

Marco Tarini - Computer Graphics 2024/2025
 Università degli Studi di Milano

Poisson Reconstruction e Octree

79

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

		ELEMENTI DISCRETI			CONTINUI
		regolari «a griglia»	semi-regolari o irregolari		
			elementi simpliciali	elementi non simpliciali	
SUPERFICIALI	2-manifold <i>«rappresenta una vera superficie»</i>	Height Field Range Scan	Triangle Mesh	Polygonal Mesh Quad-Mesh Quad dominant Mesh	Subdivision surface Parametric Surface (come B-spline)
	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)

80

Strutture volumetriche come dataset intermedio

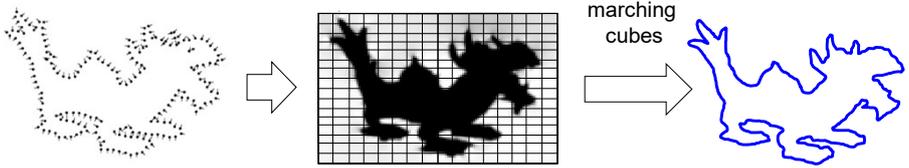
- ✓ «Poisson reconstruction»
 - ⇒ Un algoritmo per convertire una point cloud in una mesh triangolare che consiste nel passare attraverso a un struttura volumetrica a voxel
- ✓ Idea:
 - ⇒ 1. Stendere una griglia volumetrica (o «lattice») nel volume coinvolto
 - ⇒ 2. Memorizzare una «signed distance function» 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...
 - ⇒ $f(\mathbf{p}) = 0$ (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



81

Poisson Reconstruction

Esempio in 2D



Point Cloud

SDF
Signed Distance Field
Griglia 2D di voxel
(1 voxel = 1 scalar)

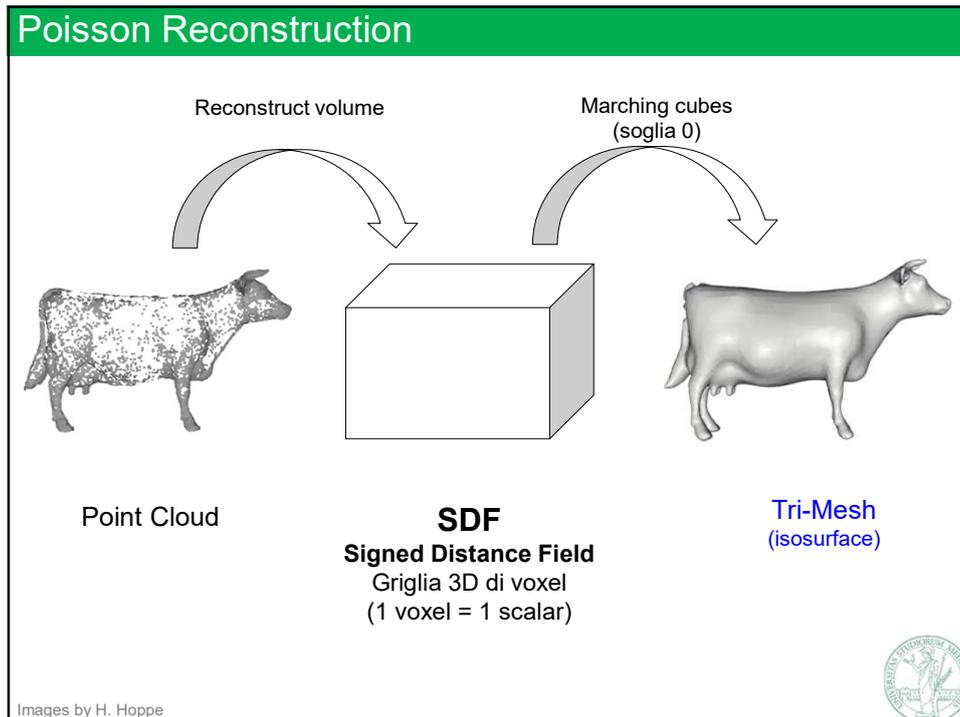
marching cubes

Tri-Mesh
(isosurface)

