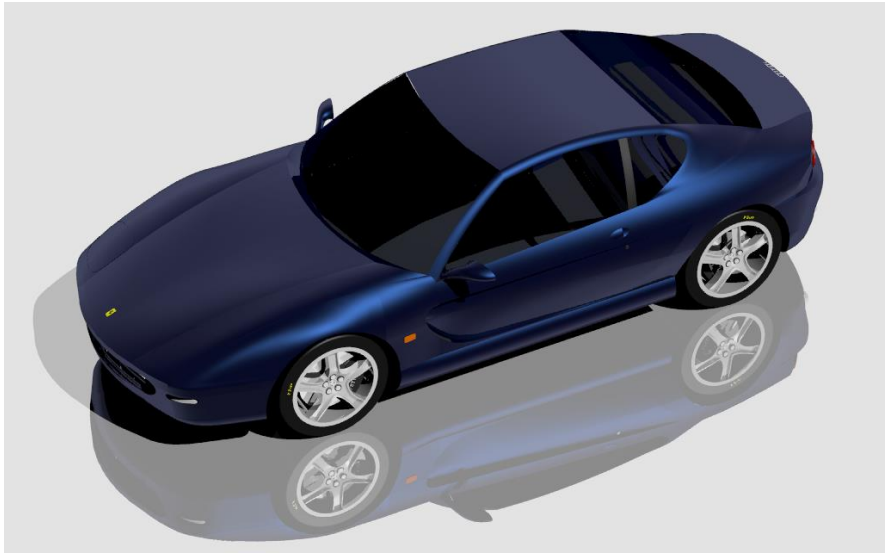
- **Caratteristiche**
  - topologia è l'elemento portante
  - dominio delle geometrie trattate:
    - descrizione precisa: analitica (vedi lezione curve e superfici)
    - descrizione approssimata: faccette
- **Vantaggi**
  - è semplice ottenere informazioni sui singoli elementi
  - è facile visualizzare il solido creato
  - adatto per generare viste, con eliminazione automatica delle parti nascoste
  - è una rappresentazione completa
- **Svantaggi**
  - richiede notevole spazio di memoria
  - editing "diretto" difficile
  - non è una rappresentazione unica

