Visualization: Volume rendering

Visualization

> Visualization is any technique for creating images, diagrams or animations to **communicate a message**.

Wikipedia

Scientific data visualization
- Abstract
- Physics (fluids, ...)
- Medical (X-Rays, MRI, Images, ...)
- Technics (Mechanics, ...)
- ...

Visualization : Problematic

- Complex Data: non visualizable (density, tensors, ...)
- Large amount of data : 10, 100 To (landscape, connections, scanners, ...)
- Noisy data (medical, ...)

Goal: Being able to visualization what is **significant**, **usefully**, and **efficiently**.

Data types

- Scalar field (temp, pression, ...)
- Vectorial field (vitesse, orientation, ...)
- Tensorial field (mechanical constraints, curvature, ...)

Goal: Do we define the data on a surface, on a volume ?

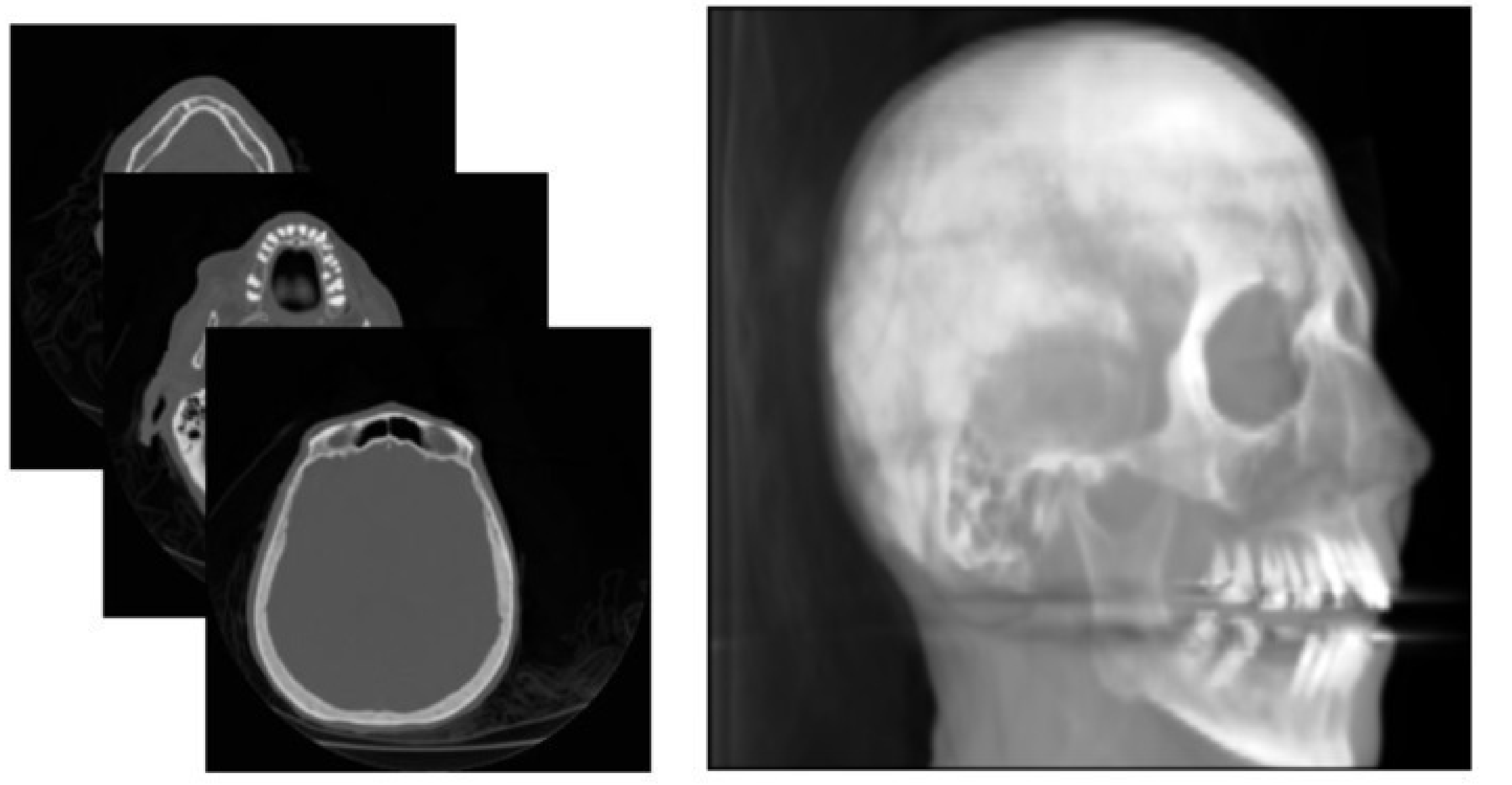
Scalar field

Surface of the domain or internal caracteristics

004

Scalar field

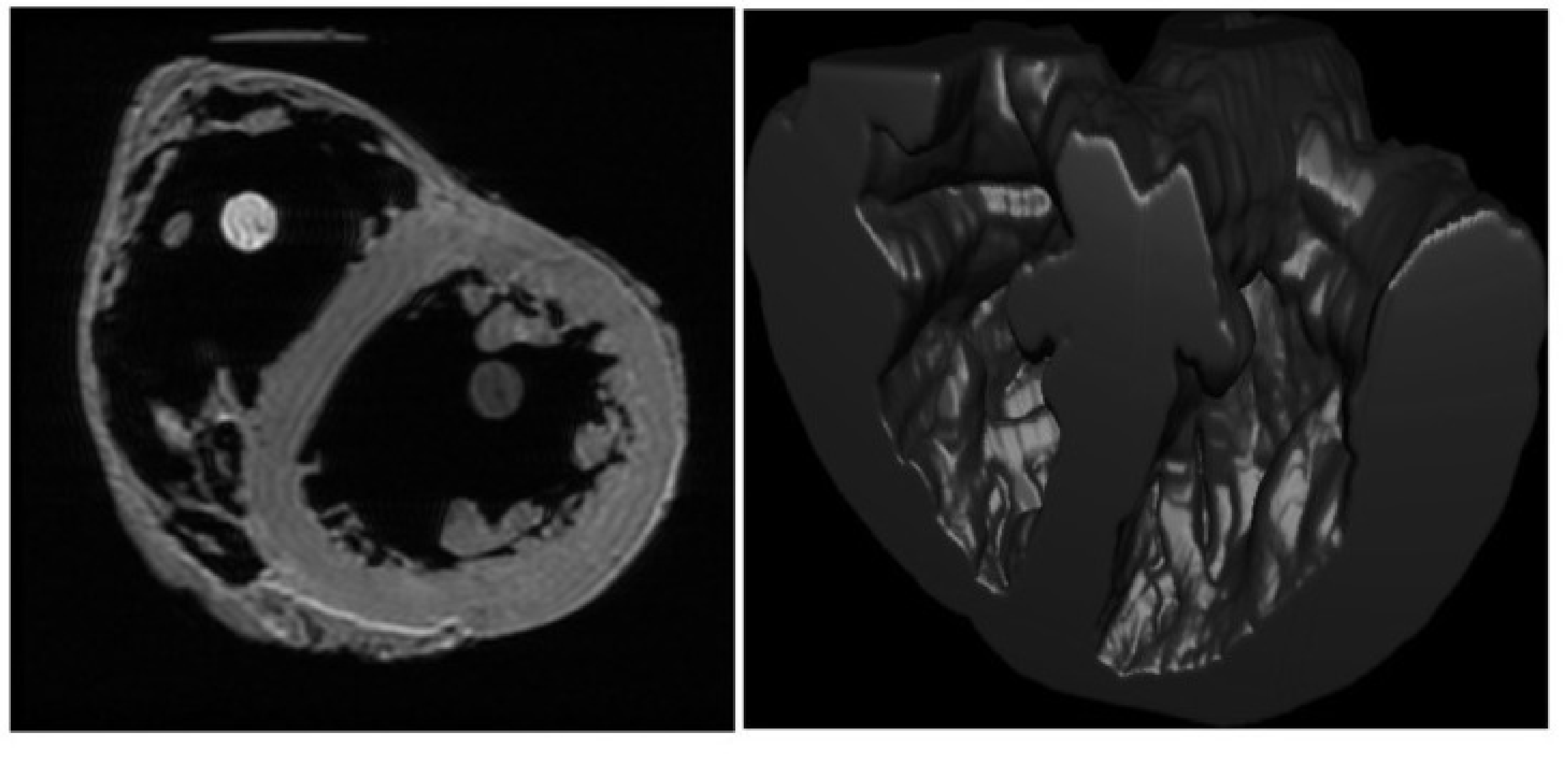
2D section or volume rendering (isosurface, 3D textures, ...)



005

Scalar field

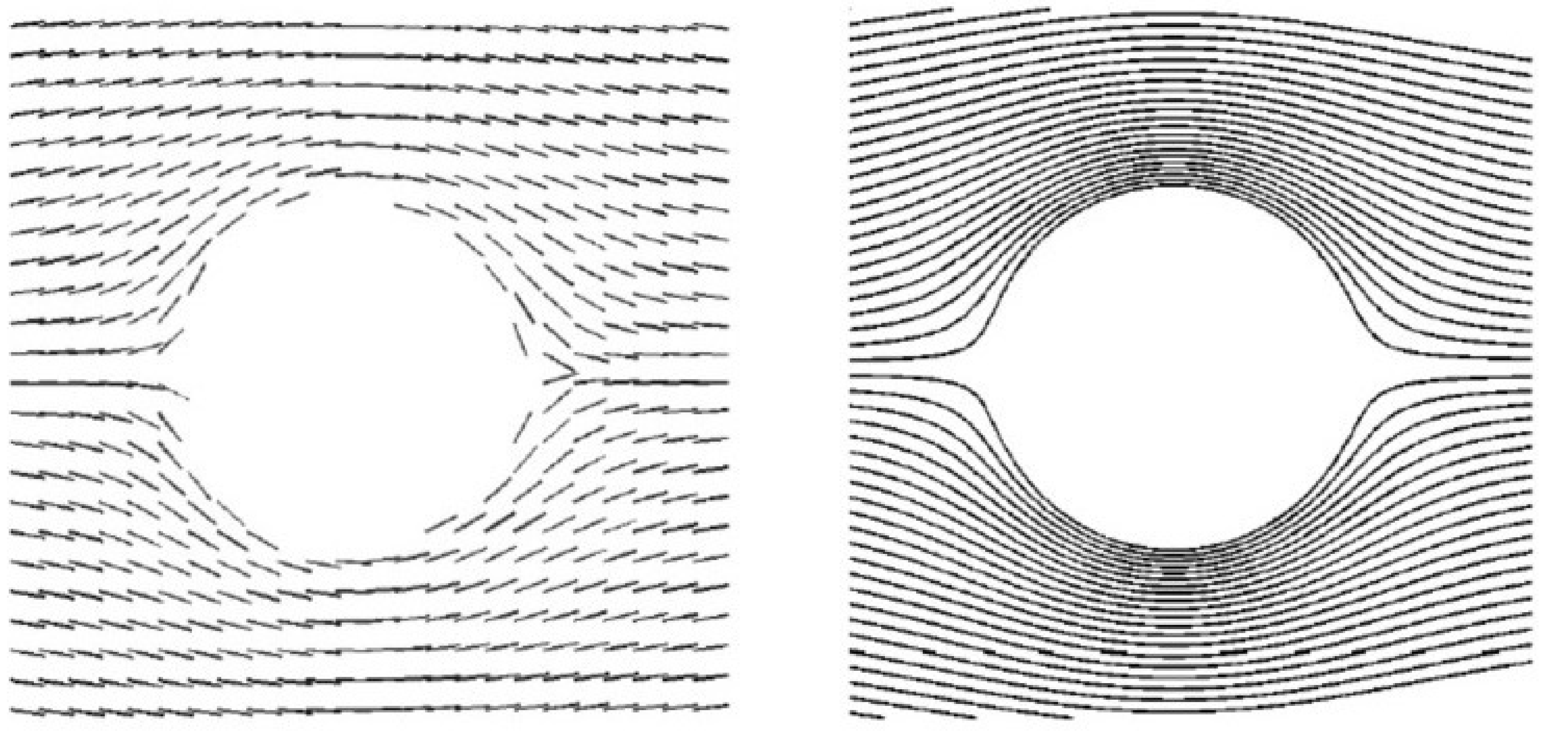
2D Section or 3D Isosurface



006

Vectorial field

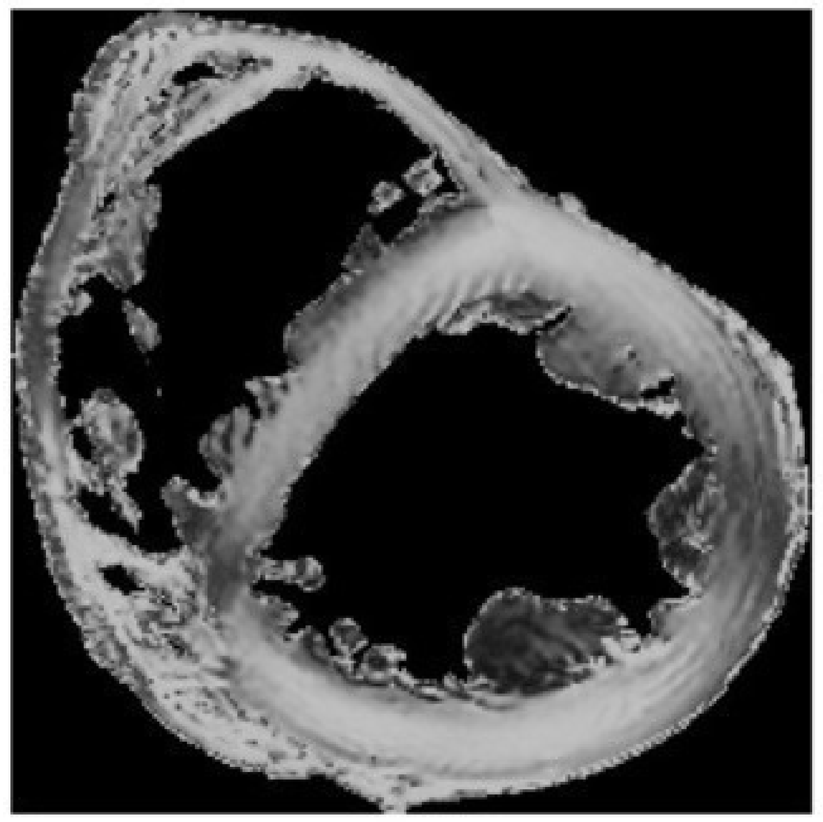
Vectors or trajectories



007

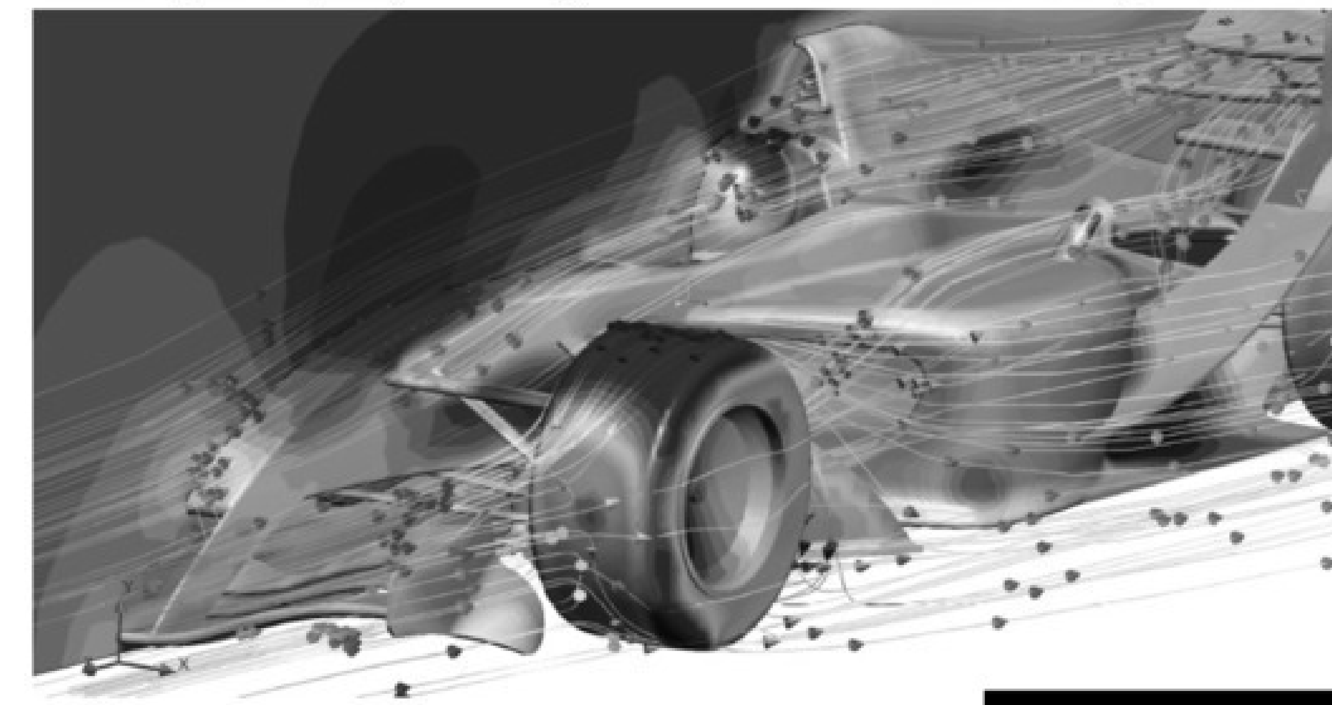
Vectorial field

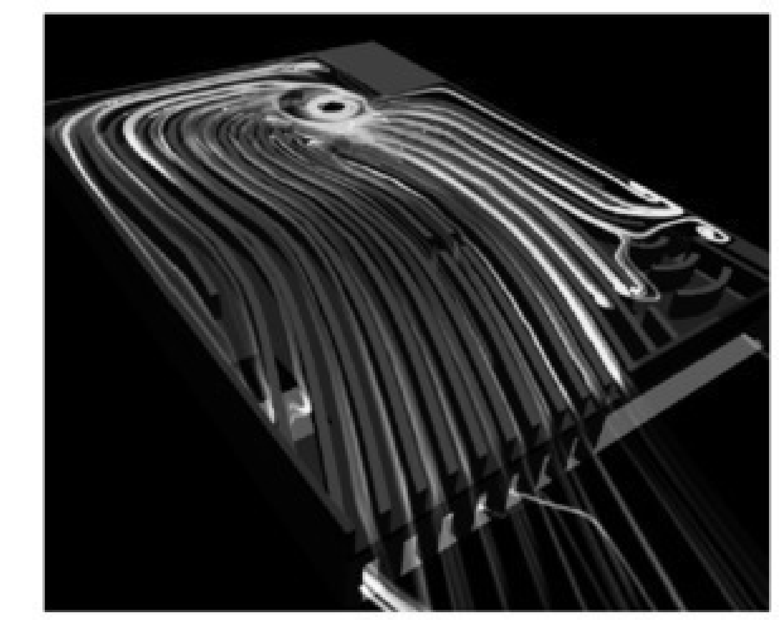
Vectors or trajectories (stream lines can represent real data)



