


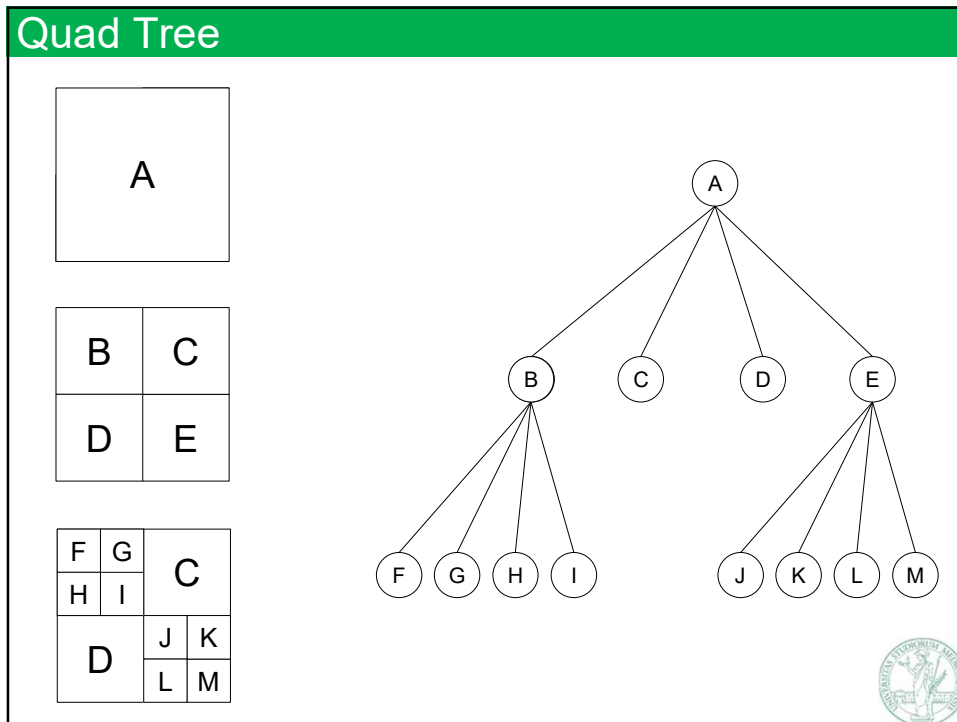
94

### Dataset volumetrici a griglia adattivi

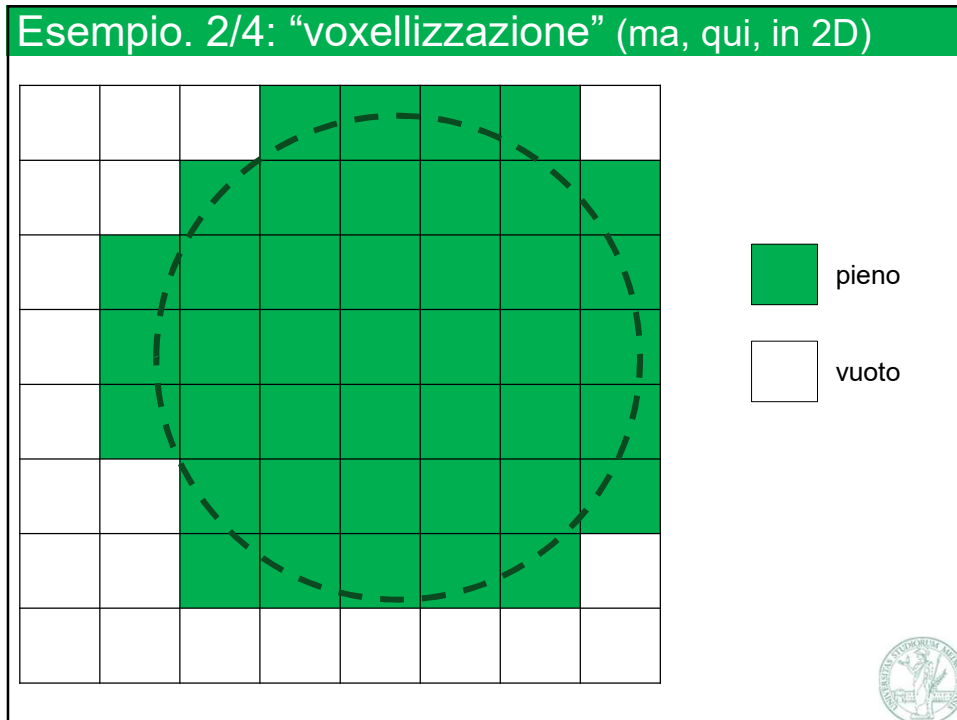
- ✓ Il problema della «curse of dimensionality» è legato ad una caratteristica del dato a voxel: la sua risoluzione non è *adattiva*
- ✓ Esistono strutture dati **gerarchiche** che sono più compatte, grazie alla loro risoluzione adattiva
- ✓ Una delle più diffuse è l'**oct-tree** (o anche **octree**)
  - ⇒ Il soggetto della prossima lezione!
  - ⇒ Vediamo prima una sua versione 2D: il «**quad-tree**»
  - ⇒ Cioè: in questi primi esempi, ipotizziamo di voler rappresentare, in forma di quad-tree, un dataset “voxelizzato” (ma in 2D) con 1 bit per voxel (1 = pieno, 0 = vuoto) (in sostanza, un’immagine in bianco e nero)



