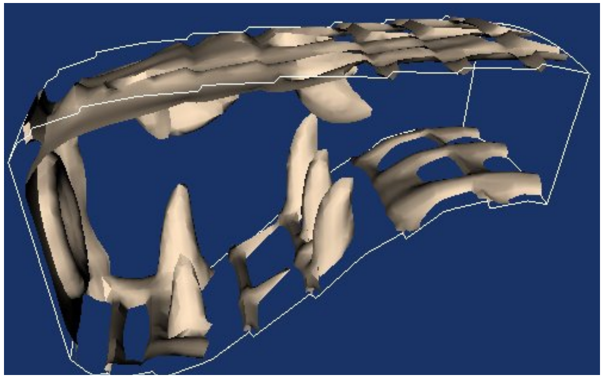
Contouring & Implicit Modelling

Visualisation – Lecture 9

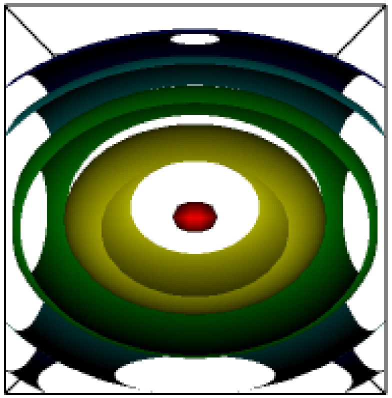
Taku Komura

Institute for Perception, Action & Behaviour
School of Informatics

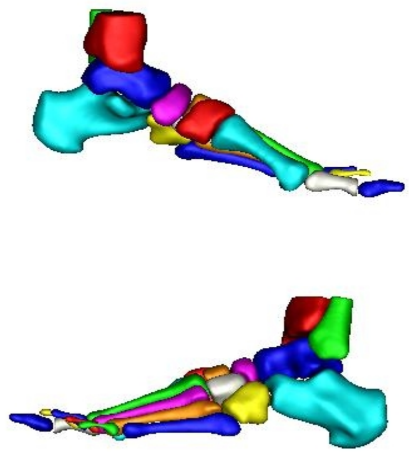
Brief Recap

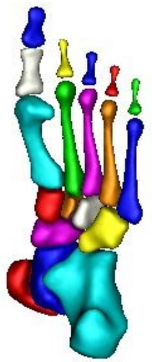


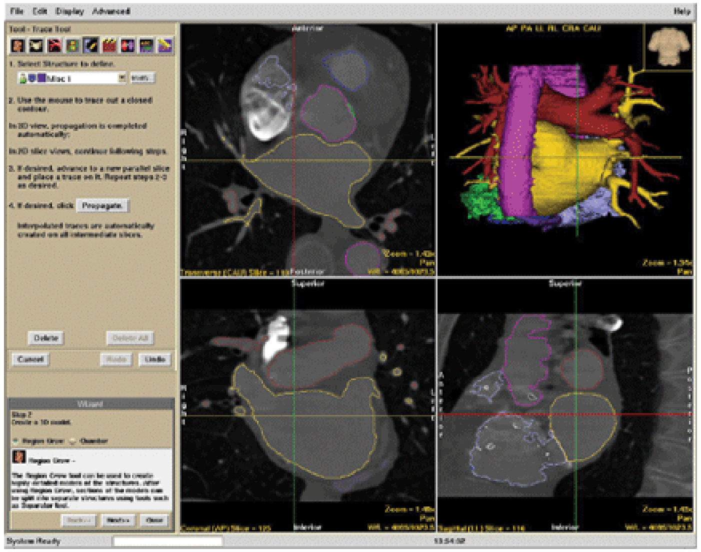
- **Contouring**
  - lecture 6 - Marching Cubes
- **Implicit Functions**
  - lecture 3 – visualisation of a Quadric
- **Today**
  - **Medical Segmentation**

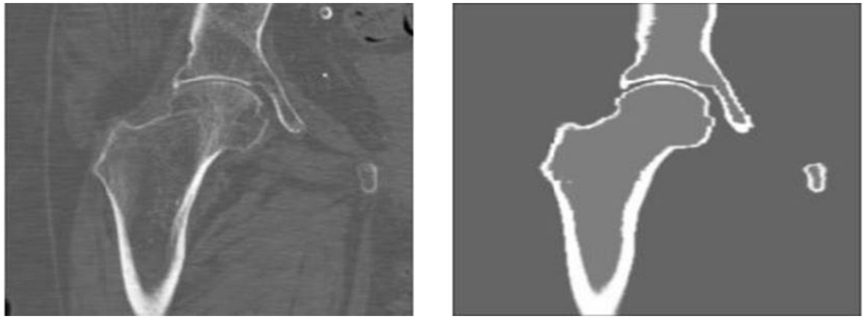


Medical Segmentation





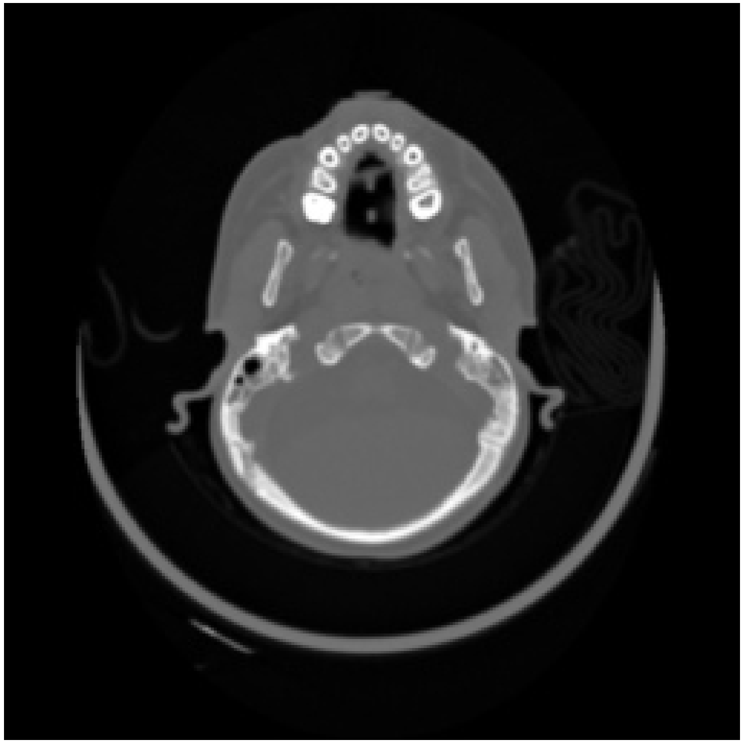




- Segmenting the data into different tissues
- The data can be either volume / image