Images by Micheal Kazhdan



83



84

Signed distance field

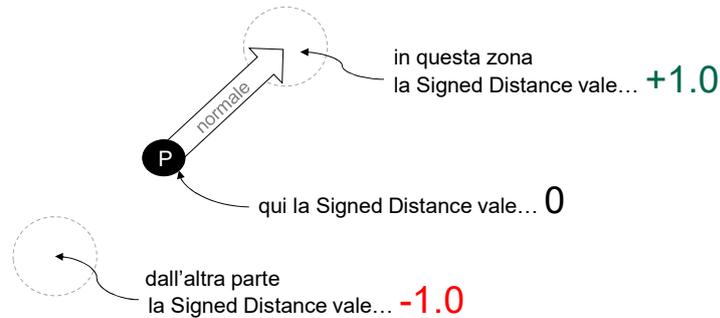
- ✓ Per definizione, la distanza (da qualcosa) è un valore reale *positivo*
- ✓ Definiamo però **Distanza con segno (signed distance)** di un punto da una superficie chiusa un numero il cui valore assoluto è la distanza, e il cui segno riflette se il punto in questione sia dentro (neg) oppure fuori (pos)
- ✓ Un **campo** di distanze con segno (**Signed Distance Field – SDF**) è una struttura dati che per ogni punto di un volume modella la distanza con segno di quel punto (da una superficie chiusa)
- ✓ Memorizziamo un **SDF** come un volume di voxel in cui per ogni voxel registriamo la sua Signed Distance
- ✓ E' un modo (dispendioso) di memorizzare una superficie chiusa

The diagram shows a curved surface labeled 'superficie'. The region inside the surface is labeled 'DENTRO' and the region outside is labeled 'FUORI'. A point inside the surface has a signed distance of -1.4, indicated by a red dashed line. A point outside the surface has a signed distance of +2.3, indicated by a green dashed line. Right-angle symbols show the perpendicular distance from the points to the surface.

85

Da nuvola di punti a SDF

- ✓ Ogni punto della nuvola (provisto come sappiamo di normale) fornisce un'indicazione sul valore della SDF in quel punto nel suo intorno:



- ✓ L'idea della Poisson Reconstruction è costruire un SDF (di voxel) che «mette insieme», e in qualche modo media fra loro, tutte le indicazioni di questo tipo



