Segmentation & Modeling

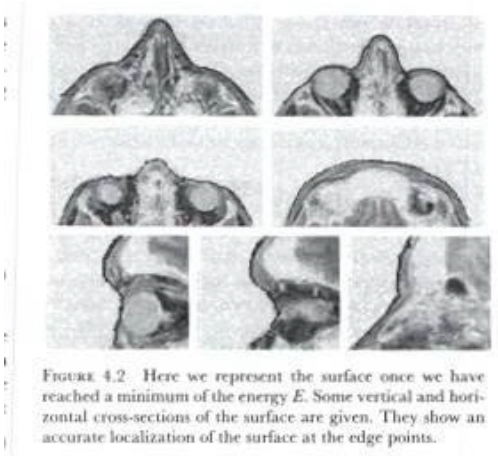


FIGURE 4.2 Here we represent the surface once we have reached a minimum of the energy E. Some vertical and horizontal cross-sections of the surface are given. They show an accurate localization of the surface at the edge points.

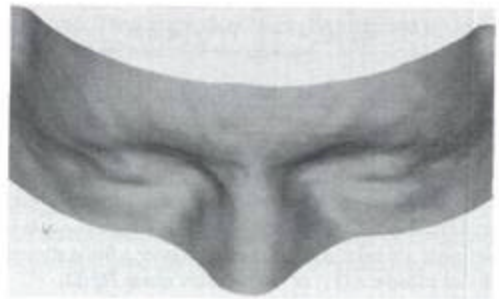


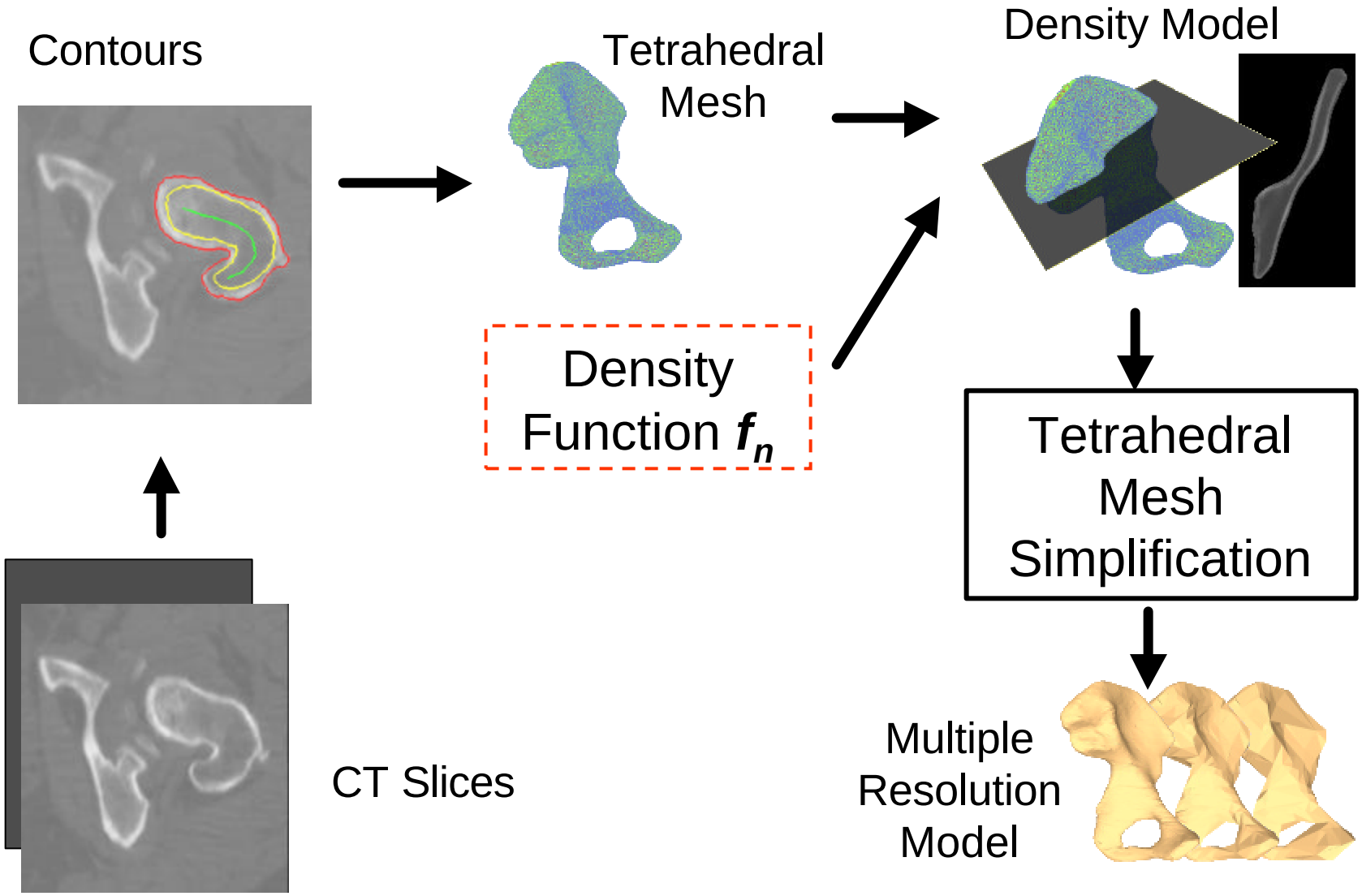
FIGURE 4.3 A 3D representation of the surface depicted in figure 4.2.

Images

Segmented Images

Models

Example: Bone Modeling from CT



Segmentation

- Process of identifying structure in 2D & 3D images
- Output may be
  - labeled pixels
  - edge map
  - set of contours

Approaches

- Pixel-based
  - Thresholding
  - Region growing
- Edge/Boundary based
  - Contours/boundary surface
  - Deformable warping
  - Deformable registration to atlases