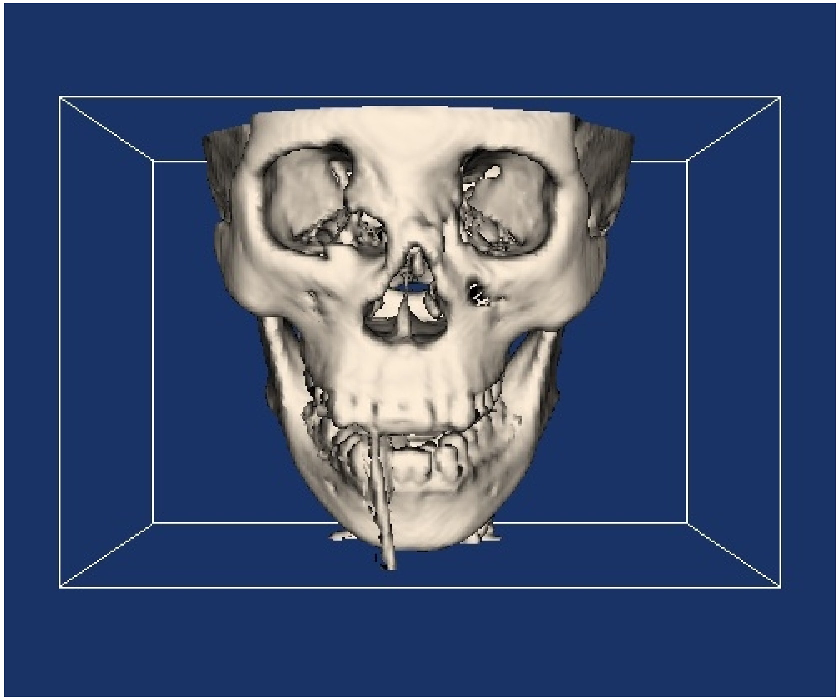
Segmentation of CT imaging

2D slice as image



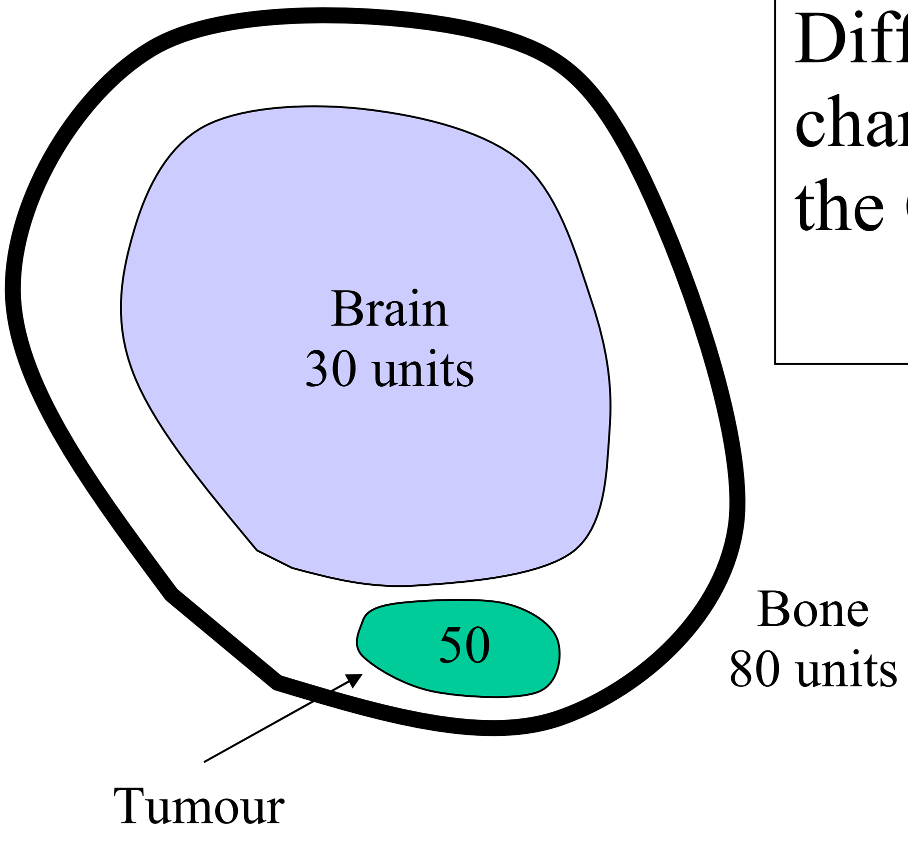
Greyscale image colourmap

3D contour of bone density



Marching Cubes

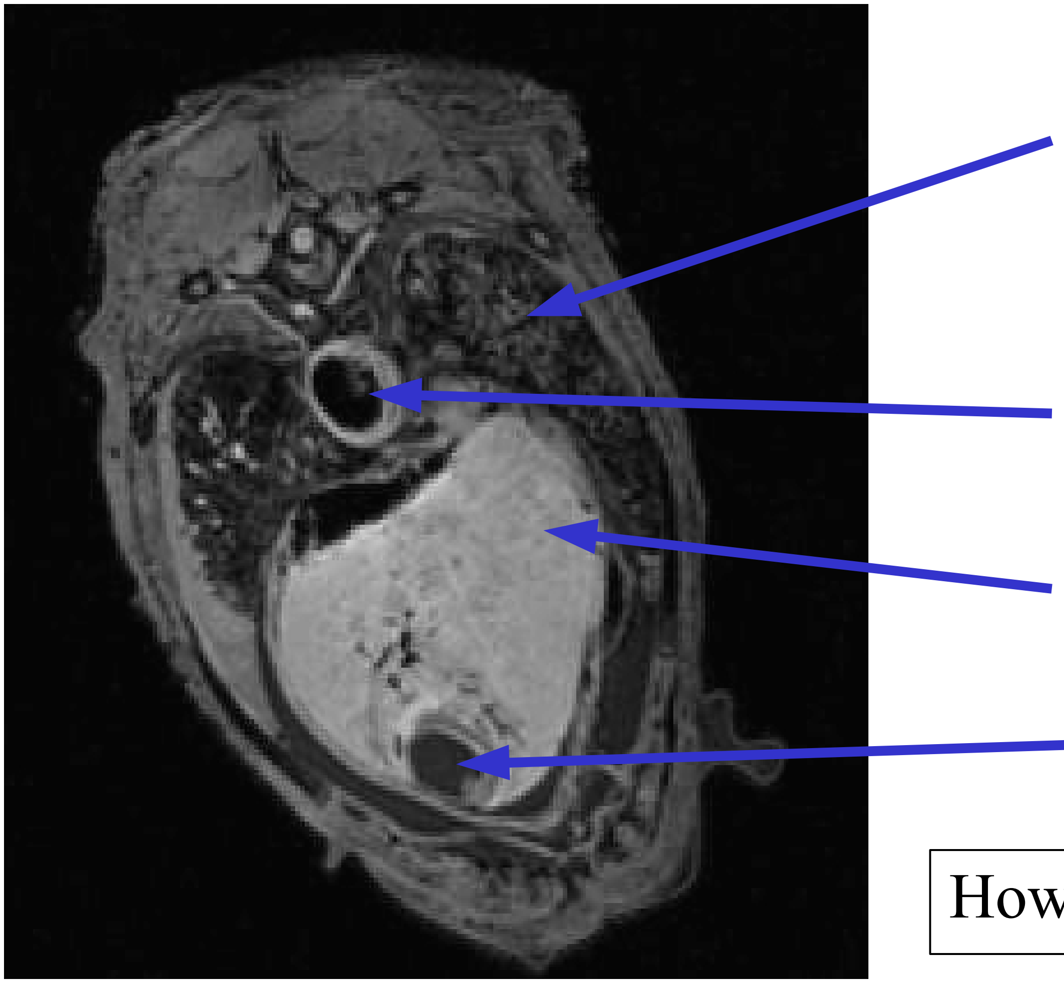
- CT value corresponds to density → easy to segment
- Different tissues have different CT values



Different tissues can be visualized by changing the transparency according to the CT values

How to segment MRI data?

A Dog's thorax

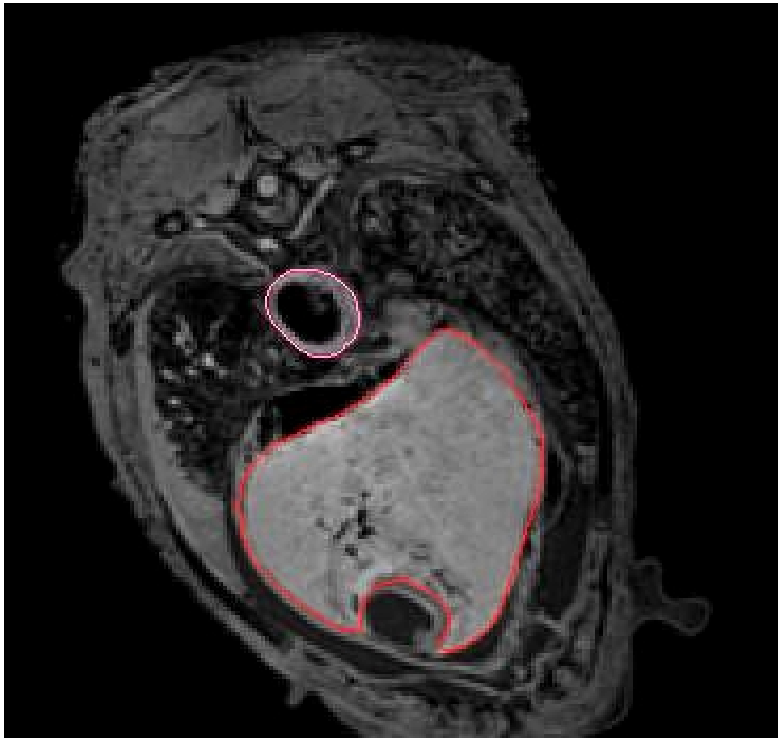


How to extract an organ here?

- MRI values not *directly* proportional to density (instead NMR signal intensity)

Image Processing on MRI Data

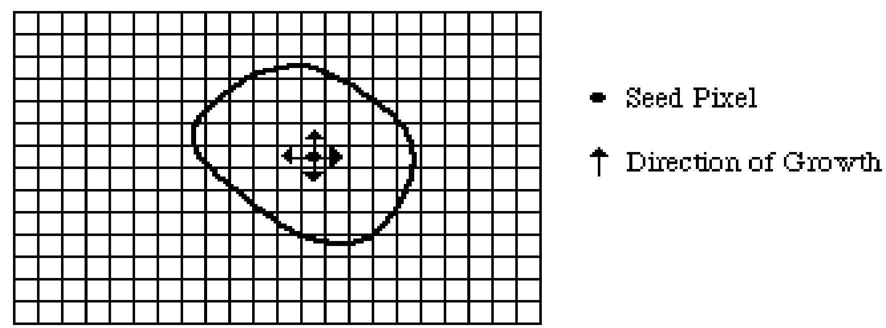
- From image processing / computer vision
  - segment image slices
    - **noise removal** (smoothing)
    - **edge detection** (changes in colour)
    - **Region growing**
    - **thresholding** – all scalar values in between upper and lower limits


Liver outlined in red

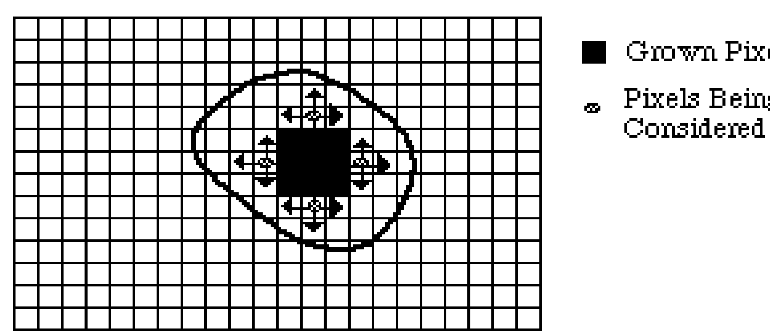
- Image processing
  - not strictly visualisation
  - additional tool in visualisation pipeline
  - see Advanced Vision course

Region Growing

- start from initial image patch
- grow outwards including similar regions
- stop when boundary or distinct change in value reaching