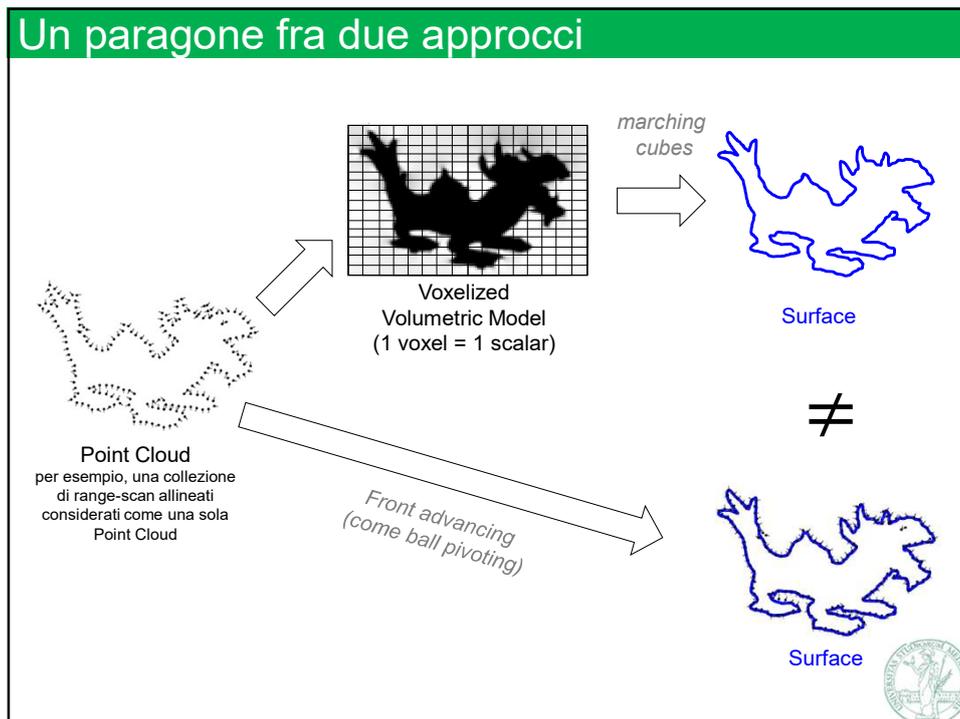
86

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) o triangolazioni di Delaunay aggiungono una connettività di triangoli per unire i punti della nuvola con triangoli (potenzialmente, scardandone alcuni) -- il modo che abbiamo visto nella prima lez sulle mesh
 - ⇒ 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)



87



88

Da nuvola di punti a mesh

- ✓ Nei **metodi diretti** (ball-pivoting, o altri front advancing, o triangolazioni di Delaunay) 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

89

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)