Vectorial field

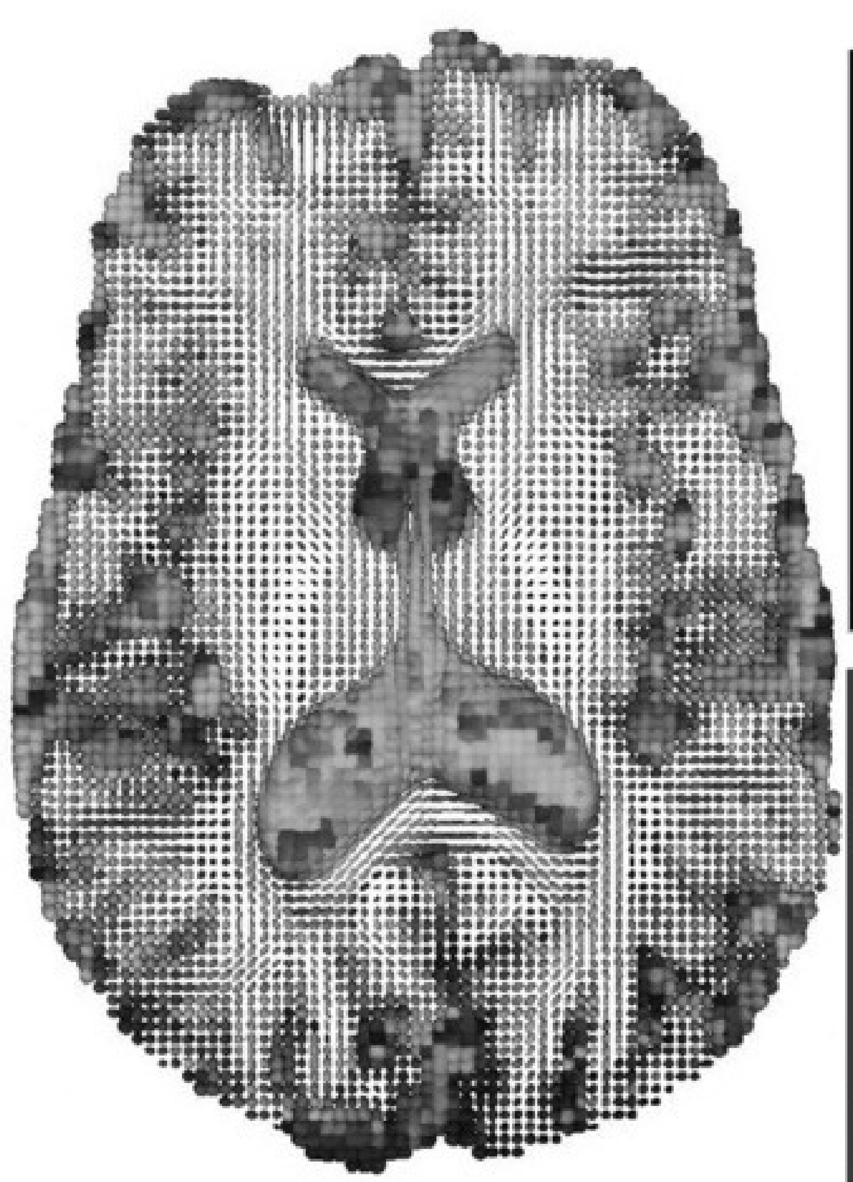
Complex physically based simulation (streamlines, ...)

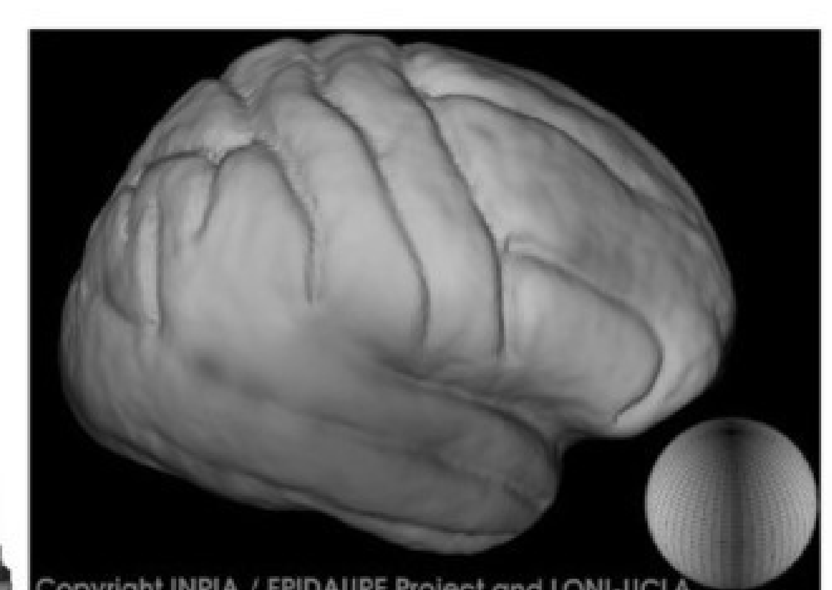


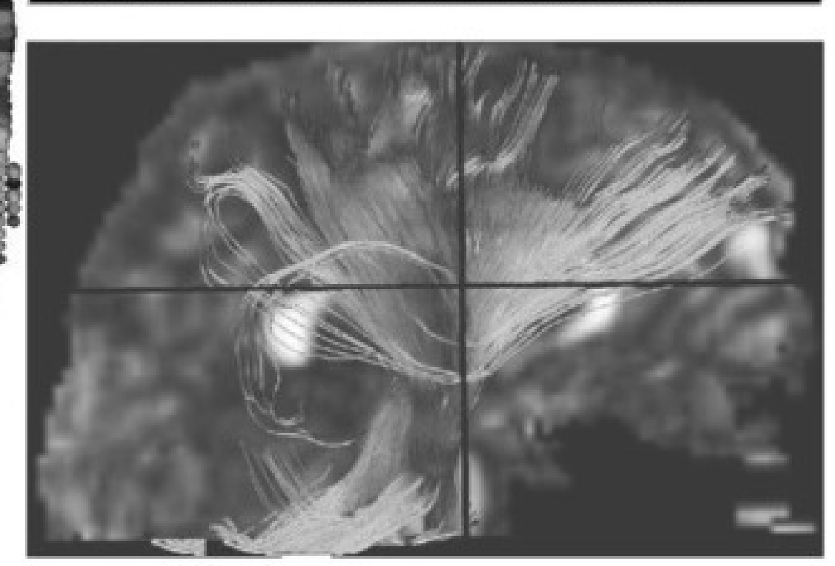


Tensorial field

Symetric matrix 3x3 (Ellipsoid, glyphs, orientation, fiber-tracking, ...)

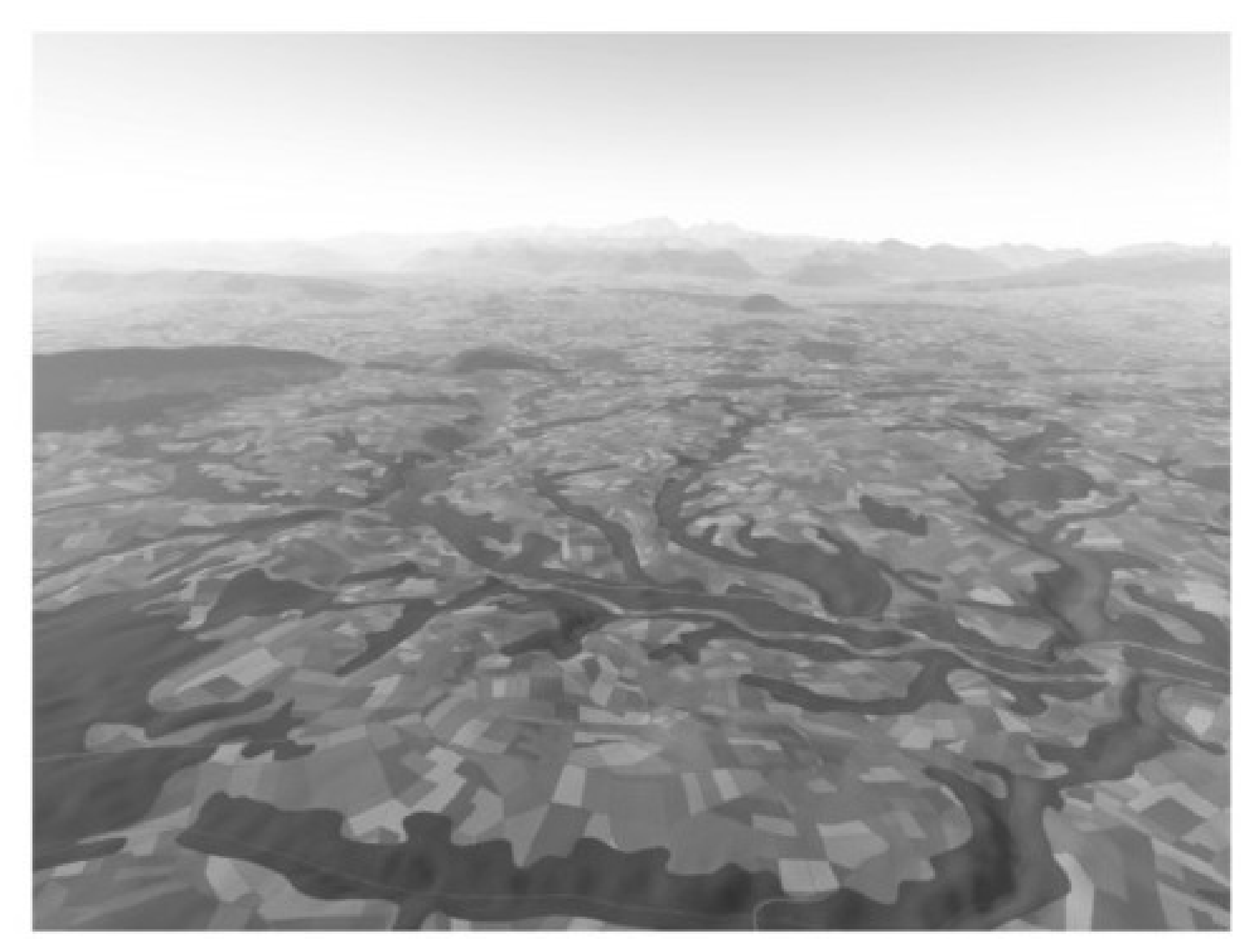


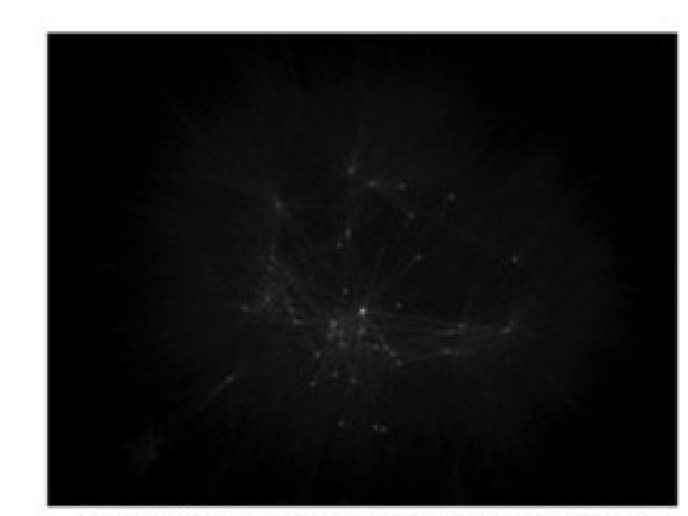


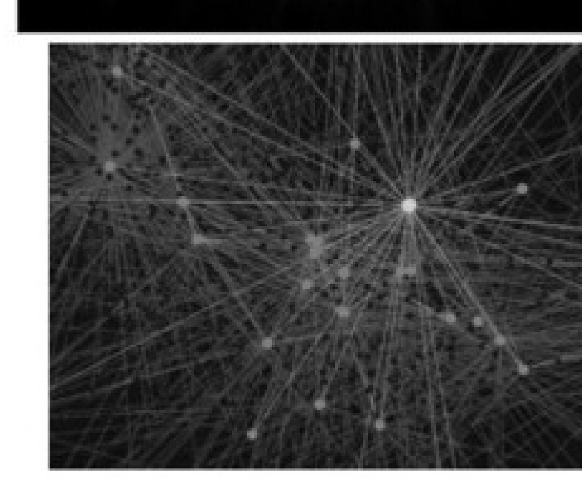


Large data

Data acquired from physical scanners are often too large
(geography, networks, ...)

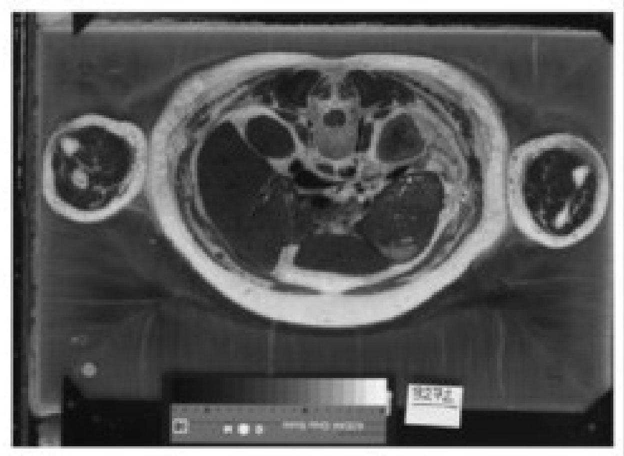
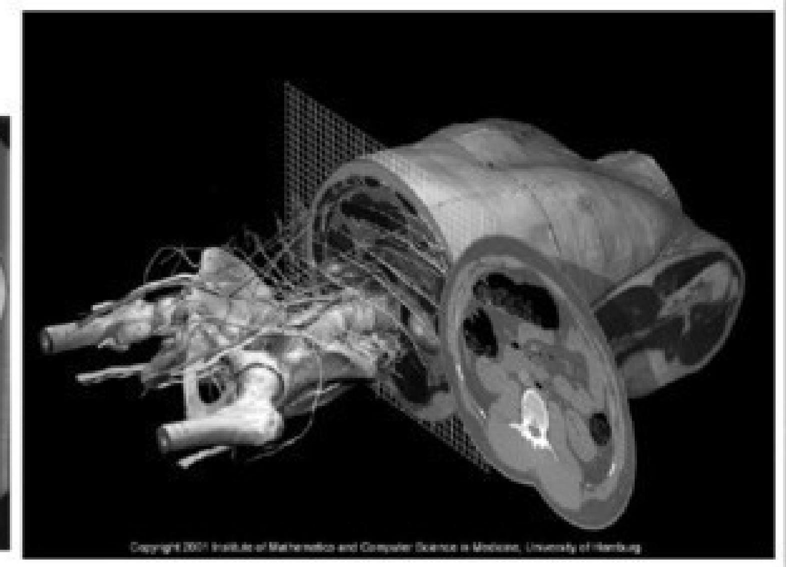






Large data

Visible Human Project, 40Go (slices of 0.33mm)


Classification

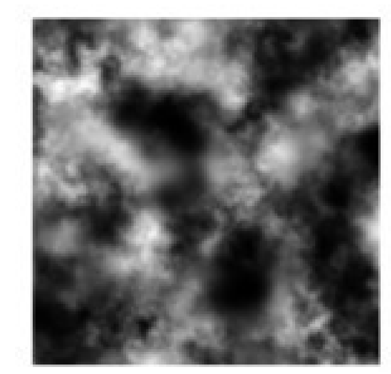
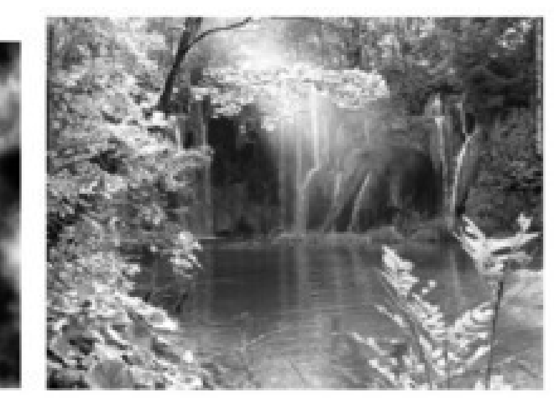
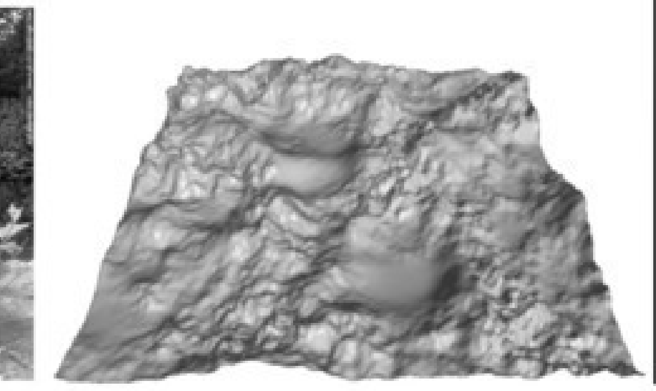
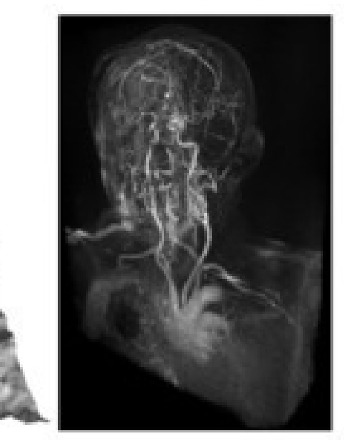
Visualize $f : \begin{cases} \mathbb{R}^v & \rightarrow & \mathbb{R}^d \text{ embeeded in } \mathbb{R}^n \\ u & \mapsto & f(u) \end{cases}$

d=1	scalar field
d>1	vectorial field
d=$(i \times j)$	matrix field

v=1	lineic field
v=2	surfacic field
v=3	volume field

Common special cases

v d n	
2 1 2	B&W image
2 3 2	Color image (texture)
2 1 3	Heigh-field (mountains)
3 1 3	Volume density

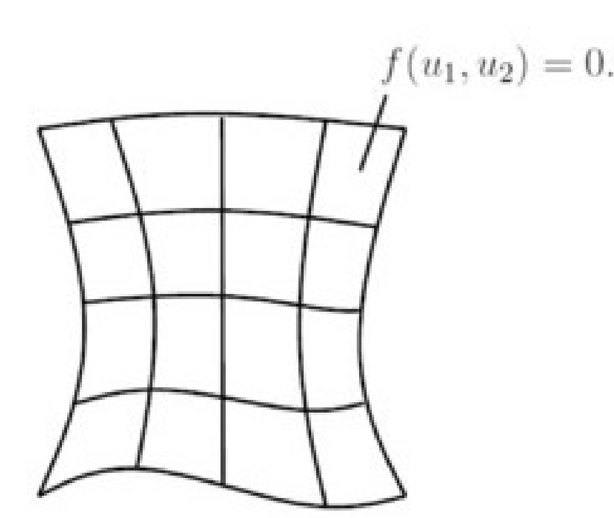
Surfacic scalar data

Surfacic scalar data : Notation

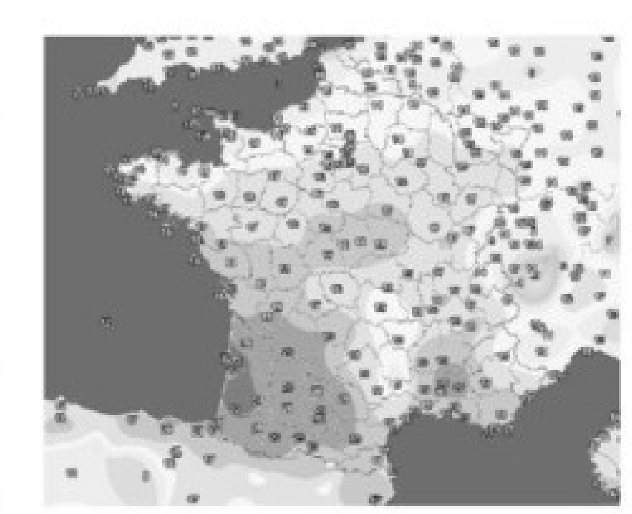
We call a density $f(u_1, u_2) = I \in \mathbb{R}$

