Marco Tarini - Computer Graphics 2023/2024
Università degli Studi di Milano

Poisson Reconstruction e Octree

110

Da Nuvola di punti a Triangle Mesh passando per…

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

111

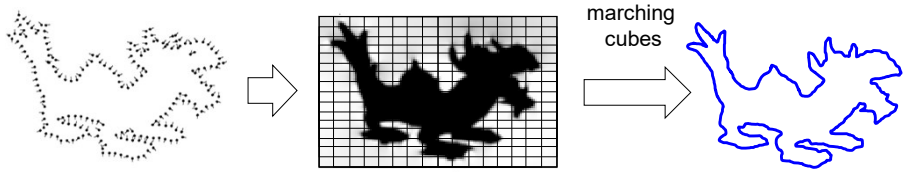
Strutture volumetriche come dataset intermedio

- ✓ 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 $\mathbf{p}$ e normale $\hat{\mathbf{n}}$ «vuole» che…
  - ⇒ $f(\mathbf{p}) = \mathbf{0}$ (cioè, il valore della funzione in $\mathbf{p}$ sia 0)
  - ⇒ $\nabla f(\mathbf{p}) = \hat{\mathbf{n}}$ (cioè, il gradiente della funzione in $\mathbf{p}$ sia il vettore $\hat{\mathbf{n}}$)
  - ⇒ Si tratta quindi di computare un volume di voxel che rappresenti una funzione f che aderisca il meglio possibile a queste richieste, per tutti i punti
  - ⇒ cioè, una funzione che sia una SDF

112

Poisson Reconstruction

Esempio in 2D



Point Cloud

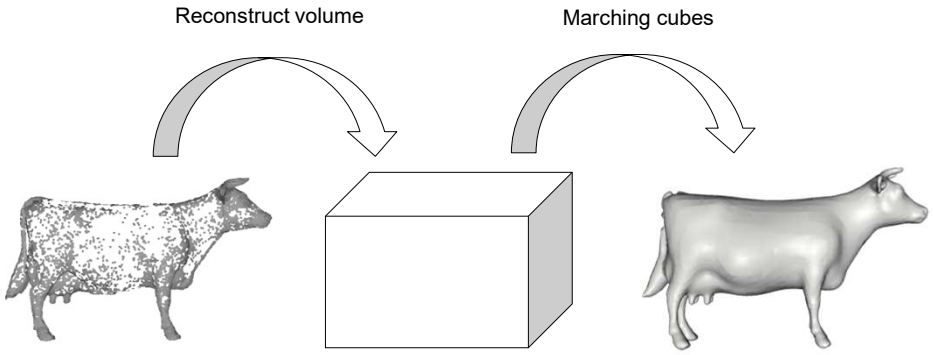
SDF
Signed Distance Function
campionata nei voxel
(1 voxel = 1 scalar)
anche detto
Signed Distance Field

Tri-Mesh
(isosurface)

Images by Micheal Kazhdan

114

Poisson Reconstruction



Point Cloud

SDF
Signed Distance Function
campionata nei voxel
(1 voxel = 1 scalar)
anche detto
Signed Distance Field

Tri-Mesh
(isosurface)