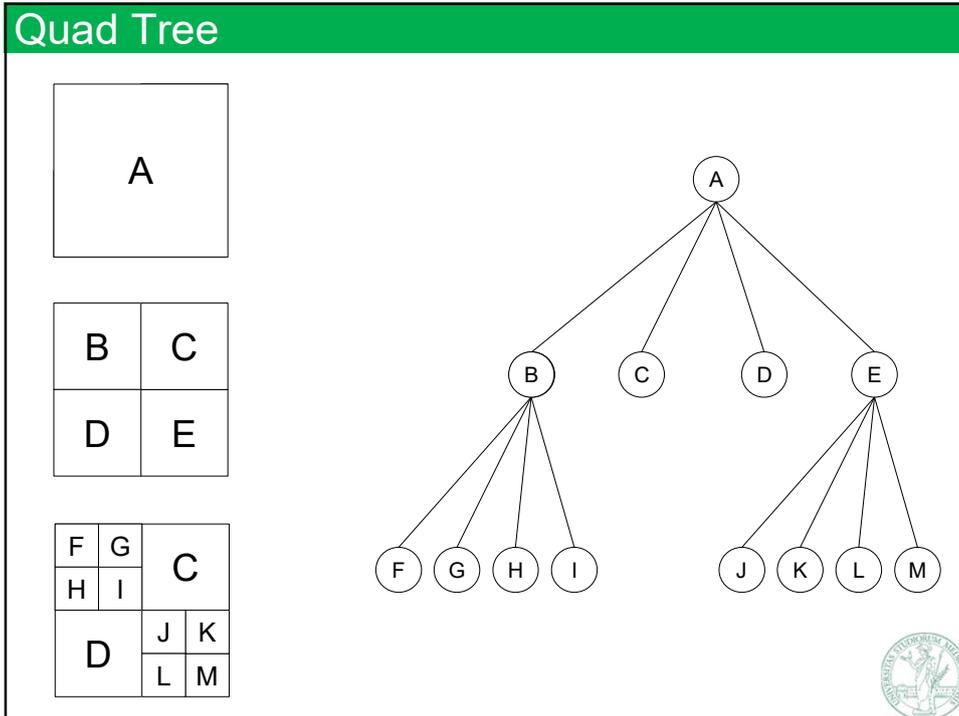
90

Dataset volumetrici a griglia adattivi

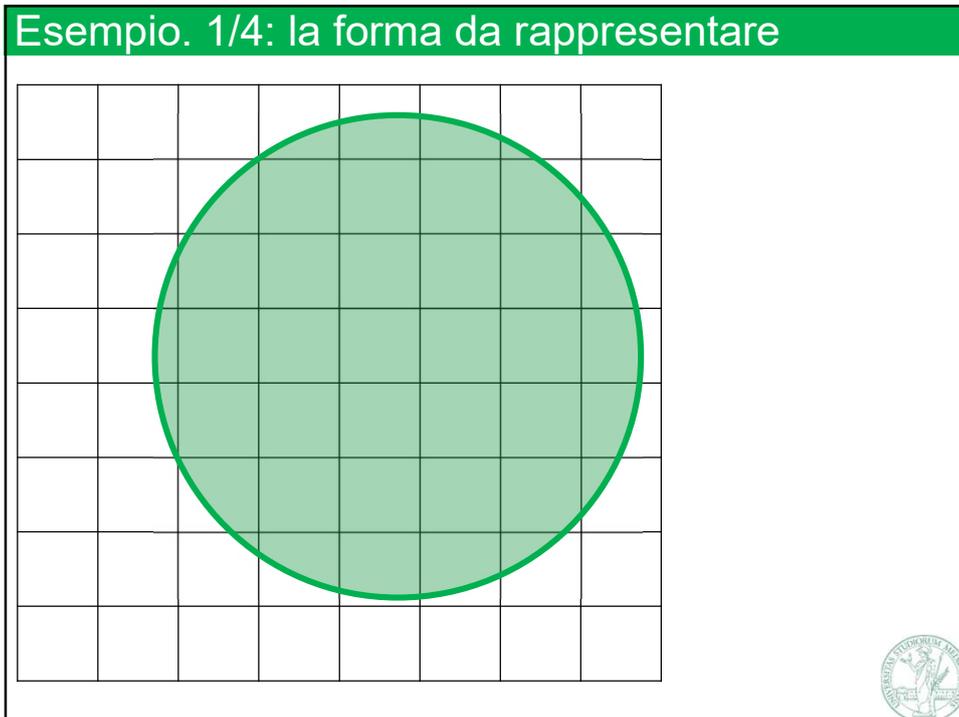
- ✓ Il problema della «*curse of dimensionality*» è legato ad una caratteristica del dato a voxel: la sua risoluzione non è *adattiva*
- ✓ Esistono strutture dati gerarchiche che sono più compatte, grazie alla loro risoluzione adattiva
- ✓ Una delle più diffuse è l'oct-tree
 - ⇒ Vediamo prima una sua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