Thresholding

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

Thresholding

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

Thresholding

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

Thresholding

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

Region Growing

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

Region Growing

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

Region Growing

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

Region Growing

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

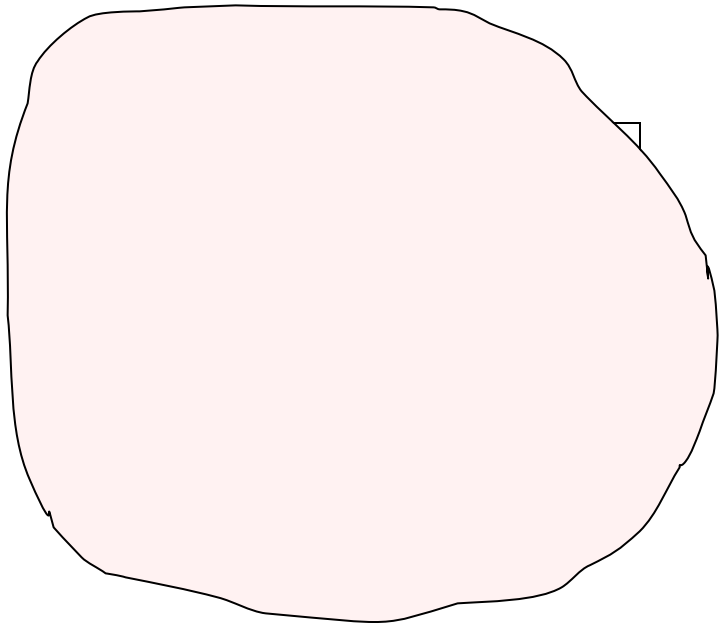
Region Growing

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

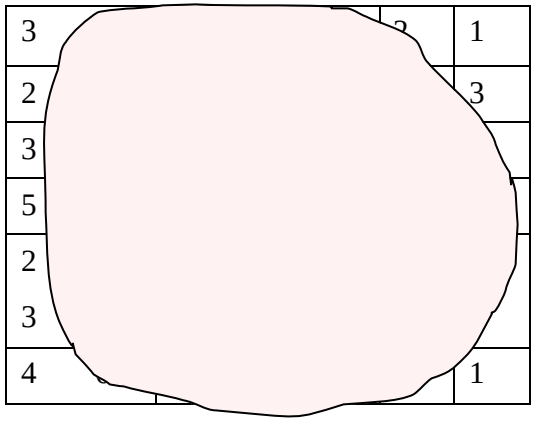
Deformable Surfaces

3	5	7	3	4	2	1
2	4	9	10	22	9	3
3	5	12	11	15	10	3
5	6	11	9	17	19	1
2	3	11	12	18	16	2
3	6	8	10	18	9	5
4	6	7	8	3	3	1

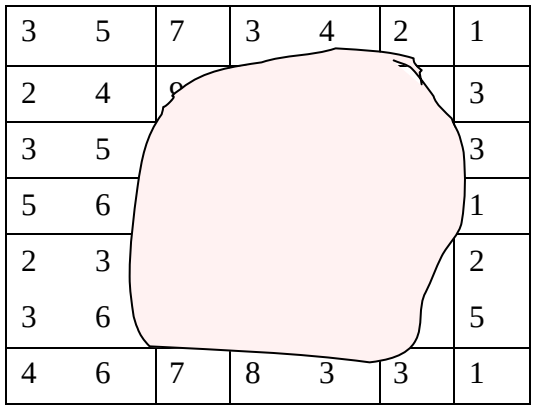
Deformable Surfaces



Deformable Surfaces



Deformable Surfaces



Deformable Surfaces

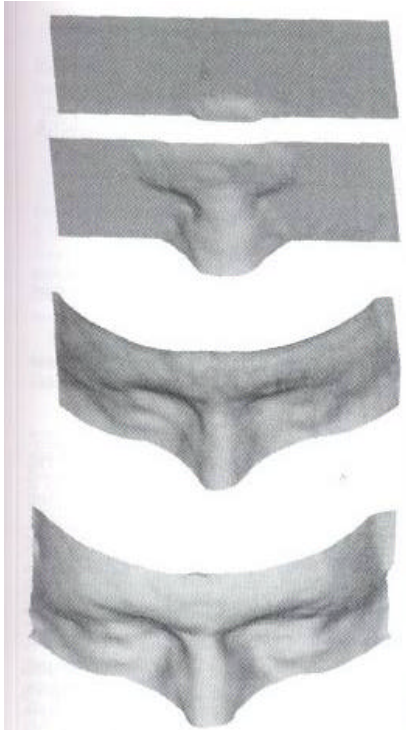


FIGURE 4.4 Evolution of the 3D surface "falling" on a 3D MRI image of a head. The initial surface is a plane on the border of the image.

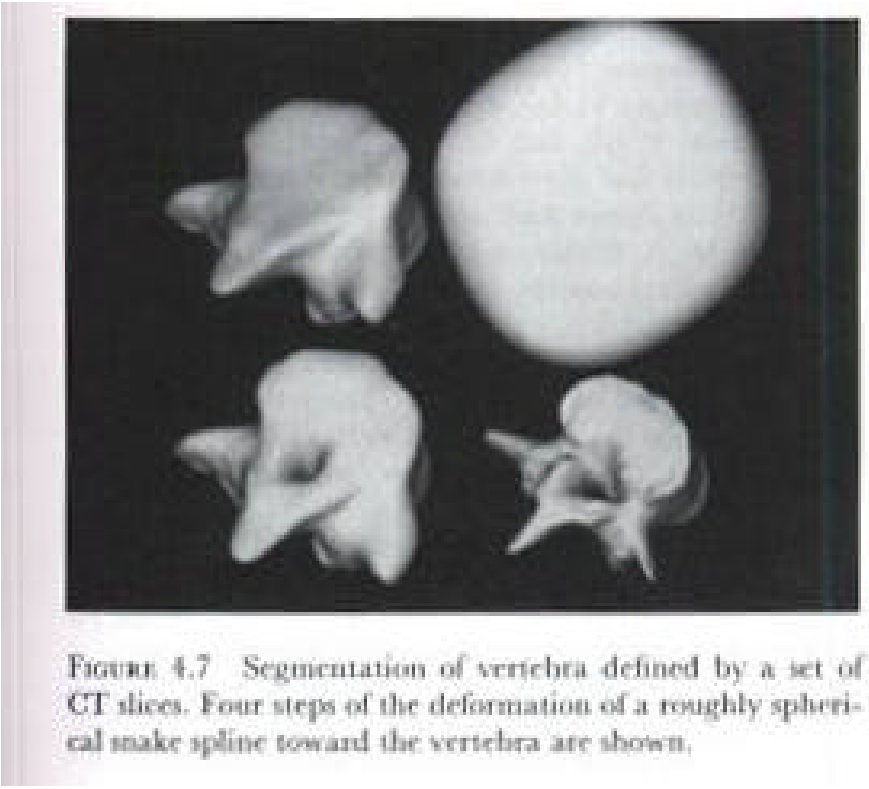
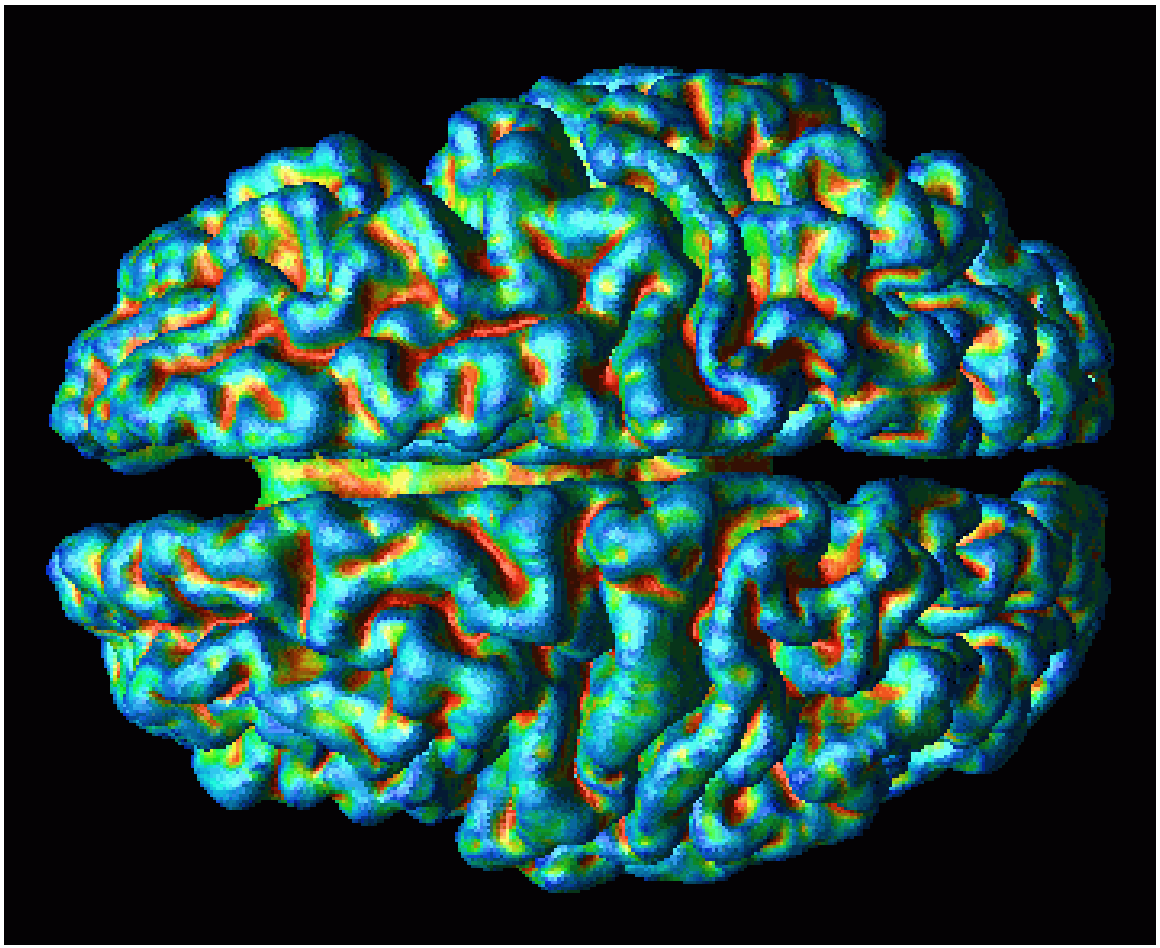