Images by H. Hoppe

115

Da nuvola di punti a mesh: sommario

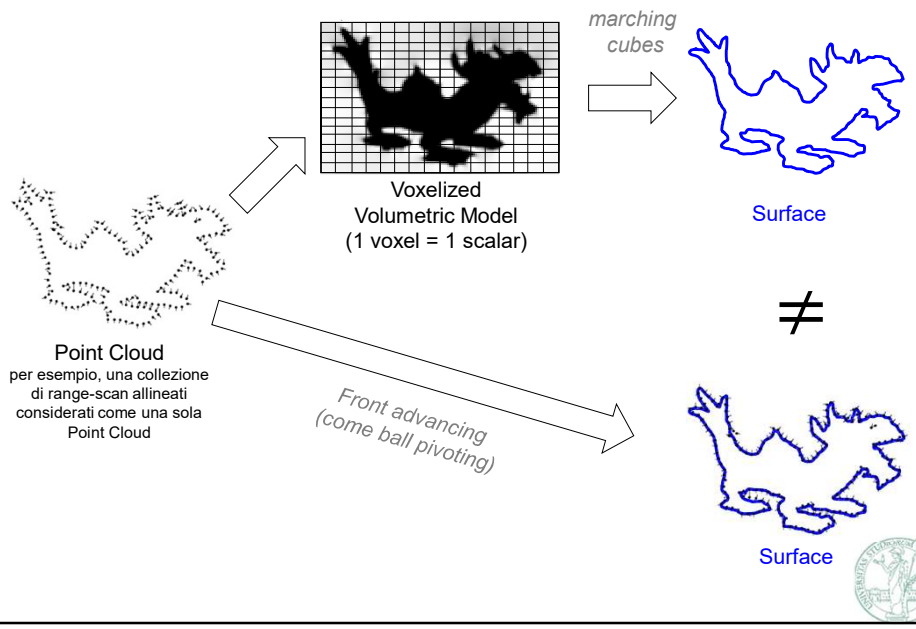
✓ Modi per convertire una nuvola di punti in una mesh

⇒ 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)

116

Un paragone fra due approcci



117

Da nuvola di punti a mesh

- ✓ Nei **metodi diretti** (ball-pivoting, o altri front advancing) si mantengono i punti della point-cloud inalterati, come vertici della mesh
- ✓ Questo è problematico, 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

118

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
  - ⇒ 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 avesse avuto una risoluzione adattiva,
    la mesh prodotta perde questa caratteristica;
  - ⇒ i vertici vengono ricampionati:
    è uno svantaggio, quando i punti originali sono molto accurati;
  - ⇒ è molto oneroso in termini di computazione e memoria,
    a meno che non venga scelta una risoluzione molto bassa,
    che produce mesh a risoluzione bassa;
    (è la solita «*curse of dimensionality*» della rappresentazione voxel)
  - ⇒ vediamo ora un modo generale per rimediare a quest'ultimo problema
    (per qualsiasi struttura basata su voxel – non solo per questo uso)

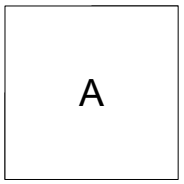
119

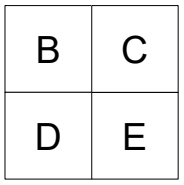
Dataset volumetrici a griglia adattivi

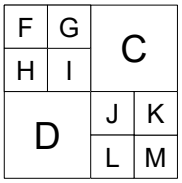
- ✓ Il problema della «curse of dimensionality» è legato ad una caratteristica del dato a voxel: la sua risoluzione non è *adattiva*
- ✓ Esistono strutture dati gerarchiche che sono più compatte, grazie alla loro risoluzione adattiva
- ✓ Una delle più diffuse è l'oct-tree
  - ⇒ Vediamo prima una sua 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)