Very often: $f(x, y) = I$

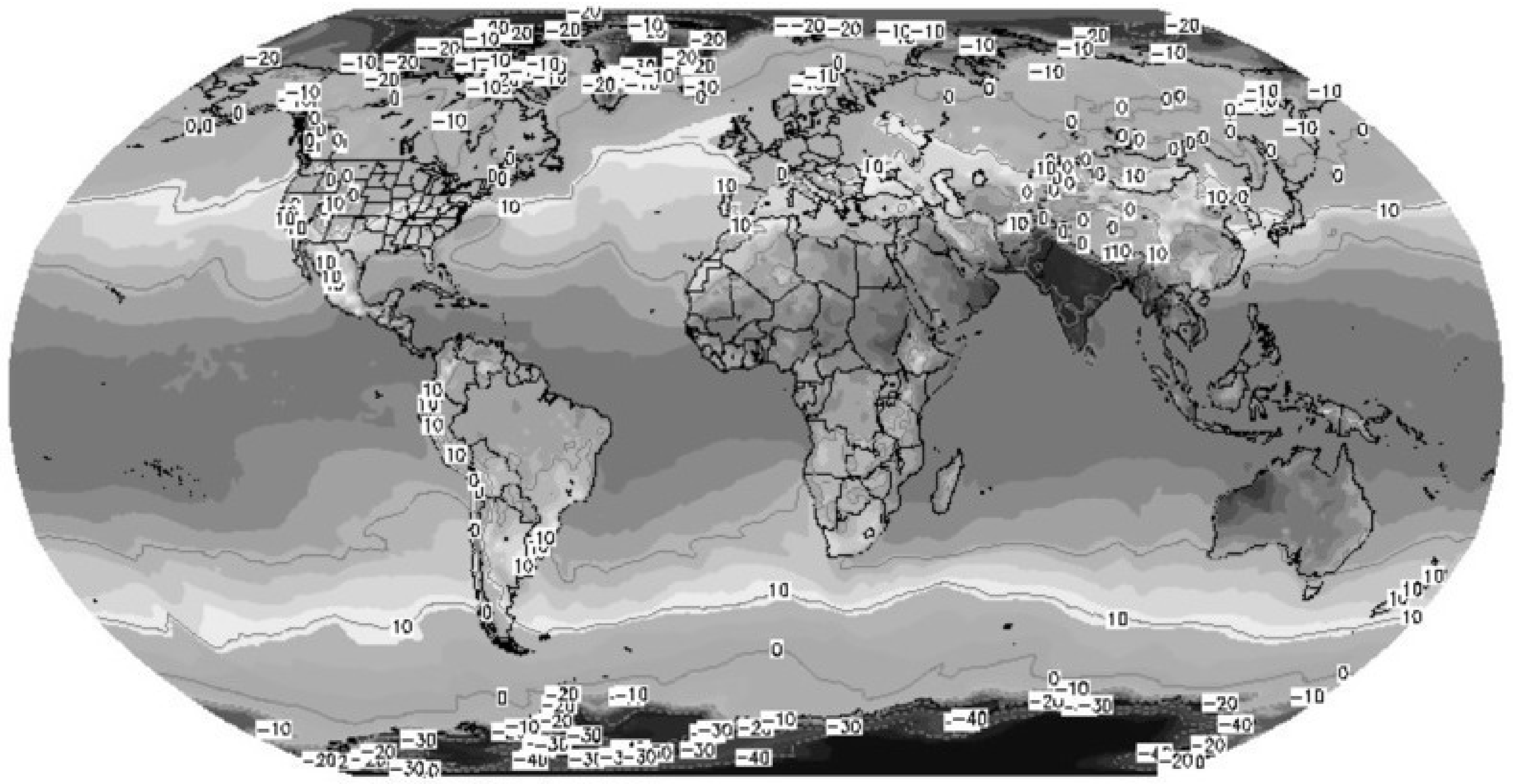
After discretization: $f(k_x\, \Delta x, k_y\, \Delta y) = I_{k_x, k_y}$



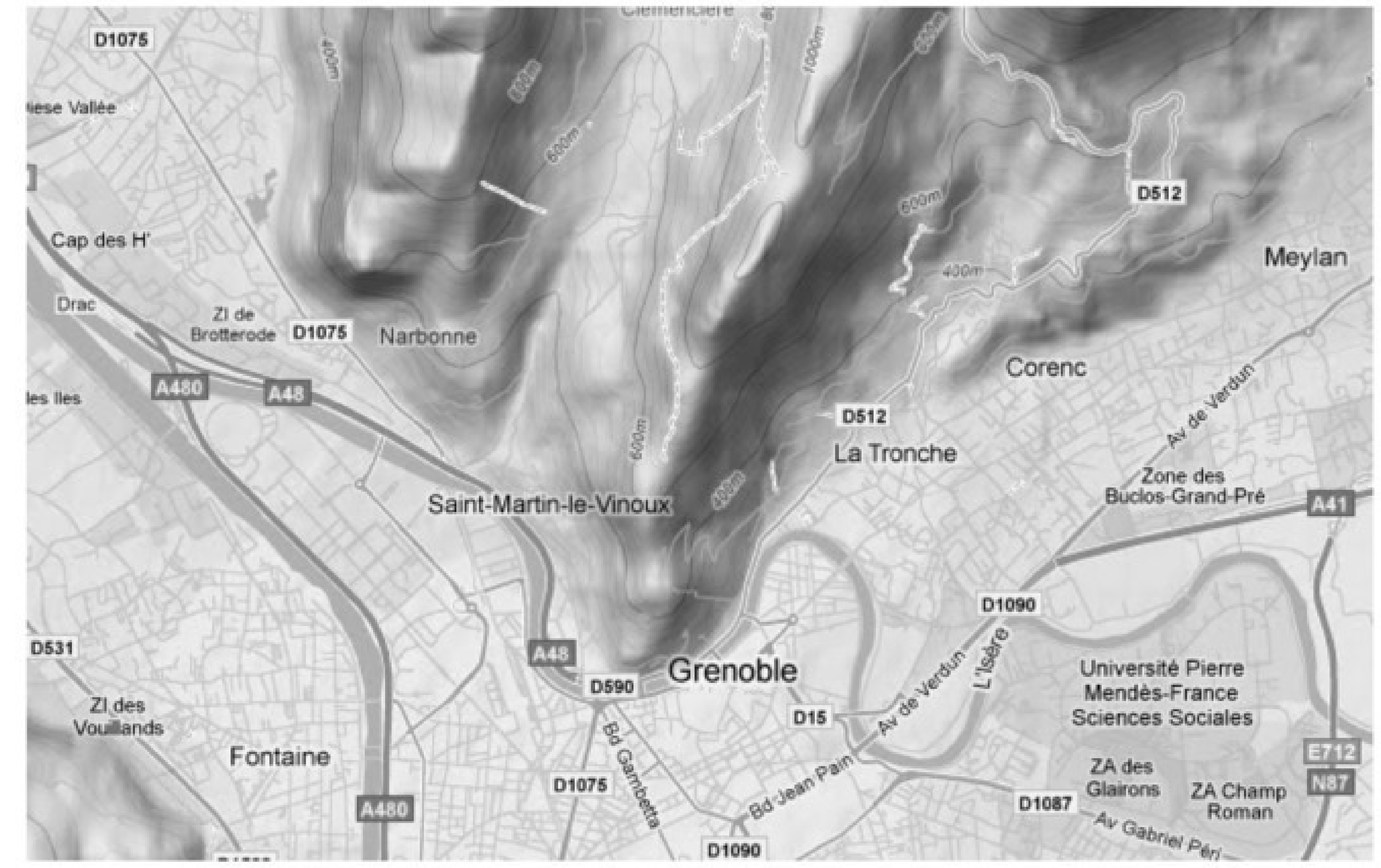
0.5	−0.2	1.1
1.5	0.5	0.9
−0.1	0.0	0.7



Surfacic scalar data : Example

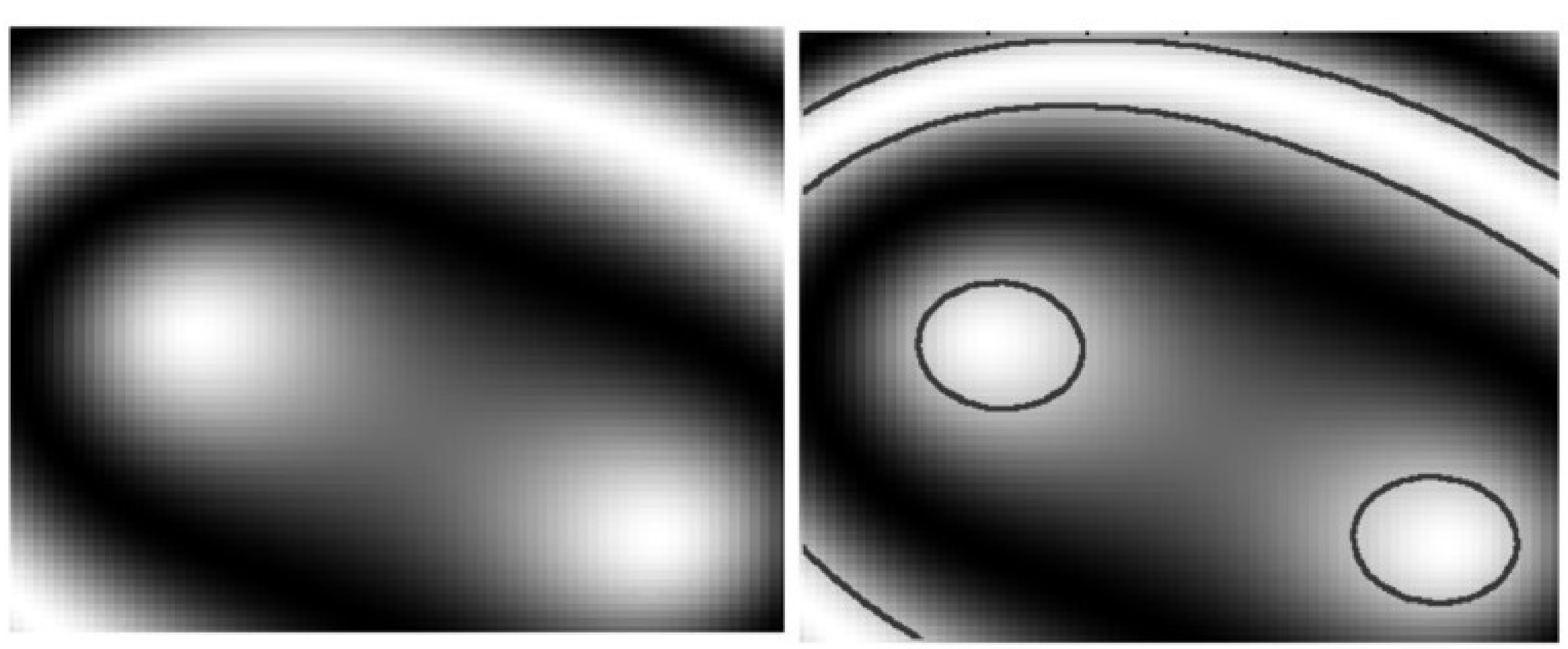


Surfacic scalar data : Example



Isolines

Goal: Trace curves on a specific value
Called: isolines, isocurves, level sets, ...

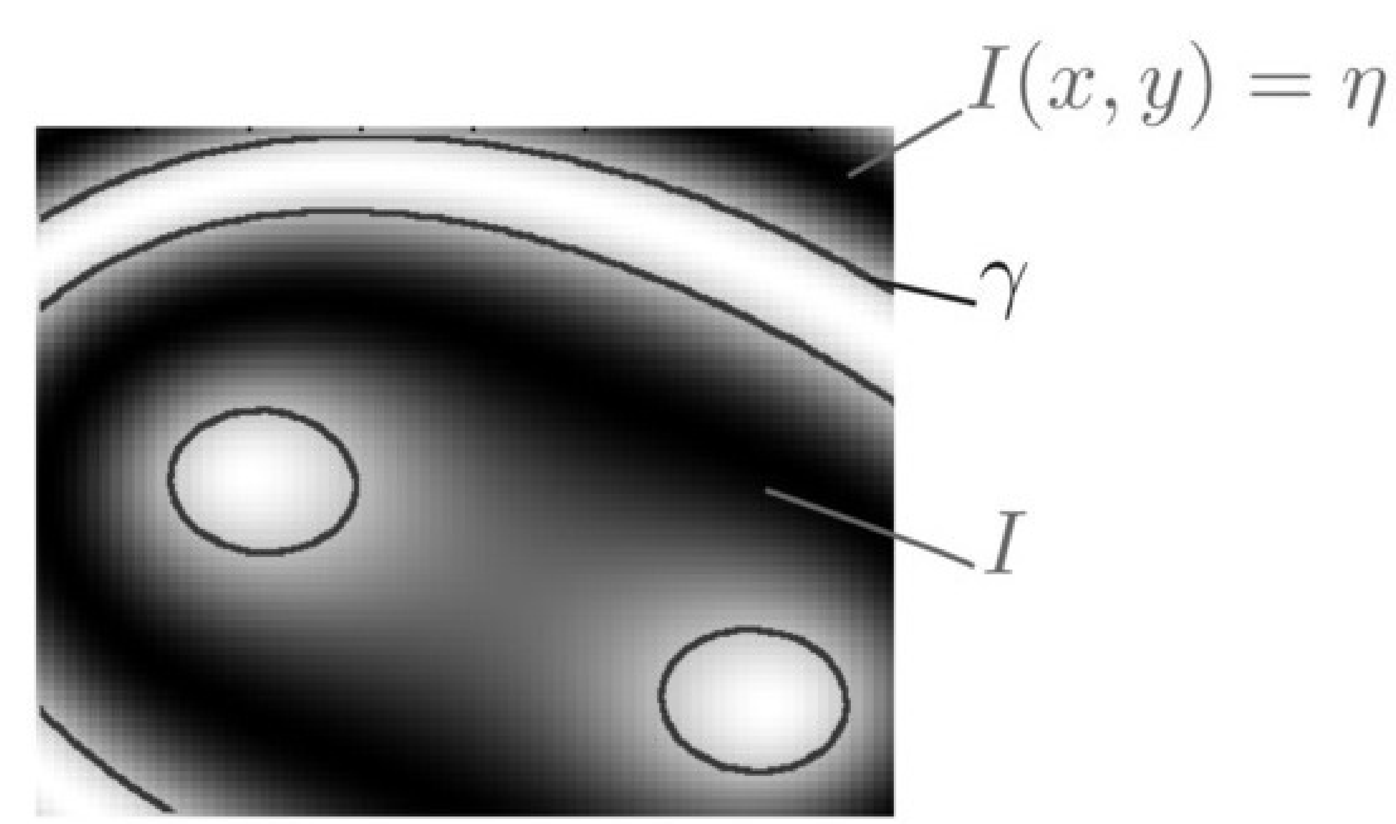


Isolines : input/output

Input: Scalar values on a regular discrete grid + isovalue η

Output: Set of curves $\{\gamma = (x,y) \in \mathbb{R}^2 | I(x,y) = \eta\}$

(degenerated cases: points, regions)



Example: continuous cases

For $\eta = 0$

$$F_1 = 1$$

$$F_2 = 1$$

$$F_3 = (x - x_0)^2 + (y - y_0)^2 - r_0^2$$

$$F_4 = F_3(x_0, y_0, r_0) + F_3(x_1, y_1, r_1)$$

$$F_5 = F_3(x_0, y_0, r_0) \times F_3(x_1, y_1, r_1)$$

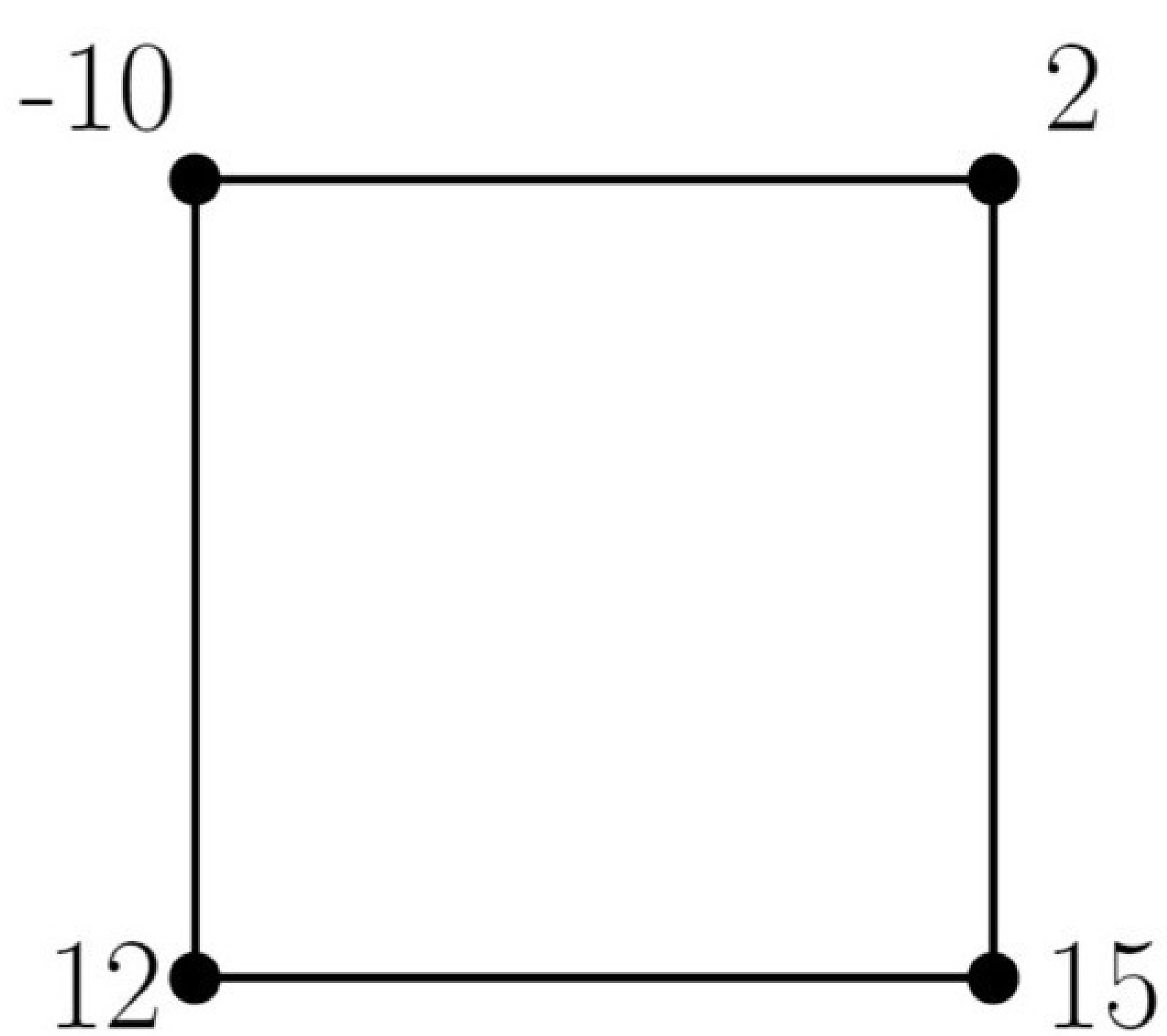
We can defines a curve by its implicit equation
+ Arbitrary topology