91



92



96

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

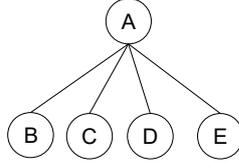
pieno
 vuoto



97

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

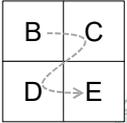


```

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



A

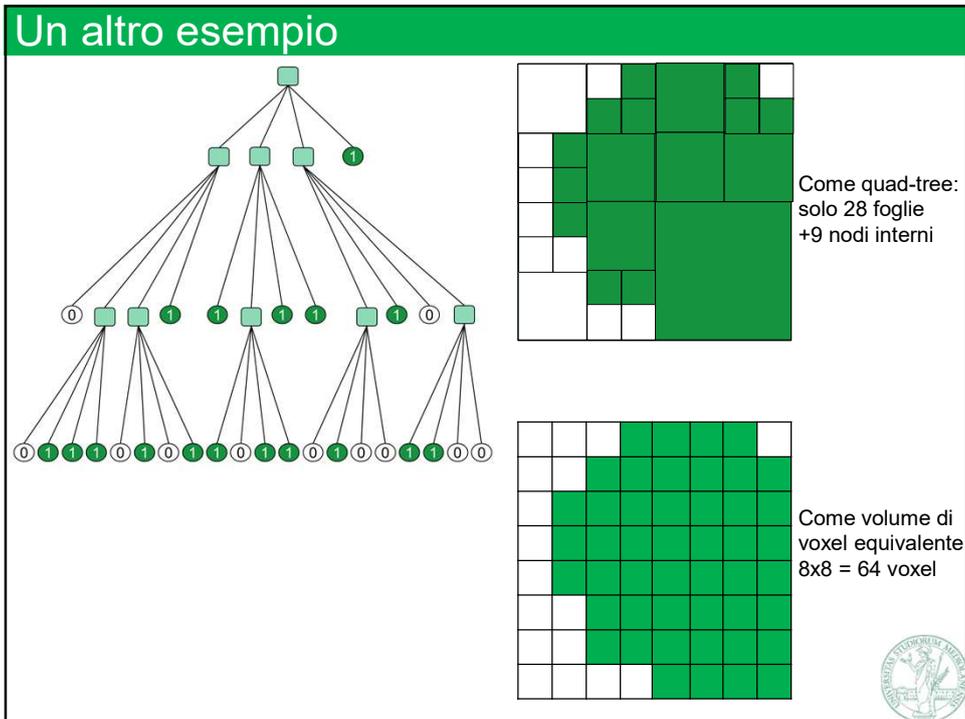
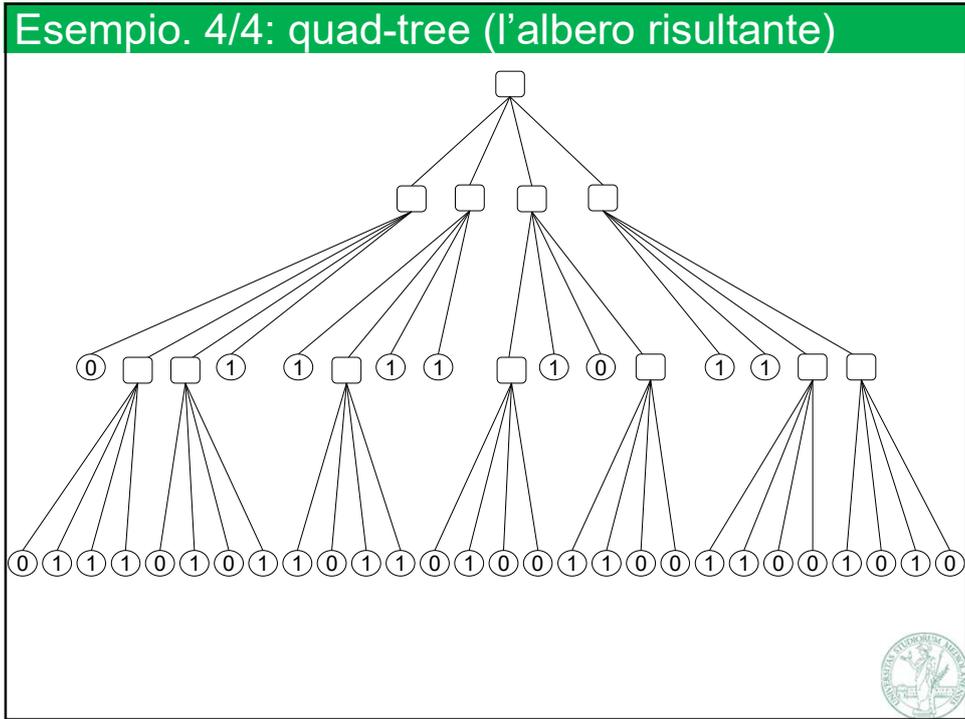


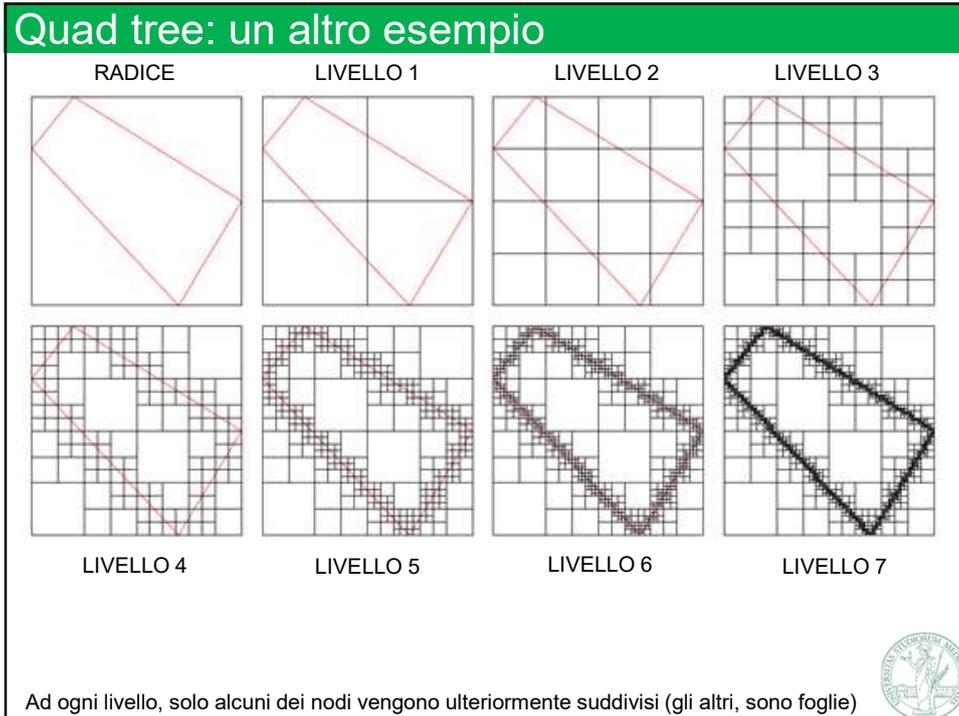
```

