


Marco Tarini - Computer Graphics 2022/2023  
 Università degli Studi di Milano

## Poisson Reconstruction e Oct-Tree

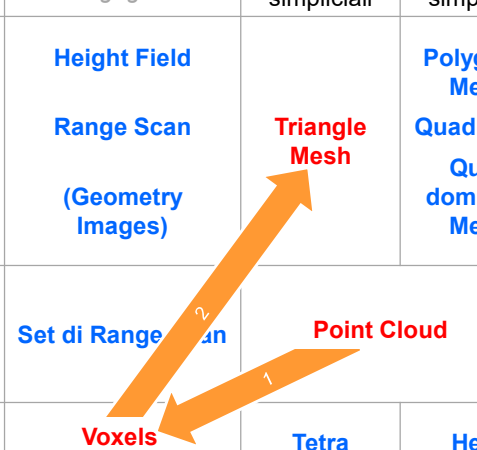


The slide features a background watermark of the University of Milan logo, which includes a figure holding a teapot. In the foreground, there is a wireframe teapot on the left and a black point cloud of a dragon-like creature on the right.

110

Da Nuvola di punti a Triangle Mesh passando per...

		ELEMENTI DISCRETI			CONTINUI
		regolari <i>«a griglia»</i>	semi-regolari o irregolari		
			elementi simpliciali	elementi non simpliciali	
SUPERFICIALI	2-manifold <i>«rappresenta una vera superficie»</i>	Height Field Range Scan (Geometry Images)	Triangle Mesh	Polygonal Mesh Quad-Mesh Quad dominant Mesh	Subdivision surface Parametric Surface (es. B-splines)
	non-manifold <i>«non rappresenta una sup»</i>	Set di Range Scan	Point Cloud		
VOLUMETRICI	(3-manifold)	Voxels Solid Textures	Tetra Mesh	Hexa Mesh	Implicit model (es. CSG)



Two orange arrows originate from the 'Voxels' cell. Arrow 1 points to the 'Point Cloud' cell, and arrow 2 points to the 'Triangle Mesh' cell.

111

## Da nuvola di punti a mesh

- ✓ Modi per convertire una nuvola di punti in una mesh
  - ⇒ Modo diretto: algoritmi di Front Advancing (come Ball Pivoting)  
si aggiunge una connettività di triangoli per connettere i punti della nuvola (potenzialmente, scardandone alcuni)  
-- il modo che abbiamo visto fin'ora
  - ⇒ Modo indiretto:  
si converte la nuvola di punti in un dataset volumetrico di voxel  
(che campiona nel volume una stima della *Signed Distance Function - SDF*)  
e si estrare da questo dataset una mesh poligonale (con algoritmo Marching Cubes con soglia 0)



112

## Strutture volumetriche come dataset intermedio

- ✓ Poisson reconstruction
  - ⇒ Un modo comune di convertire una point cloud in una mesh che consiste nel è passare attraverso a un struttura volumetrica a voxel
- ✓ Idea:
  - ⇒ 1. Stendere una griglia volumetrica (o «lattice») nel volume coinvolto
  - ⇒ 2. Memorizzare una «signed distance function» nel volume (1 valore scalare per voxel, che memorizza una stima della distanza dalla superficie, negativo se il punto è dentro la superficie)
  - ⇒ 3. estrarre iso-superficie (attraverso marching cubes)
- ✓ Nel passo 2: ogni punto della nuvola di posizione  $p$  e normale  $n$  «vuole» che...
  - ⇒ ... il valore della funzione in  $p$  sia 0
  - ⇒ ... il gradiente della funzione in quella posizione sia allineato a  $n$
  - ⇒ Si tratta quindi di computare un volume di voxel che rappresenti una SDF



113

### Poisson Reconstruction

#### Esempio in 2D

The diagram illustrates the 2D Poisson reconstruction process. It starts with a **Point Cloud** of a cow's outline. This is processed into a **Signed Distance Function campionata nei voxel (1 voxel = 1 scalar)**, represented as a grid with black and white regions. Finally, the **marching cubes** algorithm is used to extract the **Surface (isosurface)**, shown as a blue outline of the cow.