Marching squares

-61	-45	-42	-52	-72	-91	-99	-89
-17	8	13	-2	-34	-69	-94	-98
25	57	64	43	2	-45	-84	-99
51	87	94	71	25	-30	-76	-99
51	87	94	71	25	-30	-76	-99
25	57	64	43	2	-45	-84	-99
-17	7	13	-2	-34	-69	-94	-98
-61	-45	-42	-52	-72	-91	-99	-89

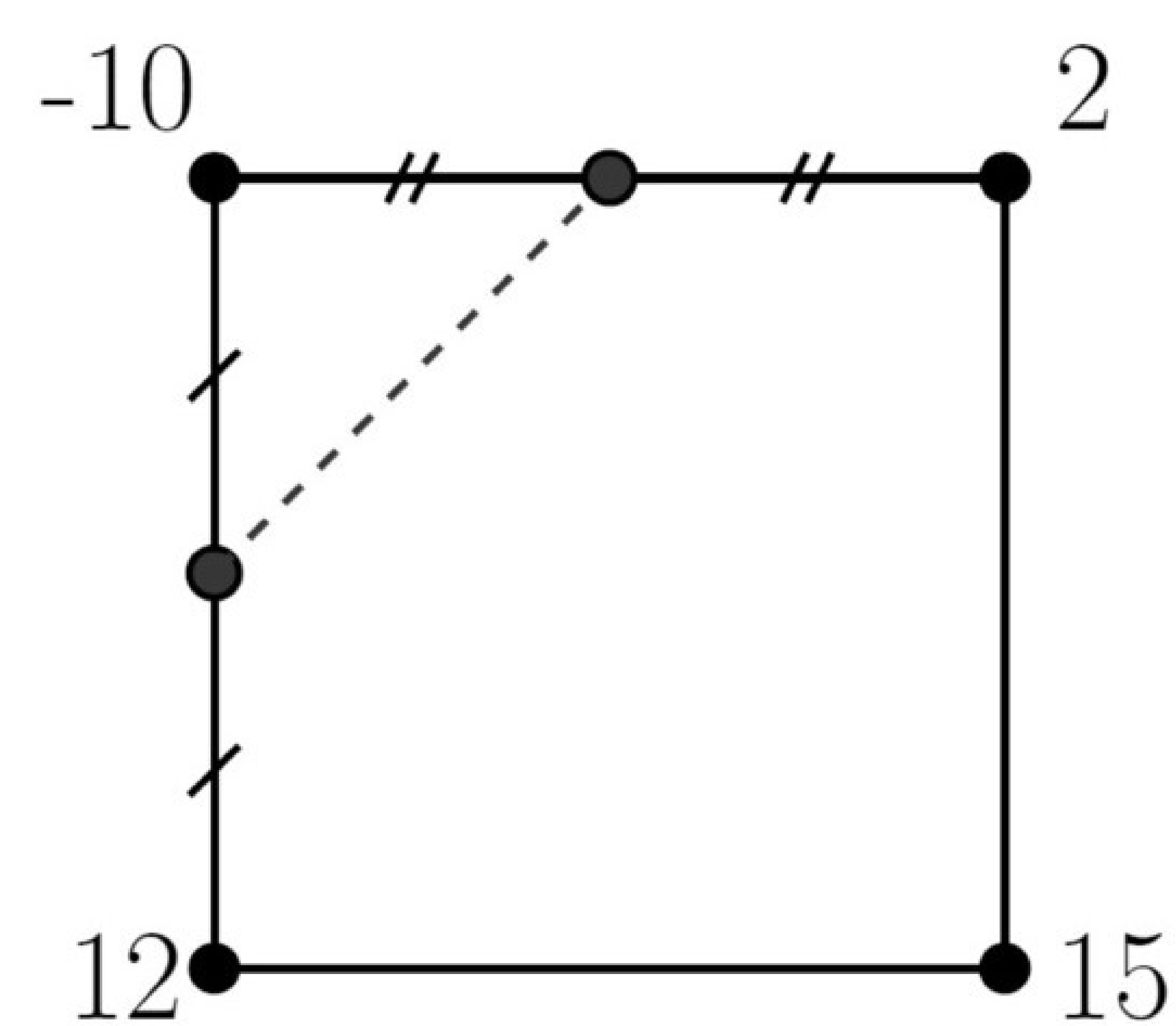
Marching squares

Which curve should we consider?



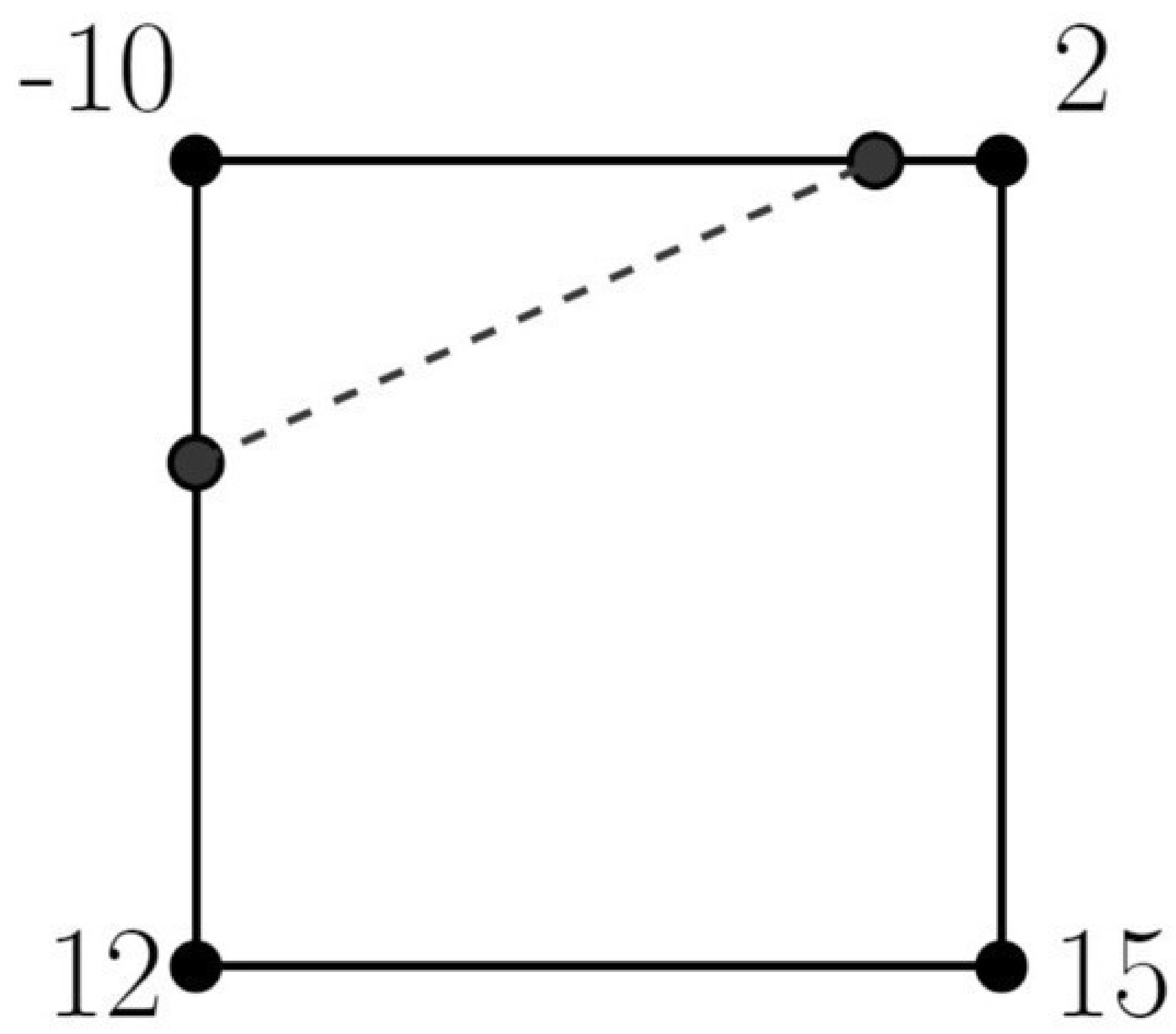
Marching squares

Middle of the edges



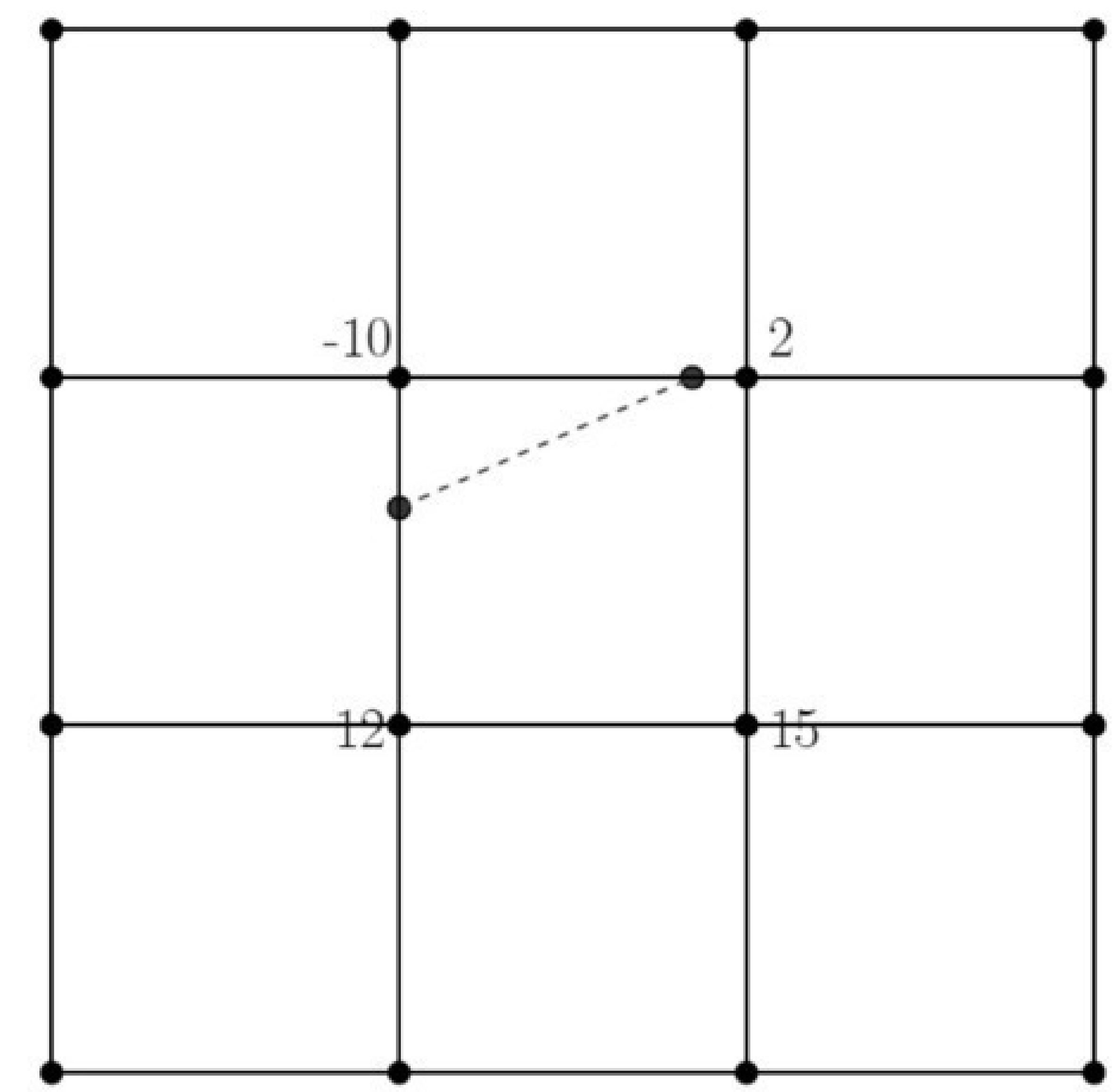
Marching squares

Interpolation (bi-)linear



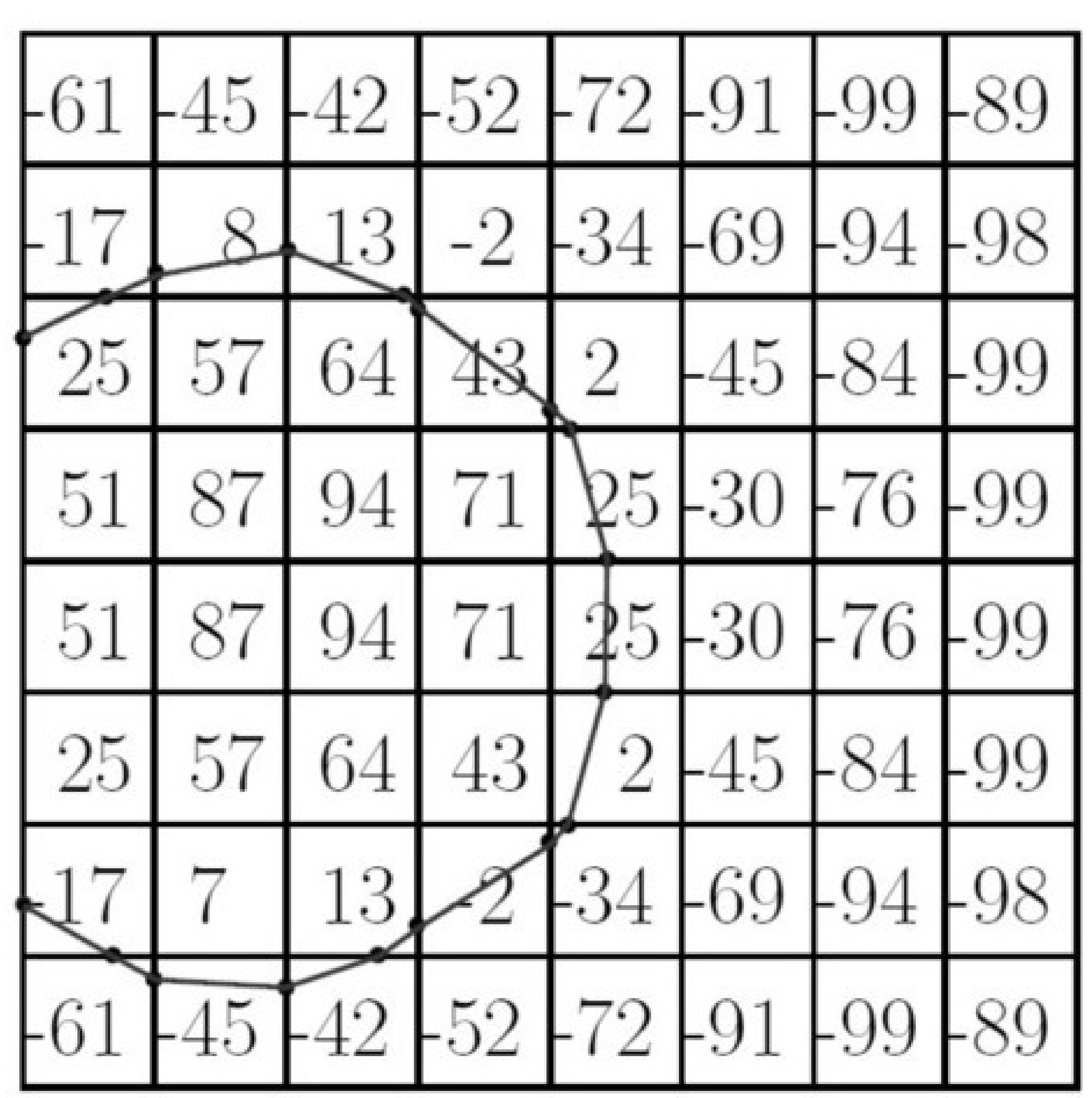
Marching squares

Other interpolation (cubic, spline, etc)