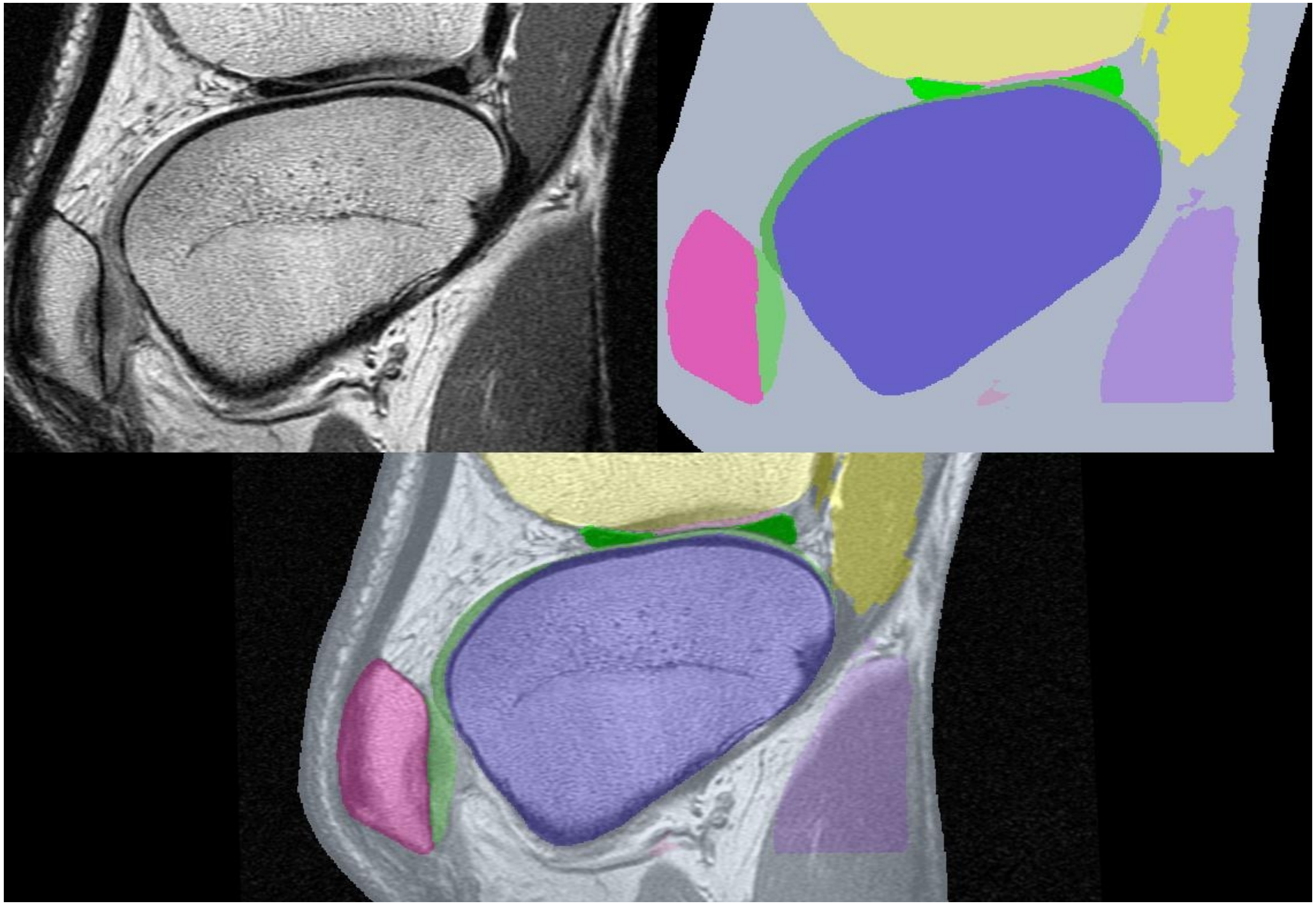


(a) Start of Growing a Region



(b) Growing Process After a Few Iterations

Example : MRI segmented knee

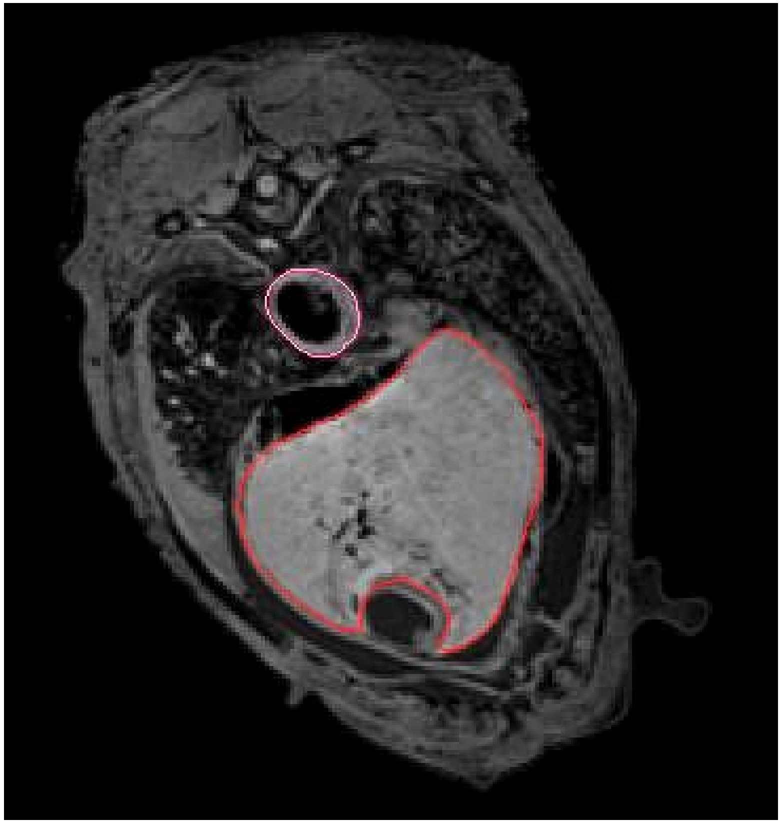


Courtesy : Brigham & Women's Hospital

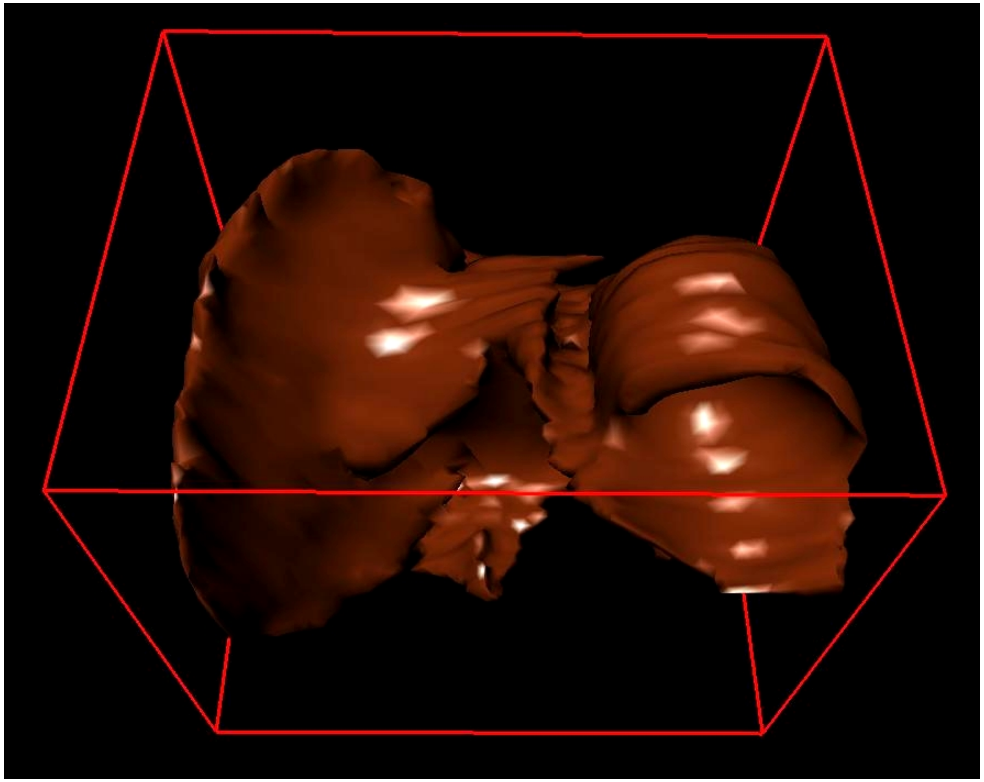
- Original MRI slice (top, left), Segmentation (top, right), Overlay (bottom).

"Segmentation" of dog organs

Liver outlined in red



3D model of Liver.

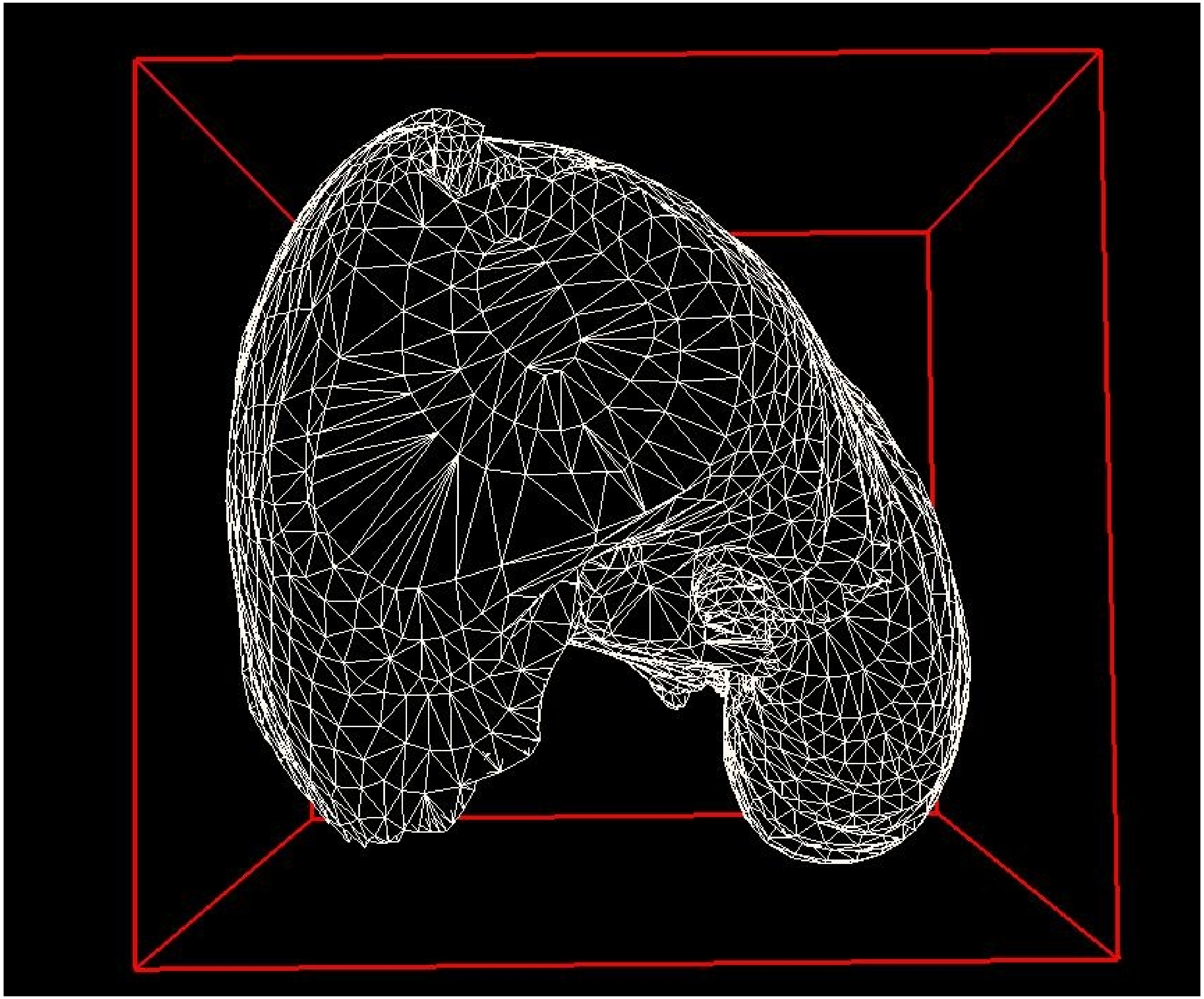


- **Segmentation** of organ in each image slice (2D)
  - visualisation of segmented 2D image stack as 3D volume

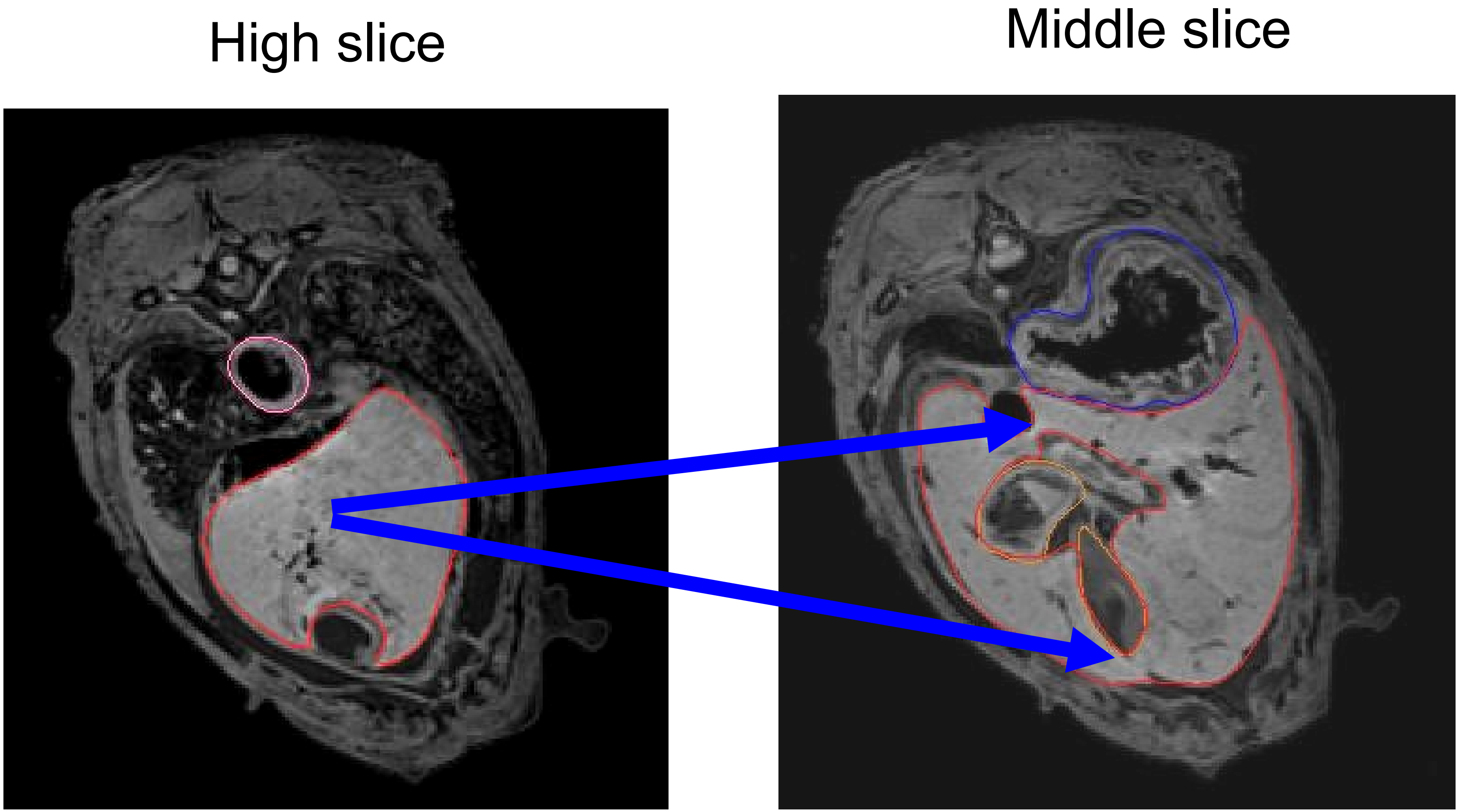
Triangulated 3D model of dog liver

Vertices in one contour need to be matched with those in the next slice to produce **triangulated mesh**.

