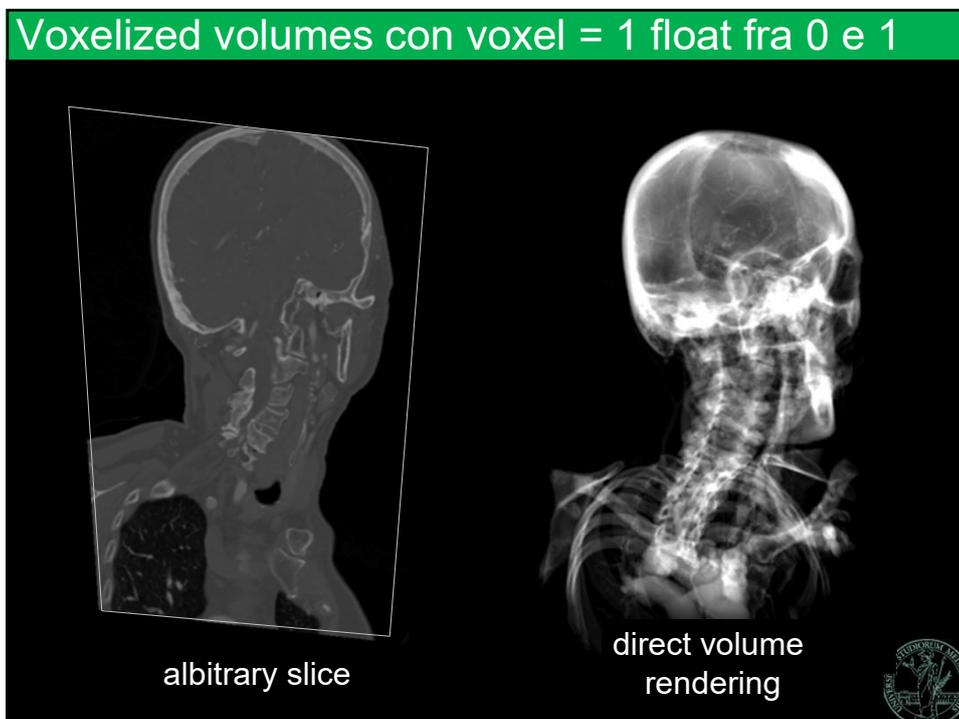


Una (imperfetta) categorizzazione dei tipi di modelli digitali 3D

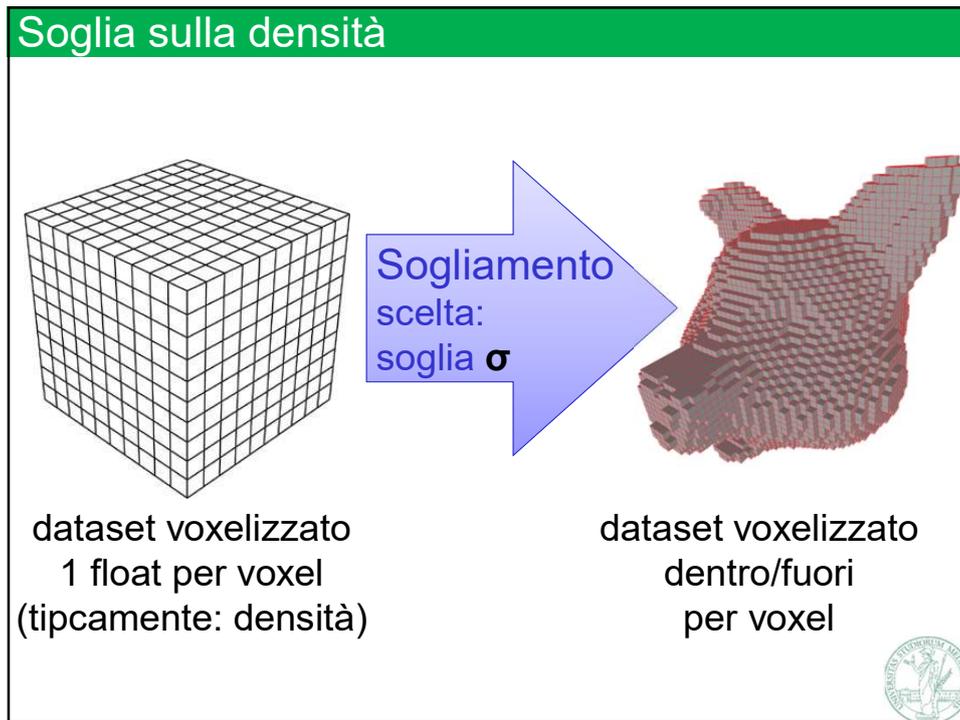
		ELEMENTI DISCRETI			CONTINUI
		regolari «a griglia»	semi-regolari o irregolari		
			elementi simpliciali	elementi non simpliciali	
SUPERFICIALI	2-manifold <i>«rappresenta una vera superficie»</i>	Height Field Range Scan Geometry Images	Triangle Mesh	Polygonal Mesh Quad Mesh Quad dominant Mesh	Subdivision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