Images by Micheal Kazhdan

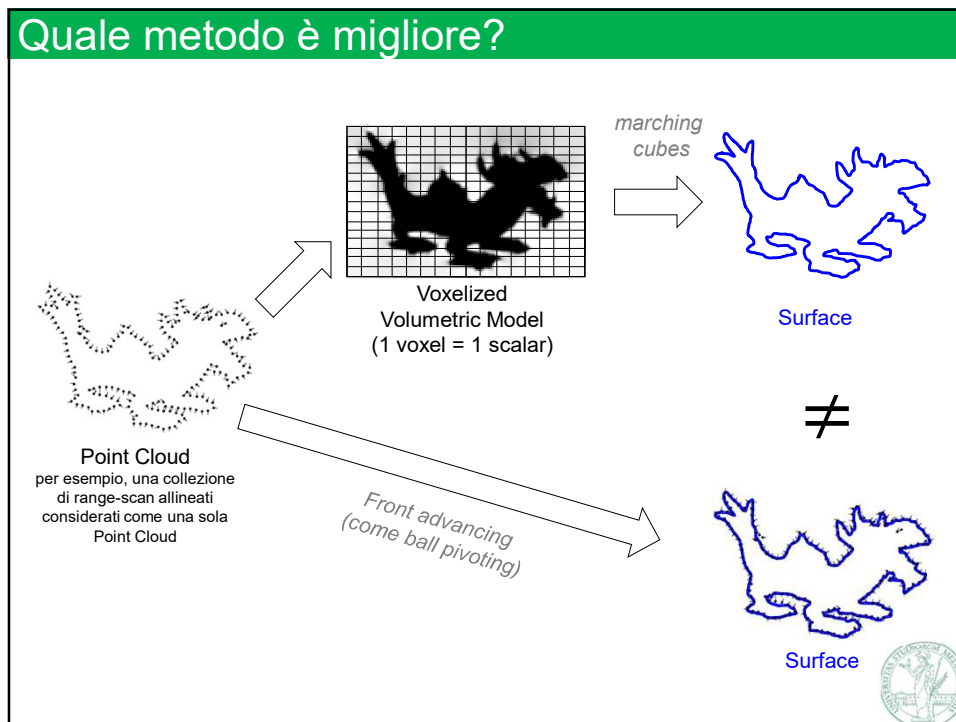
114

### Poisson Reconstruction

The diagram illustrates the 3D Poisson reconstruction process. It starts with a **Point Cloud** of a cow. This is processed into a **Signed Distance Function campionata nei voxel (1 voxel = 1 scalar)**, represented as a 3D volume. Finally, the **Marching cubes** algorithm is used to extract the **Mesh (isosuperficie)**, shown as a smooth 3D model of the cow.

Images by H. Hoppe


115



116

### Da nuvola di punti a mesh

- ✓ Nei **metodi diretti** (ball pivoting, etc) si mantengono i punti della point cloud inalterati, come vertici della mesh
- ✓ questo è un problema quando la point cloud presenta difetti come:
  - ⇒ rumore o outliers
  - ⇒ difetti di allineamento (se la nuvola di punti è ottenuta allineando e unendo due nuvole separate, come spesso è il caso)
  - ⇒ densità diverse non volute, accidentali (per es, due point-cloud parziali sovrapposte presentano molti più campioni nelle parti ripetute)
  - ⇒ parti mancanti
- ✓ Sono tutti difetti comuni nelle point-cloud generate da acquisizioni 3D



117

## Da nuvola di punti a mesh

### ✓ Con la **Poisson Reconstruction**...

- ⇒ vengono prodotti nuovi vertici che in sostanza mediano le posizioni dei punti nella nuvola di input, abbattendo rumore e difetti in modo naturale;
- ⇒ viene sempre prodotta una mesh two-manifold, chiusa, e ben orientata

### ✓ Ma...

- ⇒ la mesh prodotta è molto irregolare; (anche se i punti di input fossero molto ben distribuiti)
- ⇒ anche se la nuvola di punti fosse stata a risoluzione adattiva, la mesh prodotta non la è più;
- ⇒ i vertici vengono ricampionati: è un difetto, se i punti originali sono molto accurati;
- ⇒ è molto oneroso in termini di computazione e soprattutto memoria, a meno che non venga scelta una risoluzione molto bassa – *curse of dimensionality* della rappresentazione voxel in azione
- ⇒ vediamo ora un modo generale per rimediare a questo problema (per qualsiasi struttura basata su voxel, non solo per questo uso)



118

## Dataset volumetrici a griglia adattivi

✓ Il problema della «curse of dimensionality» è dovuto all'uso di risoluzione di un dataset di voxel non è *adattiva*