Problem - different number of vertices in each curve

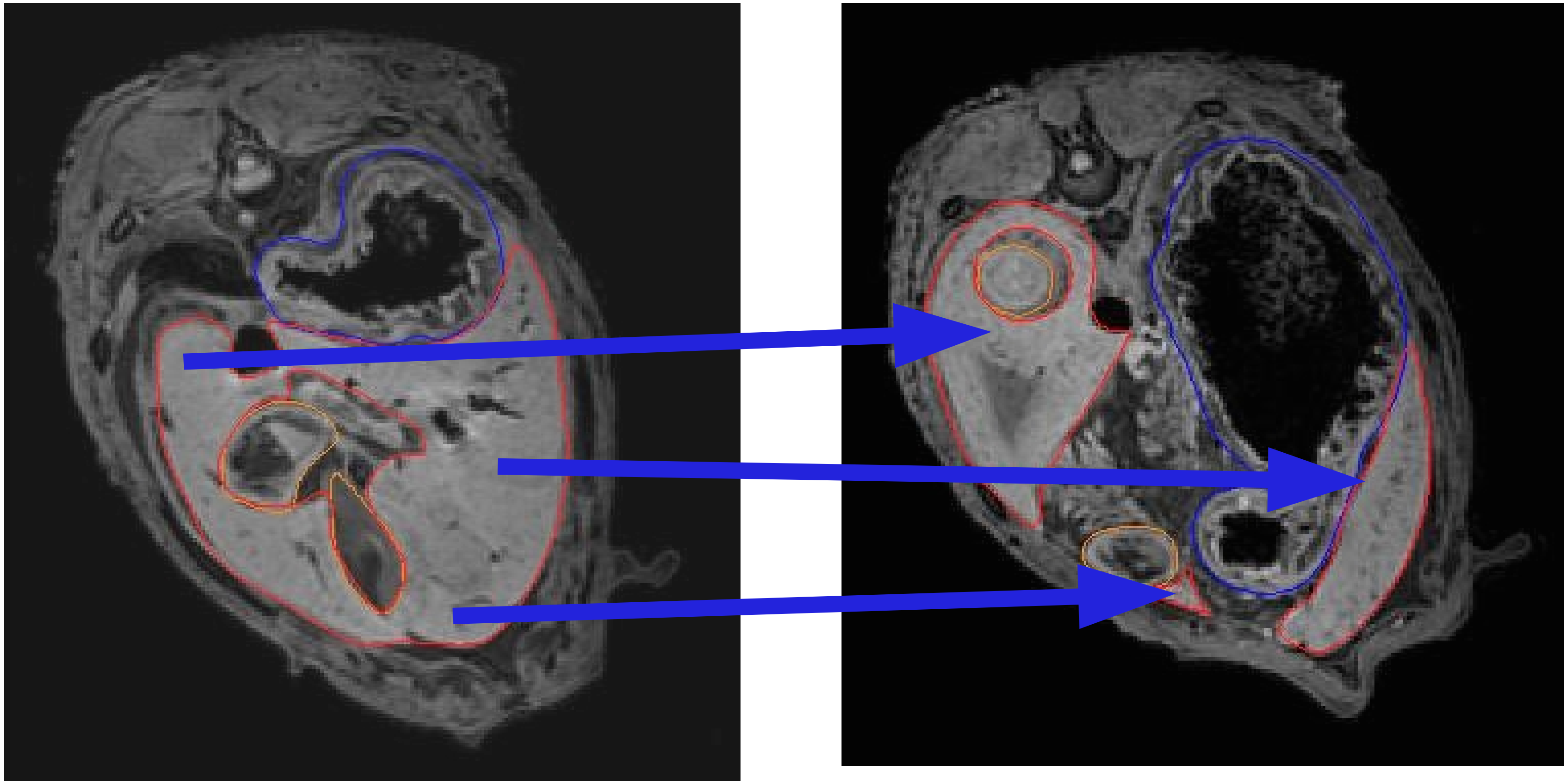


Problem 1 : Grow “hole”



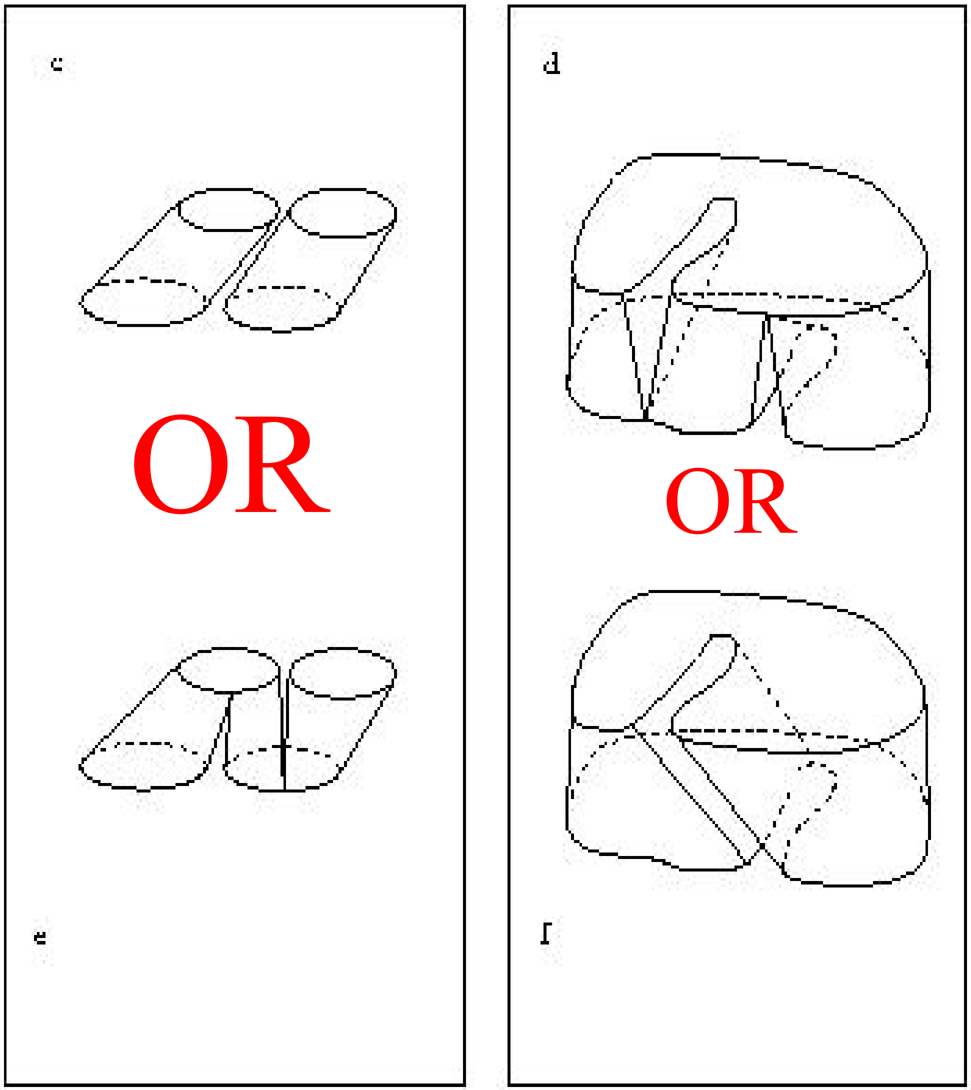
Problem 2 : split contours

Middle slice | Low slice



- Known as the **"branching problem"**.

Possible Ambiguities

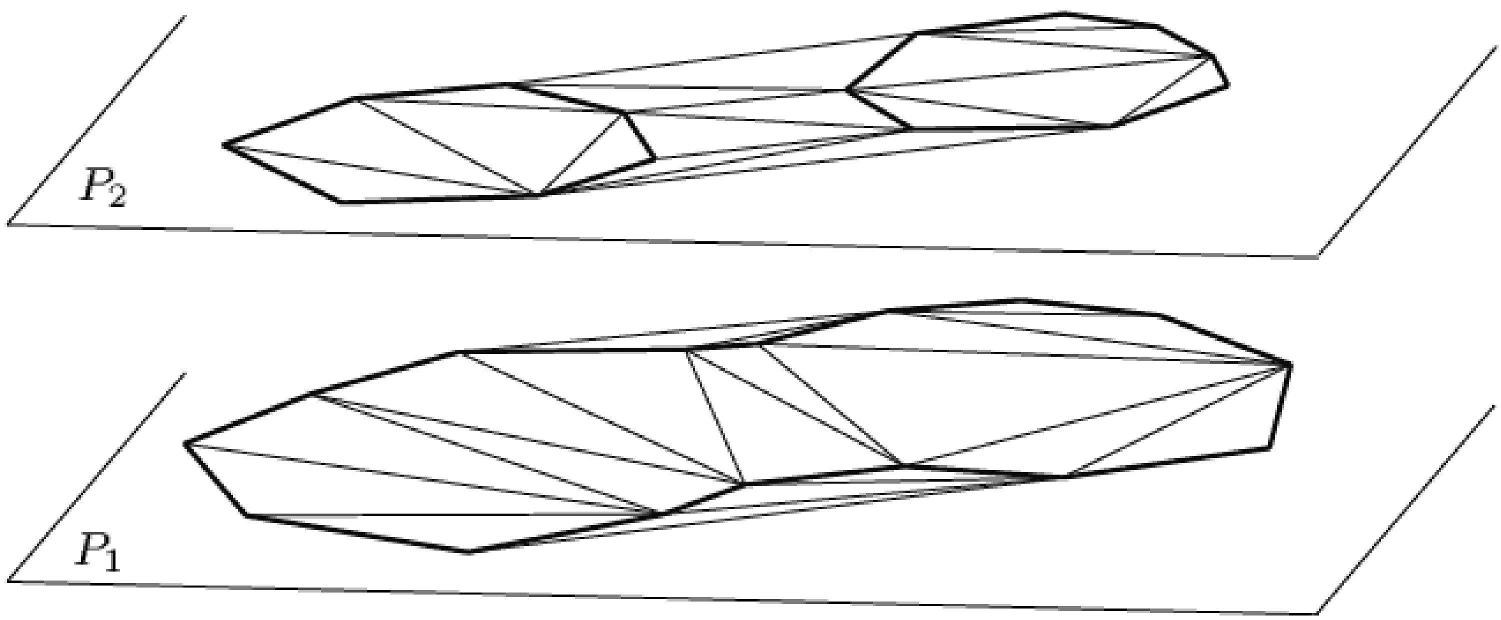


[Geiger '93]