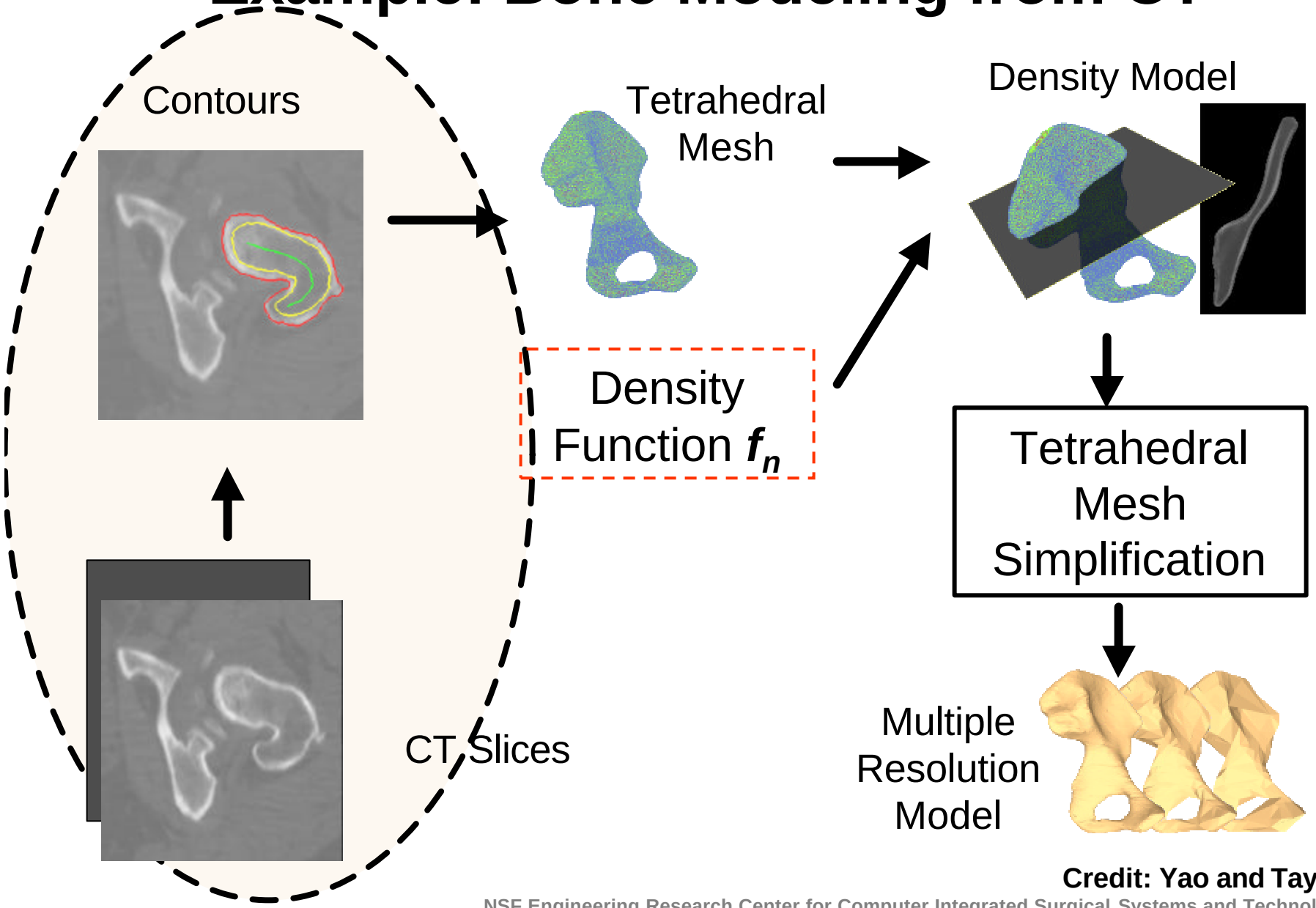
FIGURE 4.7 Segmentation of vertebra defined by a set of CT slices. Four steps of the deformation of a roughly spherical snake spline toward the vertebra are shown.