CONTINUANDO DALL'ULTIMA LEZIONE...

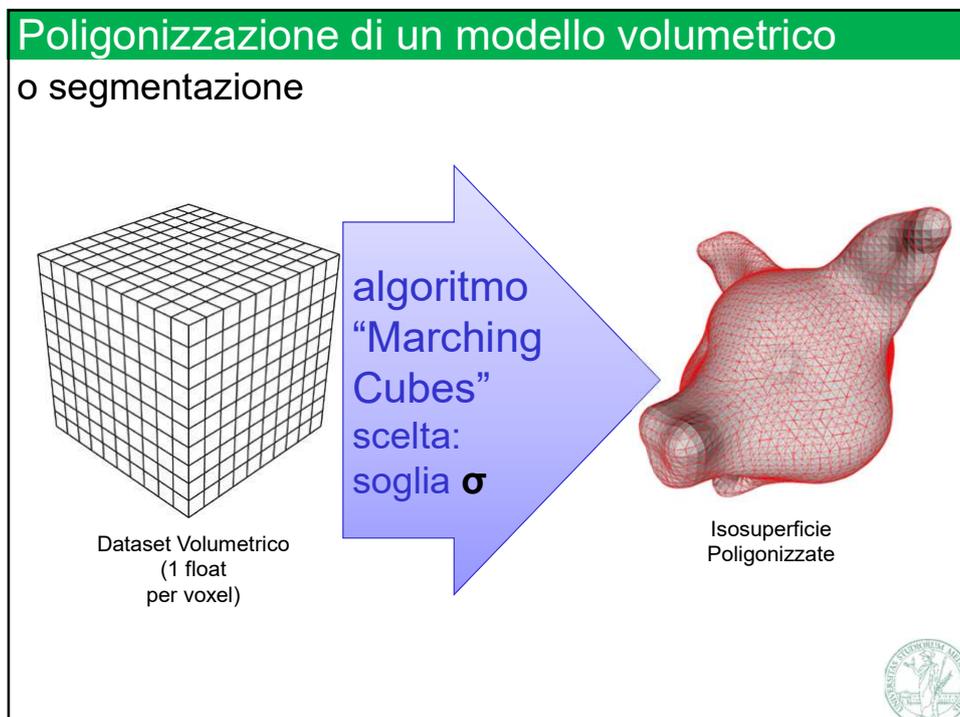
28



35



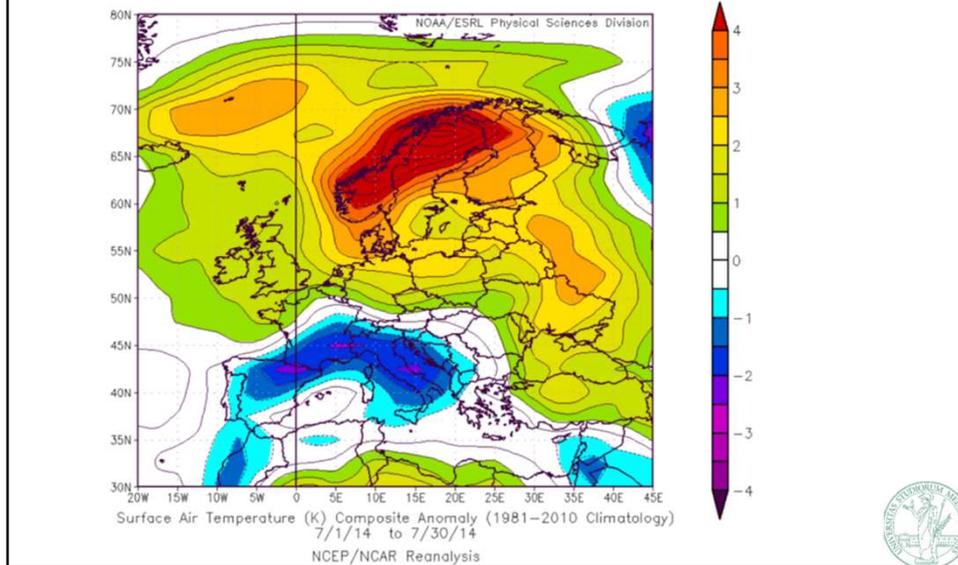
36



37

Esempio di isolina

✓ Isolinee di pressione



38

Isolinee e isosuperfici

✓ Su un piano (2D):

- ⇒ ogni punto del piano ha un valore scalare (esempio: pressione, o altezza – height field)
- ⇒ prendo tutta la regione 2D con valore $> \sigma$
- ⇒ il bordo di questa regione è una linea ... i cui punti hanno valore tutti σ e che racchiude tutti i valori di valore $> \sigma$
- ⇒ è la « isolina di valore σ » (linea di isovalori)