- Contour connection method needs to handle these cases
- **Solution:** NUAGES [Geiger '93]
  - fill convex hull of contour with tetrahedra
  - discard those outside the contours or with only one edge on each level
  - 3D contour connection = remaining exterior tetrahedra faces

NUAGES : Triangulate Contour

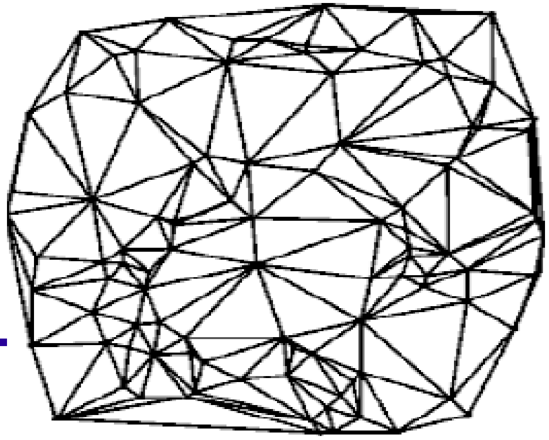
- **Triangulate** contour vertices using *Delaunay Triangulation*
  - no inside / outside constraint
  - forms planar convex hull of each contour



[Geiger '93]

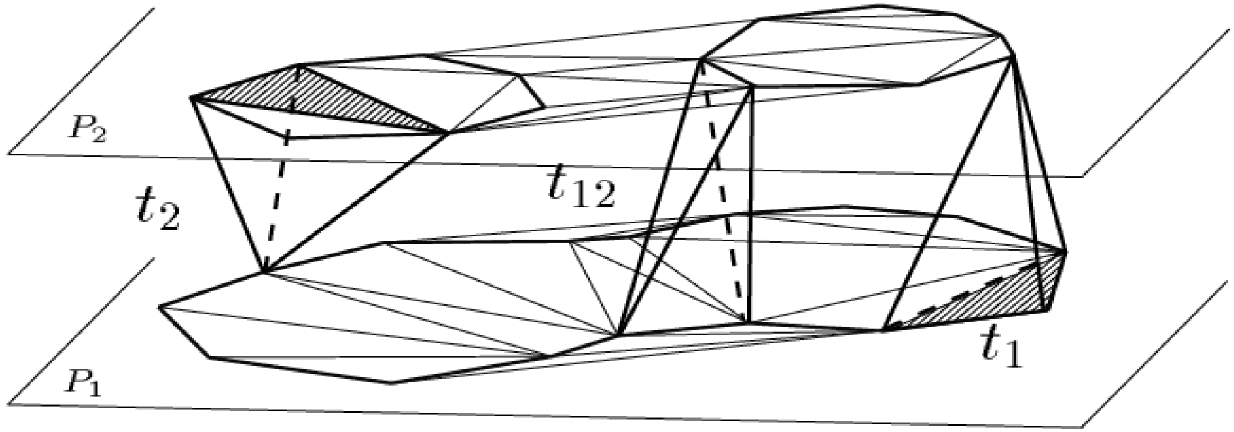
Delaunay triangulation

- Input : vertices
- Output : triangles composed by the vertices
- The triangulation DT(P) of P such that no point in P is inside the circumcircle of any triangle in DT(P)
- Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation
- Gives nice set of triangles for finite element analysis
- Used very often



NUAGES : Form Tetrahedra

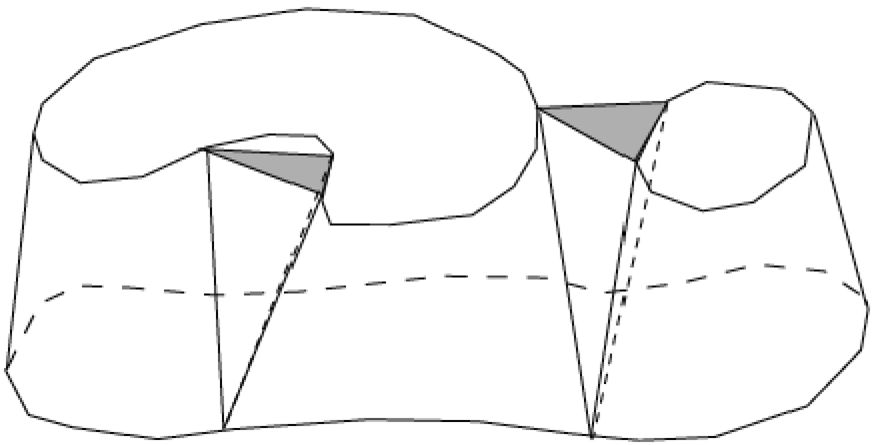
- **Form tetrahedra** by joining closest vertex on opposite contour

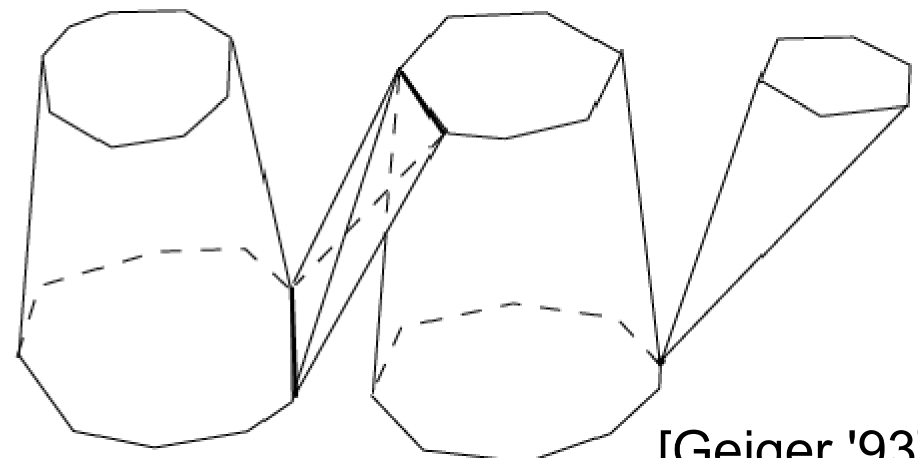


[Geiger '93]

  - tetrahedra t1 and t2 formed with closest opposite vertex enclosed within area of triangle
  - tetrahedra t12 formed by two edges that cross
  - Vertices are added if necessary

NUAGES : Remove Tetrahedra

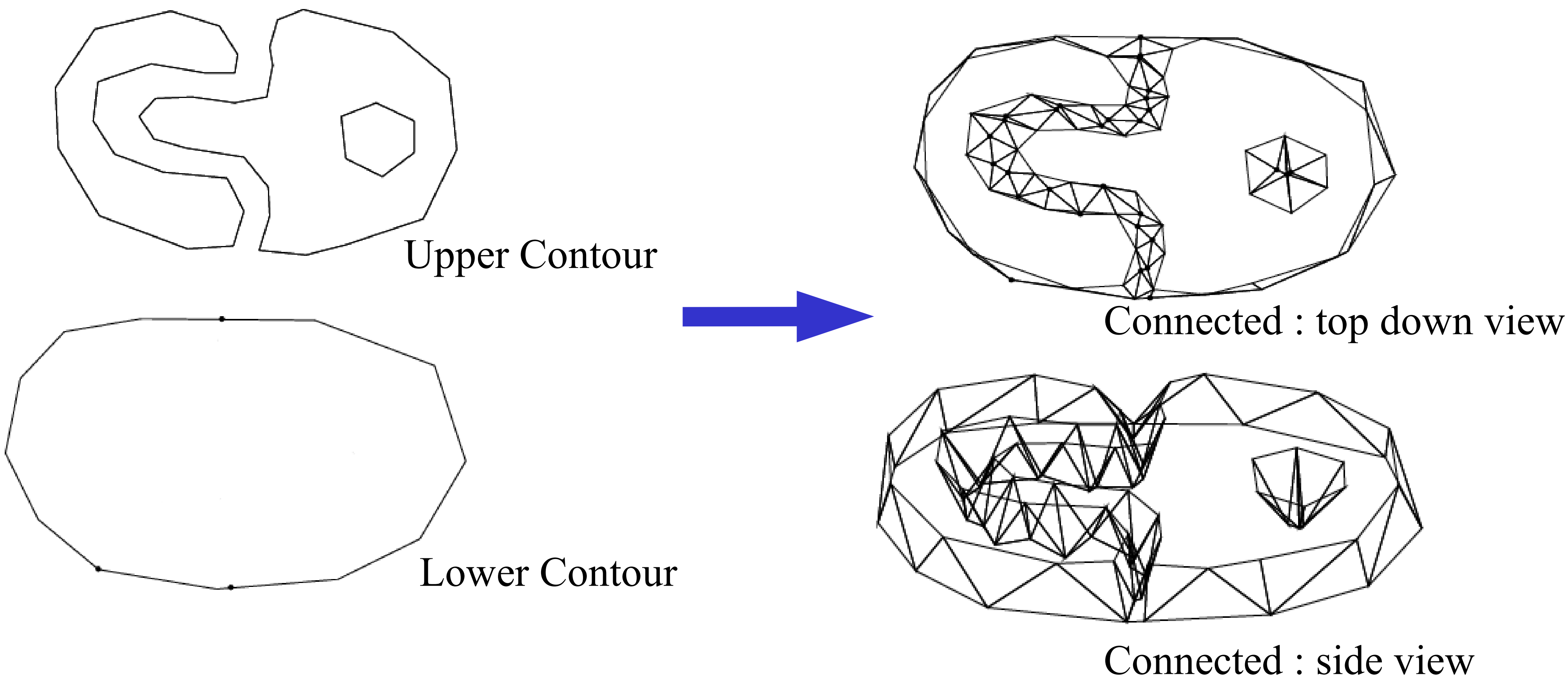




[Geiger '93]

- **Remove** : Tetrahedra with an edge outside of the contour.
- **Remove** : Tetrahedra with only one edge on each contour plane – non-solid connection