Deformable Surfaces



Credit: Prince & Davatzikos

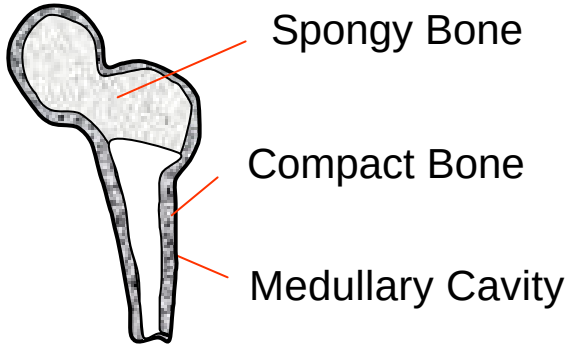
Example: Bone Modeling from CT



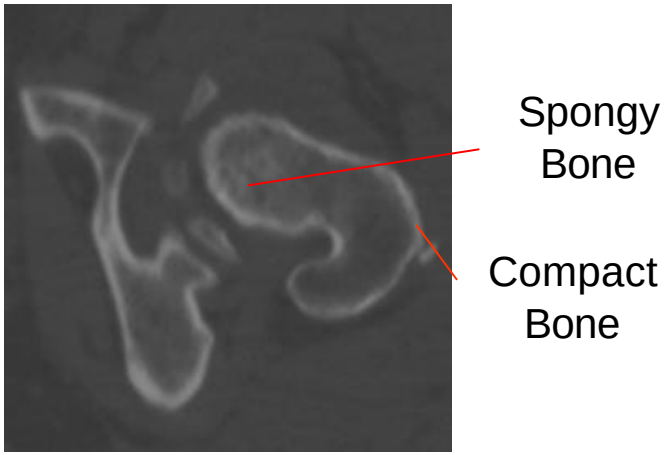
Credit: Yao and Taylor

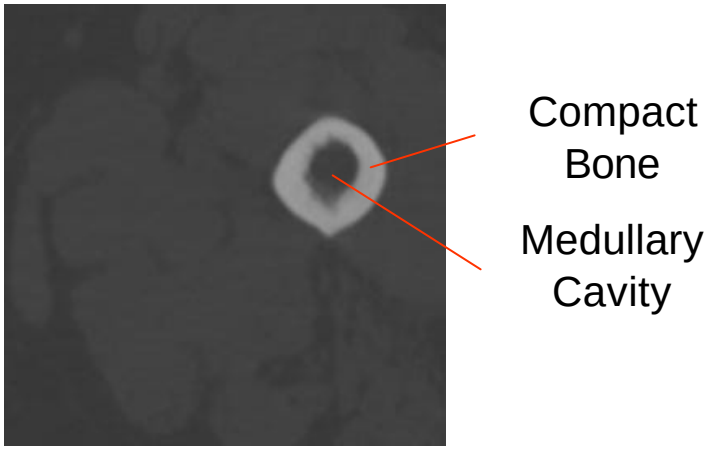
Bone Structure

- Compact bone
- Spongy bone
- Medullary Cavity



Bone Structure



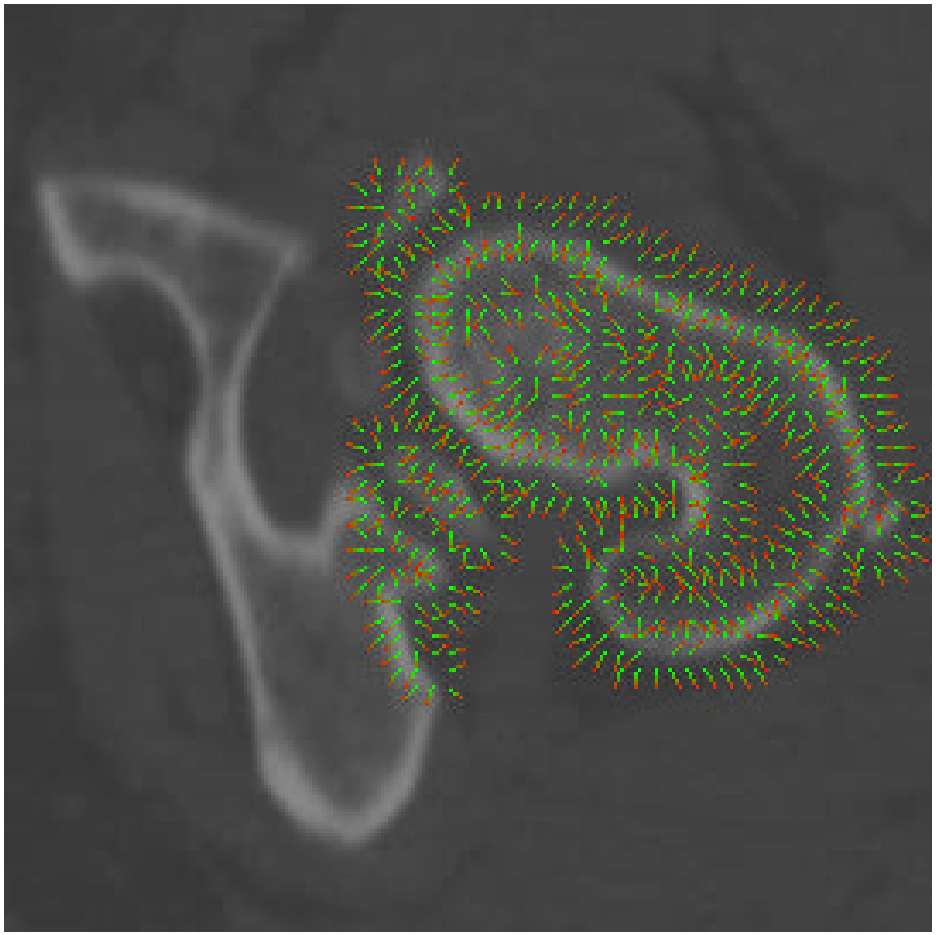


Credit: Yao and Taylor

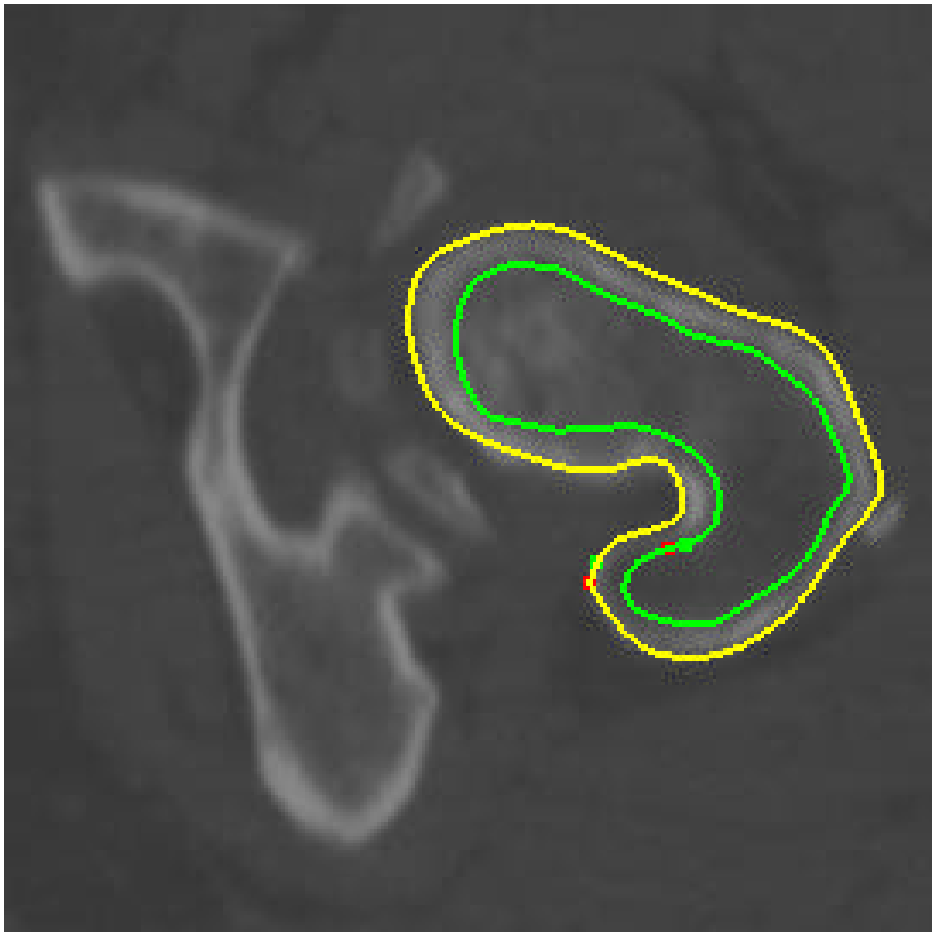
Bone Contour Extraction

- Deformable Contour Algorithm (Snake)
- $F = F_{internal} + F_{image} + F_{external}$
  - $F_{internal}$: the spline force of the contour
  - F_{image} : the image force
  - $F_{external}$: an external force
- Semi-automatic

Bone Contour Extraction



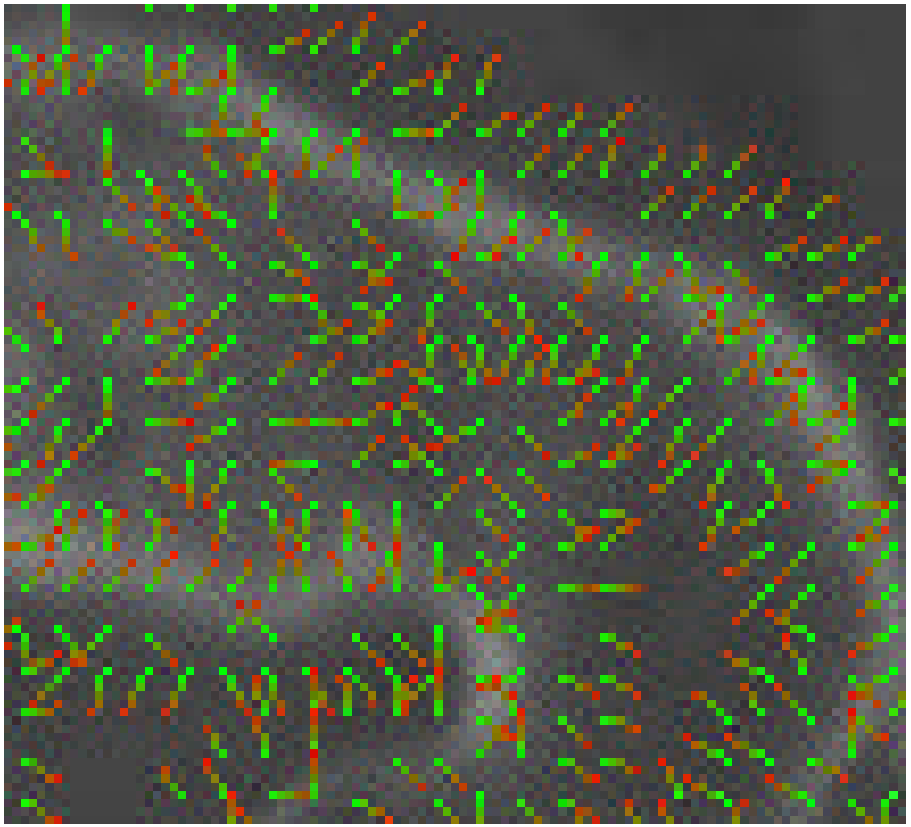
Needle graph of Image force