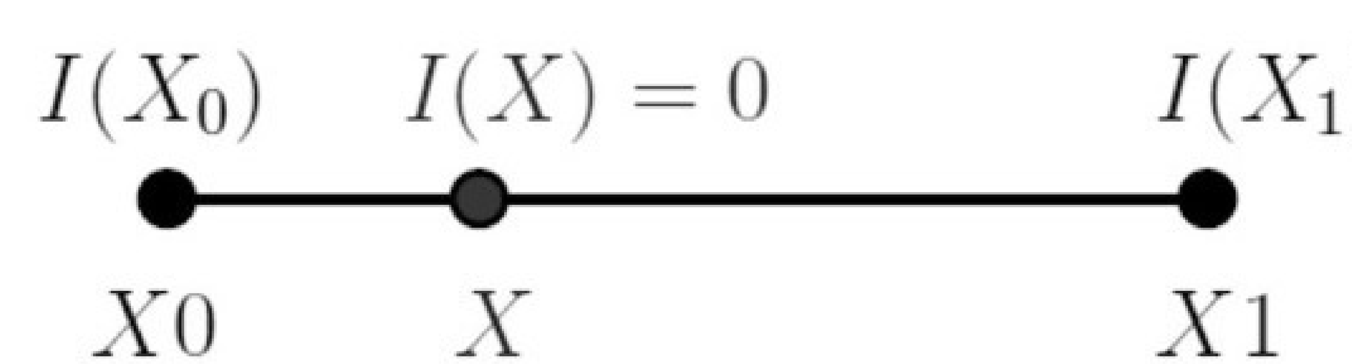
Marching squares

Result for the previous grid



Interpolation

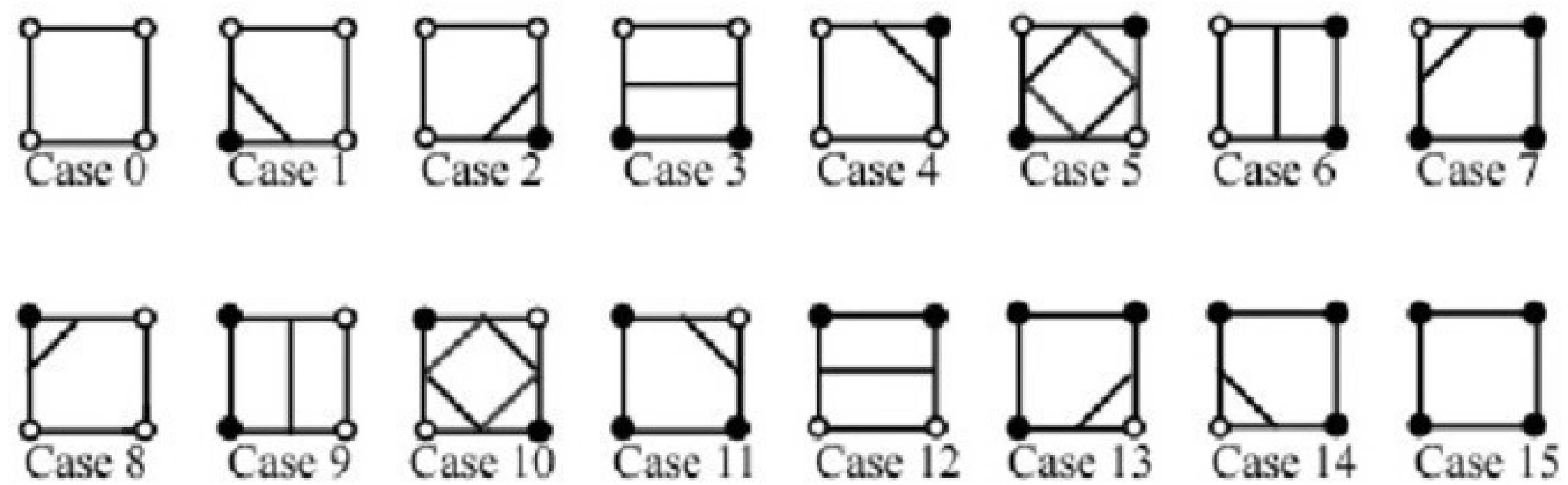
Zeros finding in linear interpolation



$$X = \frac{I(X_1)X_0 - I(X_0)X_1}{I(X_1) - I(X_0)}$$

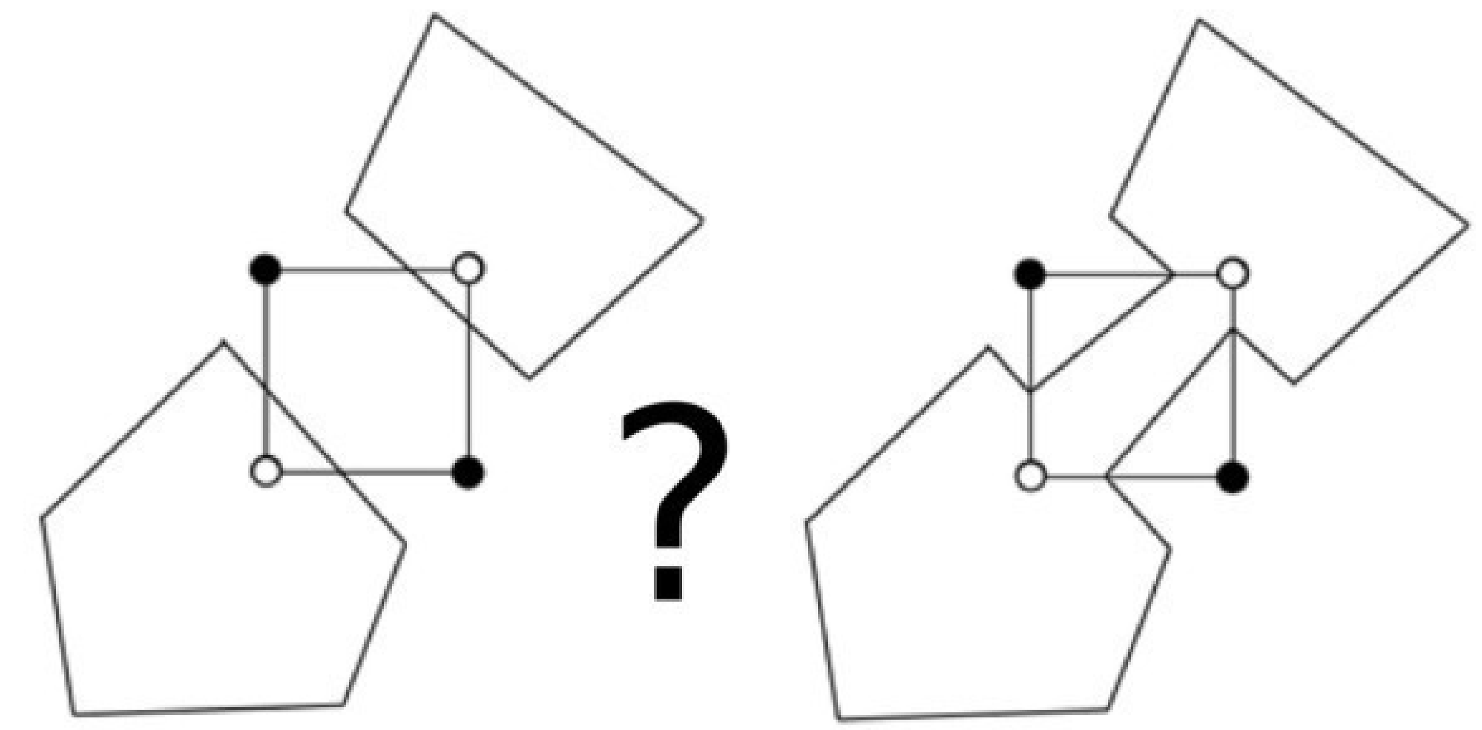
Marching square

For a single cell: 16 different possibilities



Marching square

Some undeterminated case



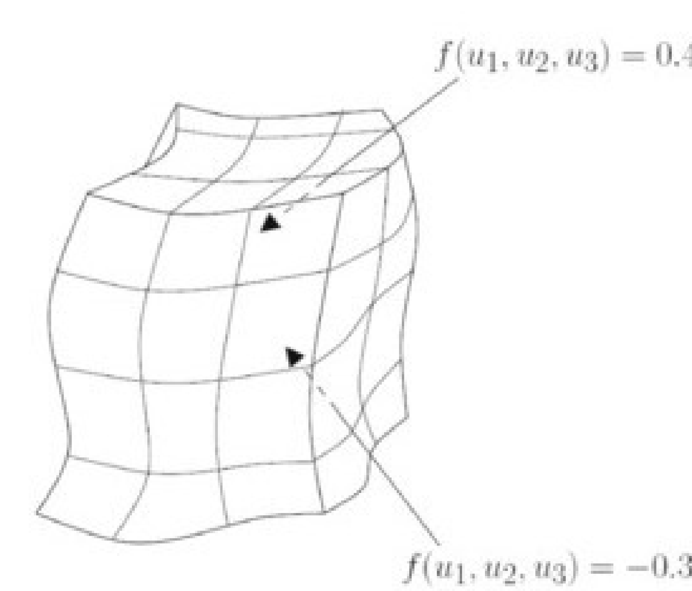
Volume data

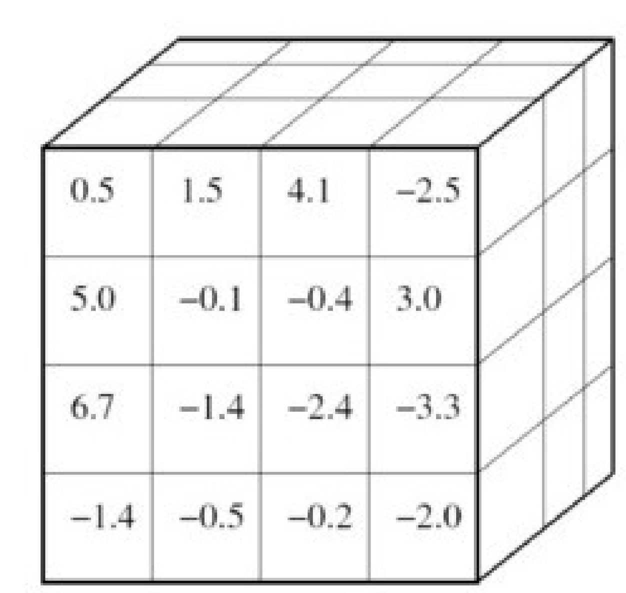
Volume data : Notation

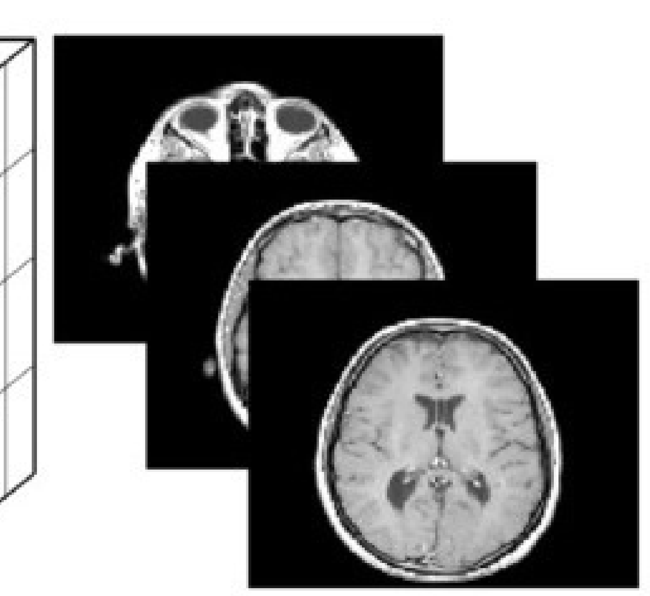
For volume field: $f(u_1, u_2, u_3) = I \in \mathbb{R}$

Very often: $f(x, y, z) = I$

After discretization: $f(k_x\,\Delta x, k_y\,\Delta y, k_z\,\Delta z) = I_{k_x,k_y,k_z}$

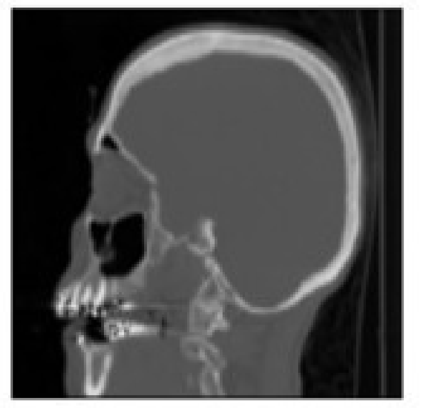




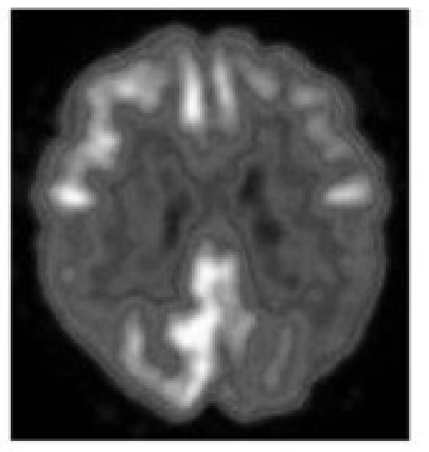


Medical Imaging Modalities

X-Ray
Anatomical
Absorbtion measurement
(inverse problem)



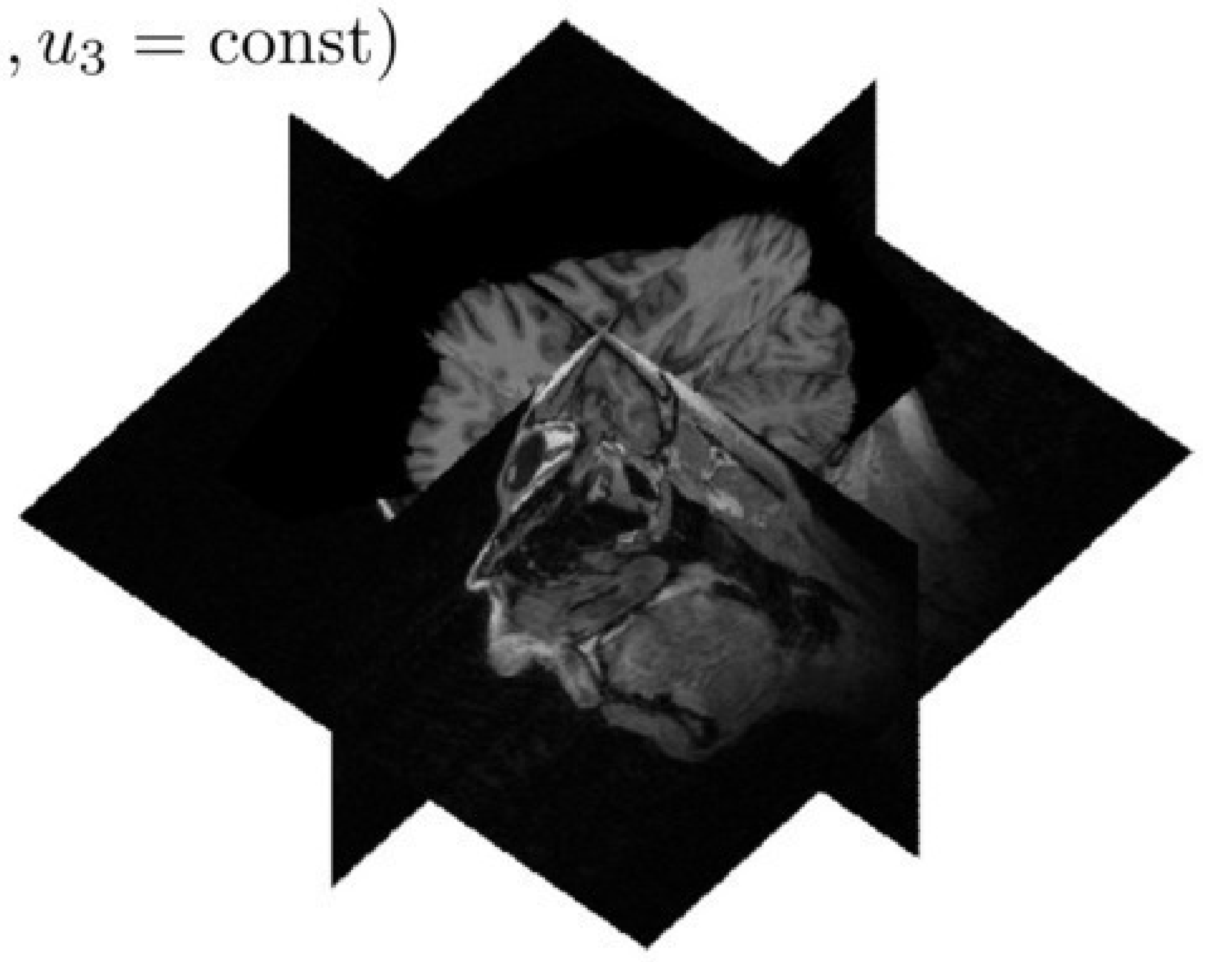
Nuclear *(PET, SPECT)*
Functional
Attenuated emission
(inverse problem, noise)



MRI
Anatomical *(MRI, Angiography)*
or Functional
Density measurement *(direct)*

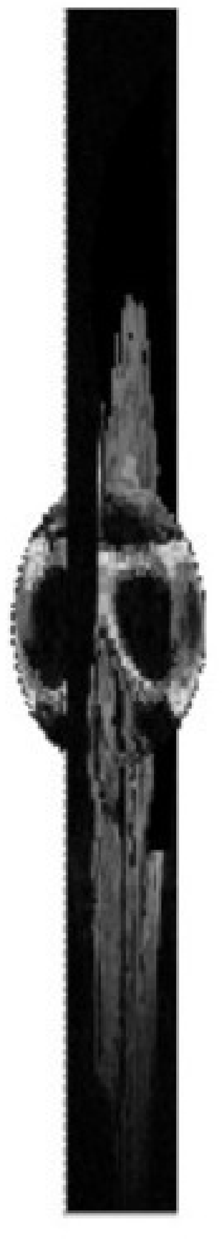
Slicing visualization

Idea: Slice some surfaces on the volume
Encode the field as a color (gray level, texture, etc)
Draw $I(u_1 = \text{const}, u_2, u_3)$
$I(u_1, u_2 = \text{const}, u_3)$
$I(u_1, u_2, u_3 = \text{const})$



Slicing visualization

We can use more general surfaces
Which is the best surface ?



Marching cubes

A common surface is the iso-surface

The isosurface of isovalue η of the I function is the set

$$\{(x, y, z) \in \mathbb{R}^3 | I(x, y, z) = \eta\}$$

In making η evolving, we obtain different surfaces

How to triangulate such implicit surface ?

Marching cubes : Examples

For $\eta = 0$

$$F_1 = 1$$
$$F_2 = 0$$
$$F_3 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r_0^2$$
$$F_4 = F_3(x_0, y_0, z_0, r_0) + F_3(x_1, y_1, z_1, r_1)$$
$$F_5 = F_3(x_0, y_0, z_0, r_0) \times F_3(x_1, y_1, z_1, r_1)$$

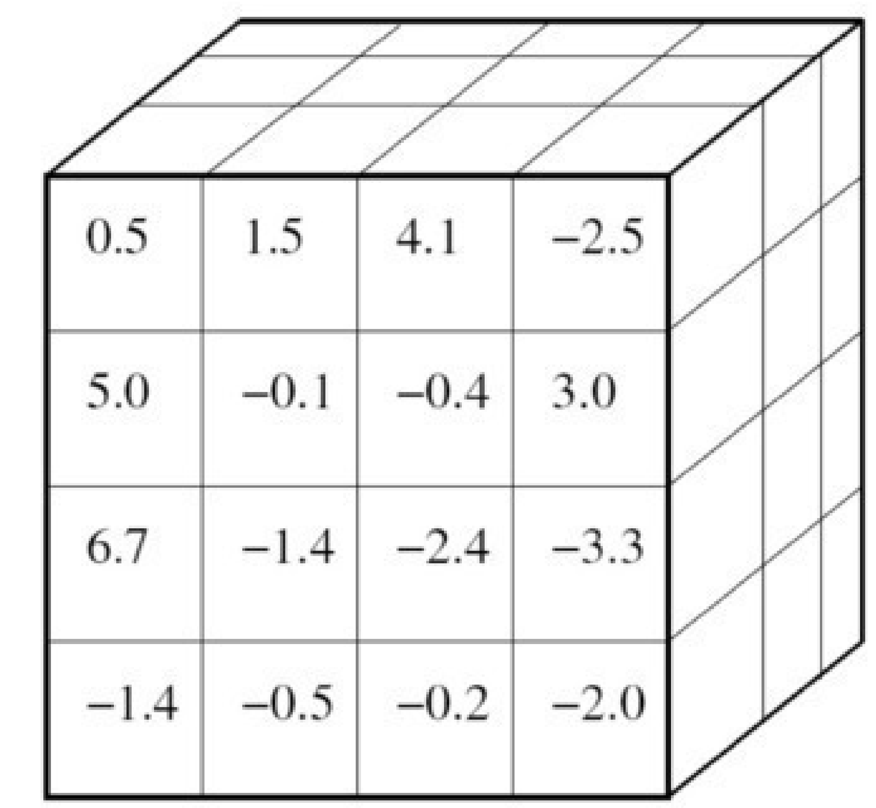
A surface can be defined by its equation
+ Arbitrary topology

Marching cubes : Intro

Goal: Build a triangulated surface from a discrete volumetric scalar field given by $I(x, y, z) - \eta$

First software patent in CG in 1985 from Lorensen & Cline.

Input data: 3D Grid in (x,y,z) of (Ni,Nj,Nk) voxels.



