

H1 Modelli Impliciti

H2 Descrizione generale

Sono modelli **continui**, non discreti.

Questi modelli sono definiti (*implicitamente*) da una **funzione primitiva** f :

$$f: \mathbb{R}^3 \mapsto \mathbb{R}$$

Dato un punto $p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$, il valore $f(p)$ definisce se p è dentro o fuori

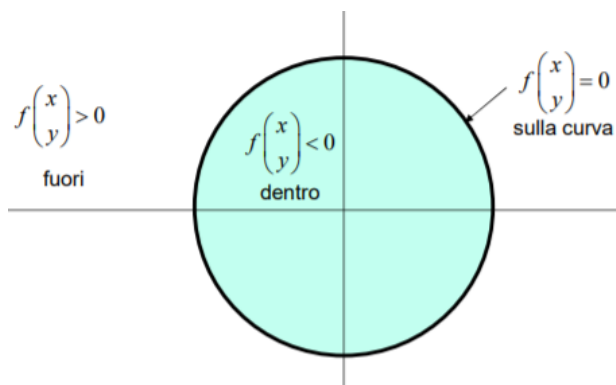
dall'oggetto:

- $f(p) < 0 \iff p$ è dentro
- $f(p) > 0 \iff p$ è fuori
- $f(p) = 0 \iff p$ è sulla superficie

Si dice **superficie implicita** la superficie che delimita il modello implicito.

Quindi, si parla di **superficie implicita** nel 3° caso, cioè è l'insieme di tutti i punti p tali che $f(p) = 0$.

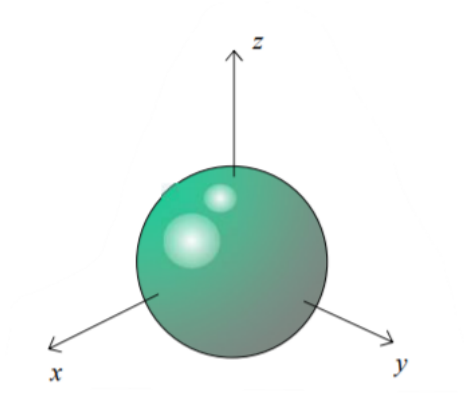
Possiamo fare un'esempio semplificato di questo tipo di rappresentazione riducendoci a 2 dimensioni:



dove f è così definita:

$$f\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = x^2 + y^2 - r^2$$

Espandendo di nuovo alle 3 dimensioni, avremo:



abbiamo 2 possibili primitive per la sfera centrata nell'origine:

$$f \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sqrt{x^2 + y^2 + z^2} - r \quad (1)$$

cioè:

$$f(p) = \|p - 0\| - r$$

oppure:

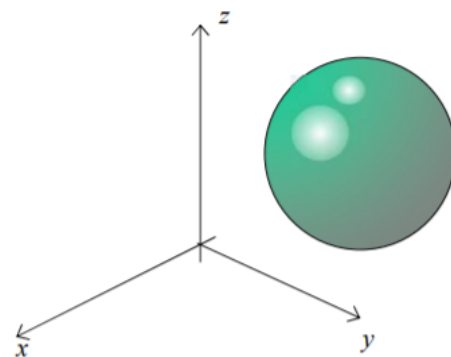
$$f \begin{pmatrix} x \\ y \\ z \end{pmatrix} = x^2 + y^2 + z^2 - r^2 \quad (2)$$

cioè:

$$f(p) = \|p - 0\|^2 - r^2$$

le funzioni (1) e (2) sono la stessa superficie.

Invece per una generica sfera nello spazio centrata nel punto c :



$$f(p) = \|p - c\| - r \quad (3)$$

Da notare come la funzione (3) rispecchi perfettamente la definizione di superficie di una sfera:

La superficie della sfera è il luogo dei punti p che distano r dal suo centro c

In base al grado della primitiva, avremo superfici implicite di diverse categorie:

- Superfici algebriche : $f(p)$ è un *polinomio*
- Superfici quadratiche : $f(p)$ è di *grado 2*
- Superfici cubiche : $f(p)$ è di *grado 3*
- Superfici quartiche : $f(p)$ è di *grado 4*
- e così via...

H2 Normali

Come definiamo le normali di una superficie implicita? Più precisamente, dato un punto p sulla superficie, cioè $f(p) = 0$, quanto vale la normale della superficie in p ?