            graph TD
              B --- C
              D --- E
              B --- D
              C --- E
            
```

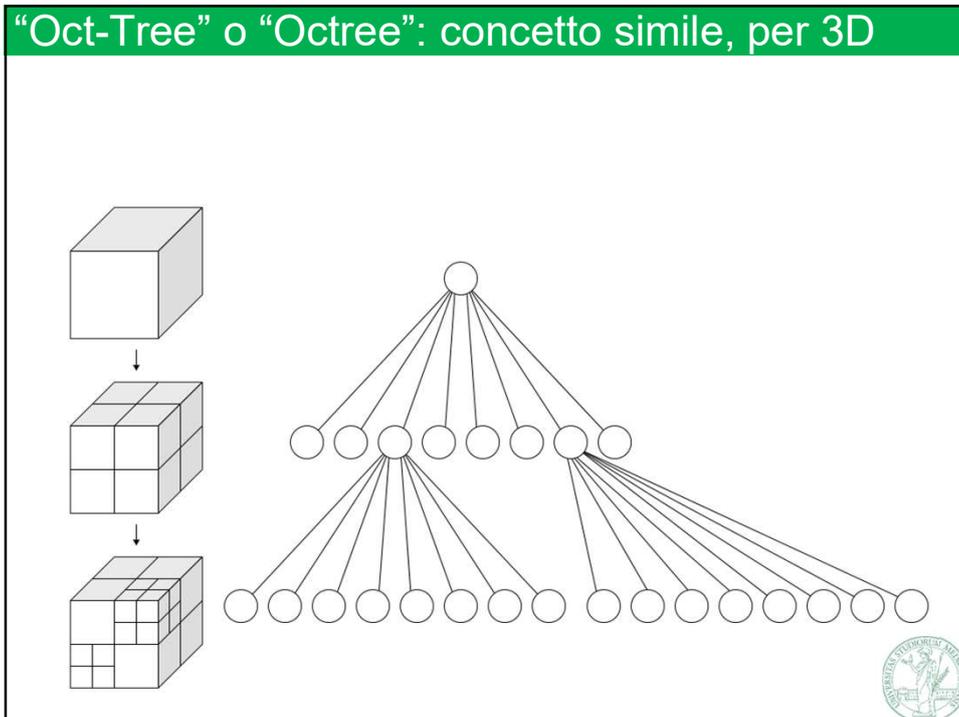


98





102



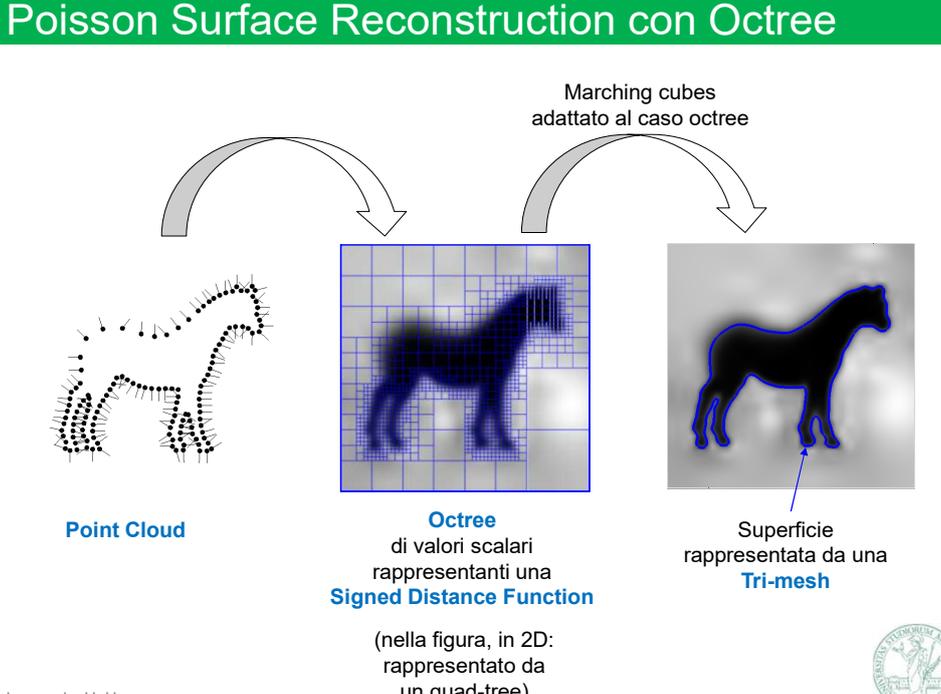
Octree: sommario

- ✓ 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 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 $O(N^2) = O(2^{2n})$



104

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

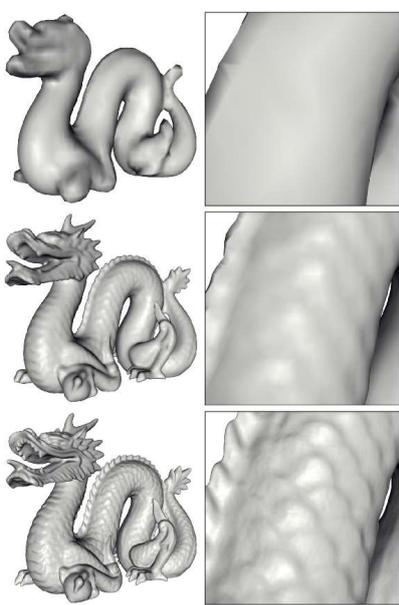
105

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 $256 \times 256 \times 256$

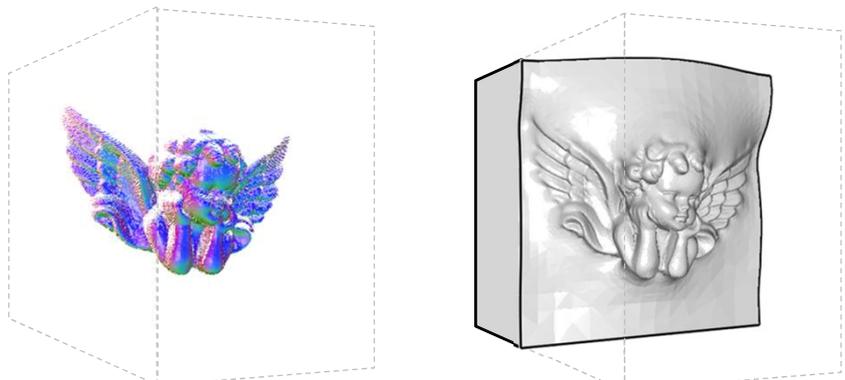
Profondità $n = 10$,
corrisponde a dataset
di voxel $1024 \times 1024 \times 1024$



107

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



108