95



96



101

### Esempio. 3/4: quad-tree (scomposizione)

0	0	1	1	1	0
	1	1		1	1
0	1	1	1	1	
0	1				
0	1	1	1	1	
0	0				
0	1	1	1	1	0
	0	0	0	0	0

```

graph TD
    A((A)) --- B((B))
    A --- C((C))
    A --- D((D))
    A --- E((E))
            
```

102

### Esempio. 4/4: quad-tree (l'albero risultante)

103

### Un altro esempio

Come quad-tree:  
solo 28 foglie  
+9 nodi interni

Prova a seguire nel dettaglio l'esempio, ricordando che, nel disegno:

nota l'ordine!

Come volume di voxel equivalente:  
8x8 = 64 voxel

105

### Quad tree: un altro esempio

RADICE (LVL 0)

LIVELLO 1

LIVELLO 2

LIVELLO 3

LIVELLO 4

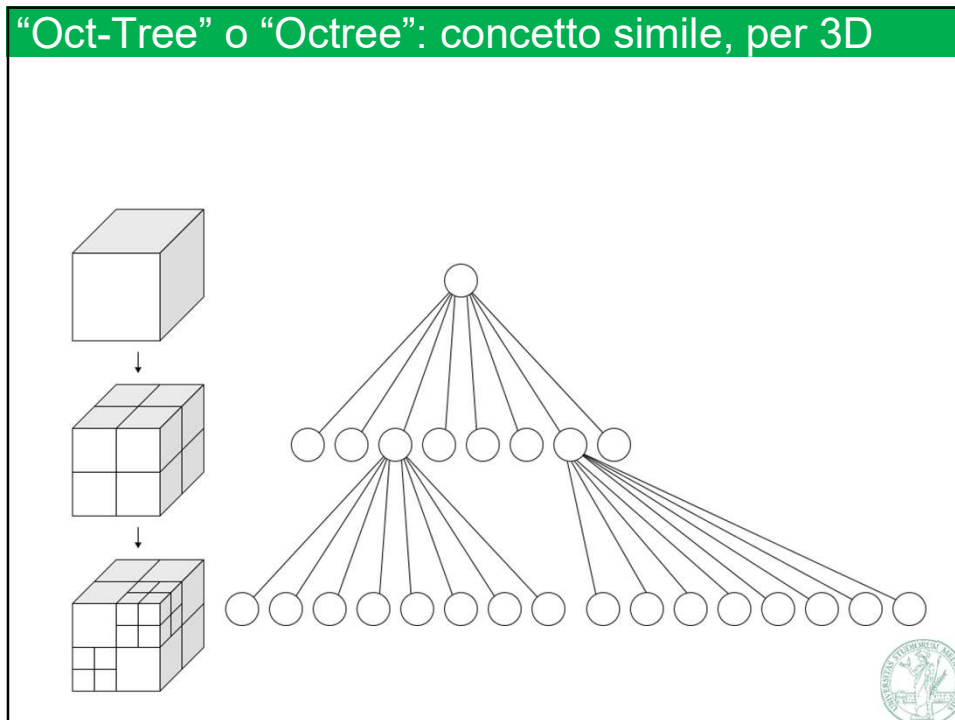
LIVELLO 5

LIVELLO 6

LIVELLO 7

Ad ogni livello, solo alcuni dei nodi vengono ulteriormente suddivisi (gli altri, sono foglie di quel livello).  
 In questo esempio, l'octree codifica un disegno composto da 4 linee (il resto è vuoto).