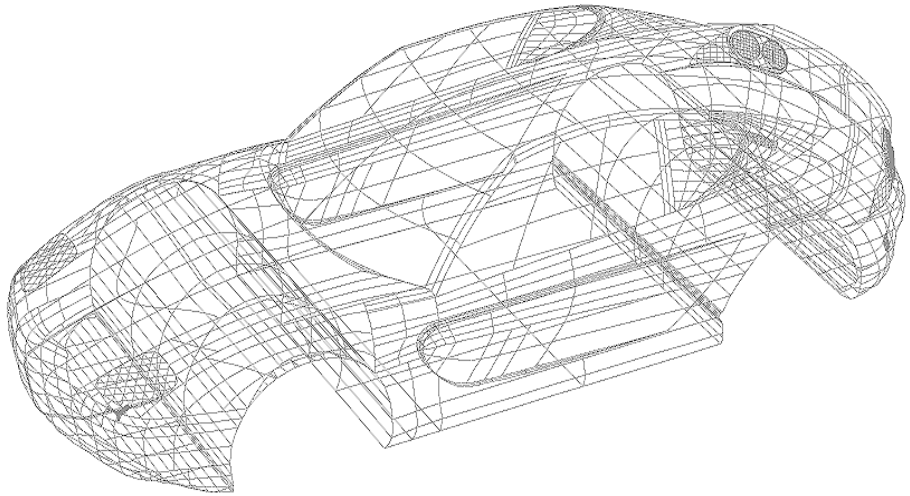
Brep – rappresentazioni wireframe

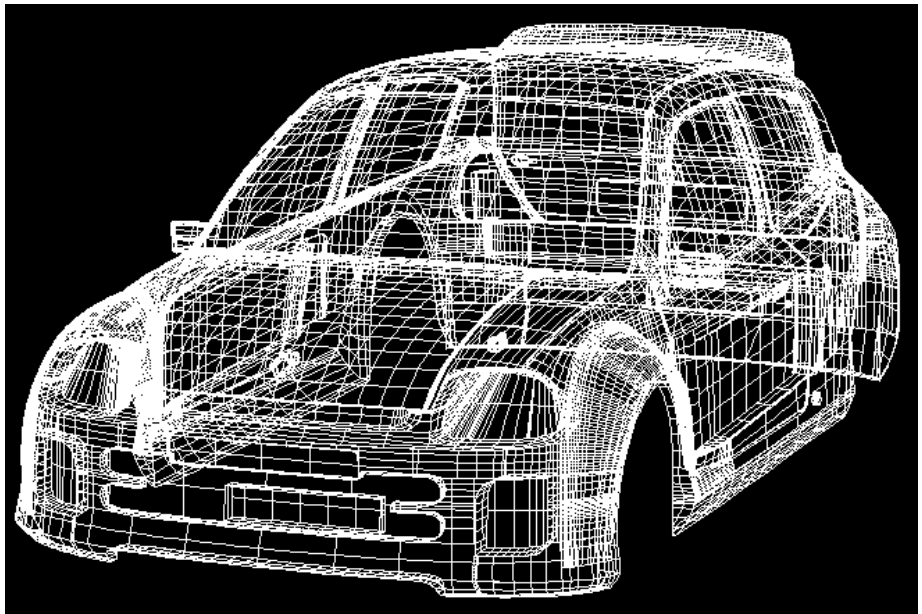




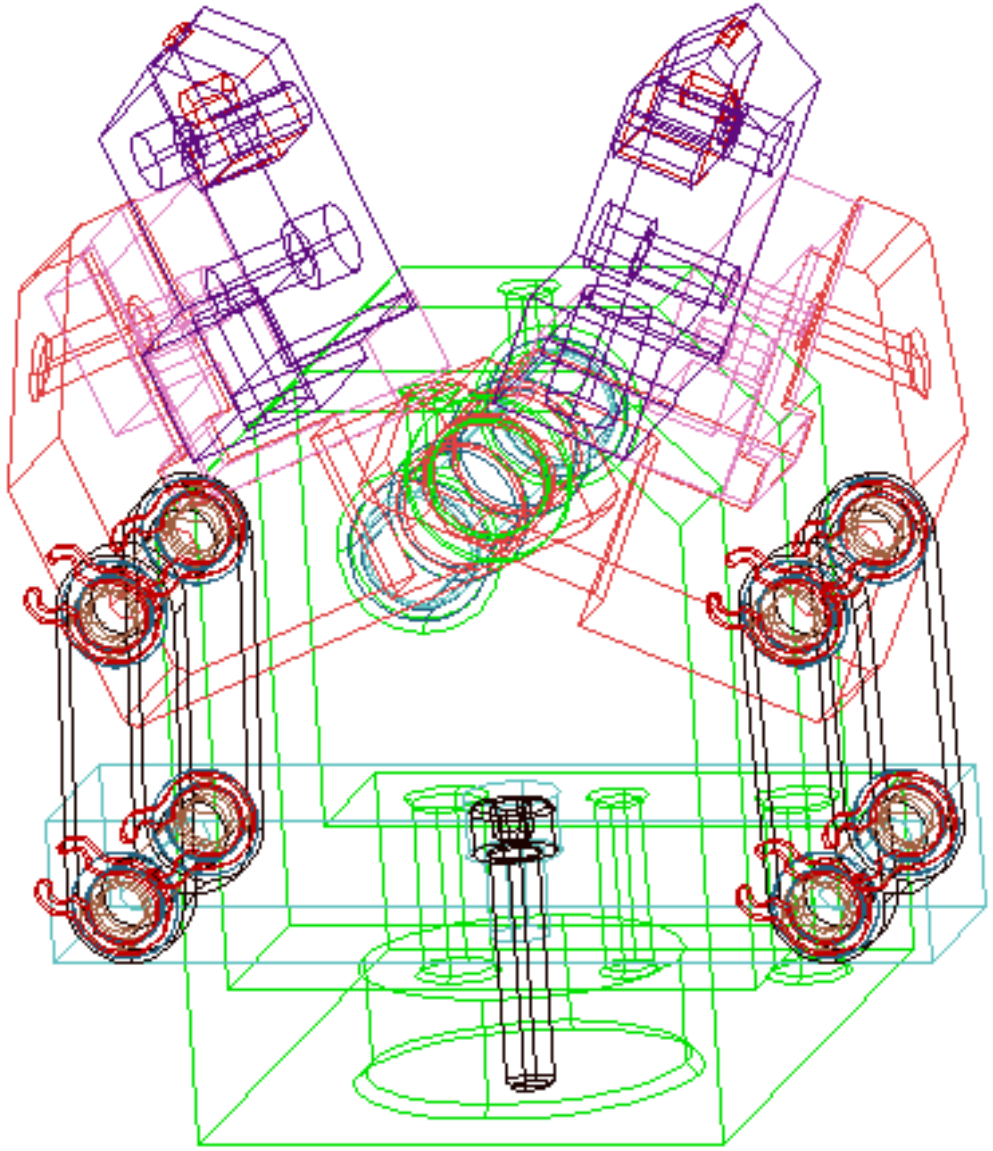
Brep – rappresentazioni wireframe





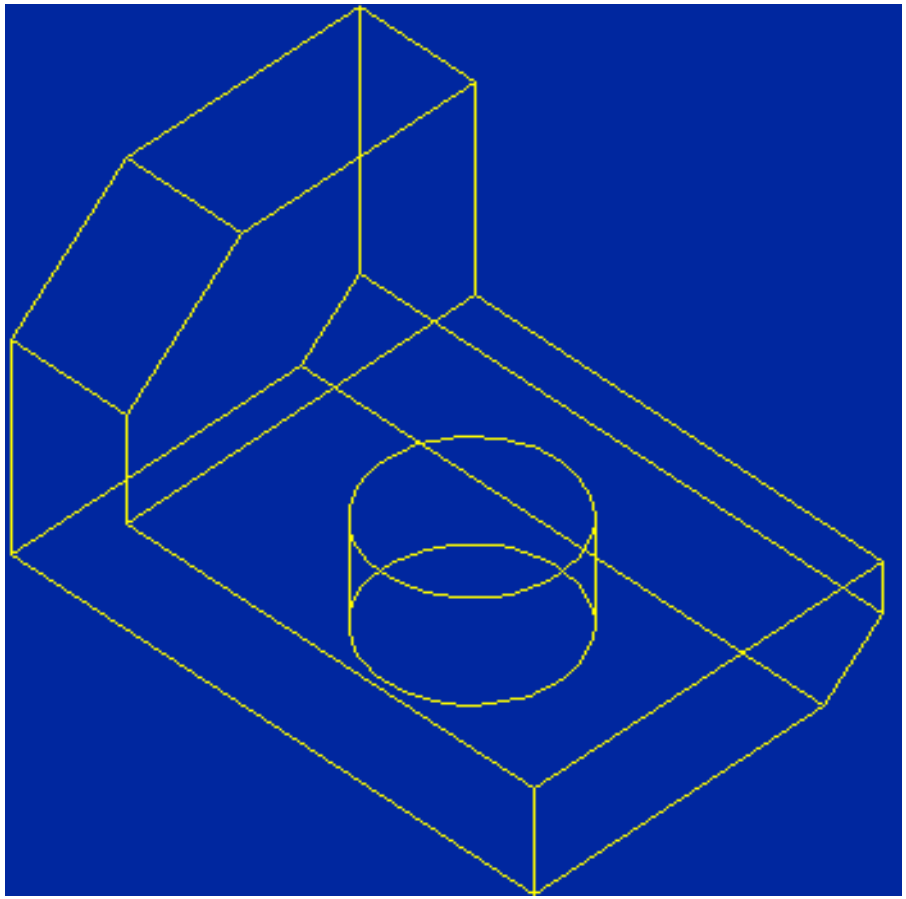


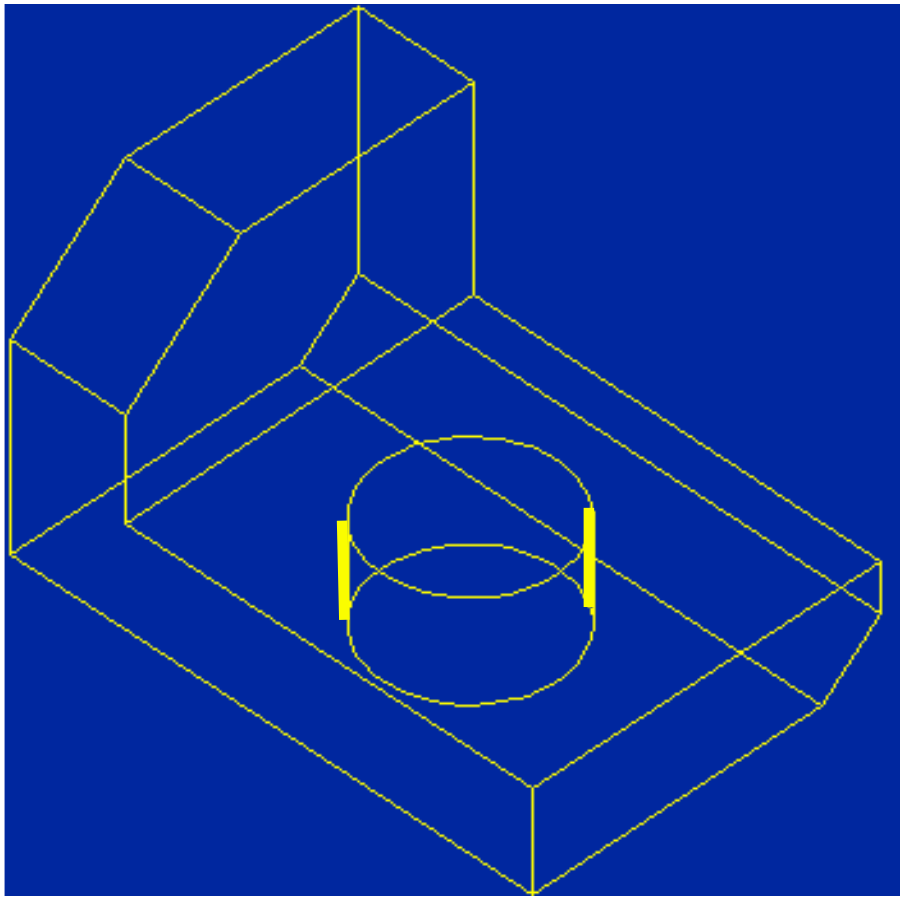
Brep – rappresentazioni wireframe



Brep – rappresentazioni wireframe

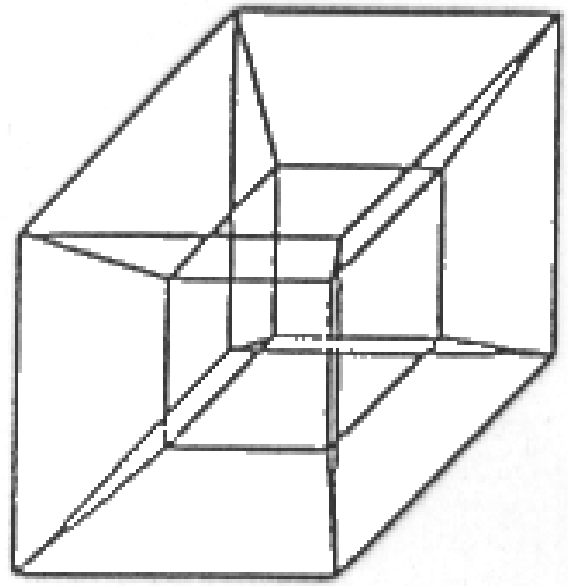
Spigoli reali e spigoli fittizi usati per superfici non planari