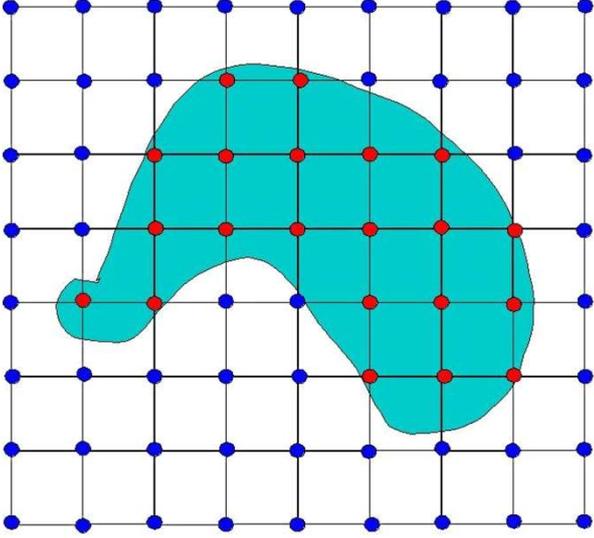
✓ Su un volume (3D):

- ⇒ ogni punto dello spazio ha un valore scalare (esempio: densità, pressione, temperatura...)
- ⇒ prendo la regione 3D con valore $> \sigma$
- ⇒ il bordo di questa regione è una superficie... i cui punti hanno tutti valore σ che racchiude tutti i valori di valore $> \sigma$
- ⇒ è la « iso-superficie di valore σ »

39

Algoritmo marching squares (per 2D)

✓ Sogliare voxels



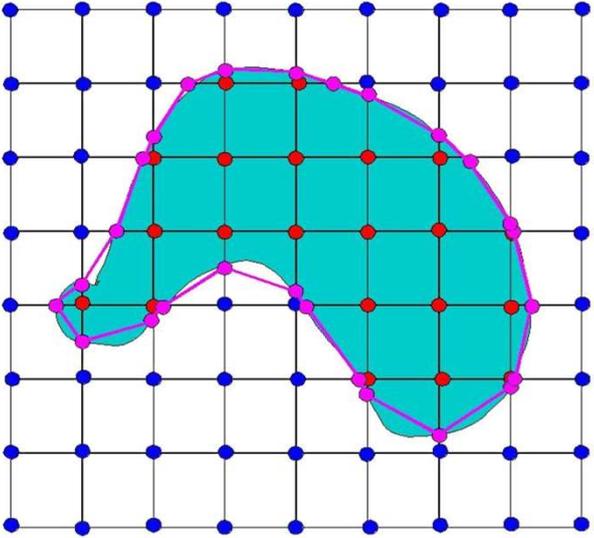
- voxel con valore $< \sigma$
- voxel con valore $\geq \sigma$