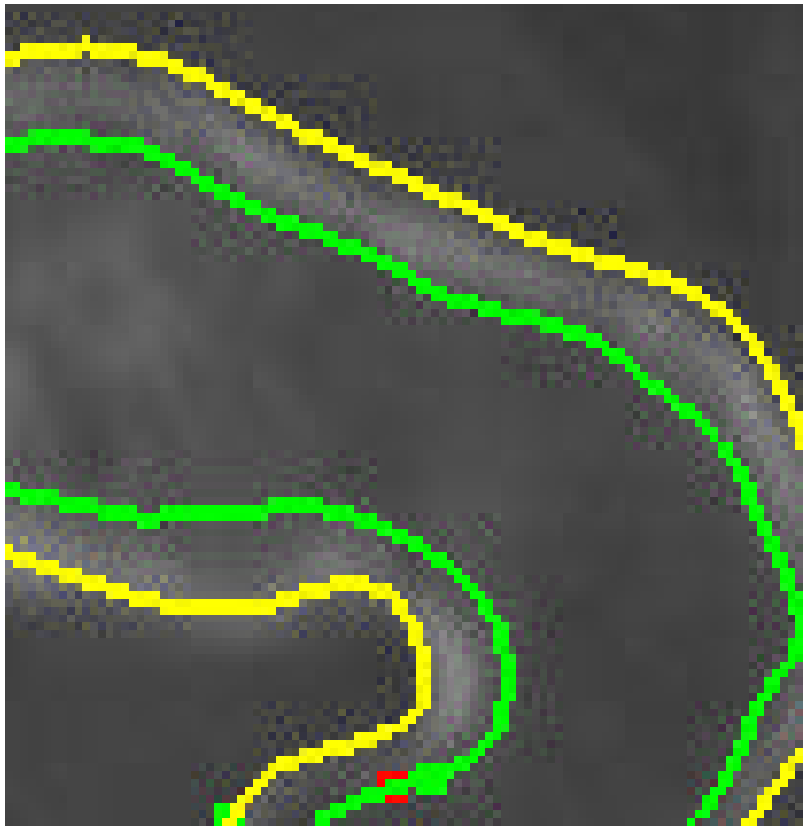
Bone Contours

Credit: Yao and Taylor

Bone Contour Extraction
Closer-up view



Needle graph of Image force



Bone Contours

Credit: Yao and Taylor

Modeling

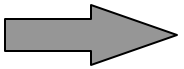
- Representation of anatomical structures
- Models can be
  - Images
  - Labeled images
  - Boundary representations

FROM VOXELS TO SURFACES

Representing solids:

- B-REP - surface representation,
  d/s of vertices, edges, faces.
- CSG- composition of primirive solids

---

binary image  **B-REP representation**

Surface construction algorithms:

- 2D-based algrorithms
- 3D-based algorithms

Surface Representations

- Implicit Representations

- Explicit Representations
  - Polyhedra
  - Interpolated patches
  - Spline surfaces
  - ...