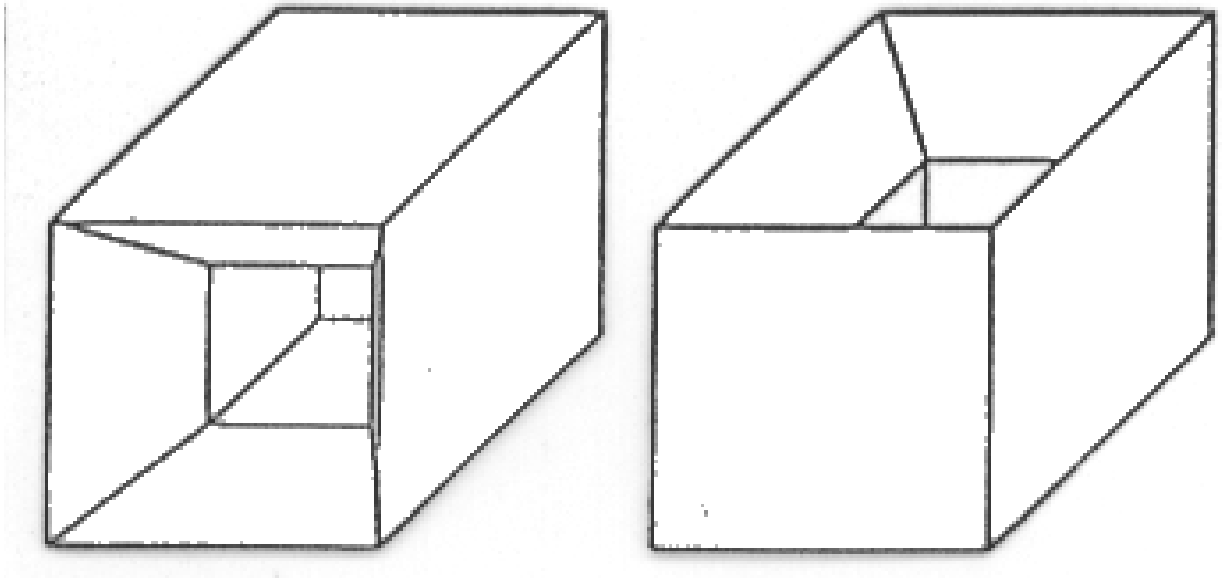




Brep – rappresentazioni wireframe

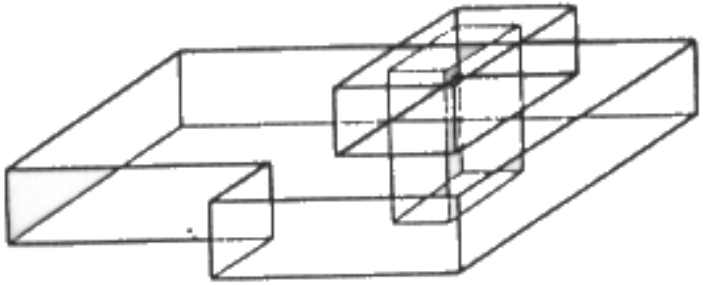
Ambiguità





Brep – rappresentazioni wireframe

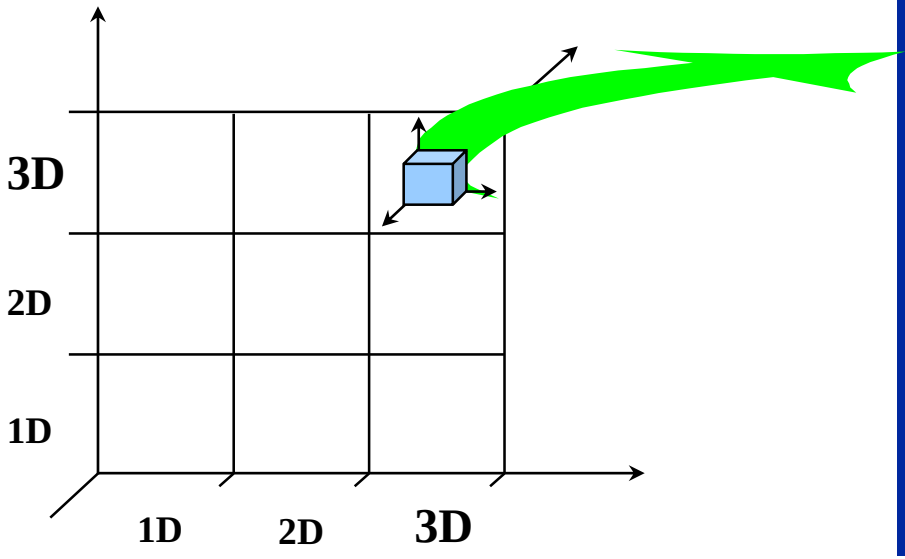
- **Vantaggi**
  - forniscono un modello unico da cui derivare i disegni 2D (rappresentazioni)
  - permette diversi tipi di visualizzazione (viste ortogonali, prospettiche e multiple)
  - semplici, facili da modificare
  - basso costo computazionale

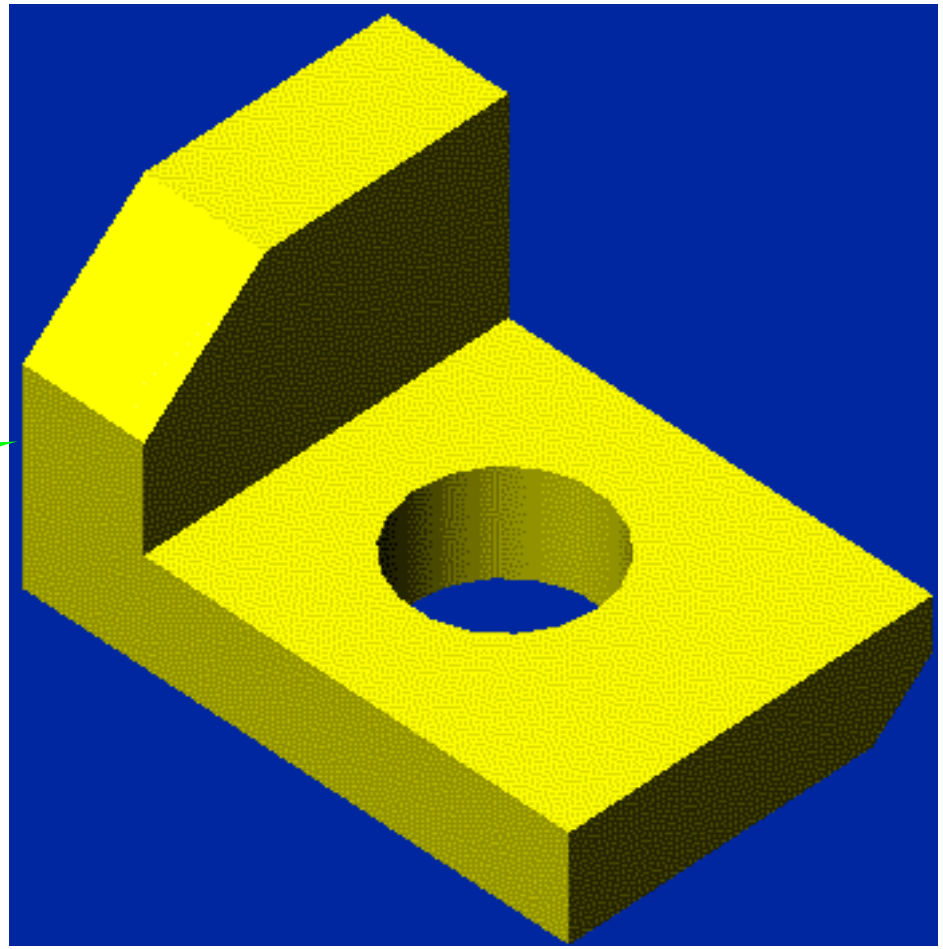


- **Svantaggi**
  - geometria povera (solo spigoli)
  - **ambiguità** nella rappresentazione
  - possibilità di creare oggetti privi di senso
  - non è facile distinguere le linee visibili
  - molto complicati da leggere e interpretare
  - mancanza di standard