41

Algoritmo marching squares (per 2D)

✓ Variante: trovare intersezioni e unirle



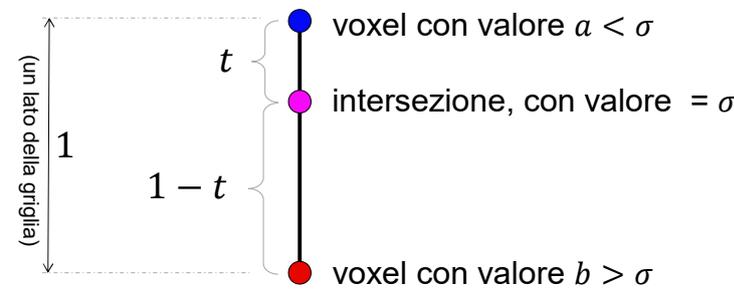
- voxel con valore $< \sigma$
- intersezione
- voxel con valore $\geq \sigma$



44

Algoritmo marching squares (per 2D)

✓ Come trovare le intersezioni



$a < \sigma$ (voxel con valore $a < \sigma$)
 $= \sigma$ (intersezione, con valore $= \sigma$)
 $b > \sigma$ (voxel con valore $b > \sigma$)

✓ Ipotesi: segnale interpolato linearmente:

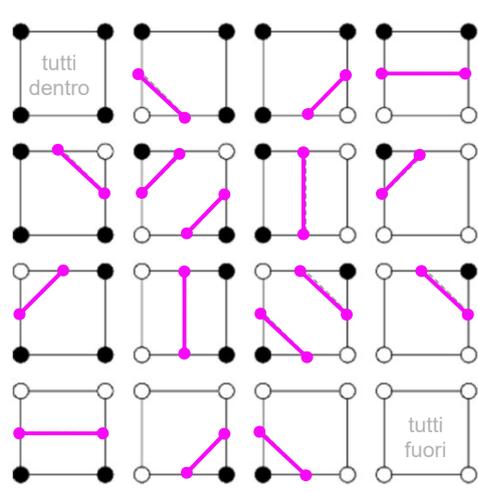
$$a(1-t) + bt = \sigma \quad \Leftrightarrow \quad t = \frac{\sigma - a}{b - a}$$


45

Algoritmo marching squares (per 2D)

Come trovare i segmenti che connettono le intersezioni?

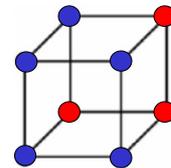
- ✓ Consideriamo un quadrato fra 4 voxel
- ✓ Ogni suo vertice è «dentro» $< \sigma$ o «fuori» $\geq \sigma$
- ✓ Per ogni combinazione, (e sono solo $2^4 = 16$) decido, una volta per tutte, i segmenti da costruire per unire le intersezioni
- ✓ ottengo questa tabella:




46

Generalizzando a 3D: algoritmo marching cubes

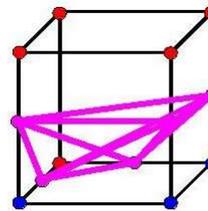
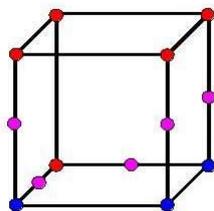
- ✓ Ogni voxel della griglia è dentro o fuori
 - ⇒ valore $>$ o $<$ della soglia prescelta
- ✓ Ogni edge della griglia (orientato lungo la X, Y o Z), che connetta un vertice dentro ad uno fuori, ha una intersezione con l'isosuperficie cercata
 - ⇒ la si trova trovata come nel caso 2D
 - ⇒ per ciascuna intersezione, creo un vertice della mesh
 - ⇒ ho ottenuto la **geometria** della mesh!
- ✓ Come ottengo la **connettività**?
 - ⇒ Scompongo la griglia in cubetti $1 \times 1 \times 1$
 - ⇒ Ogni cubo ha 8 vertici, ciascuno è dentro o fuori → $2^8 = 256$ casi
 - ⇒ Uso una tabella per analizza, in ciascun caso, quali poligoni creare per connettere i vertici sui lati della griglia