106

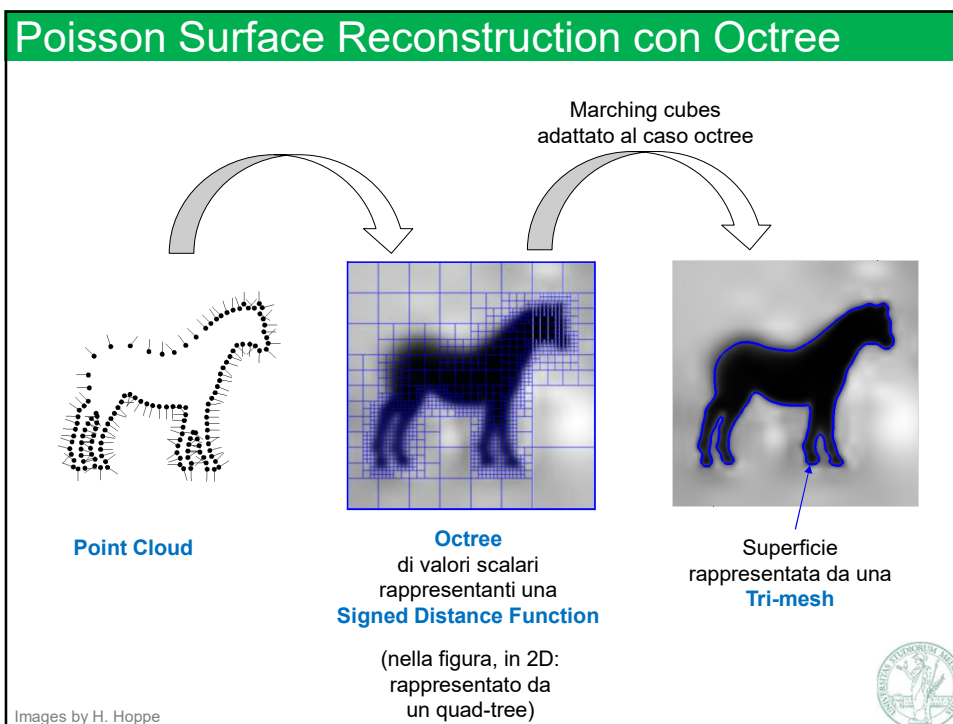


107

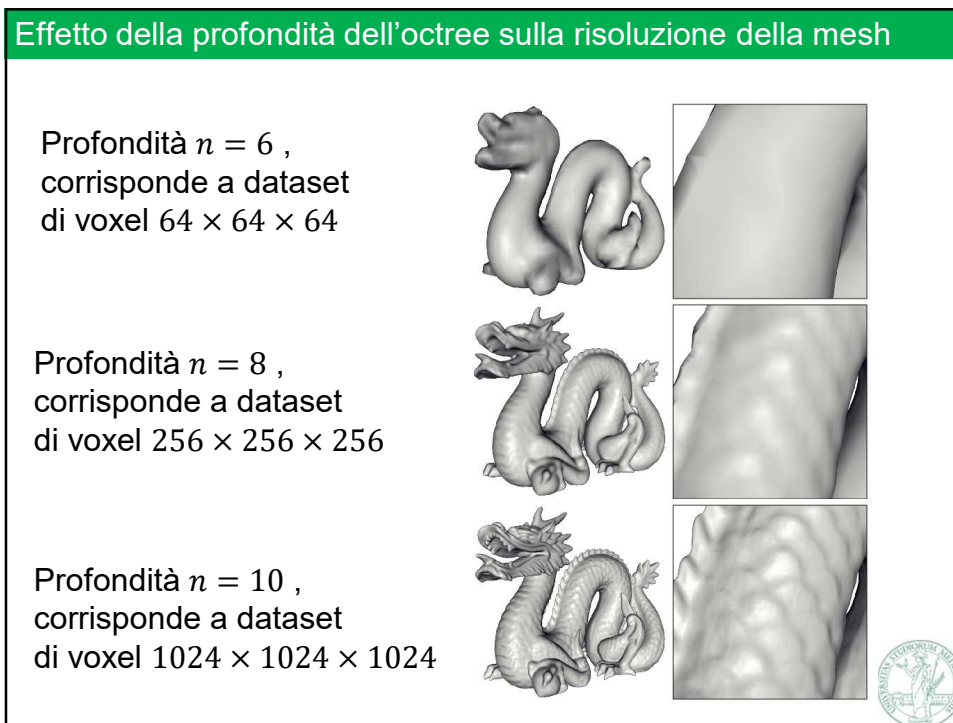
### Octree (oppure oct-tree): sommario

- ✓ Struttura ricorsiva, ad albero
- ✓ Ogni nodo è associato ad una porzione cubica di volume (di dim. variabile)
  - ⇒ La **Radice**: nodo associato all'intero volume
  - ⇒ Ogni **nodo non-foglia** ha 8 figli, uno per ciascun ottavo del padre
  - ⇒ Ogni **foglia**: memorizza un **voxel**... di dimensione che dipende dal livello (ad esempio, 1 bit – pieno o vuoto; oppure, un valore di *Signed Distance* – l'octree rappresenta un SDF)
- ✓ Per memorizzare una struttura octree di profondità  $n$  come voxel sarebbe necessaria una risoluzione lineare  $N = 2^n$  per lato, per un totale di  $N \times N \times N = 2^n \times 2^n \times 2^n = 2^{3n}$  voxel