Rappresentazioni volumetriche

- **Rappresentazioni per decomposizione o costruttive**





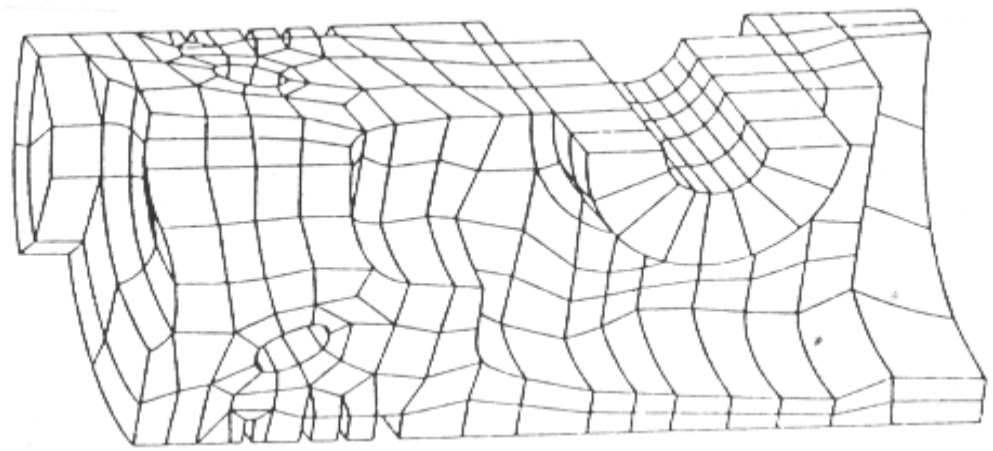
Rappr. Volumetriche - decomposizione

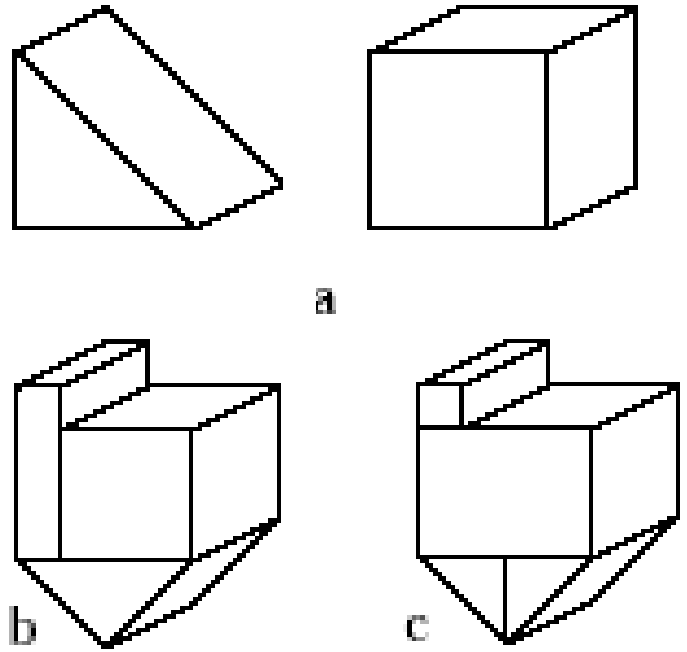
- Solido ottenuto dalla de-composizione in volumi elementari
- I volumi elementari possono avere varie forme e orientazioni
- Gli schemi più comuni:
  - Decomposizione in celle (Cell Decomposition)
    - Es. Mesh per FEM
  - Enumerazione per occupazione spaziale (Spatial-Occupancy Enumeration)
    - Es. Modelli voxel-based
  - Partizione spaziale binaria (Binary Space Partitioning)
    - Es. Modelli quadtree, octree

Rappr. Volumetriche - decomposizione

Decomposizione in celle (1/2)

- **Solido descritto da volumi elementari adiacenti tramite operatore di glueing (saldatura)**
- **Celle sono di varia natura: tetraedri, semplici poliedri, solidi più complessi**
- **Volumi elementari possono essere anche di diversa forma/orientazione**

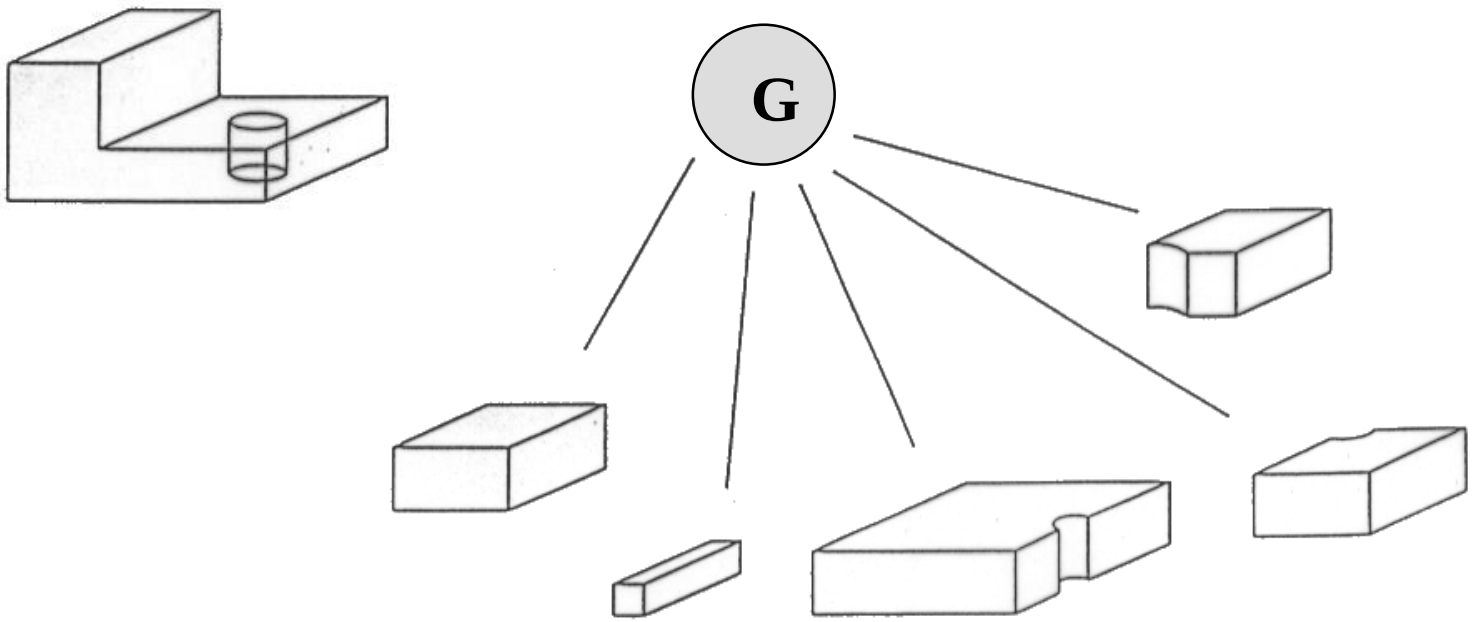




Rappr. Volumetriche - decomposizione

Decomposizione in celle (2/2)

oggetti complessi si ottengono applicando l'operatore di gluing ai volumi elementari