NUAGES : Results

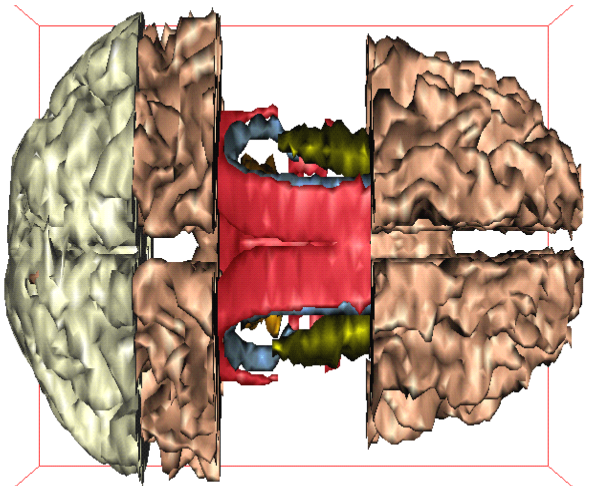


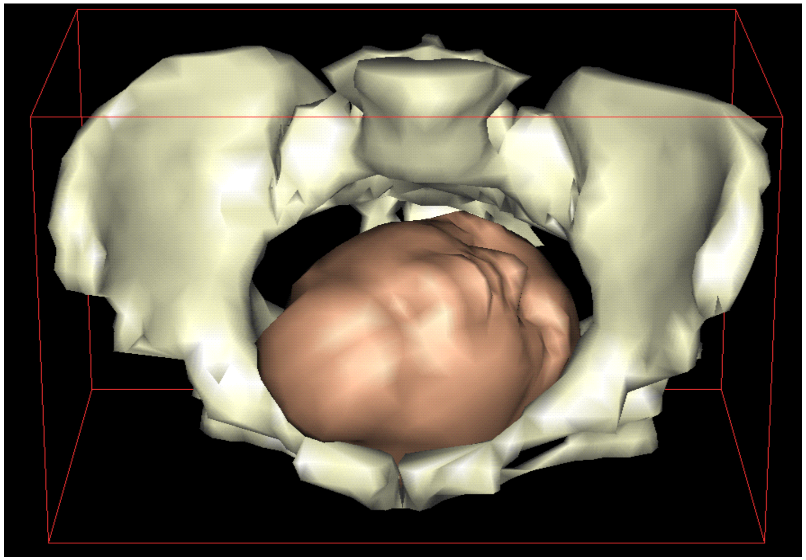
- Use **exterior triangles** of remaining tetrahedra to **form outside boundary of the shape**

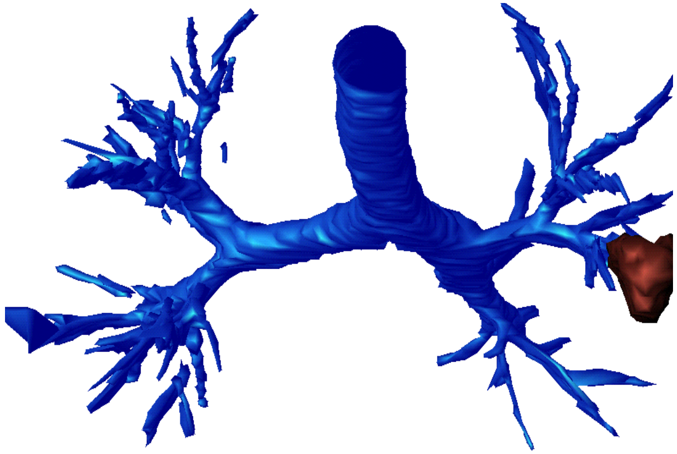
N.B. hole appears as form of a point

[Geiger '93]

NUAGES : Results

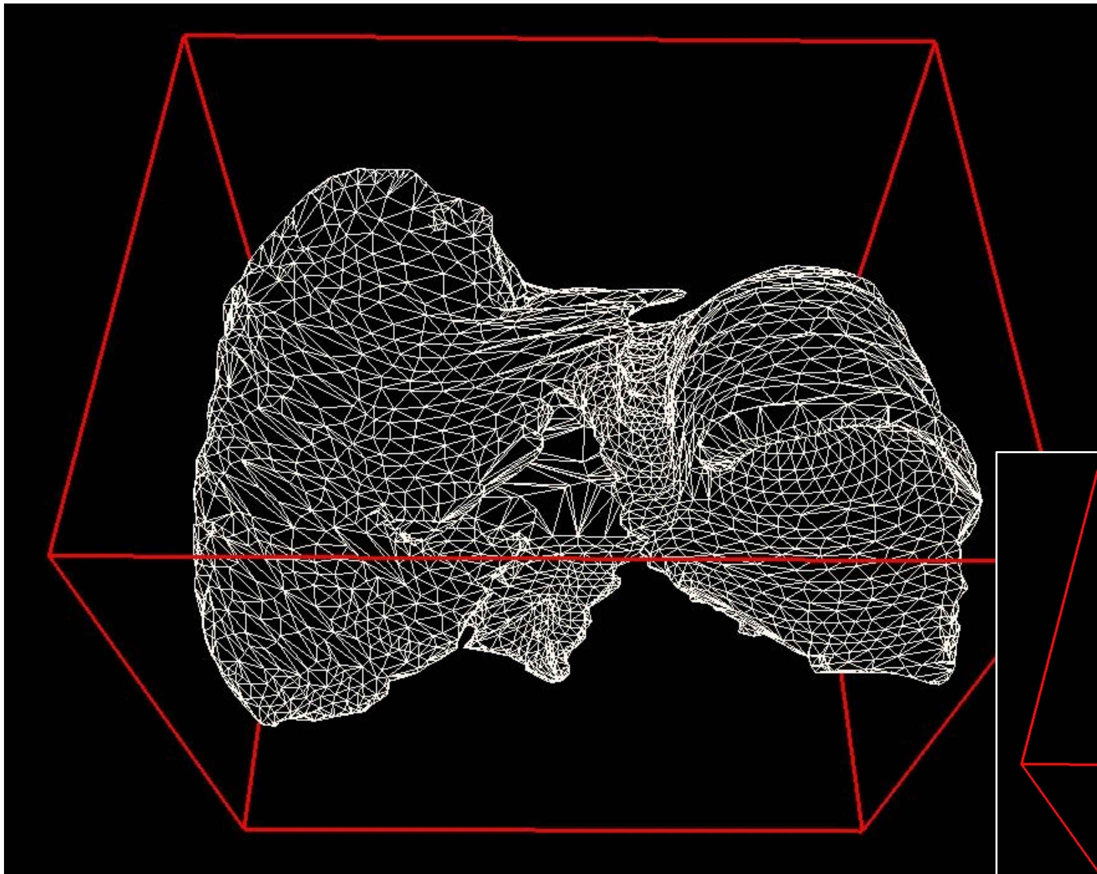


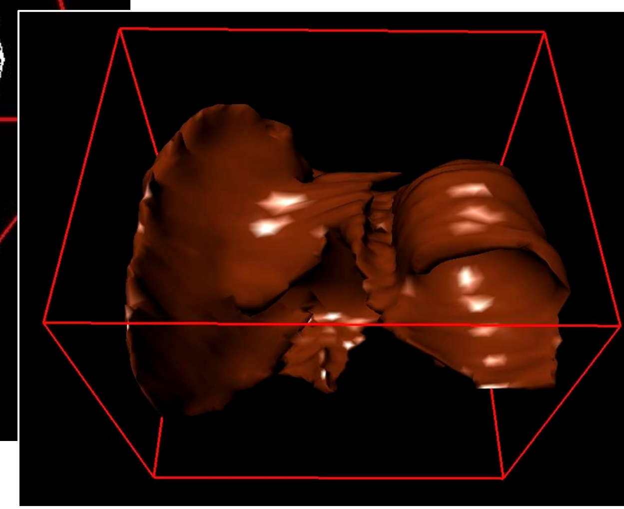




- Multiple medical image slices joined by **NUAGES**
  - http://www-sop.inria.fr/prisme/fiches/Medical/

Dog Liver : Final Model





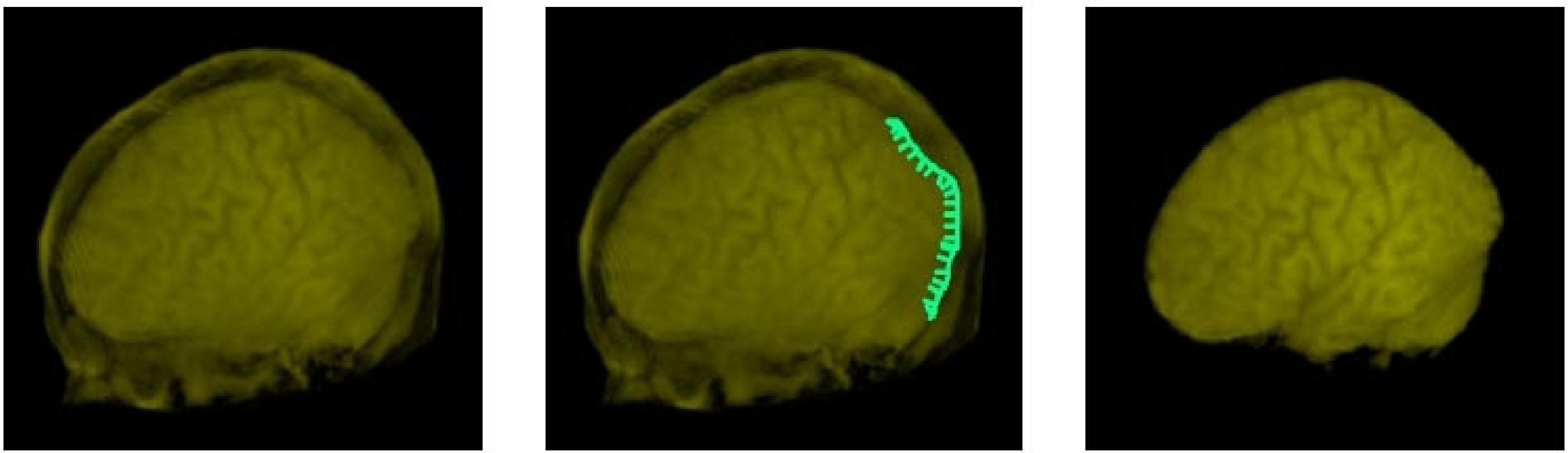
- Completed model – despite presence of contour holes/ branches (NUAGES)

Fully automatic segmentation is difficult

- although you apply all great computer vision techniques
- Segmentation is a low level procedure
  - Based on color information
  - But users have high level knowledge

Volume Catcher

- We need to give some knowledge to the system
  - A good user interface can improve the efficiency of segmentation
  - Sketch-based interface , Owada et al. I3D 2005



(a) (b) (c)

Figure 1: (a) 105 × 73 × 73 head MRI data. (b) Drawing a 2D stroke along the contour of the brain. (c) Resulting 3D region. The system automatically computes the depth of the stroke and applies constrained segmentation.