Marching cubes : Principle

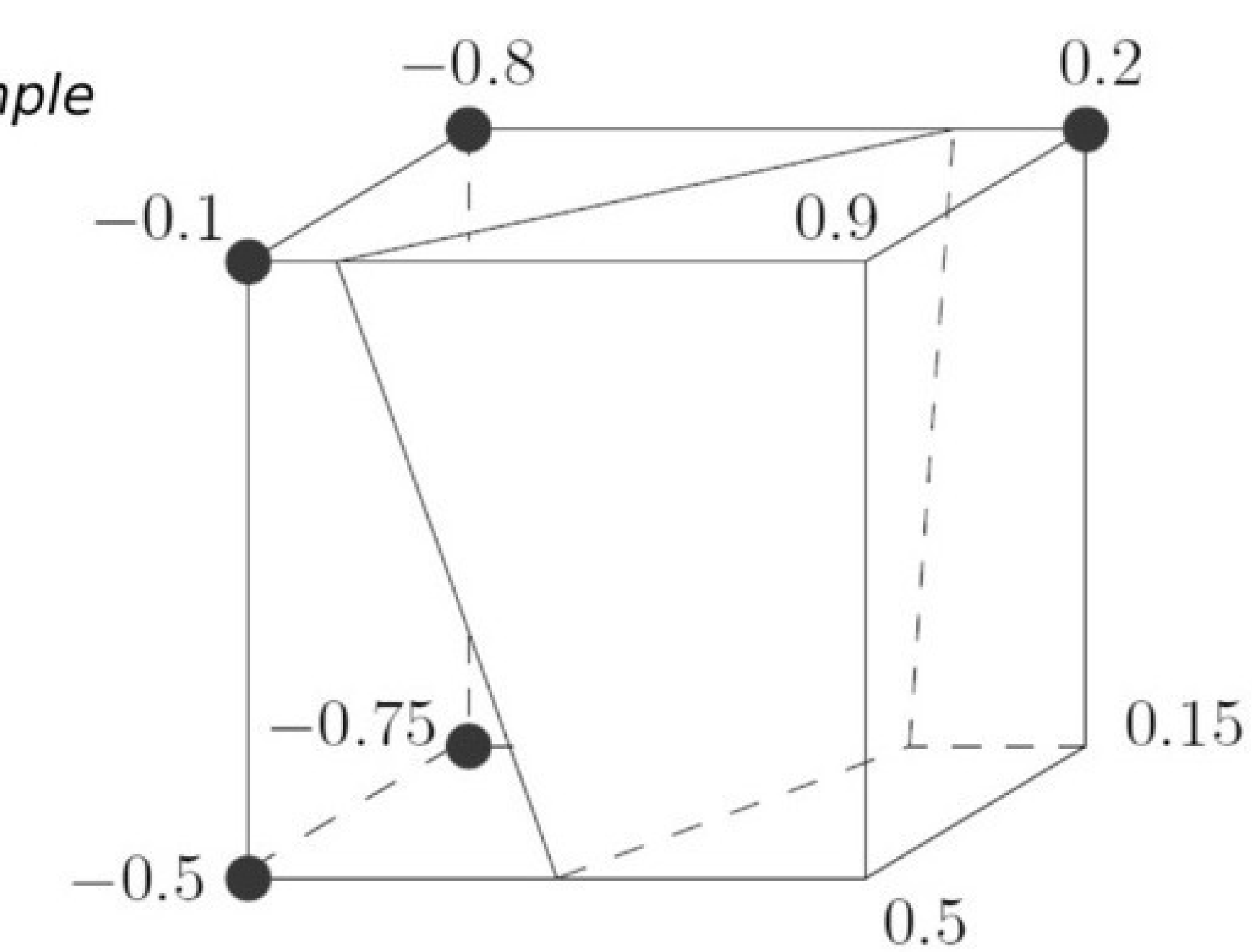
Traversal of the grid "cube by cube"

Compute the sign of $I(x, y, z) - \eta$

Check the possible cases

The 0 value is obtain by interpolation

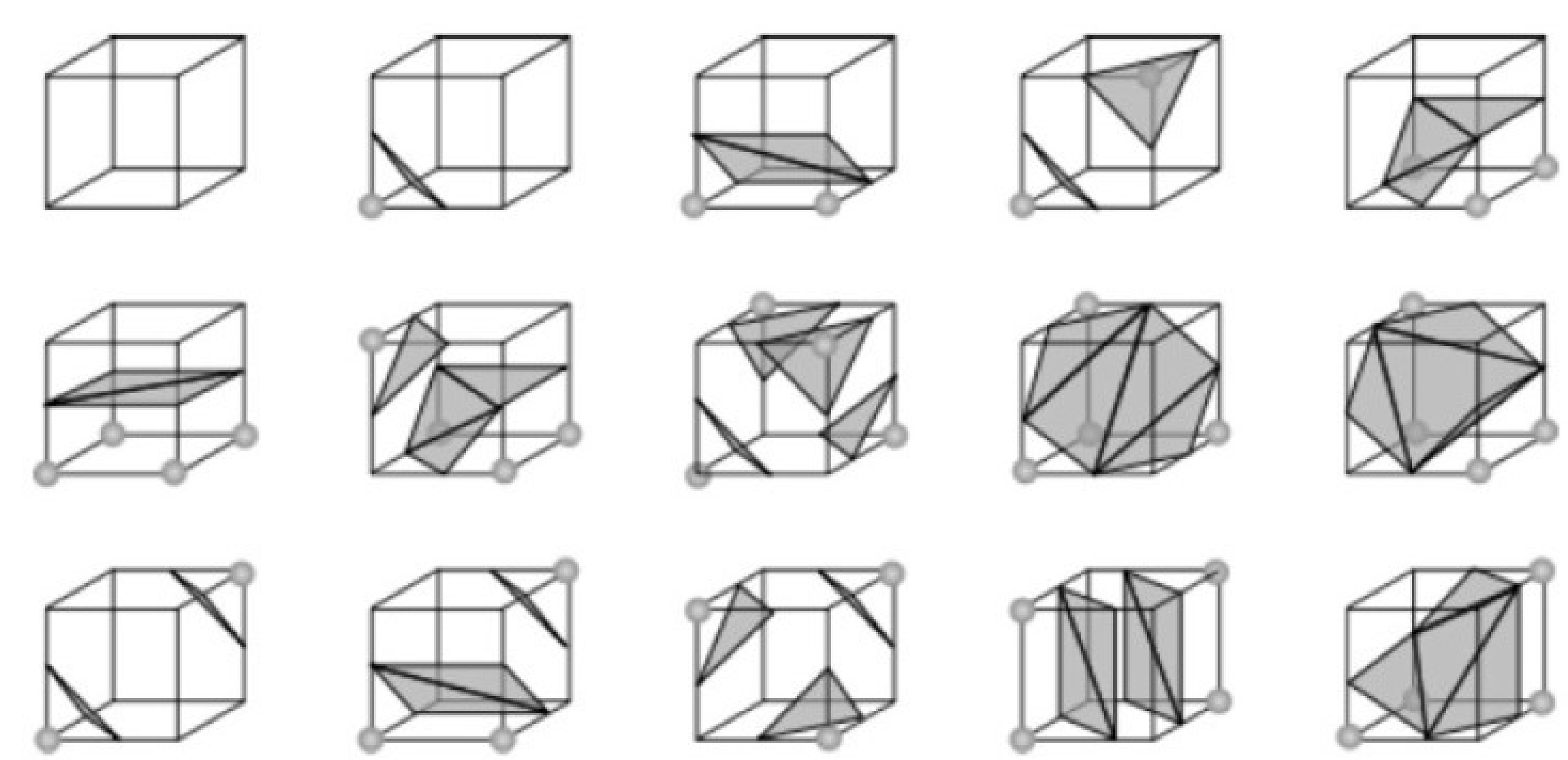
Example



Marching cubes : Different cases

A total of 256 possible cases

Only 15 basic cases



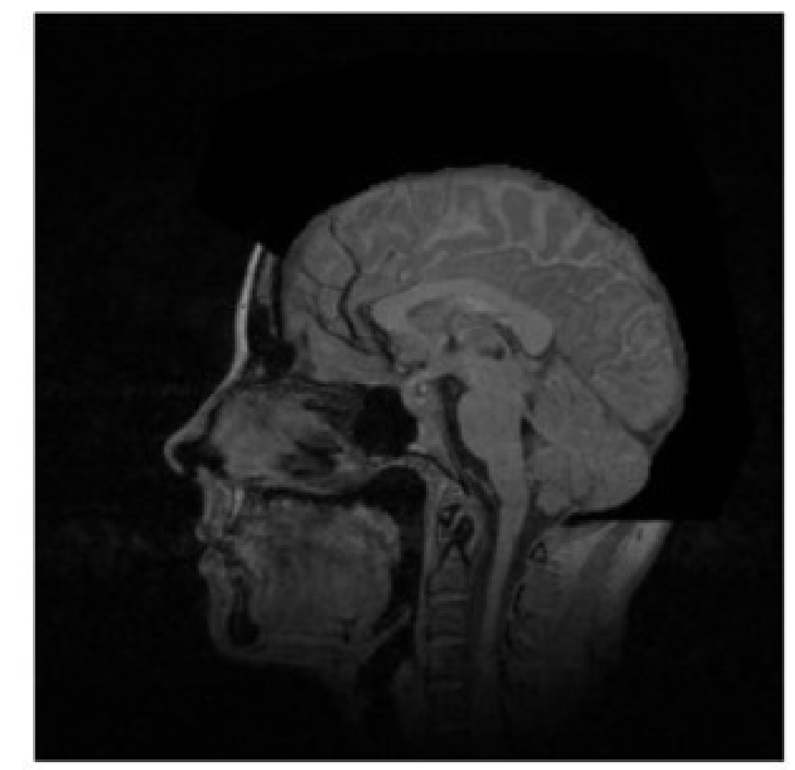
Marching cubes : Usage

+ Efficient

- Cubic aspect
  - Smooth volume
  - Smooth surface
  - Medical correctness
- Undetermined cases

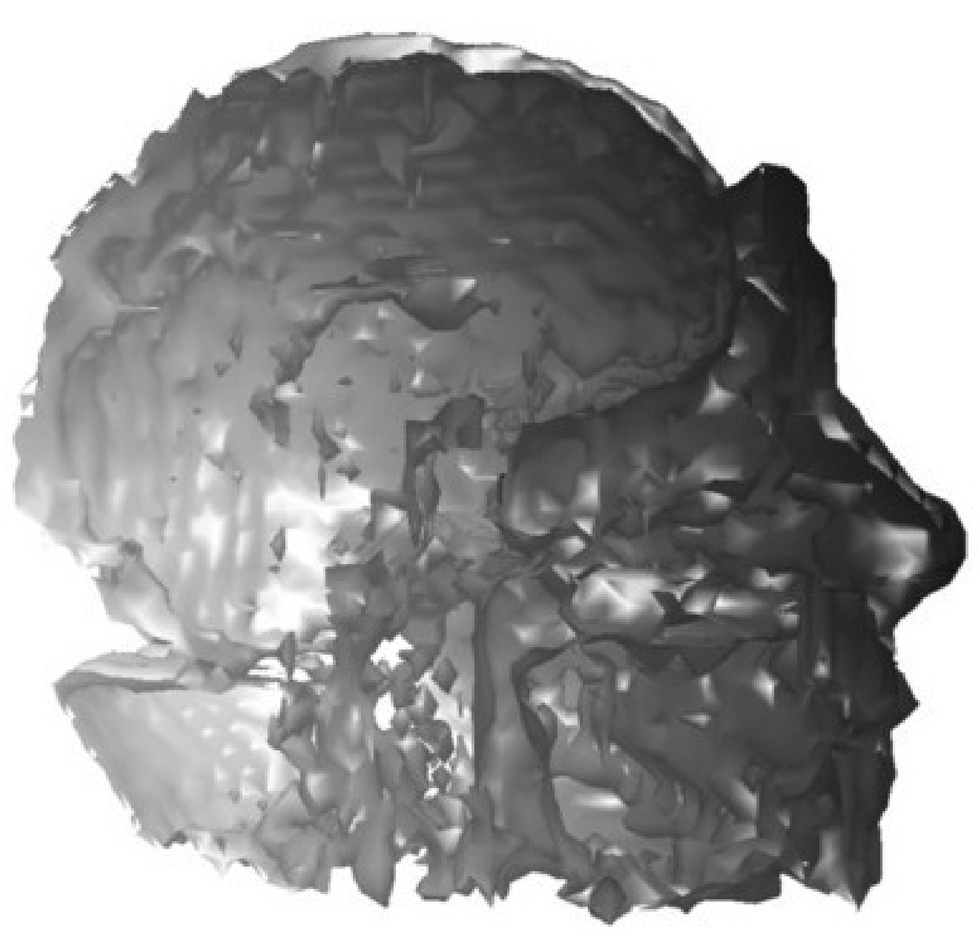
Isosurface example: MRI

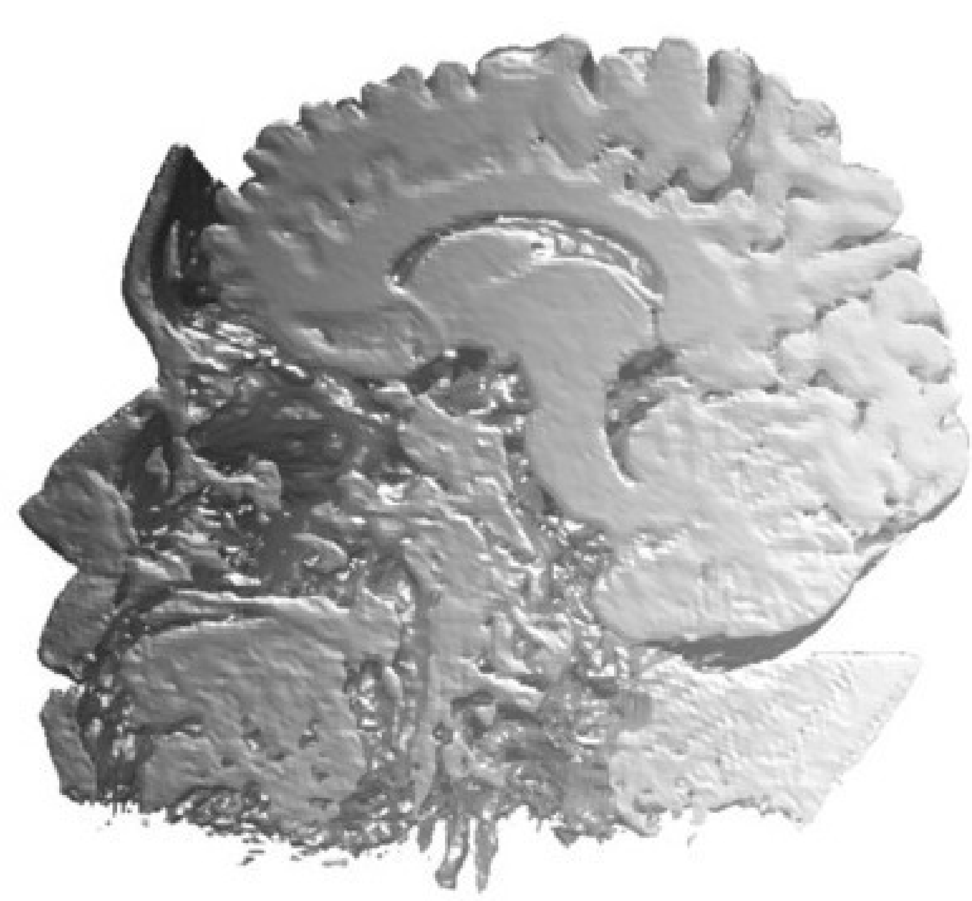
MRI Data (256 x 256 x 99)