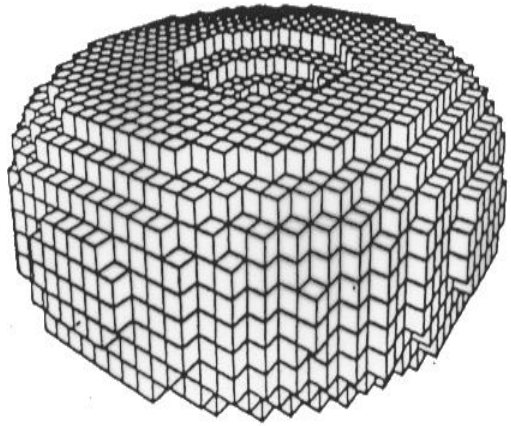


- Gluing: restrizione di operazione di unione, in cui gli oggetti non devono intersecarsi. Inoltre 2 celle devono avere almeno un vertice, uno spigolo o una faccia in comune.

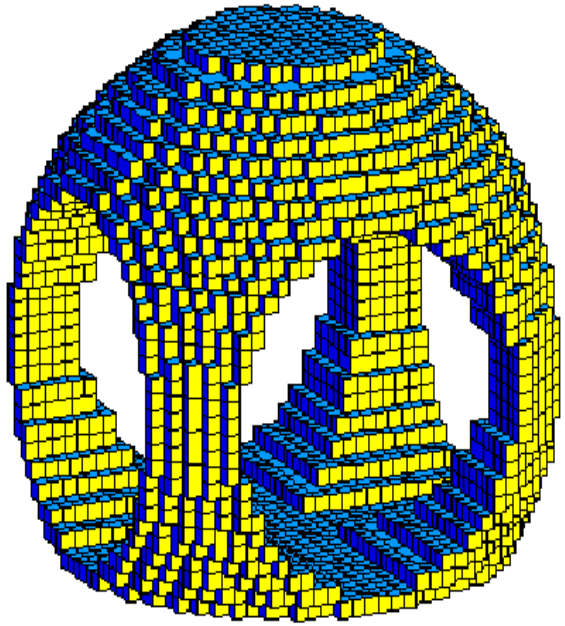
- Difficile verificarne la validità (verifica che tutte le coppie di primitive adiacenti siano "incollate")

Rappr. Volumetriche - decomposizione

Occupazione per enumerazione spaziale



- Volumi elementari: celle di stessa forma ed orientazione
- Combinati in griglie strutturate e regolari
- Ogni cella interamente occupata dal solido o no
- Esempio: modello voxel-based (voxel = cubo)



Operativamente:

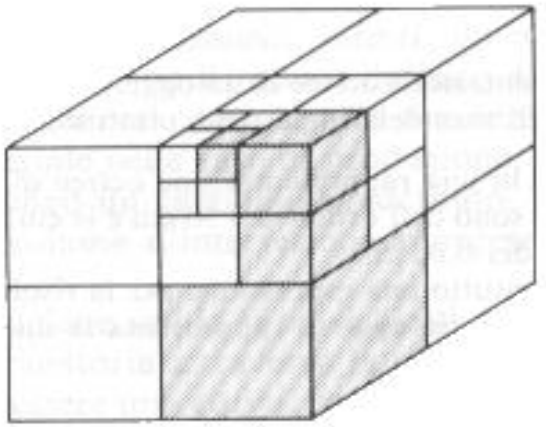
- definizione di una griglia regolare orientata lungo i 3 assi principali
- assegnazione di valore TRUE alle celle che contengono materiale, e FALSE a quelle che non lo contengono
- rappresentazioni di ogni cella mediante le coordinate di un singolo vertice

Rappr. Volumetriche - decomposizione

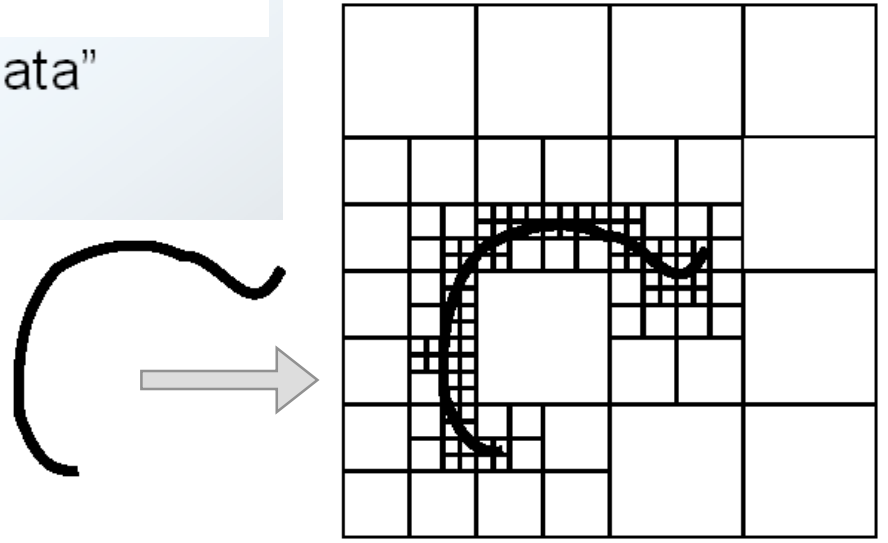
Partizione spaziale binaria (1/2)

- Suddivisione di una regione in n sottoregioni uguali ed equidirette
- Schema ricorsivo: ogni sottoregione è suddivisa a sua volta in n sottoregioni di stessa forma...
- Ottimizzazione: una sottoregione è suddivisa solo se interseca l'oggetto
- Rappresentazione comunque "approssimata"
- Metodi BSP più usati:
  - partizionamento **quad-tree** (2D)
    - suddivisione ricorsiva in 4 quadrati
  - partizionamento **octree** (3D)
    - suddivisione ricorsiva in 8 cubi