47

Generalizzando a 3D: algoritmo marching cubes

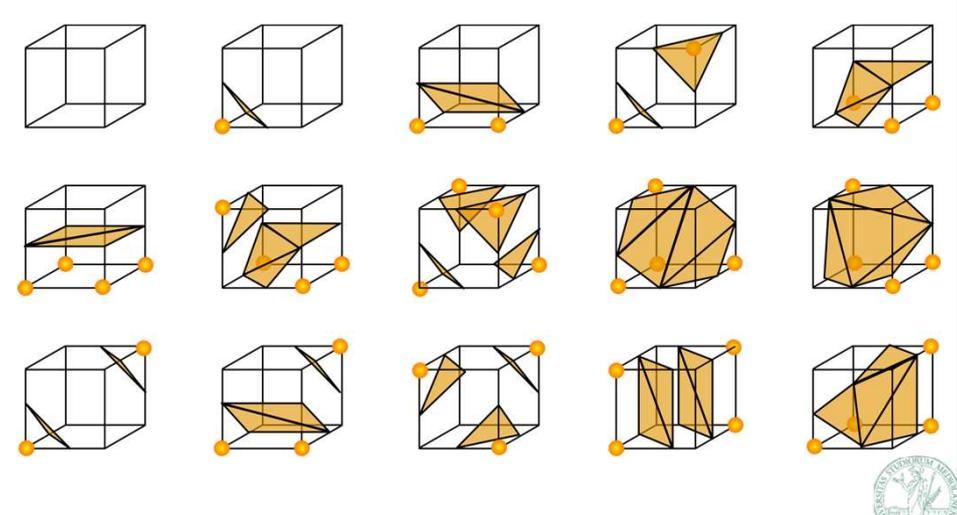
- ✓ Esempio di uno dei 256 casi



48

Marching cube table

✓ Altri esempi
⇒ Tutti gli altri sono analoghi a questi 15, per simmetria



49

Marching cubes

✓ Problema: efficienza.
⇒ Curse of dimensionality: numero cubico di cubi da analizzare

✓ Soluzione possibile: evitare di processare il gran numero di cubi fra i voxel
⇒ Molti dei cubi sono «tutti dentro» o «tutti fuori»
⇒ Cioè sono vuoti (niente intersezioni, nè facce)

✓ Far “marciare” i cubi:
⇒ Trovo un cubo non vuoto → lo processo
⇒ Passo a processare i cubi non vuoti vicini
⇒ Continuo fino ad aver completato la mesh connessa



50

Da modello voxelizzato a Mesh poligonale

Dataset Volumetrico

algoritmo
"Marching
Cubes"
scelta:
soglia σ

Iso-superficie
Triangolata
S
(di valore s)