✓ Esistono strutture dati più compatte, perchè dotate di risoluzione adattiva

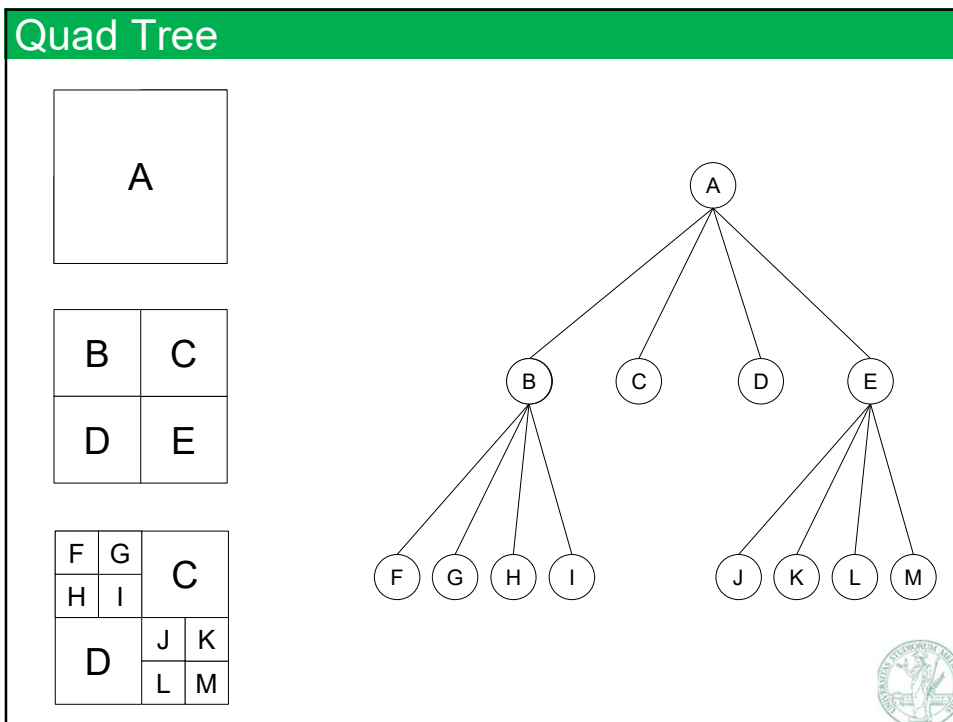
✓ Una delle più diffuse è l'oct-tree

⇒ Vediamo prima una sua versione 2D:  
il quad-tree

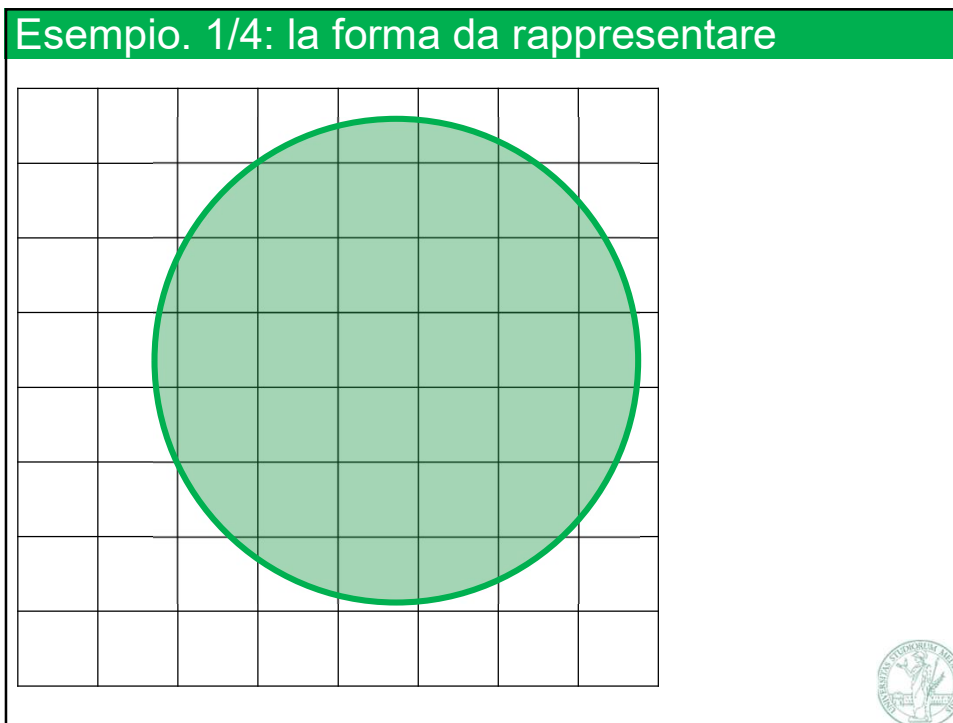
⇒ In questi primi esempi, ipotizziamo di voler rappresentare in forma di quad-tree un dataset "voxelizzato" (ma in 2D) con 1 bit per voxel (1 = pieno , 0 = vuoto)