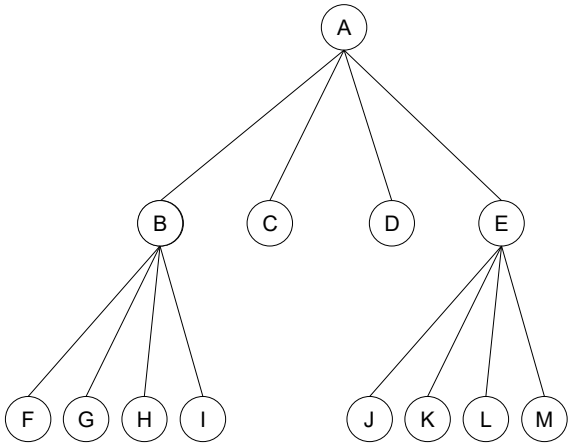
120

Quad Tree



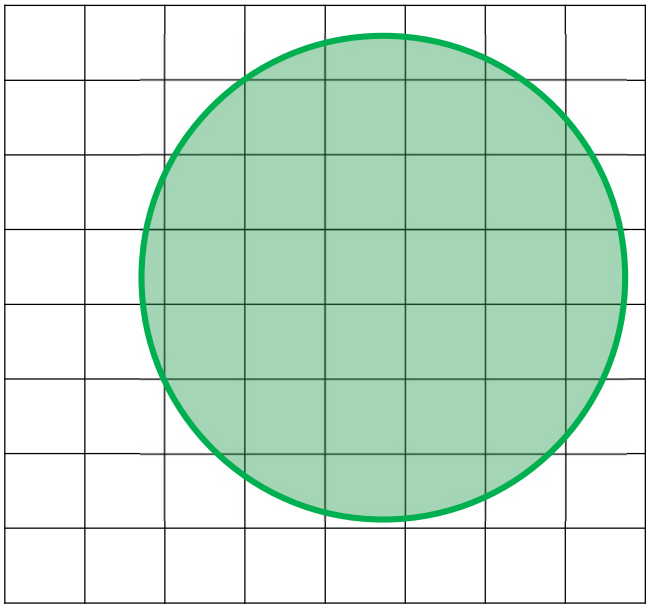






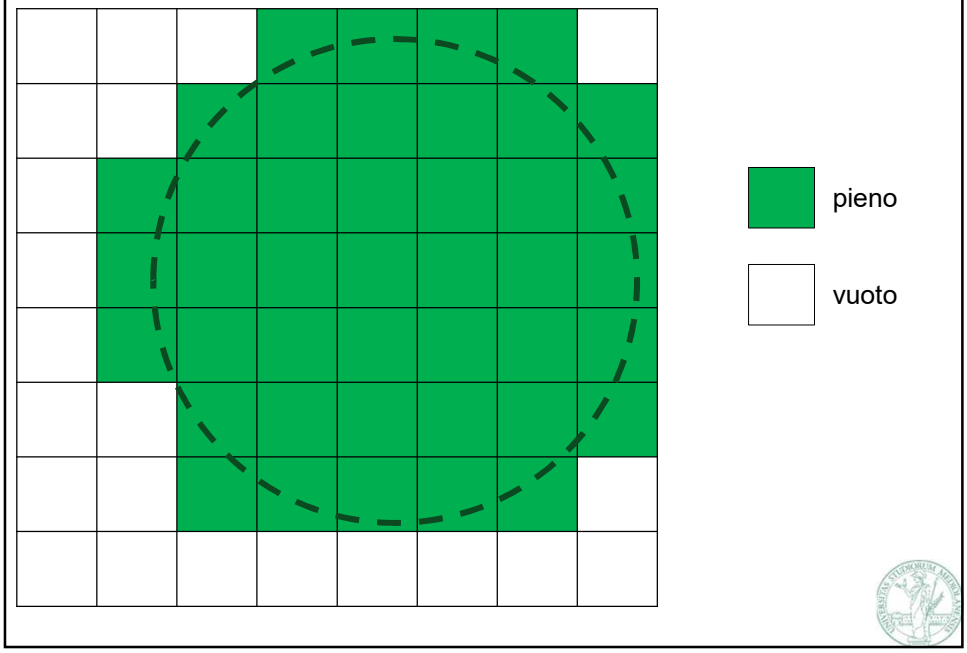
121

Esempio. 1/4: la forma da rappresentare



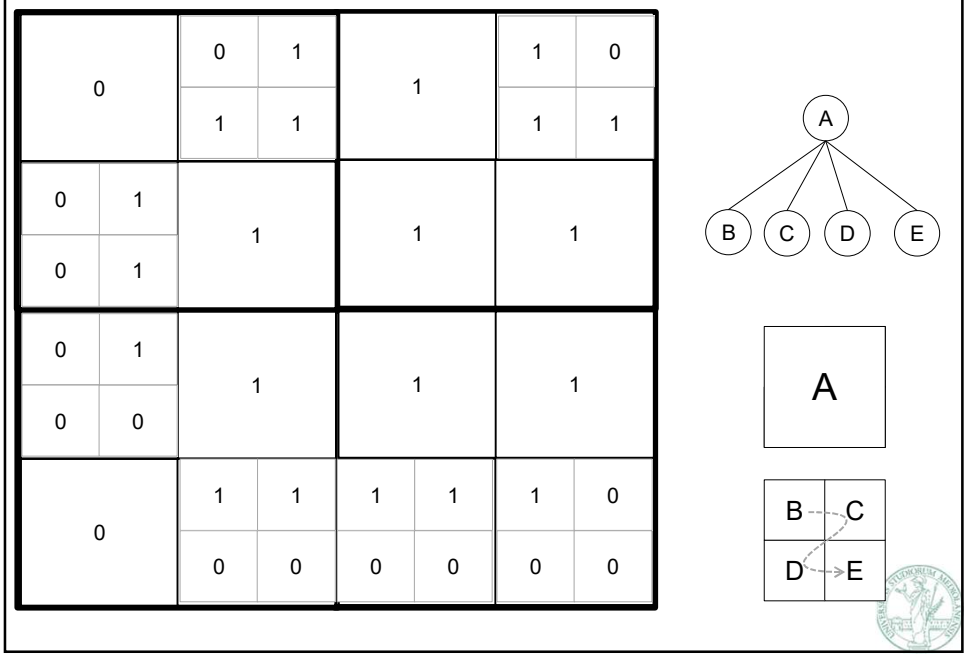
125

Esempio. 2/4: “voxellizzazione” (ma, qui, in 2D)