Specifying the Region of Interest

- The user specifies which area he/she is interested in
- The system finds out the color information inside and outside the volume data
- The color information is used for region growing/

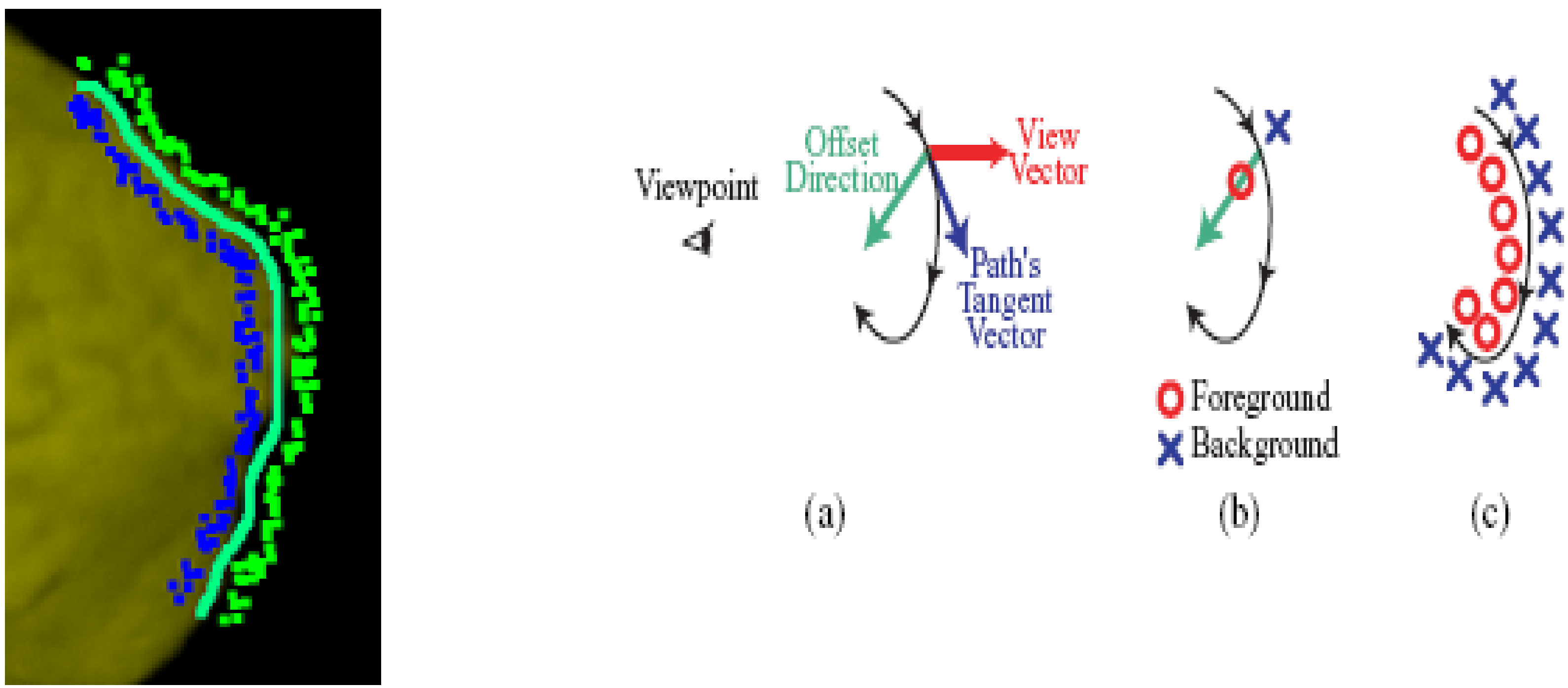
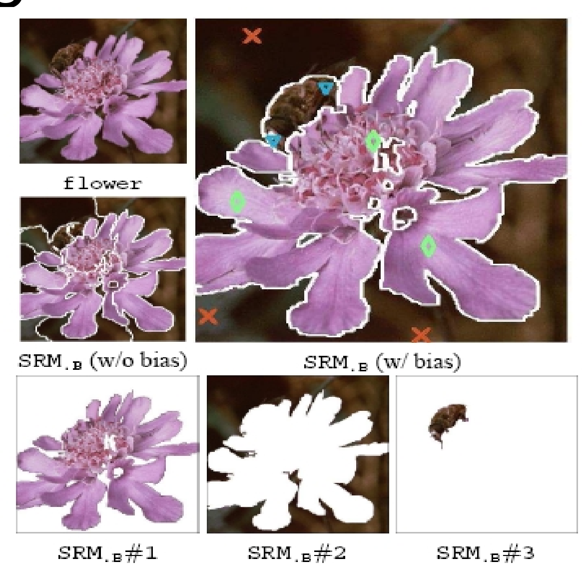


Figure 5: The constraints generated for Figure 1

Low level segmentation by color

- Statistical region growing algorithm (Nock and Nielsen, CVPR 2004)
  - segments the image into regions based on the color information
  - And the regions are merged based on the user input
- The user specifies which regions belong to the same object
- This region growing is used for the Volume Catcher

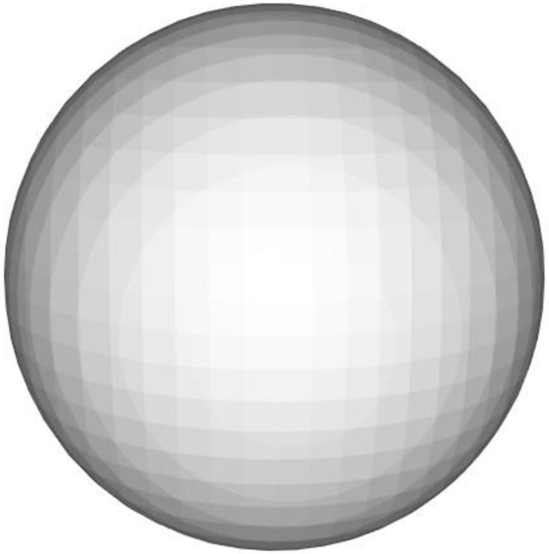


Summary - Contours

- Some 3D dataset easy to visualise with iso-surfaces
  - **CT** data values correspond to density
  - iso-surfaces from **Marching Cubes**
- Some datasets have no such correspondence
  - **MRI** data values
  - **segmentation** of organs requires **image processing**
  - image processing – performed on slices
  - **slices joined to create 3D volumetric model**
    - ambiguities need to be resolved (**NUAGES**)
  - Using biased statistical region growing to segment volume data

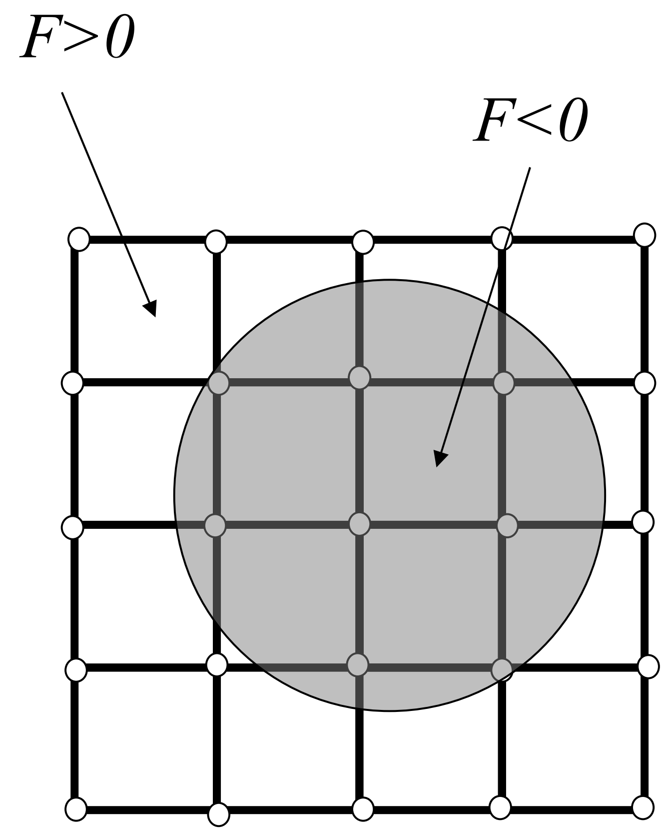
Modelling Algorithms – Implicit Functions

- **Functions of the form *F(x,y,z) = c***
  - *c* is an arbitrary constant
  - function **implies** a surface (at F=0)
  - e.g $F(x,y,z) = x^2 + y^2 + z^2 - R^2$
    defines a sphere radius R centred at the origin
- Defines **3 regions** :
  - $F = 0$ lies **on** the sphere
  - $F < 0$ lies **inside** the sphere
  - $F > 0$ lies **outside** the sphere
- **Uses** :
  - Geometric modelling
  - Selecting data (i.e. *cutting*)



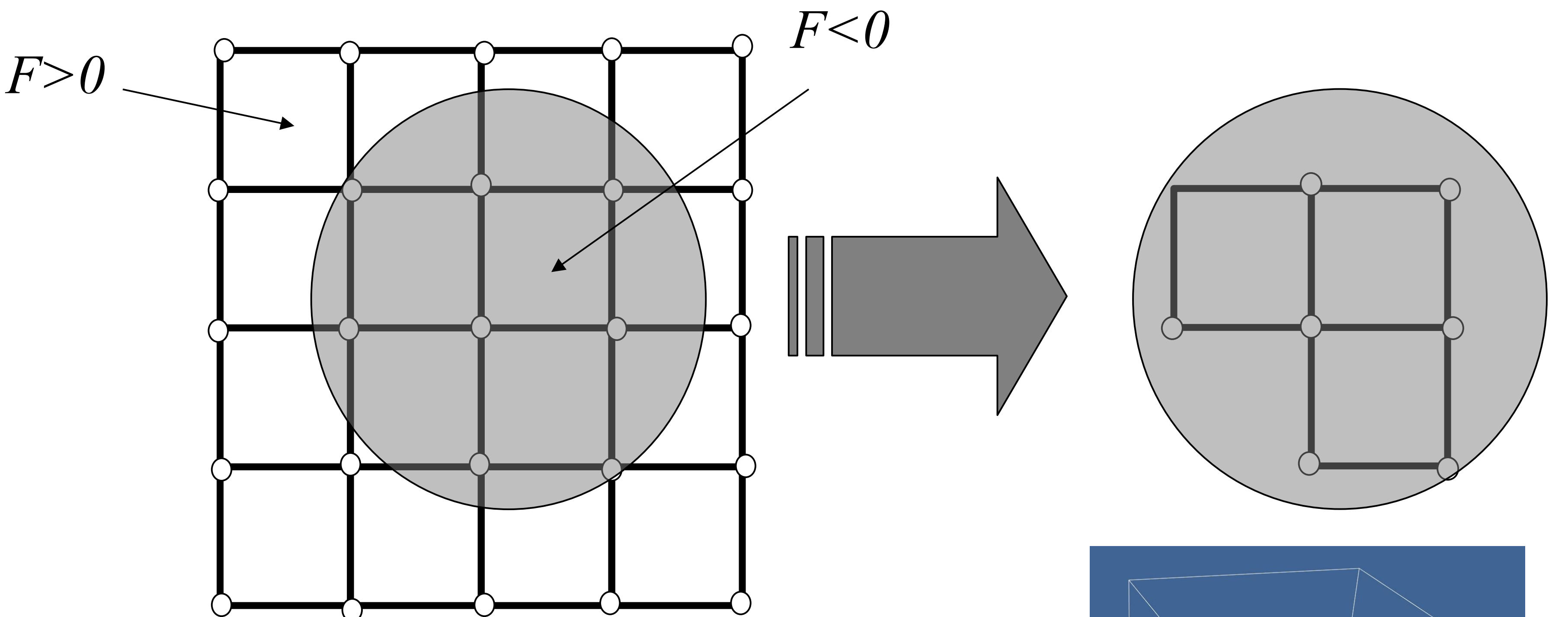
Implicit Functions for Cutting 1

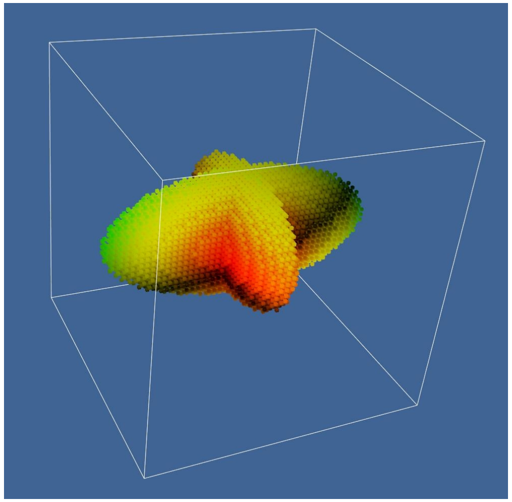
- Define the **implicit function F**
- **Evaluate *F* at every point** on grid (in set)
- **If *F* < *0*** (clipping everything inside) **remove** data
  - similarly for outside
  - iso-surface of sphere at F=0