Esempio 3D



Esempio 2D

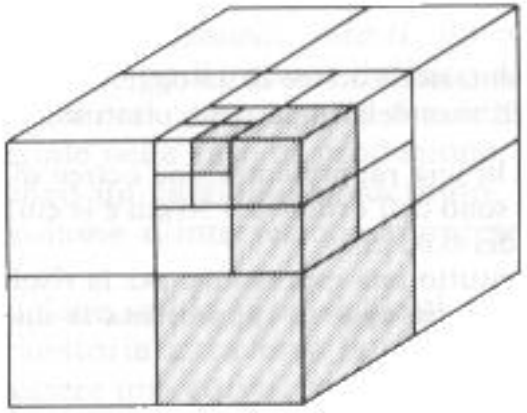


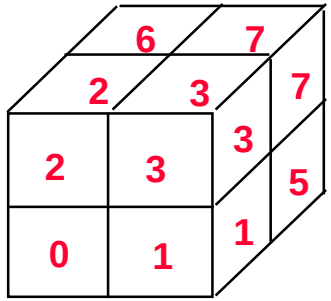
Rappr. Volumetriche - decomposizione

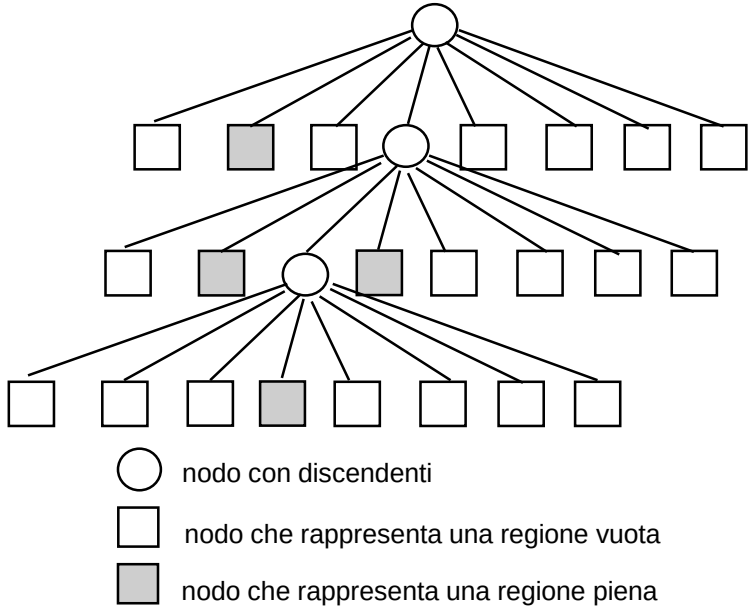
Partizione spaziale binaria (2/2)

Esempio: Modello Octree

- modello è rappresentato da un albero: ogni nodo padre ha 8 figli
- spazio è suddiviso ricorsivamente: ogni passo genera 8 figli
- ad ogni figlio viene assegnato un codice:
  - completamente contenuto nel solido
  - completamente esterno al solido
  - parzialmente contenuto nel solido







Rappr. Volumetriche - decomposizione

	Decomposizione in celle	Enumerazione spaziale	Octree
Dominio	buono	buono con approssimazione	buono con approssimazione
Validità	difficile da validare	facile da validare	facile da validare
Completezza	si	si	si
Unicità	no	si	si
Concisione	no	no	no
Facilità di creazione	non direttamente	non direttamente	non direttamente

Rappr. Volumetriche - decomposizione

- Sono usati quando serve una rappresentazione volumetrica
  (es: analisi FEM agli elementi finiti)

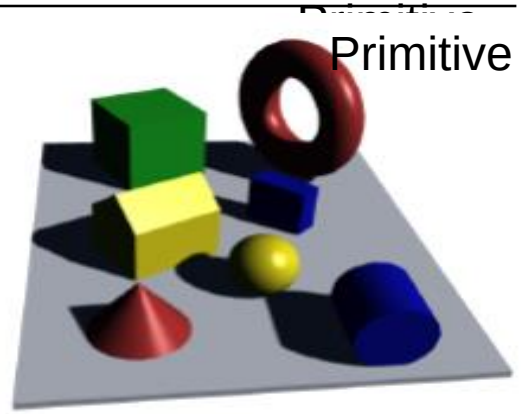
- Richiedono elevata quantita di memoria

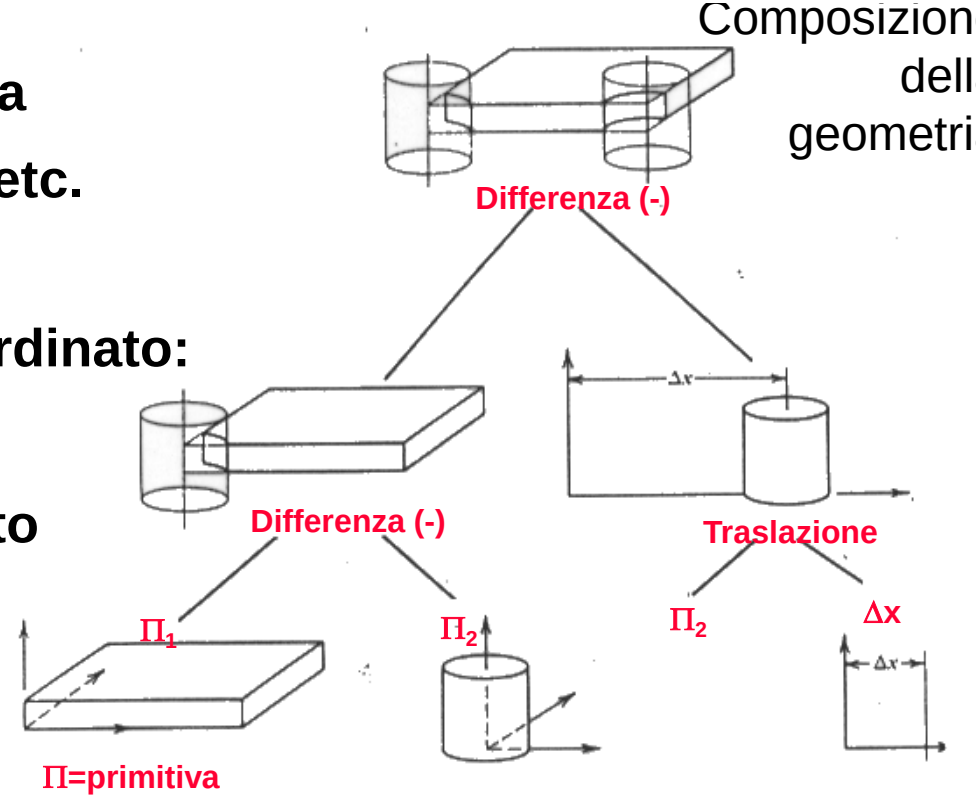
- La conversione da voxel a mesh poligonale è “facile”. Il viceversa è più complesso (computazionalmente costoso).

Rappr. Volumetriche - costruttive

CSG – Constructive Solid Geometry

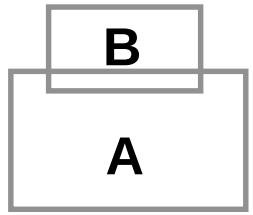
- **primitive geometriche: figure euclidee**
  - **Parallelepipedo, sfera, cilindro, cono, etc.**
- **oggetti ottenuti per composizione delle primitive euclidee**
- **composizione = operazione booleana**
  - **unione, intersezione, differenza, etc.**
  - **traslazione, rotazione, scala, etc.**
- **rappresentazione mediante albero ordinato:**
  - ***ALBERO CSG***
- **modello procedurale: viene modellato COME e non COSA**