Isosurface example: MRI

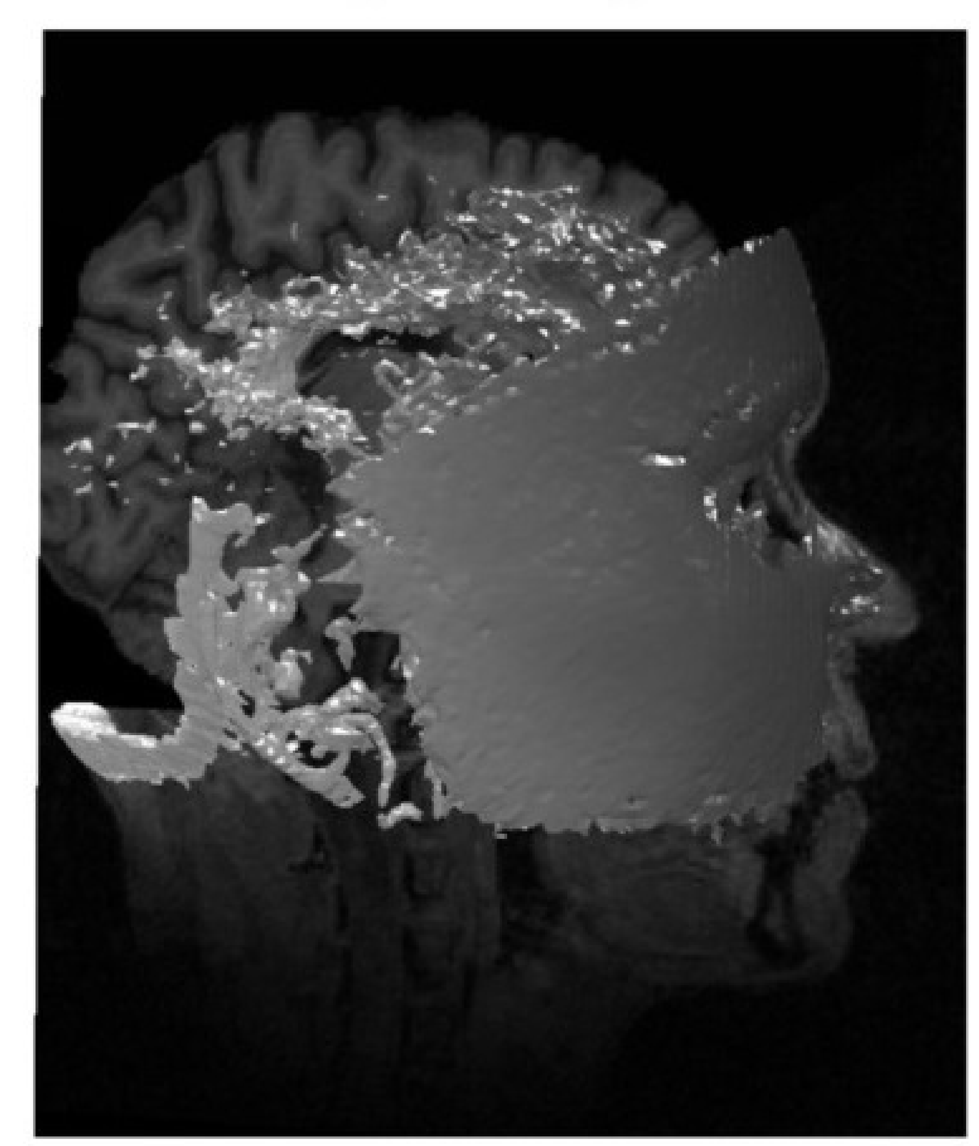
Can observe internal structure





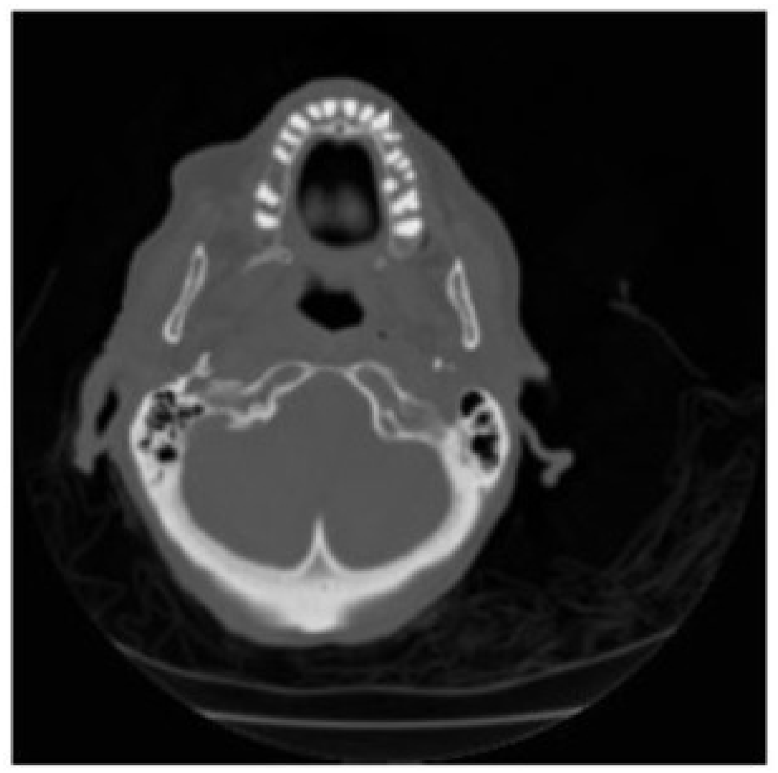
Isosurface example: MRI

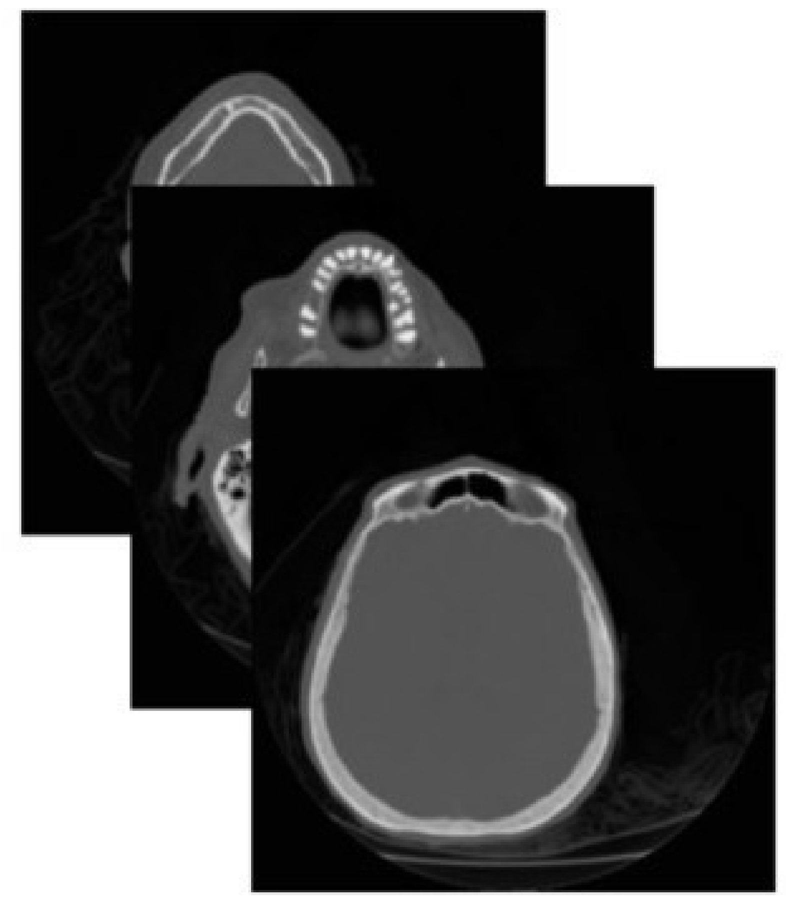
Combining Slicing + isosurface



Isosurface example: CT

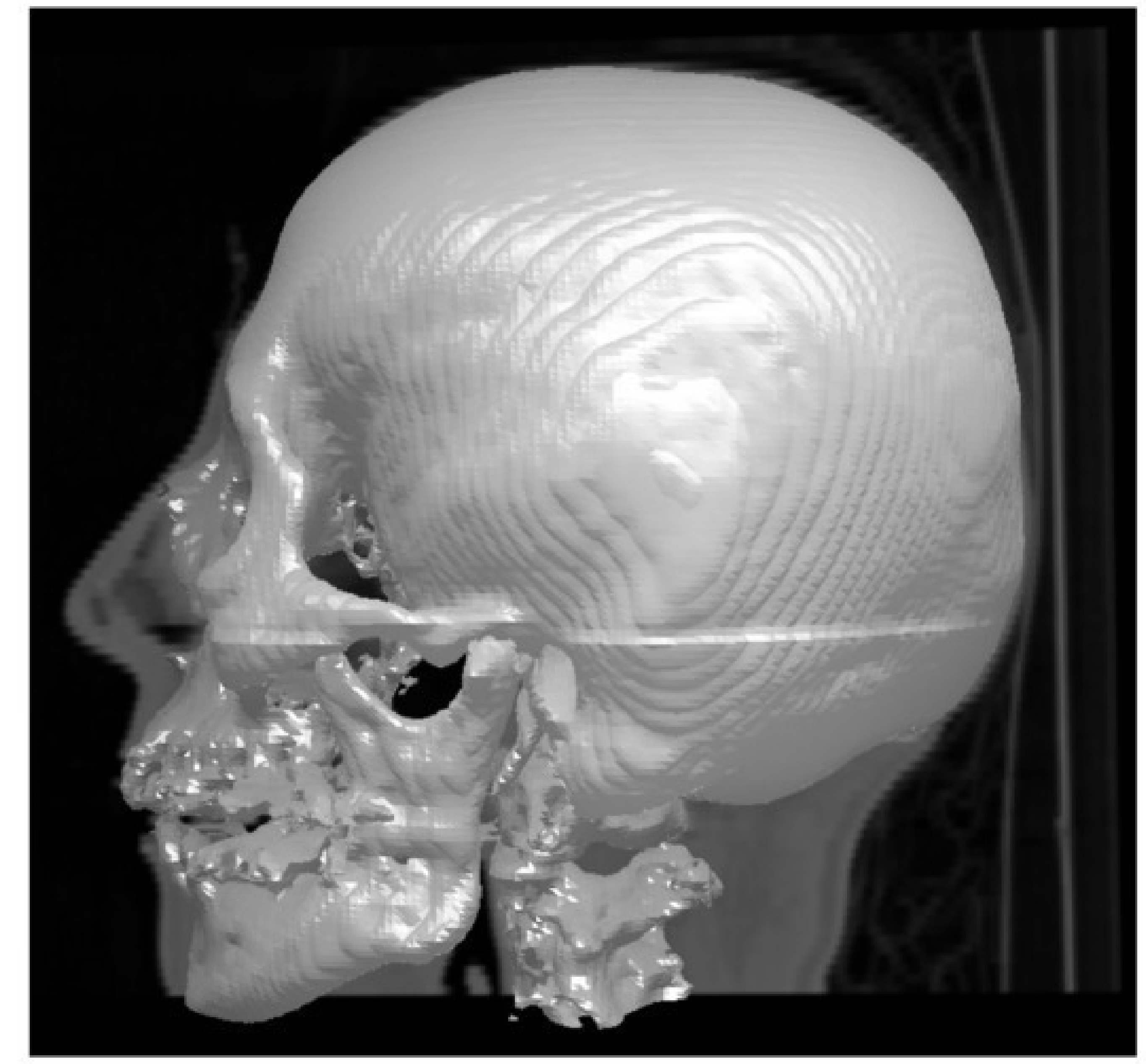
CT (X-Ray) data
Morpholocical data (skin, bone)
256 x 256 x 99





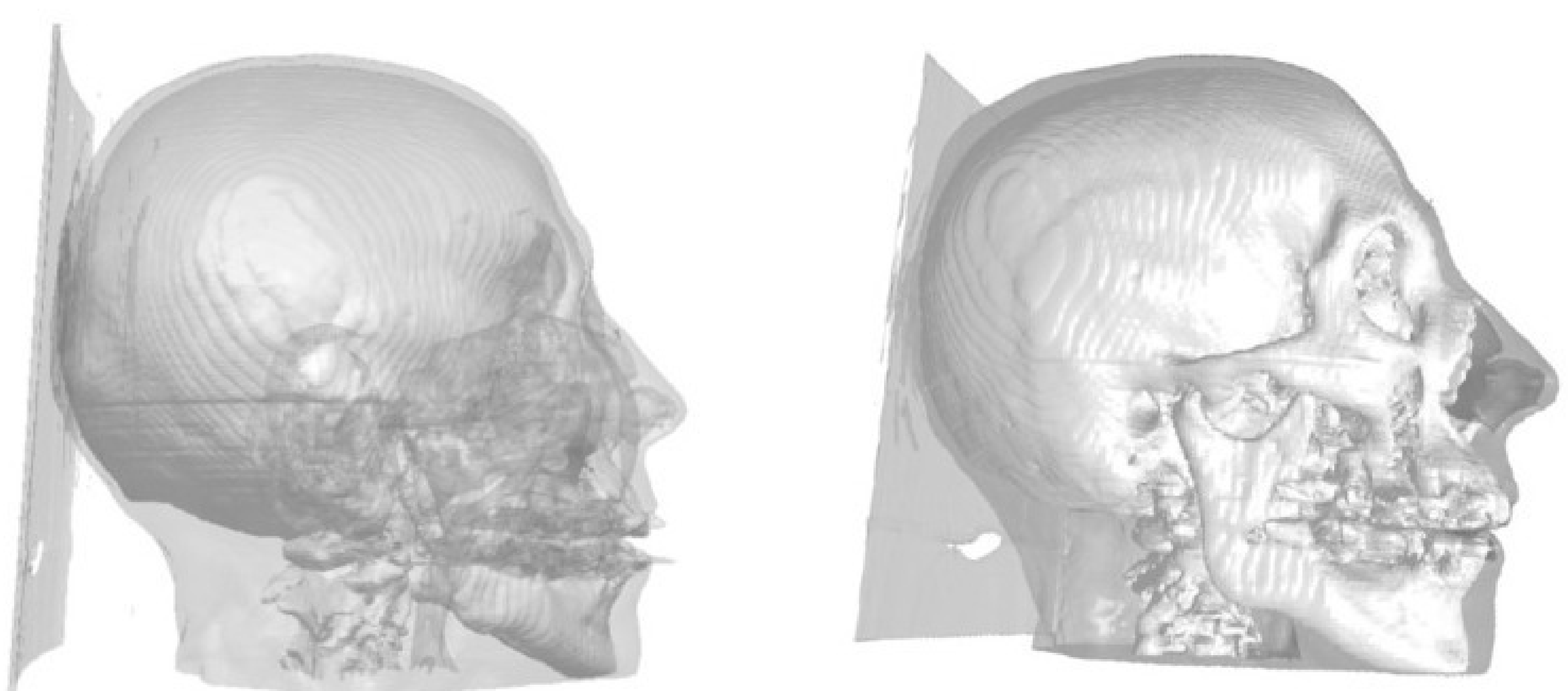
Isosurface example: CT

Combination slice + isosurface



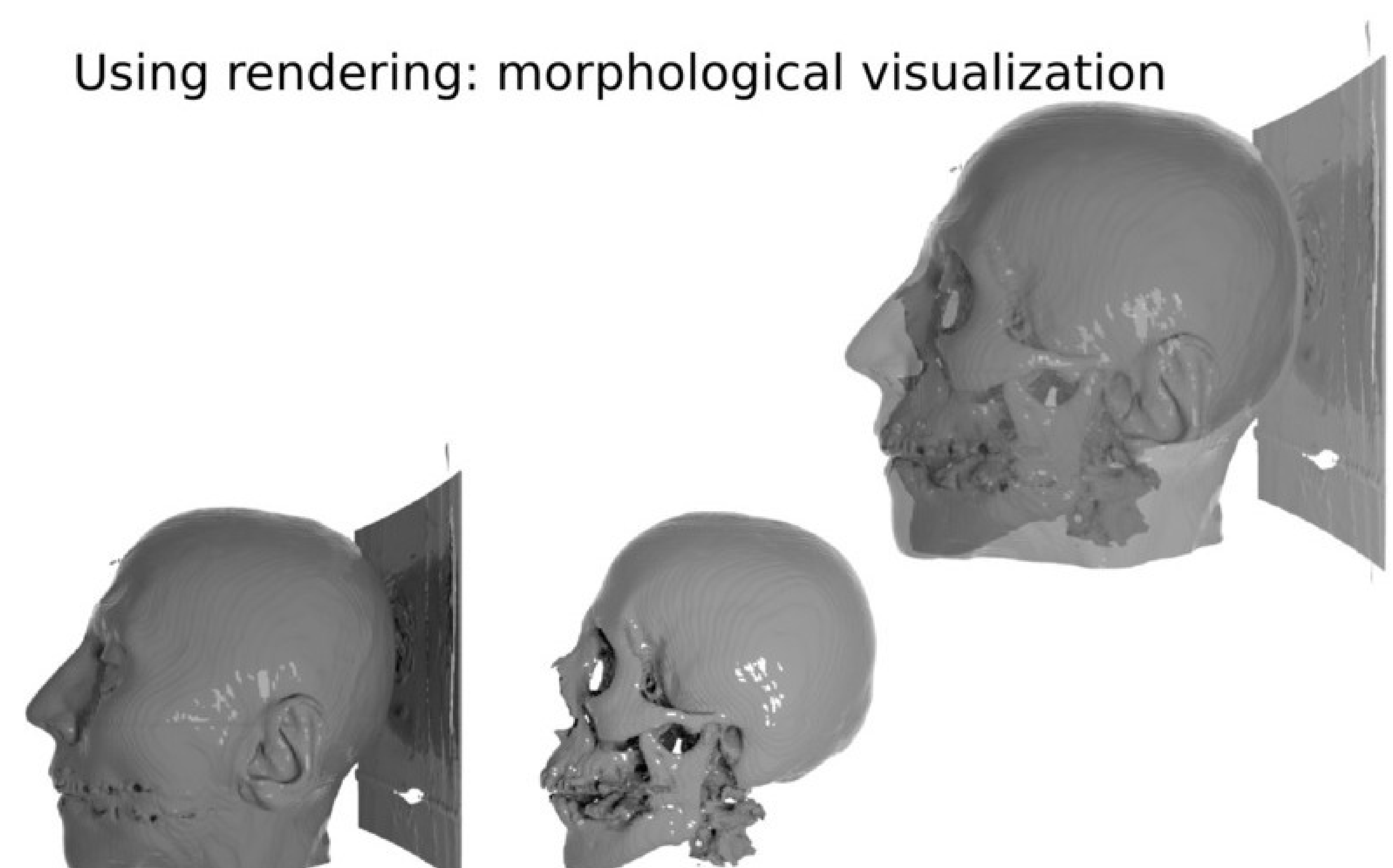
Isosurface example: CT

Accumulation of surface with transparency



Isosurface example: CT

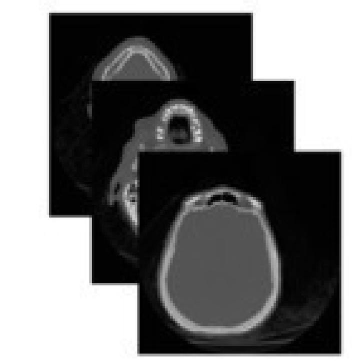
Using rendering: morphological visualization

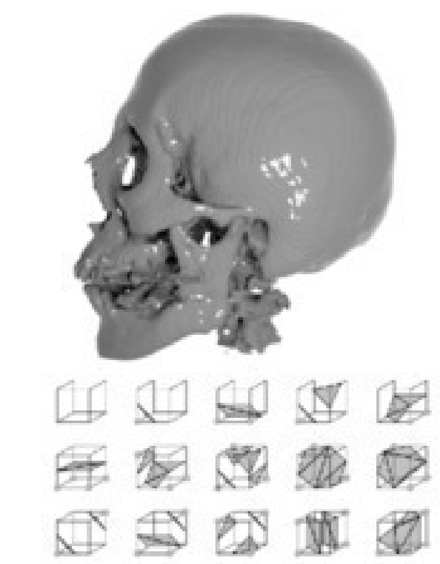


Ray casting

Ray casting

What we have seen:
- Slice in a volume
- Isosurface extraction (marching cubes/tetrahedron)





What we are going to see
- Transparency rendering = **volume rendering**

Ray casting

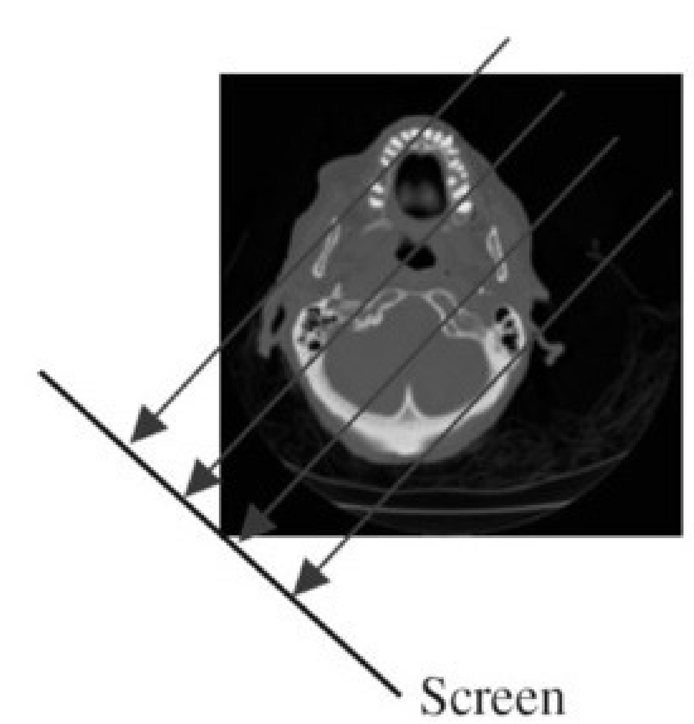
Surface rendering
- \+ Accurate
- \+ Data reduction
- \- Local information A-priori knowledge

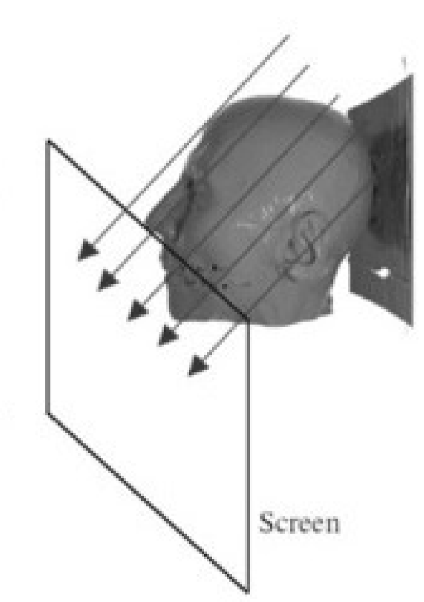
Volume rendering
- \+ Global information, direct visualization
- \- Not accurate, transparency can be tricky

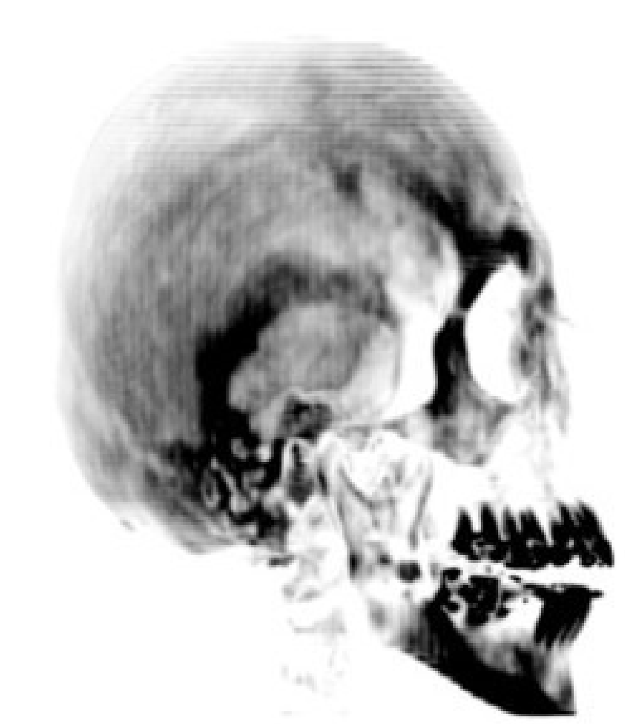
Pipe-line: Volume rendering to guide a surface extraction

Ray casting

Goal: Modeling a data acquisition using transparency
Problem: Human are not used to see transparency
General approach: Ray-tracing/casting = Throw rays and set the color as a function of path and obstacles.