119



120



124

### Esempio. 2/4: "voxellizzazione" (ma in 2D)

pieno  
 vuoto

125

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

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

```

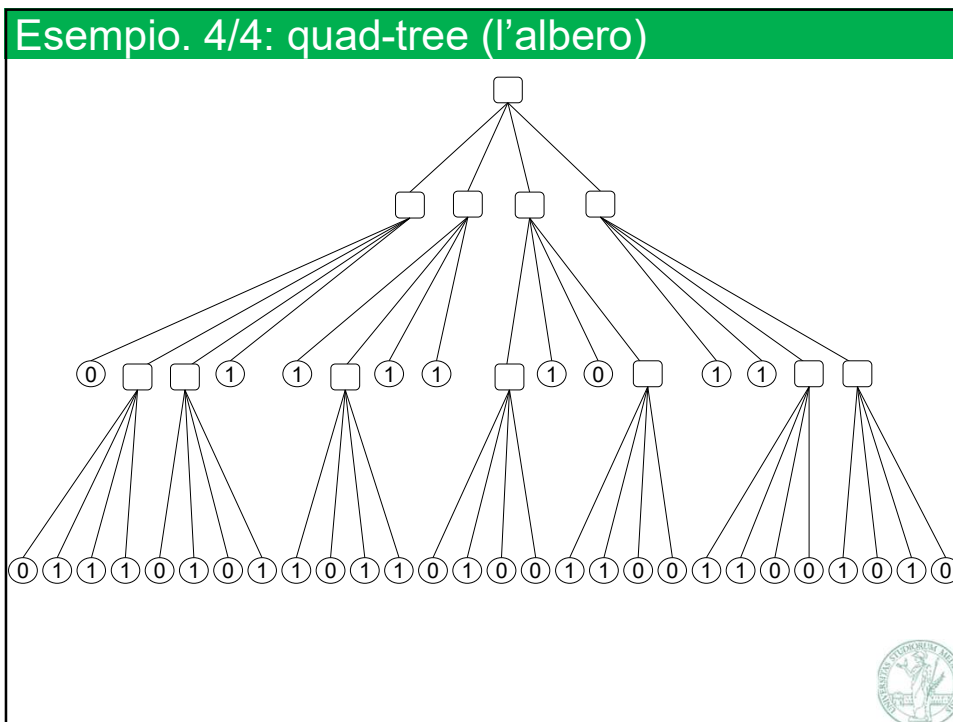
            graph TD
            A((A)) --- B((B))
            A --- C((C))
            A --- D((D))
            A --- E((E))
            
```