Rappr. Volumetriche - costruttive

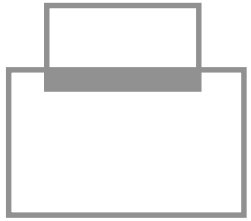
CSG – Constructive Solid Geometry



- **Unione**

$A \cup B$

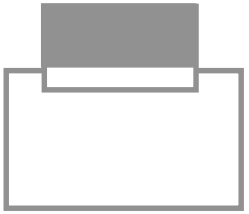
- **Intersezione**



$A \cap B$

- **Differenza**

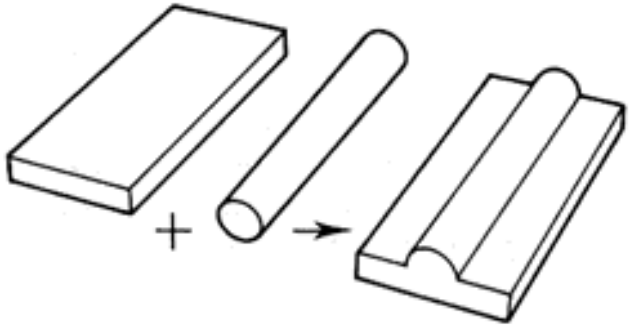
$A - B$



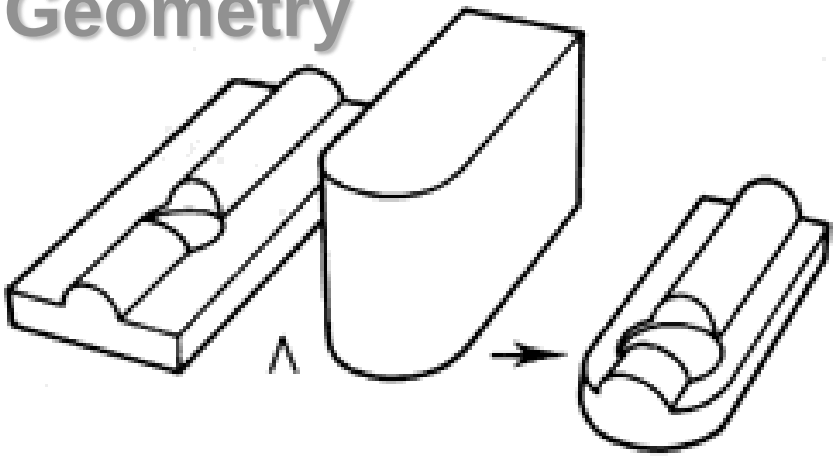
$B - A$

Rappr. Volumetriche - costruttive

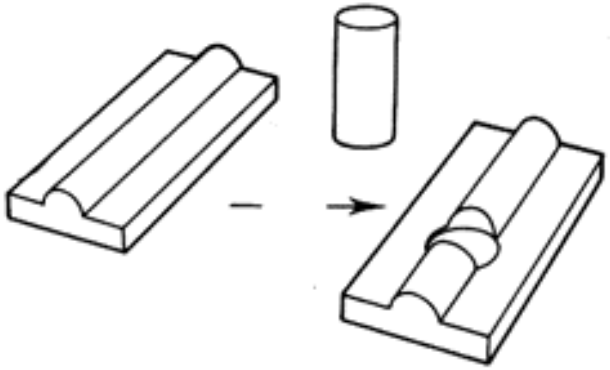
CSG – Constructive Solid Geometry



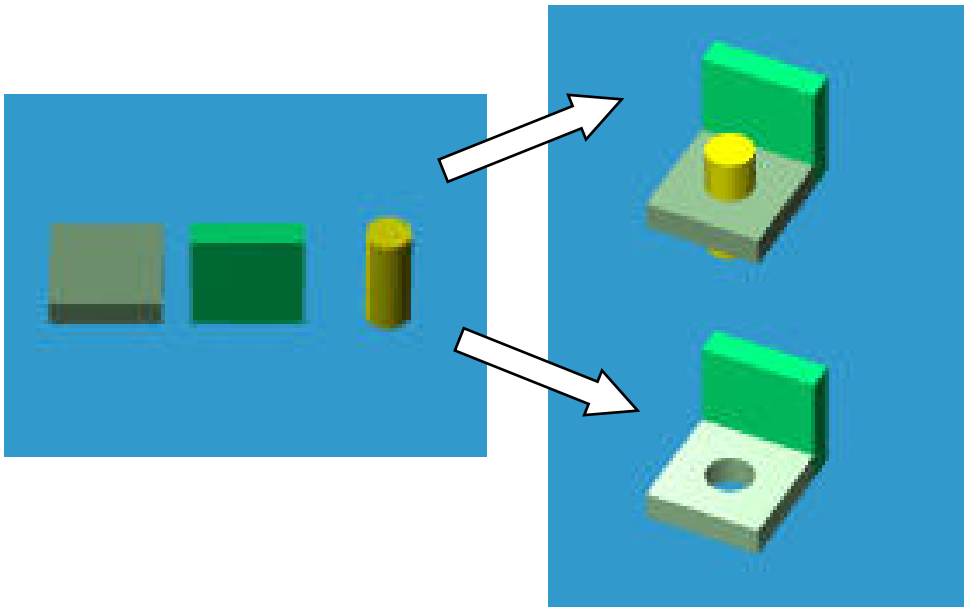
(a)



(c)



(b)



Rappr. Volumetriche - costruttive

CSG – Constructive Solid Geometry

- **Modello procedurale**
  - si descrive come ottenere il solido, non il solido stesso
- **E' possibile convertire una rappresentazione CSG in una rappresentazione mediante il contorno**
- **Vantaggi**
  - metodo procedurale
  - consistenza e validità del risultato
  - facilita' di modifica di un oggetto costruito
- **Svantaggi**
  - rappresentazione non è unica
  - non è possibile verificare se 2 solidi sono uguali o no valutando le rappresentazioni, non fornisce informazioni su facce e spigoli.
  - il numero di operatori usati per creare e modificare i solidi è limitato

Rappr. Volumetriche - costruttive

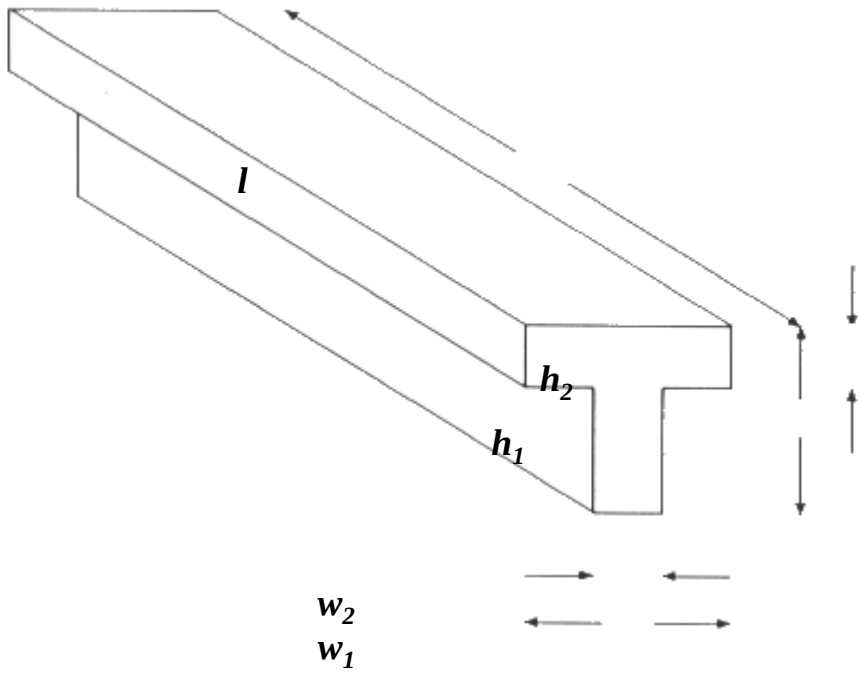
CSG – Constructive Solid Geometry

	CSG
Dominio	buono
Validità	intrinsecamente valido
Completezza	si
Unicità	no
Concisione	si
Facilità di creazione	mediante linguaggio/interfaccia

Rappr. Volumetriche - costruttive

Istanza di primitive

- **basato su famiglie di oggetti**
- **un solido è creato tramite una operazione di istanziazione, specificando alcuni parametri**
- **Usata nei sistemi CAD (es modellazione di viti, ingranaggi ecc…)**



Rappr. Volumetriche - costruttive

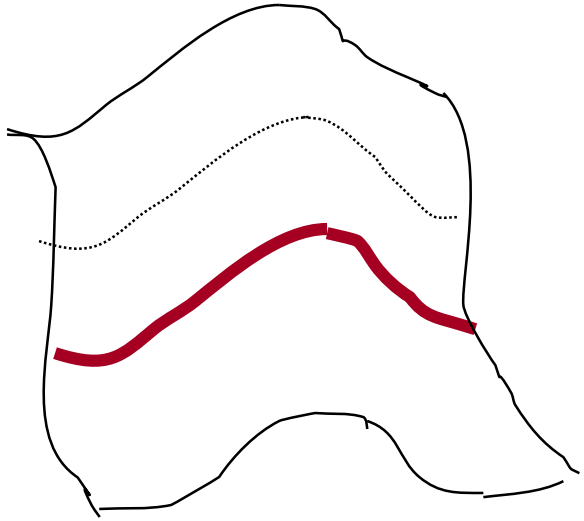
Istanza di primitive

	Istanza di primitive
Dominio	limitato
Validità	facile da validare
Completezza	si
Unicità	no
Concisione	si
Facilità di creazione	si