51

Da modello voxelizzato a Mesh poligonale

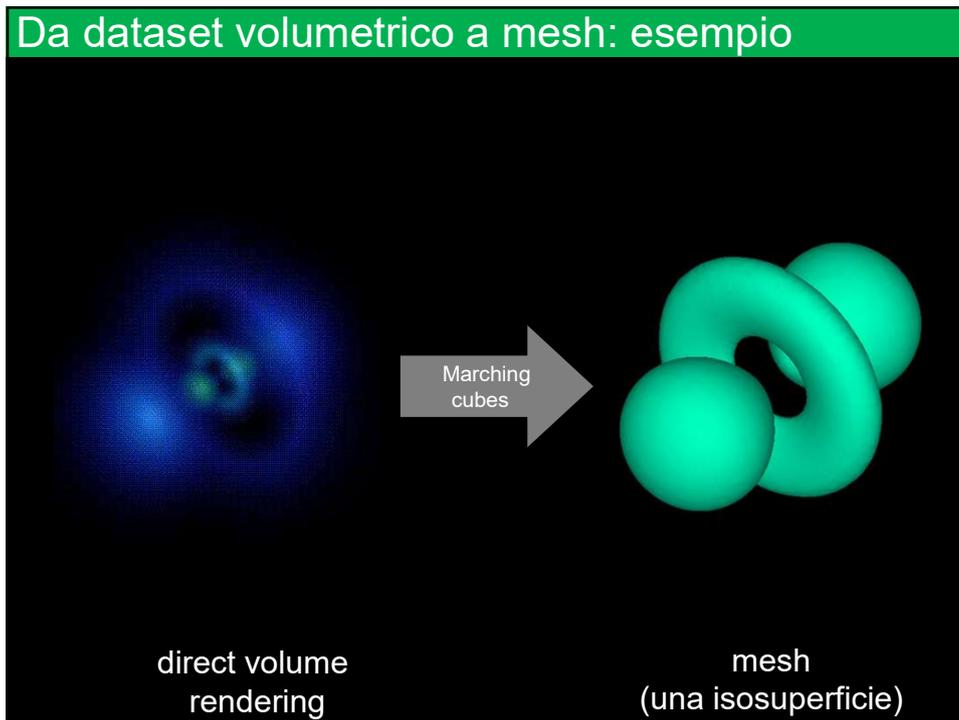
Output:
mesh chiusa,
two-manifold,
ben orientata

Ma tipicamente,
cattivo meshing:
triangoli di forma
molto lunga e
stretta, triangoli
piccoli, o anche
degeneri

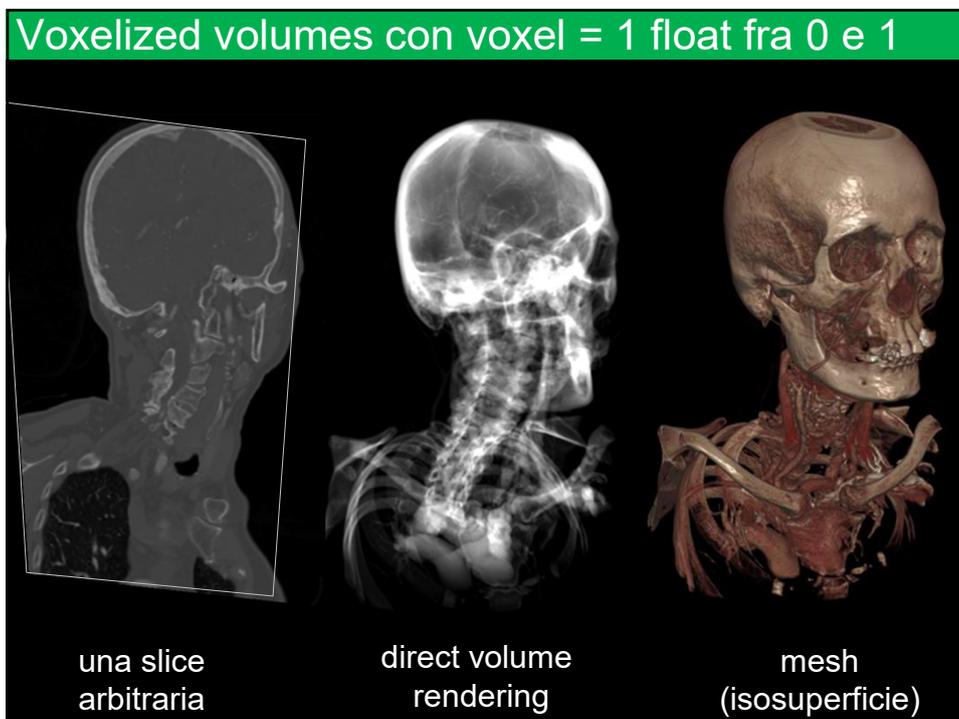
algoritmo "Marching Cubes"

Iso-superficie Triangolata S (di valore s)

52



53



55