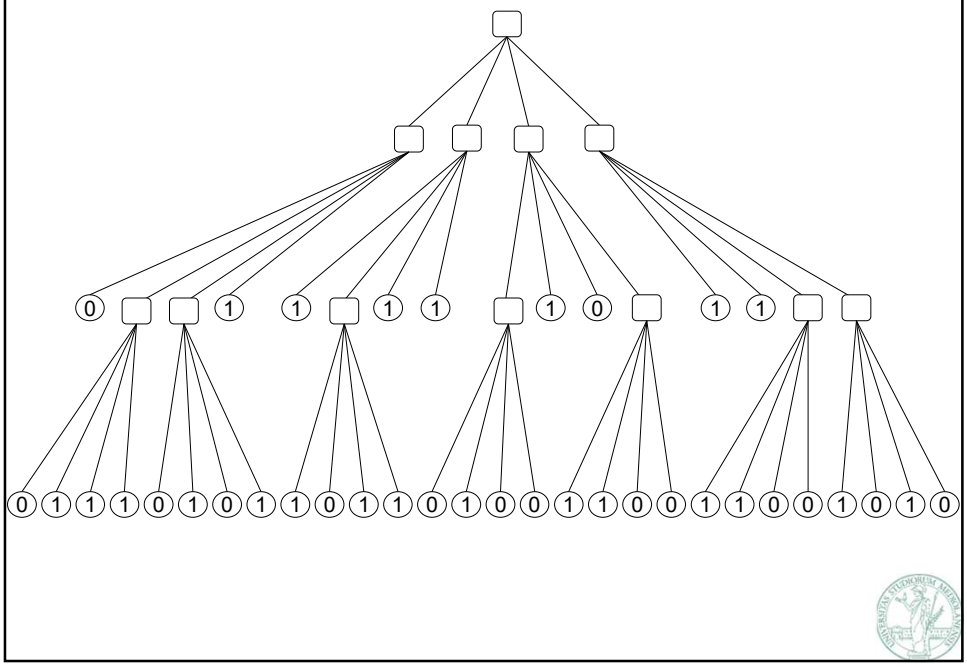
126

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



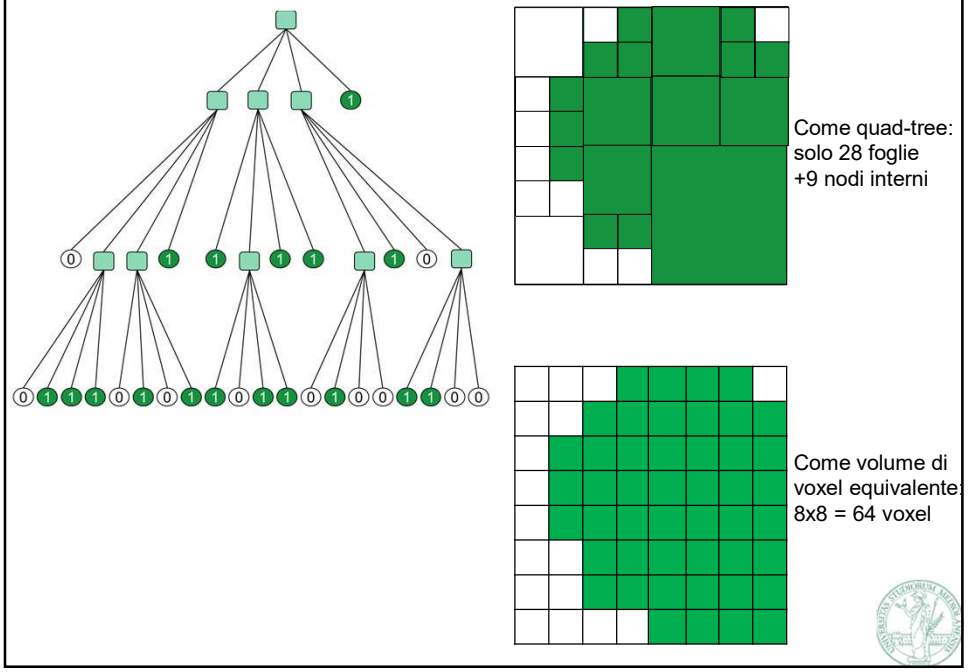
127

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