- **Cutting = Modelling Algorithm** : changes aspects of data

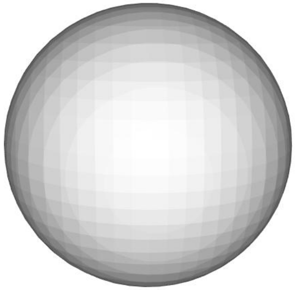
Implicit Functions for Cutting 2



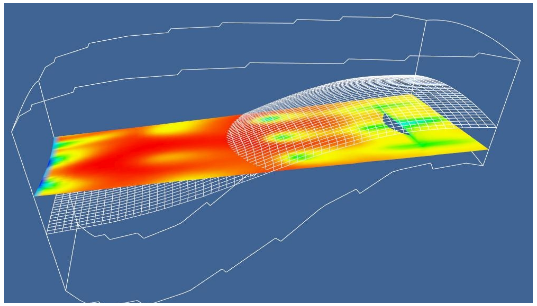


- A **topology change** occurs in the data
  - structured points to unstructured points and cells

Cutting



- **Implicit functions generate scalars**
  - 3D contouring algorithm → iso-surface
    - e.g. marching cubes to generate *F(x,y,z) = 0* and produce cut surface
- **Cutting a 3D dataset** with a plane → 2D slice
  - colour mapped or contoured
  - visualise data on (nearly) arbitrary surface



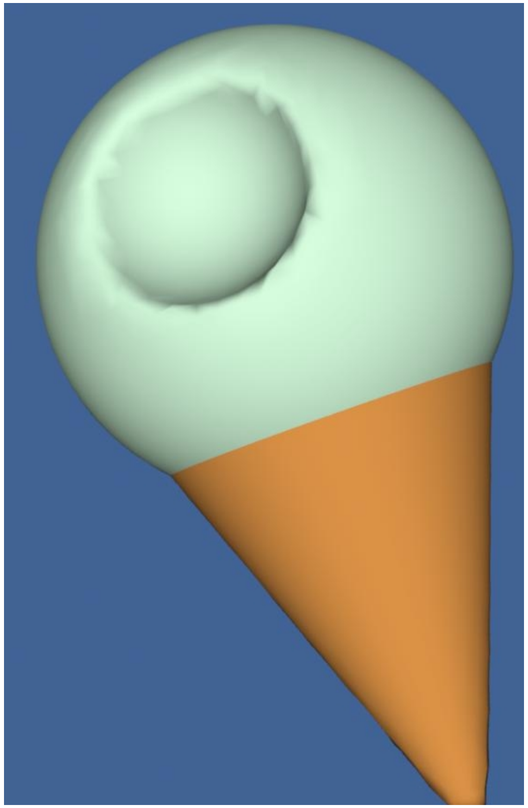
- **Cutting polygonal geometry** → lines
  - generate contour lines of 3D models
  - better visual aid than lighting & shading for 3D shape

Cutting in VTK

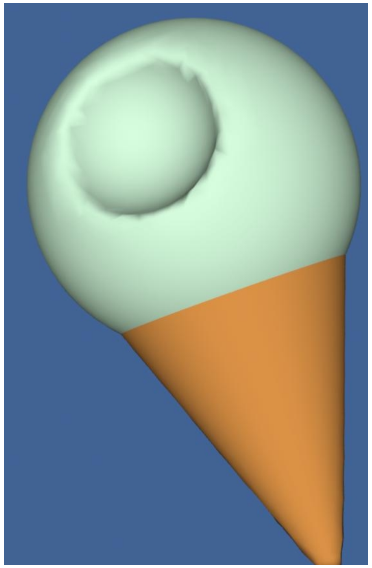
- `vktImplicitFunction`
  - sub-classes are `vtkSphere`, `vtkPlane`, `vtkQuadric`, `vtkCylinder`, `vtkCone`, ...
- `vtkCutter`
  - cuts cells with an implicit function
- `vtkClipPolyData`
  - cuts polygonal data with an implicit function

Implicit Object Modelling

- Combine multiple implicit functions
  - union, intersection, difference operations
  - e.g. $((F_1 \cap F_2) \cup F_3) - F_4$
  - e.g. two spheres and cone (right)
  - extract surface with contouring operation (in VTK pipeline)
- **Constructive Solid Geometry**



Implicit Object Modelling in VTK



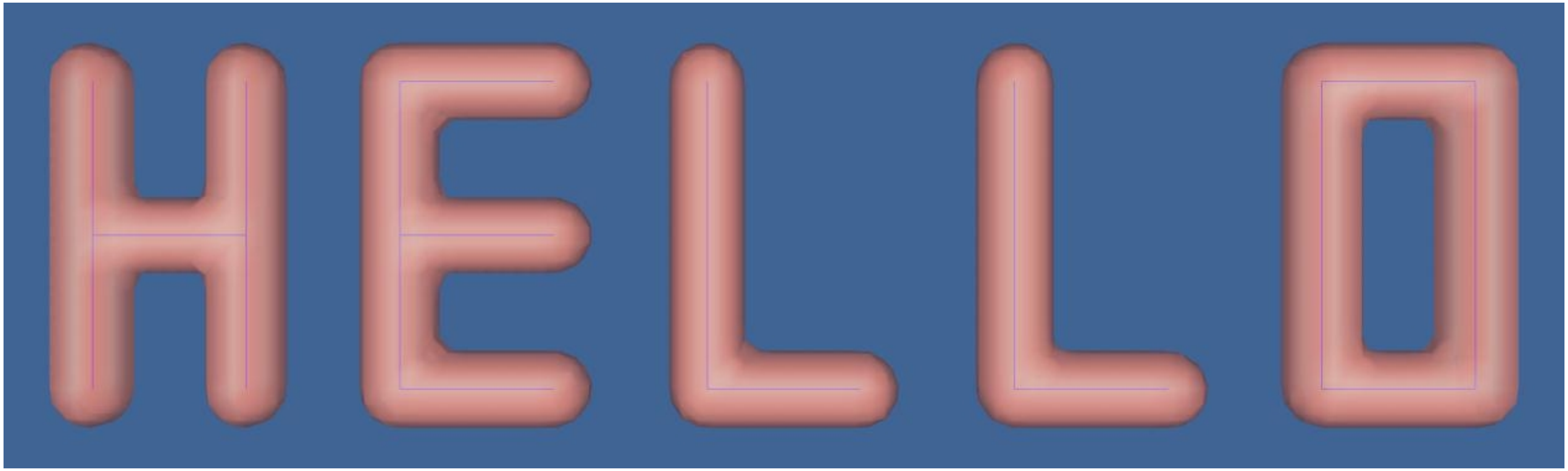
```
vtkCone      cone
vtkPlane     vertPlane
vtkPlane     basePlane
vtkSphere    iceCream
vtkSphere    bite
```

 # combine primitives to build ice-cream cone

```
vtkImplicitBoolean theCone
     theCone SetOperationTypeToIntersection
     theCone AddFunction cone
     theCone AddFunction vertPlane
     theCone AddFunction basePlane

vtkImplicitBoolean theCream
     theCream SetOperationTypeToDifference
     theCream AddFunction iceCream
     theCream AddFunction bite
```

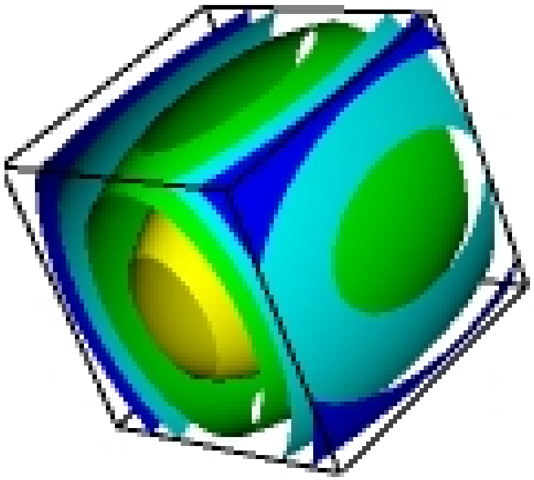
Implicit Modelling

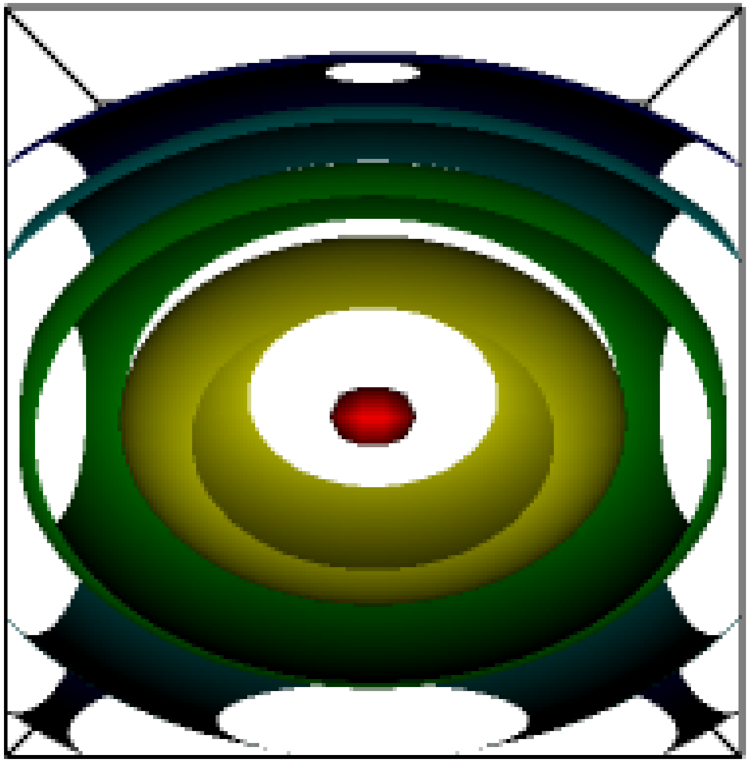


- Modelling using implicit function
  - defined as Euclidean distance from geometric primitives
    - e.g. lines spelling the word hello
      (geometry and iso-surface of function shown)
  - VTK : `vtkImplicitModeller` (`hello.tcl`)

Implicit Modelling - Functions

- All mathematical functions of form ***F(x,y,z) = c***
  - *e.g. quadric*





Summary – Implicit Functions

- Defined as $F(x,y,z) = c$
  - **Data Selection** (sub-selection)
    - reduce data → improve speed, limit visualisation
  - **Implicit Object Modelling**
    - constructive solid geometry
  - **Implicit Modelling**
    - surface implied by distance from geometric primitives
  - Visualisation of **Mathematical Functions**

***next lecture :** volume rendering*