  - ⇒ Un numero estremamente grande anche per  $n$  piccoli
  - ⇒ Per es, per  $n = 10$  ... circa 1 miliardo
- ✓ Invece, il numero totale di nodi memorizzati in un **octree** è usualmente molto minore
  - ⇒ tende ad essere quadratico (non cubico!) con la risoluzione  $N$  del modello rappresentato  $O(N^2) = O(2^{2n})$

108



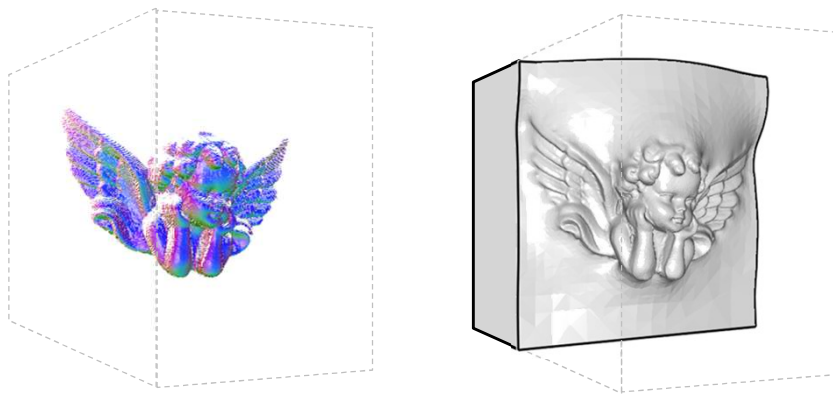
109



111


**Nota: la poisson reconstruction produce mesh chiuse**

Tecnicamente, l'isosuperficie è sempre una mesh chiusa.  
Quando la nuvola di punti di partenza descrive un solo lato dell'oggetto (tipico, nelle range scans) ...



Nuvola di punti  
(point splatting)

Mesh risultante



112