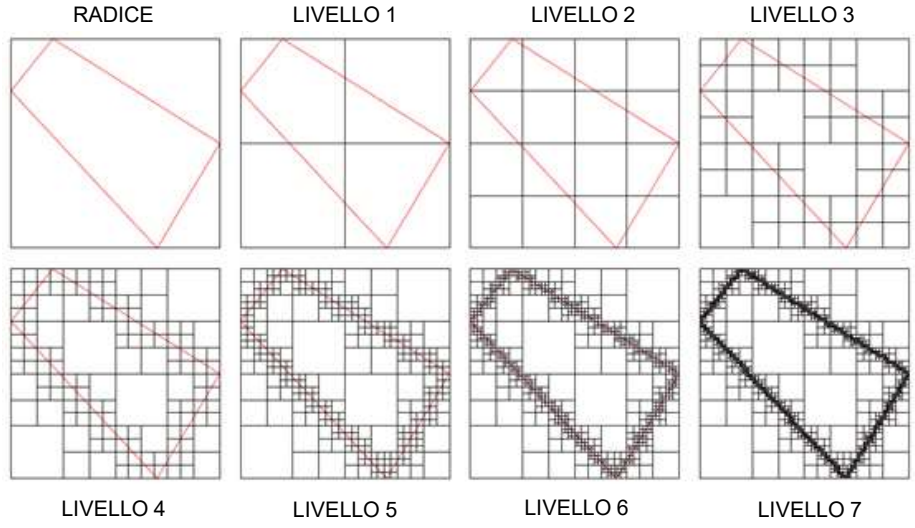
128

Un altro esempio



130

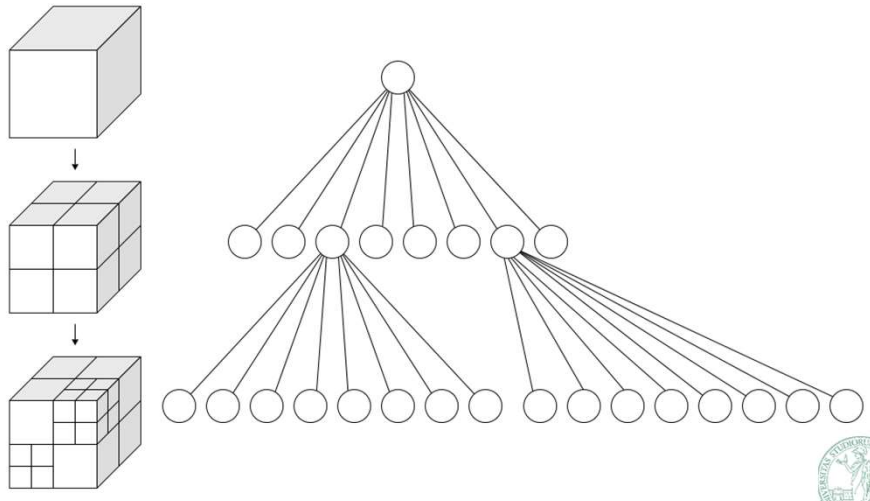
Quad tree: un altro esempio



Ad ogni livello, solo alcuni dei nodi vengono ulteriormente suddivisi (gli altri, sono foglie)