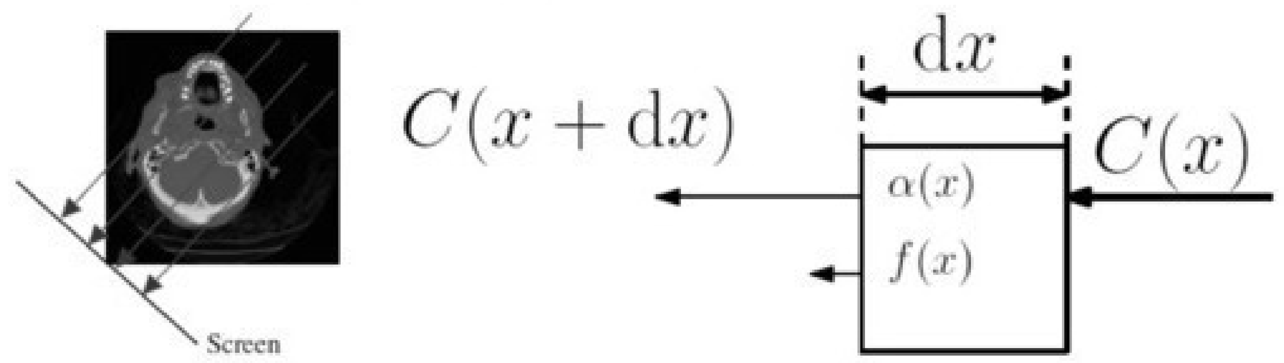






Ray casting: equations

For attenuated emission



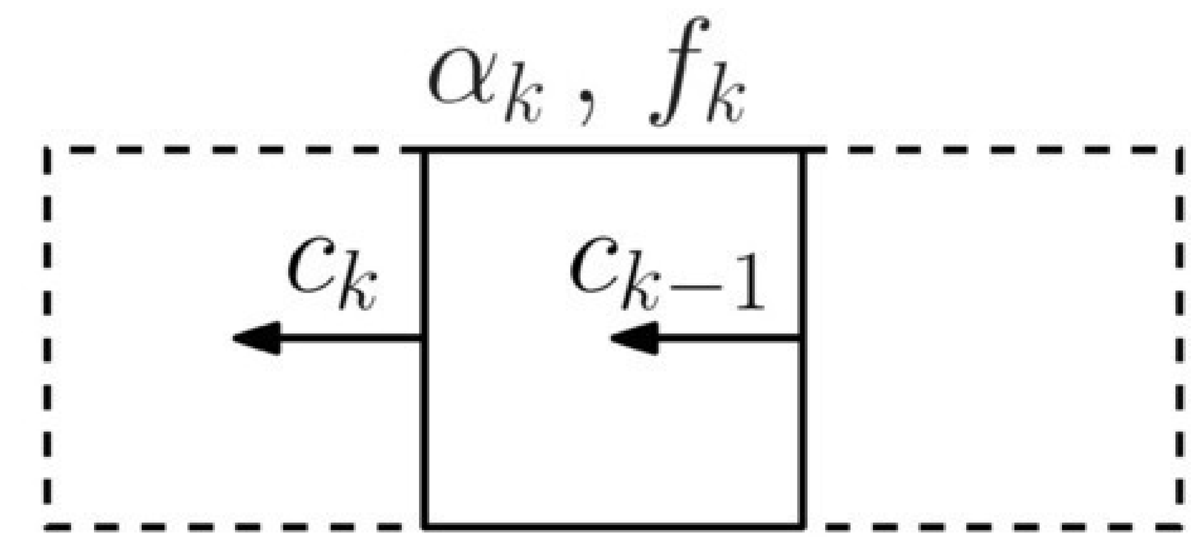
$$C(x+\mathrm{d}x) = [1-\alpha(x)\,\mathrm{d}x]\;C(x) + \;f(x)$$

$$\Rightarrow C'(x) = -\alpha(x)C(x) + f(x)$$

$$\Rightarrow C(x) = \left(\int_{x_0}^{x} f(u)\, e^{\int_{x_0}^{u}\alpha(t)\,\mathrm{d}t} + C(x_0)\right) e^{-\int_{x_0}^{x}\alpha(t)\,\mathrm{d}t}$$

For a given α, f, find C = Volume rendering
For a given C, find α, f = Tomography

Ray casting: discretization



In discrete form $c_k = (1-\alpha_k)\,c_{k-1} + f_k$

α_k, f_k are functions of the intensity I of the current voxel
Can also depends on the derivatives

ex. $\alpha_k = A\Delta x\, I_k,\ f_k = B\,\Delta x\, I_k$

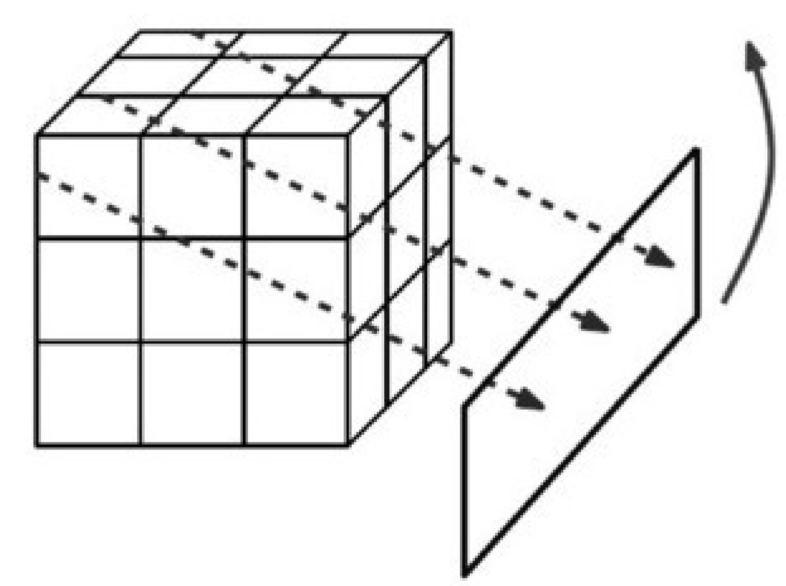
More generaly, transfert functions are used

$$\alpha_k = \mathcal{F}(I_k) \quad f_k = \mathcal{G}(I_k)$$

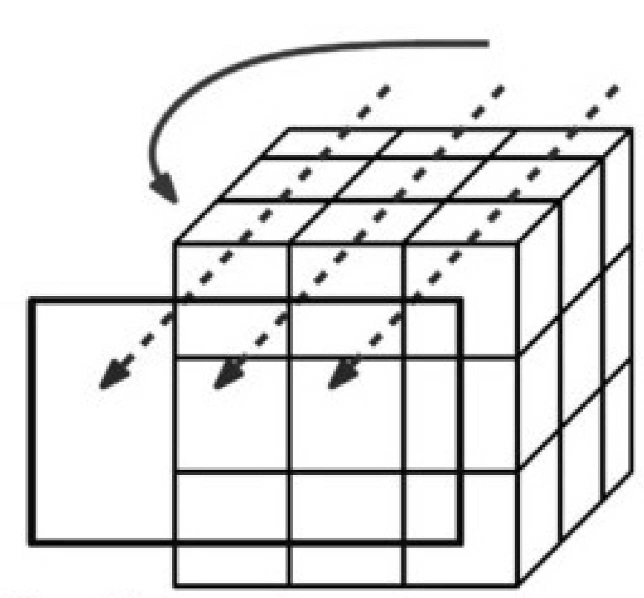
Ray casting: Implementation

2 Approches

Throw rotated lines in fixed grid



Rotate grid and integrate along fixed axis
(3D texture)



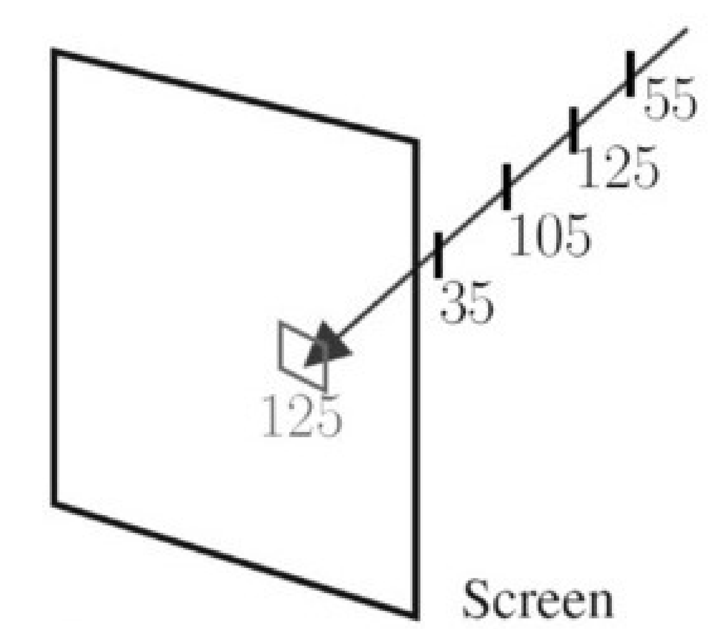
Trivial parallelisation

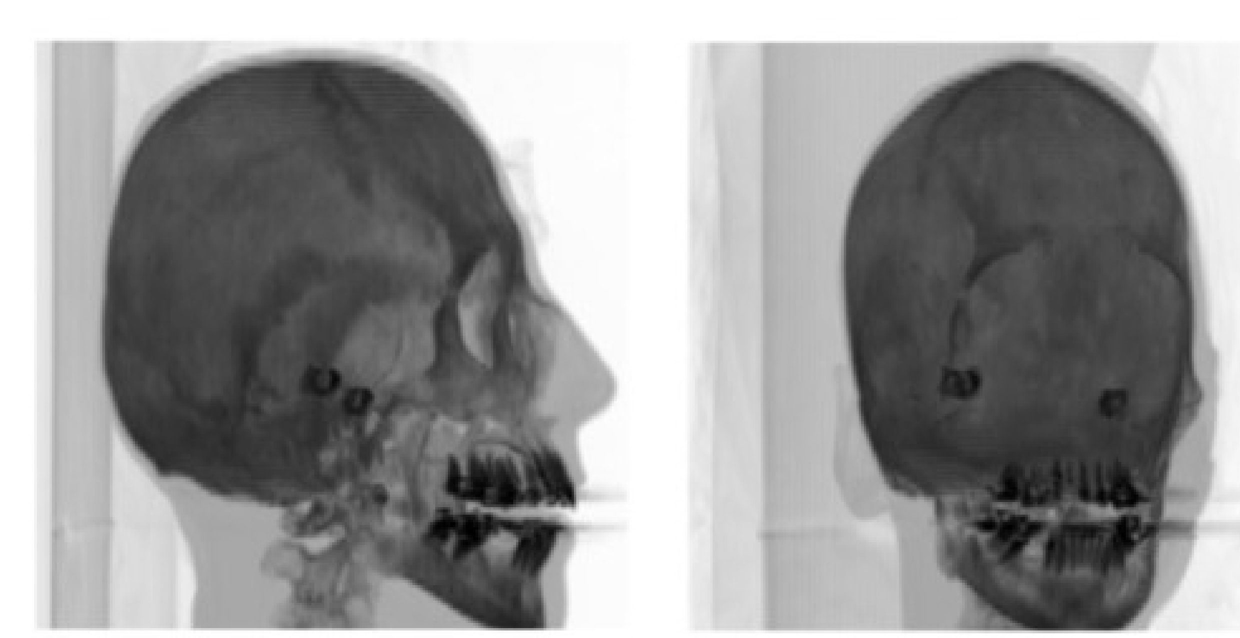
MIP

MIP = Maximum Intensity Projection $c = \max_k(I_k)$

+ Fast, simple
+ Standard in medical domain

- No information about depth
  (without motion)



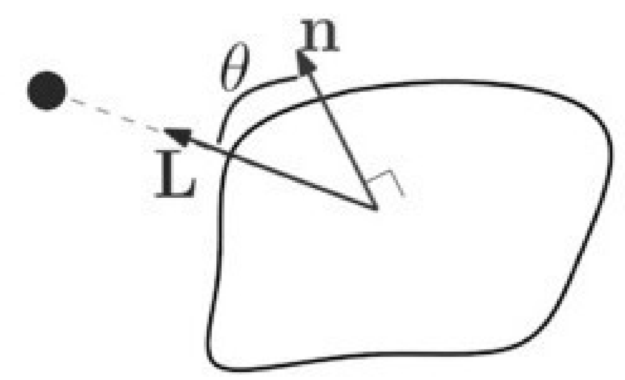


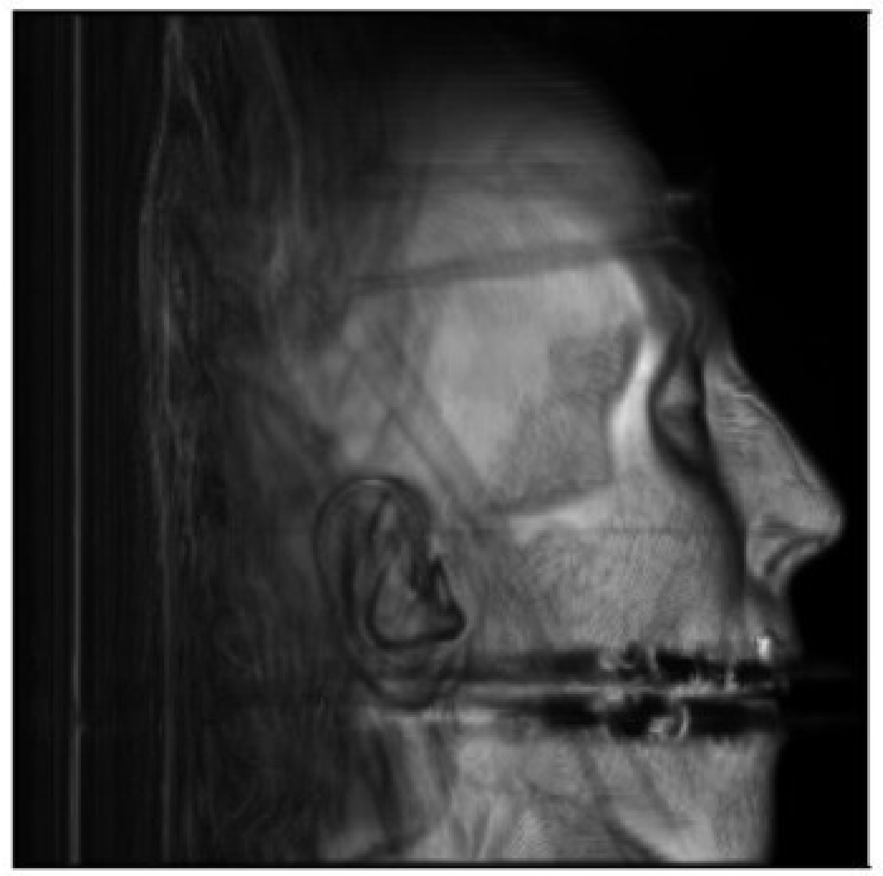
Shading

Diffuse shading $\cos(\theta) = < \mathbf{L}, \mathbf{n} >$

In a given voxel, approximates the surface normal

$$\mathbf{n} = \frac{\nabla I}{\|\nabla I\|}$$

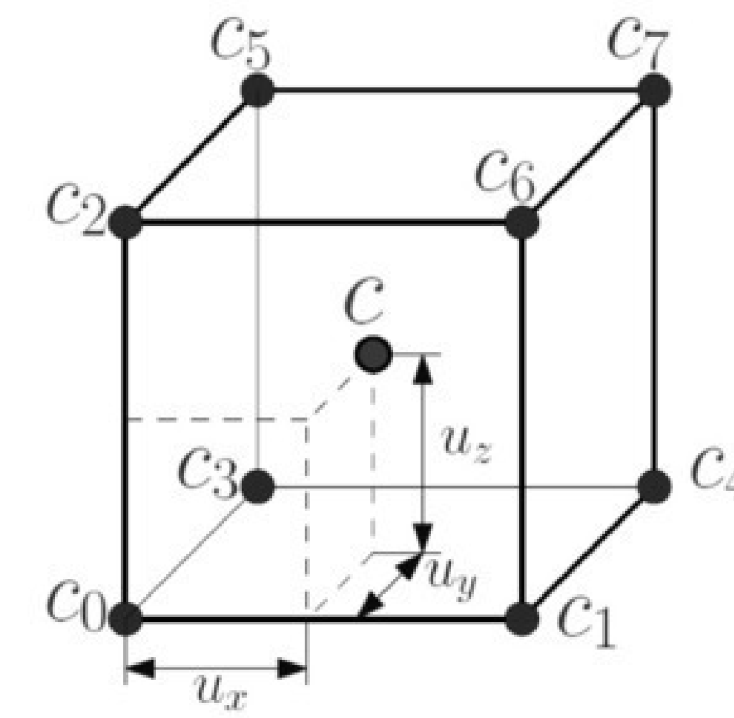




$$\nabla I = \begin{pmatrix} I(k_x+1, ky, kz) - I(k_x-1, ky, kz) \\ I(k_x, ky+1, kz) - I(k_x, ky-1, kz) \\ I(k_x, ky, kz+1) - I(k_x, ky, kz-1) \end{pmatrix}$$

Trilinear interpolation

$$\begin{aligned} c = \; & (1-u_x)(1-u_y)(1-u_z) \quad c0 + \\ & u_x(1-u_y)(1-u_z) \quad c1 + \\ & (1-u_x)(1-u_y)u_z \quad c2 + \\ & (1-u_x)u_y(1-u_z) \quad c3 + \\ & u_x u_y(1-u_z) \quad c4 + \\ & (1-u_x)u_y u_z \quad c5 + \\ & u_x(1-u_y)u_z \quad c6 + \\ & u_x u_y u_z \quad c7 \end{aligned}$$



Libraries

VTK: The Visualization ToolKit
Heavy and complete set of tools.
http://www.vtk.org

Volume rendering library (Stanford).
Standard, old.
http ://www-graphics.stanford.edu/software/volpack/

ImageVis3D (Utah).
http ://www.sci.utah.edu/cibc/software/41-imagevis3d.html

V3.
Fast on the GPU
http ://www.stereofx.org/volume.html