A

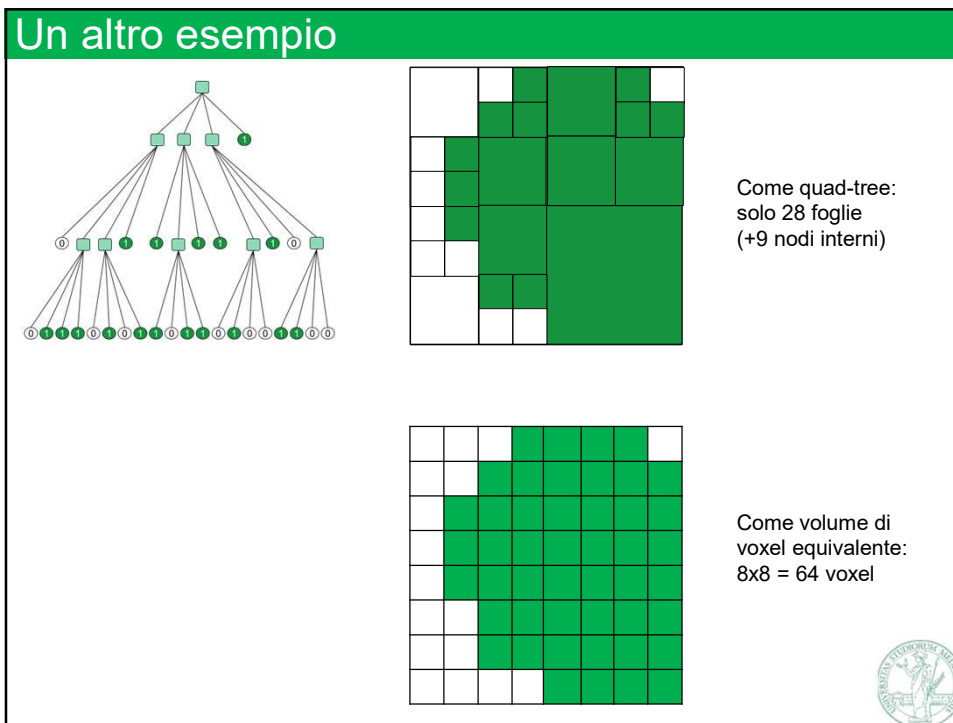
```

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

126

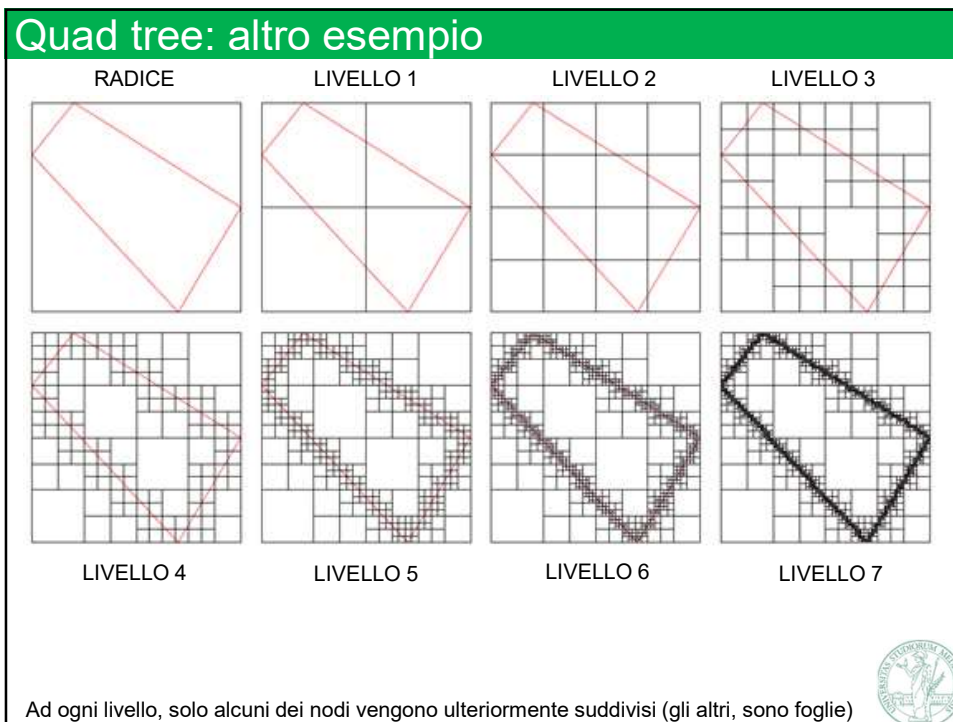


127

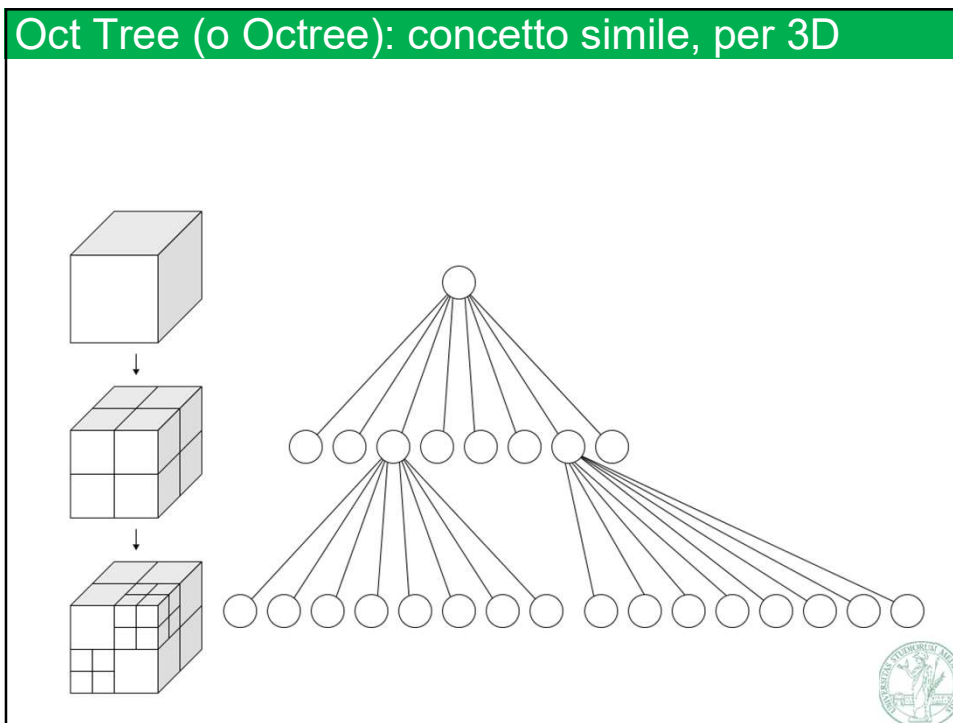


129






130



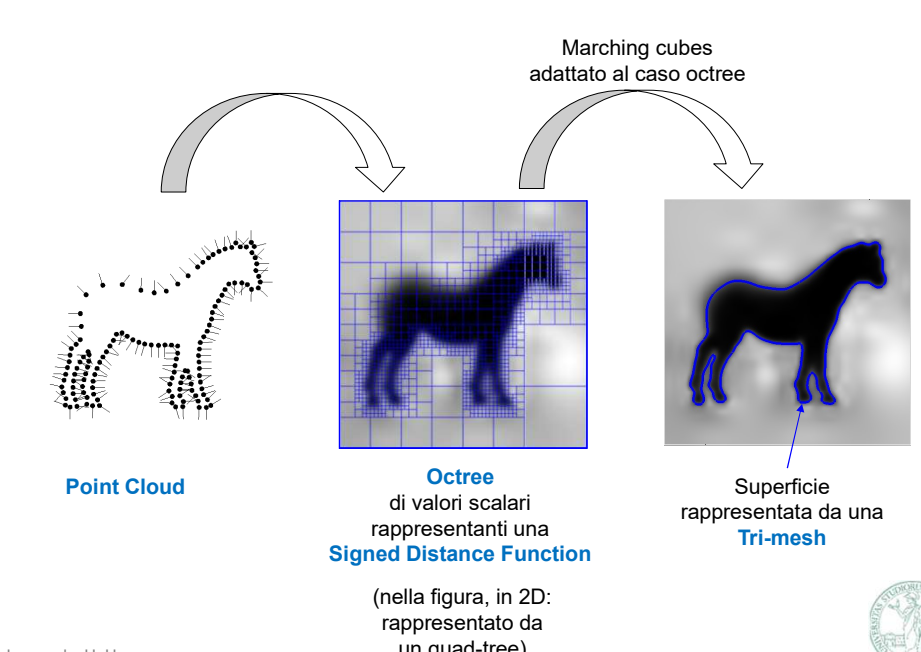
### Octree: sommario

- ✓ Struttura ricorsiva, ad albero
- ✓ Un'alternativa molto più efficiente a dati di voxel
- ✓ Ogni nodo è associato ad un cubo nel volume
  - ⇒ 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 esmpio, 1 bit – pieno o vuoto oppure un valore di signed distance field)
- ✓ Per memorizzare una struttura octree di profondità  $n$  come voxel sarebbero necessari  $2^n \times 2^n \times 2^n$  voxels =  $2^{3n}$ 
  - ⇒ Un numero estremamente grande anche per  $n$  piccoli
  - ⇒ Per es, per  $n = 10$  ... (soluzione: circa 1 miliardo)
- ✓ Invece, il numero totale di nodi memorizzati in un **octree** è usualmente molto minore
  - ⇒ tende ad essere quadratico (e non cubico!) con la risoluzione  $2^n$  del modello rappresentato



132

### Poisson Surface Reconstruction con Octree




Marching cubes  
adattato al caso octree

**Point Cloud**

**Octree**  
di valori scalari  
rappresentanti una  
**Signed Distance Function**  
(nella figura, in 2D:  
rappresentato da  
un quad-tree)

Superficie  
rappresentata da una  
**Tri-mesh**



Images by H. Hoppe

133