131

“Oct-Tree” o “Octree”: concetto simile, per 3D



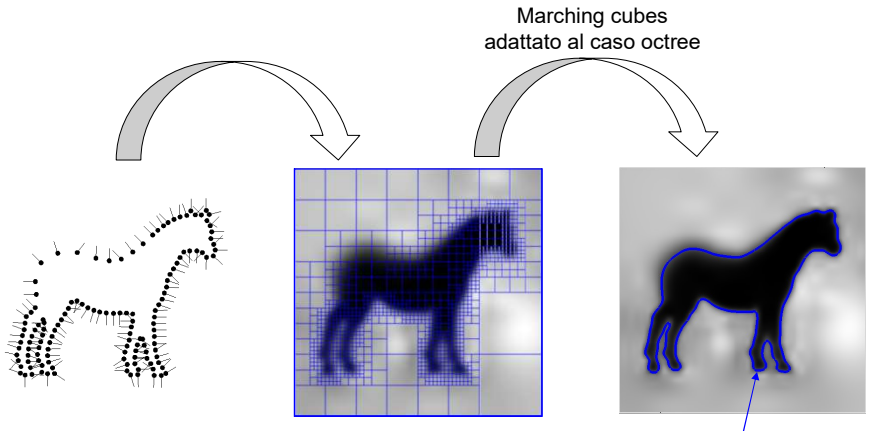
132

Octree: 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 profodità n come voxel sarebbe necessara una risoluzione lineare $\mathrm{N} = 2^n$ per lato, per un totale di $\mathrm{N} \times \mathrm{N} \times \mathrm{N} = 2^n \times 2^n \times 2^n = 2^{3n}$ 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 $\mathbf{O}(\mathrm{N}^2) = \mathbf{O}(2^{2n})$

133

Poisson Surface Reconstruction con Octree



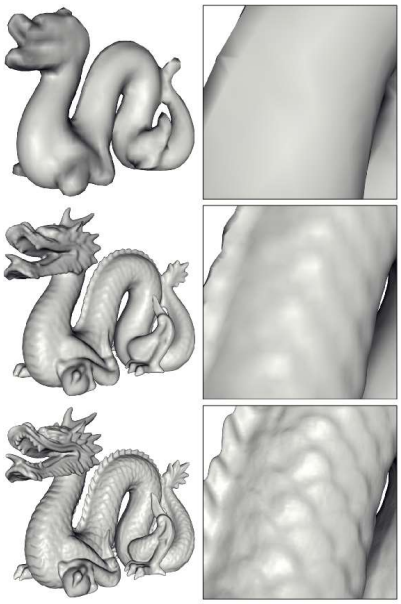
134

Effetto della profondità dell'octree sulla risoluzione della mesh

Profondità $n = 6$,
corrisponde a dataset
di voxel $64 \times 64 \times 64$

Profondità $n = 8$,
corrisponde a dataset
di voxel $256 \times 256 \times 256$

Profondità $n = 10$,
corrisponde a dataset
di voxel $1024 \times 1024 \times 1024$

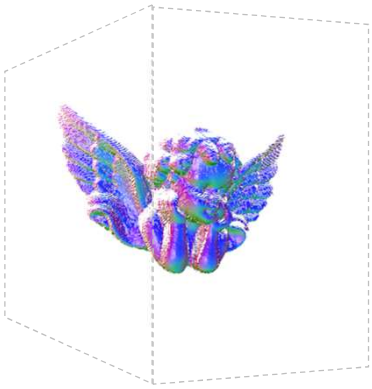


135

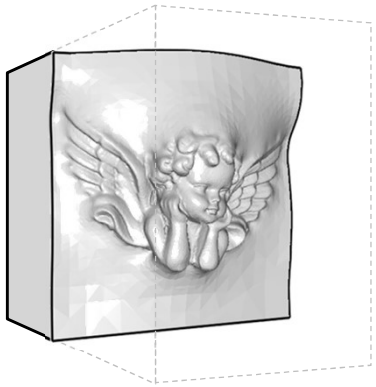
Nota: la poisson reconstruction produce mesh chiuse

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

136