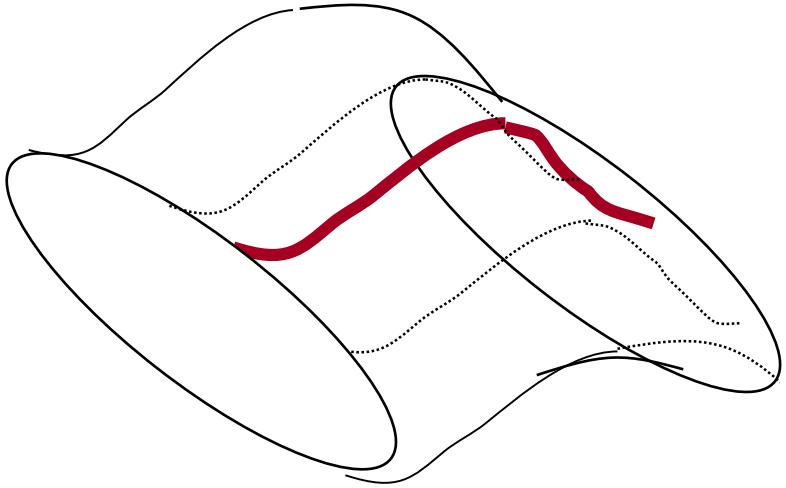
Rappresentazioni di Sweep

Sweeping:

Creazione di una geometria 2D (o 3D) considerando il volume spazzato dal movimento di entità 1D (o 2D) lungo una traiettoria fissata



Sweeping di curva 1D → superficie 2D



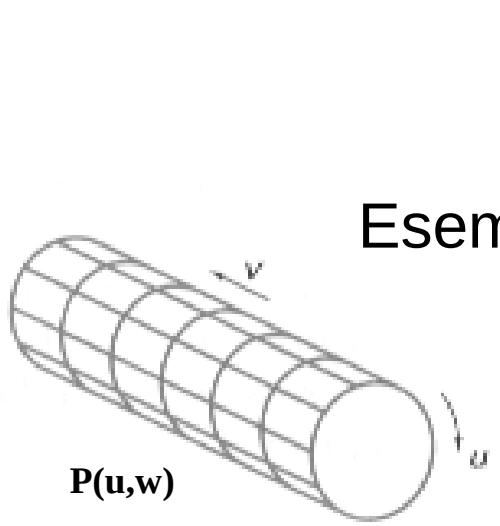
Sweeping di regione 2D → volume 3D

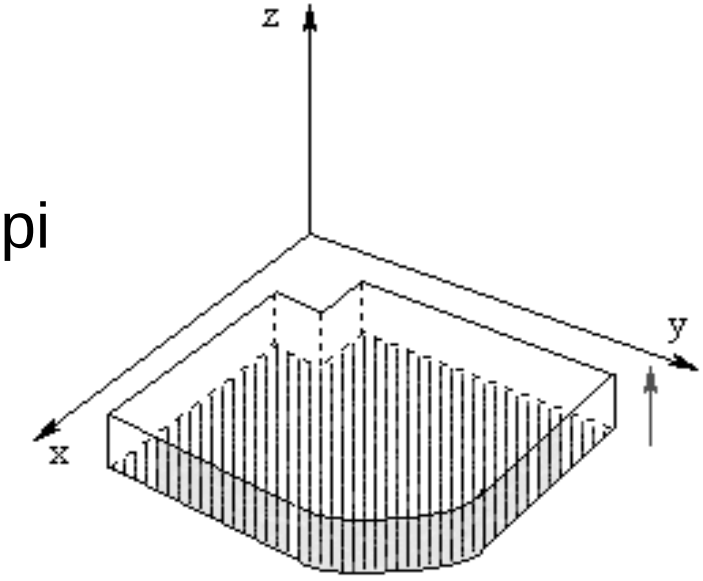
Rappresentazioni di Sweep

Sweep traslazionale (rettilineo)

- **Solido ottenuto estrudendo una superficie piana lungo una traiettoria rettilinea**
- **Solido generato è valido se il vettore di sweep è di lunghezza > 0, e non giace sul piano della superficie**
- **Tecnica utilizzate per modellare processi di lavorazione:**
  - applicazione: simulazione ed analisi di operazioni di asportazione di materiale

Esempi



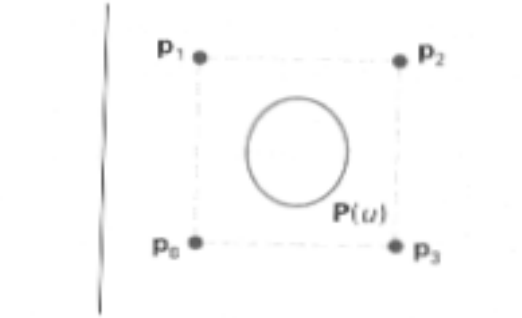


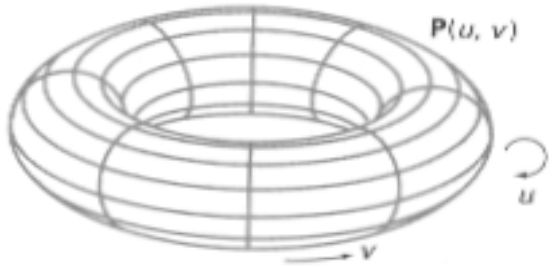
Rappresentazioni di Sweep

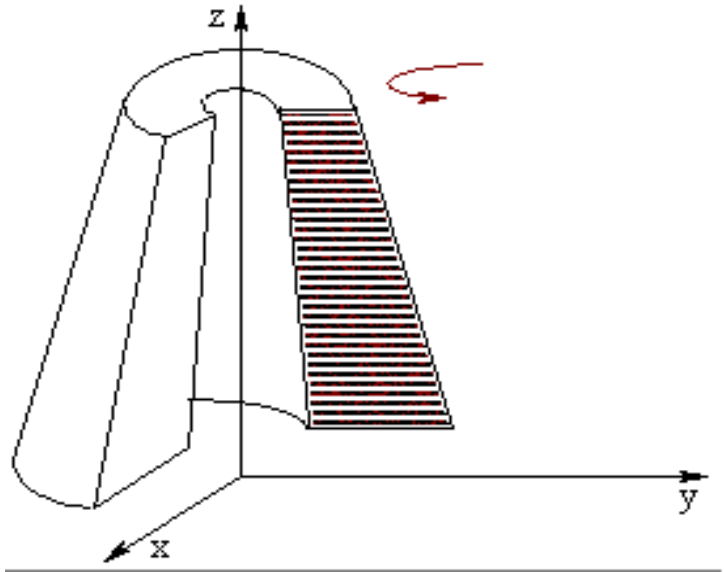
Sweep rotazionale

- **Solido ottenuto ruotando di un certo angolo una superficie piana attorno ad un asse**
- **Solido generato è valido se la superficie piana non interseca l'asse di rotazione e l'angolo di rotazione è:** $0° \leq \alpha \leq 360°$
- **Tecnica utilizzata per modellare processi di lavorazione:**

Esempi







Rappresentazioni di Sweep

Proprietà

	Sweep
Dominio	limitato
Validità	facile da validare
Completezza	si
Unicità	no
Concisione	si
Facilità di creazione	no