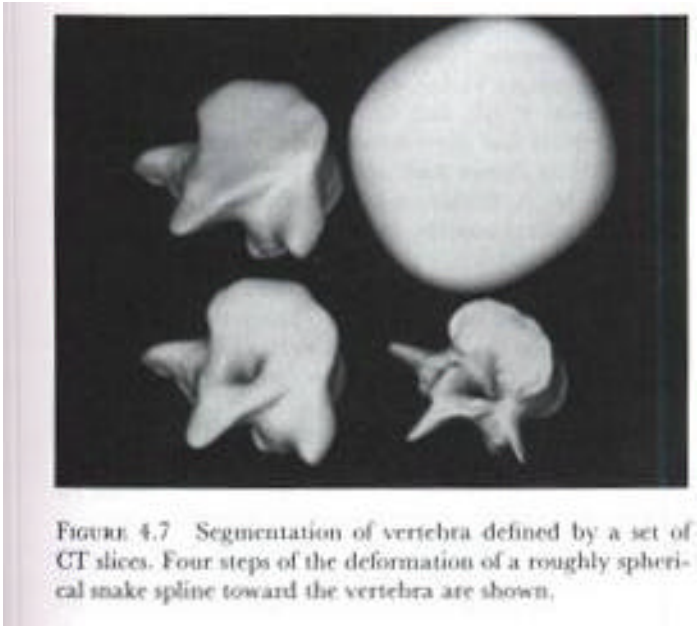
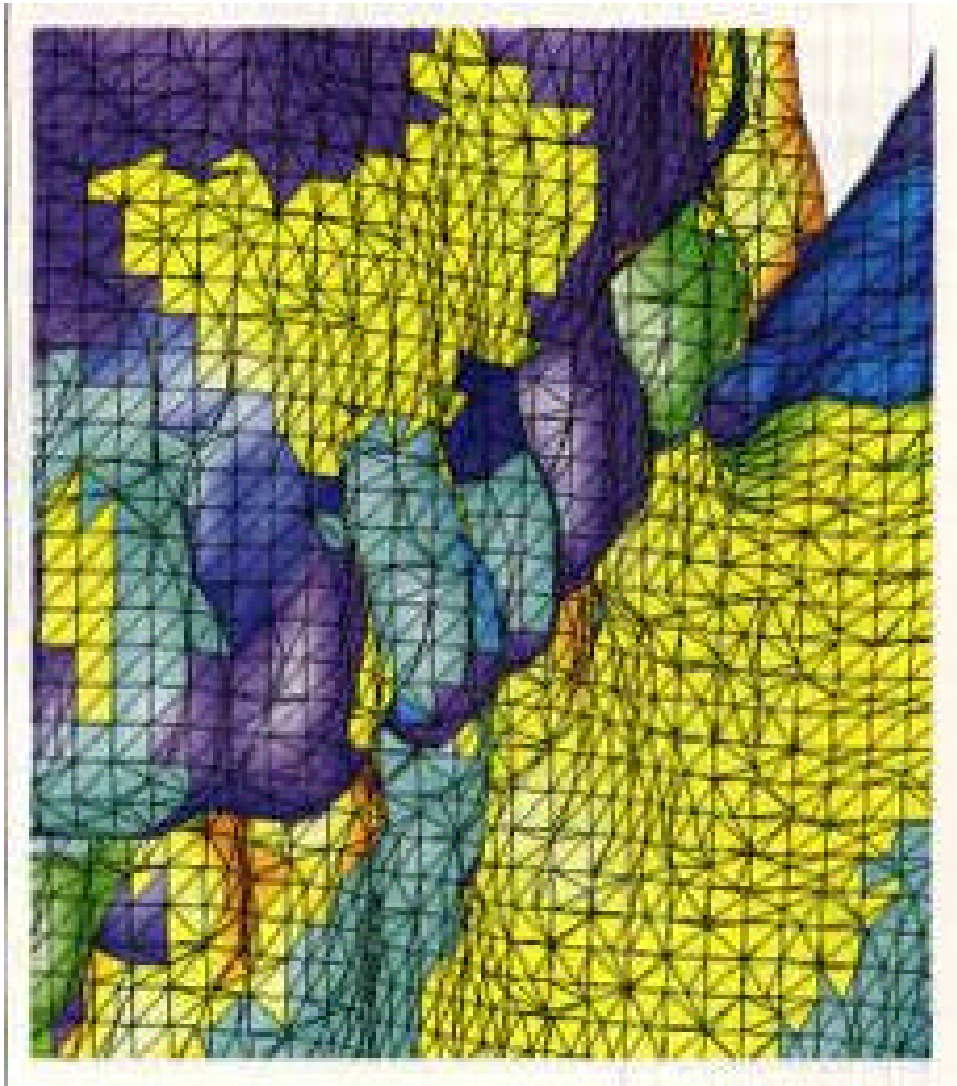


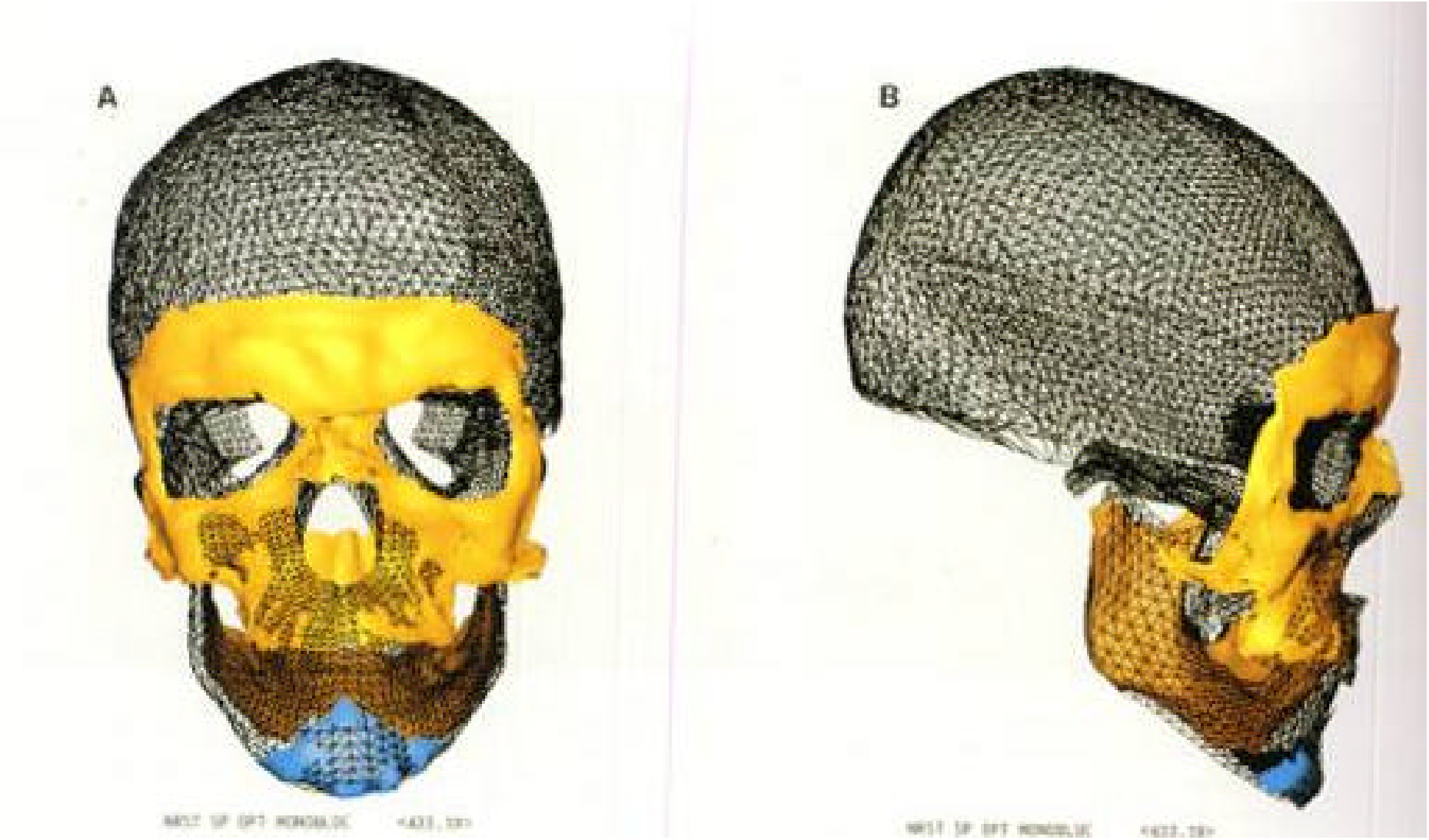
FIGURE 4.7 Segmentation of vertebra defined by a set of CT slices. Four steps of the deformation of a roughly spherical snake spline toward the vertebra are shown.