Si calcola il **gradiente** di $f(p)$ e lo si normalizza:

$$\hat{n} = \frac{\nabla f(p)}{||\nabla f(p)||}$$

dove

$$\nabla f(p) = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$$

dalla definizione di **vettore gradiente** in un punto in \mathbb{R}^3 :

Sia $f(p)$ una funzione $\mathbb{R}^3 \mapsto \mathbb{R}$

Il suo **gradiente** $\nabla f(p)$ nel punto p è definito dal vettore delle sue *derivate parziali* in x, y, z calcolate in p

Tornando all'esempio della sfera, la cui funzione primitiva è la (2), avremo il suo gradiente pari a:

$$\nabla f(p) = \nabla f \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2x \\ 2y \\ 2z \end{pmatrix} = 2p$$

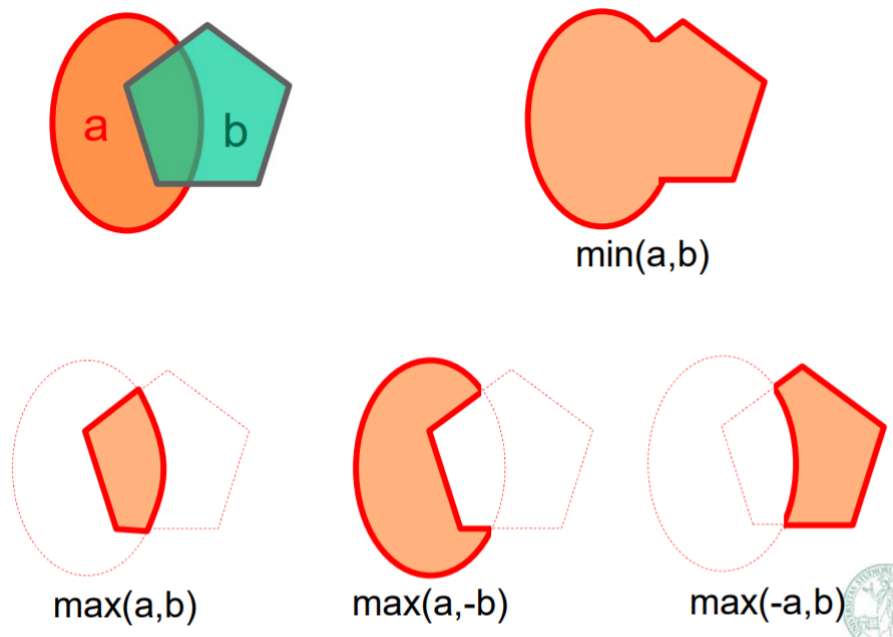
da cui la normale in p di una sfera è:

$$\hat{n} = \frac{2p}{||2p||}$$

H2 Composizione con operazioni booleane

Sembrerebbe che questo tipo di modelli non abbiano un potere espressivo molto elevato, in quanto sarebbe difficile modellare un oggetto complesso (ad es. un drago) usando un'unica funzione primitiva.

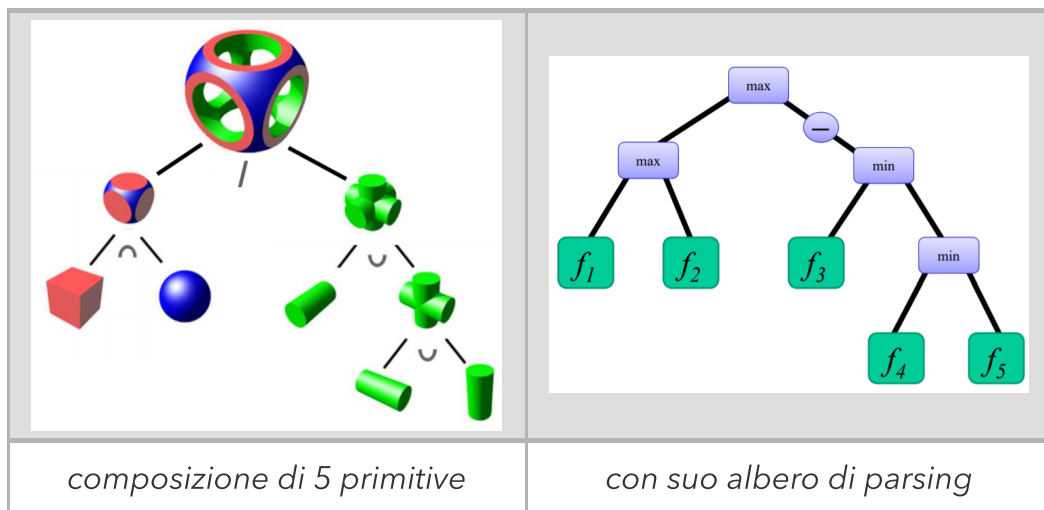
Quello che si fa è combinare più primitive per costruire figure più complesse, e per farlo vengono usate le **operazioni booleane volumetriche**, che si traducono in semplici confronti fra le primitive.



riassumendo le operazioni in una tabella:

Operazione volumetrica booleana	Vista come operazione fra insiemi di punti	Vista come operazione booleana	Operaz delle funzione primitiva
Opposto	\bar{A}	$\sim A$	$-f_A$
Intersezione	$A \cap B$	$A \wedge B$	$\max(f_A, f_B)$
Unione	$A \cup B$	$A \vee B$	$\min(f_A, f_B)$
Scavo di B da A	$A \cap \bar{B}$	$A \wedge \sim B$	$\max(f_A, -f_B)$
Scavo di A da B	$\bar{A} \cap B$	$(\sim A) \cap B$	$\max(-f_A, f_B)$

Vediamo un esempio concreto:



H3 Composizioni booleane tra mesh

Può essere utile eseguire le operazioni booleane appena viste tra mesh, ma per farlo dovremo prima **convertire una mesh in un modello implicito**. In realtà possiamo fermarci a convertire una mesh in un dataset voxelizzato, in quanto questo può essere visto come caso particolare di modello implicito. Infatti la sua primitiva f sarà definita, dentro ogni cella cubica della griglia 3D, come l'*interpolazione* degli 8 voxel memorizzati agli angoli di tale cella.

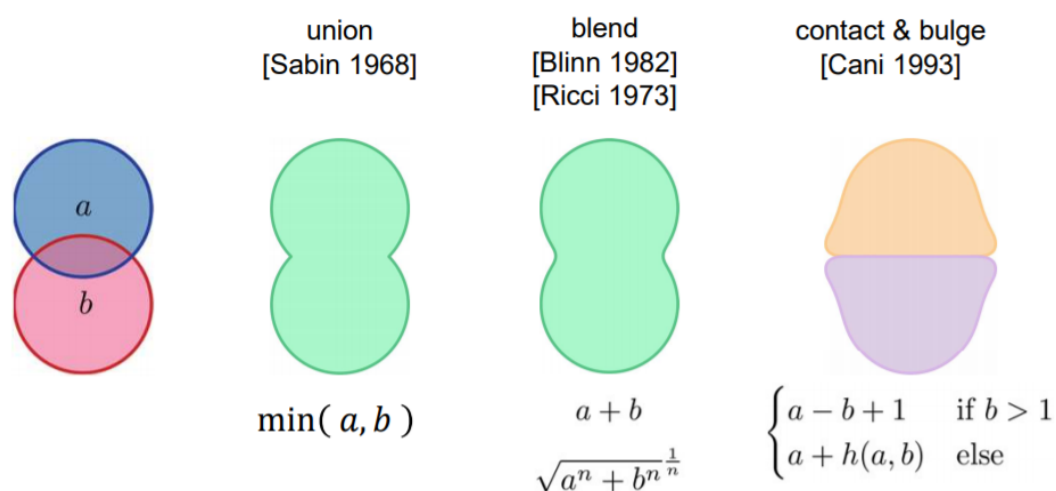
Per farlo dovremo:

1. "Immergere" la mesh in una griglia 3D di risoluzione data `resX, resY, resZ`
2. Computare per ogni vertice della griglia la sua distanza (con segno) dalla mesh

NOTA: La mesh dovrà essere 2-manifold, chiusa e ben orientata

H3 Altri tipi di composizione

Per molti oggetti può essere utile unire le primitive con operazioni diverse da quelle booleane per avere ancora più capacità espressiva. Nella prossima figura vediamo degli esempi:



H2 Altre caratteristiche

H3 Attributi

Il calcolo degli attributi avviene assegnando a ogni primitiva un attributo. Ad esempio, posso assegnare alla primitiva f_A l'attributo

`colore = rosso`, e a f_B l'attributo `colore = blu`.

H3 Rendering

Per il rendering dei modelli impliciti spesso si passa a usare il modello voxelizzato come rappresentazione di passaggio. Operativamente, per passare da un modello implicito a una mesh poligonale faremo:

1. campionare f ad una data risoluzione `resX, resY, resZ`, ottenendo un campo scalare voxelizzato
2. usare l'algoritmo *marching cubes* con $\sigma = 0$ per estrarre la relativa mesh poligonale M
3. calcolare le normali dei vertici di M attraverso i gradienti di f calcolati nelle posizioni dei vertici

Quindi si renderizza la mesh ottenuta in questo modo.

Un'alternativa per renderizzare direttamente il modello implicito è l'uso del *ray-tracing*.

H2 Vantaggi e svantaggi

- ✓ Sono rappresentazioni volumetriche, in quanto è possibile calcolare se un punto è dentro o fuori dall'oggetto.
- ✓ Sono molto compatti, in quanto basta memorizzare le primitive!
- ✓ Rappresentano superfici curve, non le approssimano come le mesh per esempio!
- ✓ Sono ottime per rappresentare oggetti artificiali o meccanici
- ✓ La continuità della rappresentazione fa sì che non vi sia risoluzione
- ✗ Il rendering non è diretto a meno di uso di *ray casting*
- ✗ Hanno comunque una capacità espressiva limitata, e non sono adatte all'editing