Source: CIS p 73 (Lavallee image)

Polyhedral Boundary Reps

- Common in computer graphics
- Many data structures.
  - Winged edge
  - Connected triangles
  - etc.

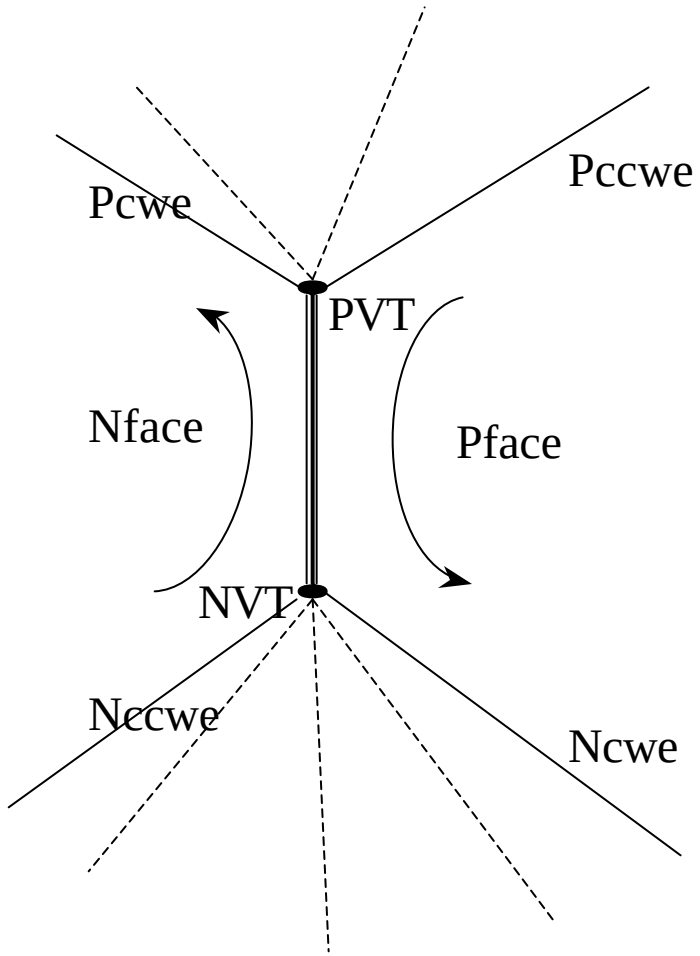




Source: C. Cutting, CIS Book

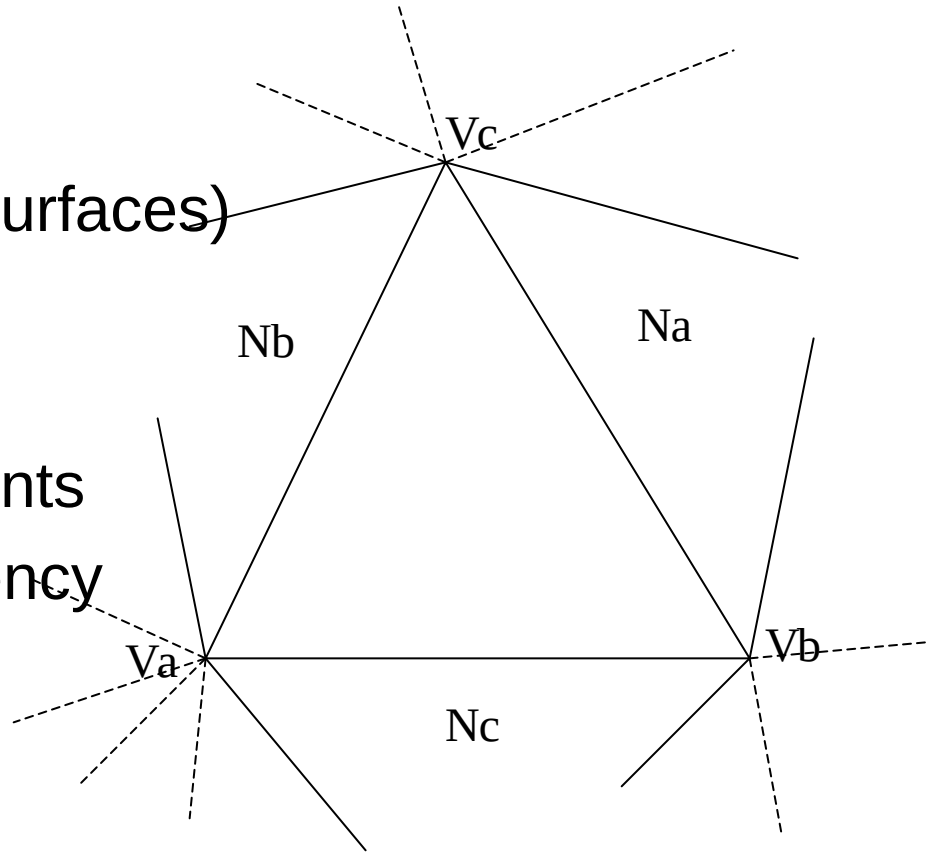
Winged Edge

- Baumgart 1974
- Basic data structures
  - winged edge (topology)
  - vertex (geometry)
  - face (surfaces)
- Key properties
  - constant element size
  - topological consistency



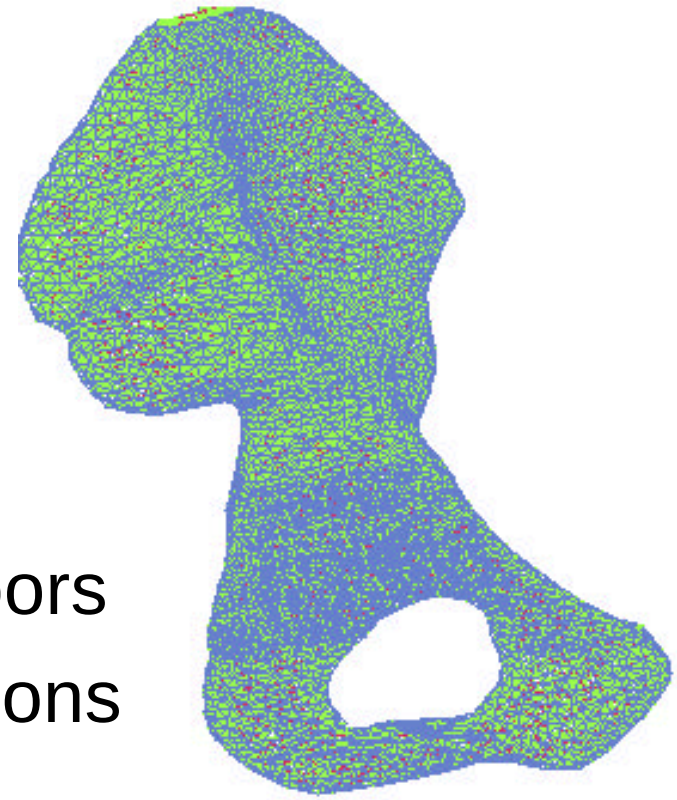
Connected Triangles

- Basic data structures
  - Triangle (topology, surfaces)
  - Vertex (geometry)
- Properties
  - Constant size elements
  - Topological consistency



Tetrahedral Mesh Data Structure

- Vertex list
  - x, y, z coordinates
  - reference to one tetrahedron
- Tetrahedron list
  - references to four vertices
  - references to four face neighbors
- Properties such as density functions



Credit: Yao and Taylor

Advantages of Tetrahedral Mesh

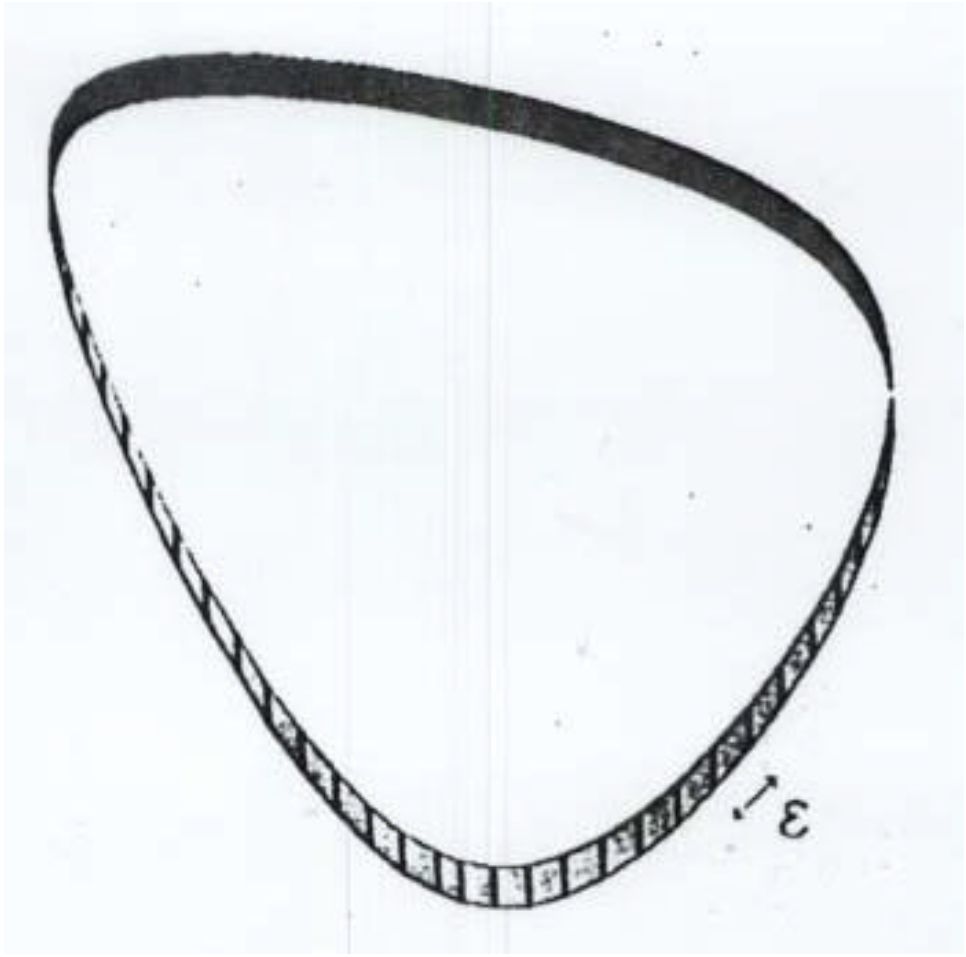
- Greatest degree of flexibility
- Data structure, data traversal, and data rendering are more involved
- Ability to better adapt to local structures
- Computational steps such as interpolation, integration, and differentiation can be done in closed form
- Finite element analysis
- Hierarchical structure of multiple resolution meshes

Credit: Yao and Taylor

Methods to Build Tetrahedral Mesh

- Cubicle Voxel Subdivision
- 3D Delaunay triangulation
- 2D Delaunay triangulation and tiling
- Tiling from contours
  - Fast (local operation)
  - Easy to optimize (metric function, constraints)
  - Adapt to anatomical structures

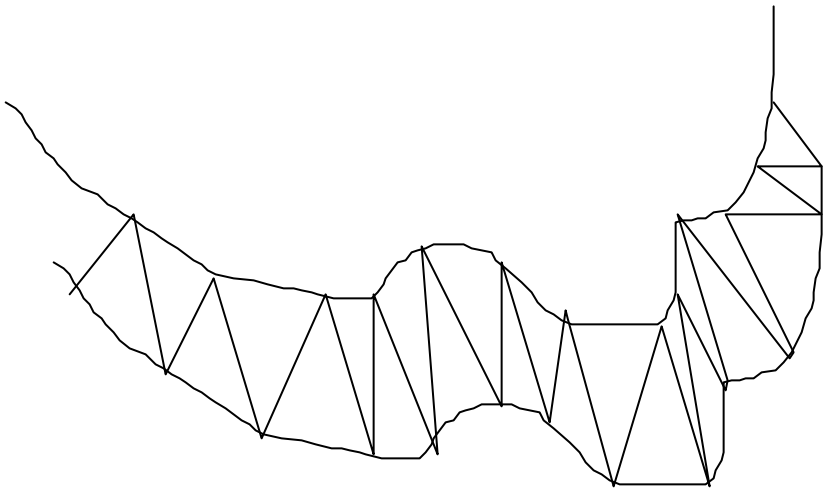
2D-based Methods



Ribbon Stacking

2D-based methods

- Treat 3D volume as a stack of slices
- Outline
  - Find contours in each 2D slice
  - Connect contours to create tiled surfaces

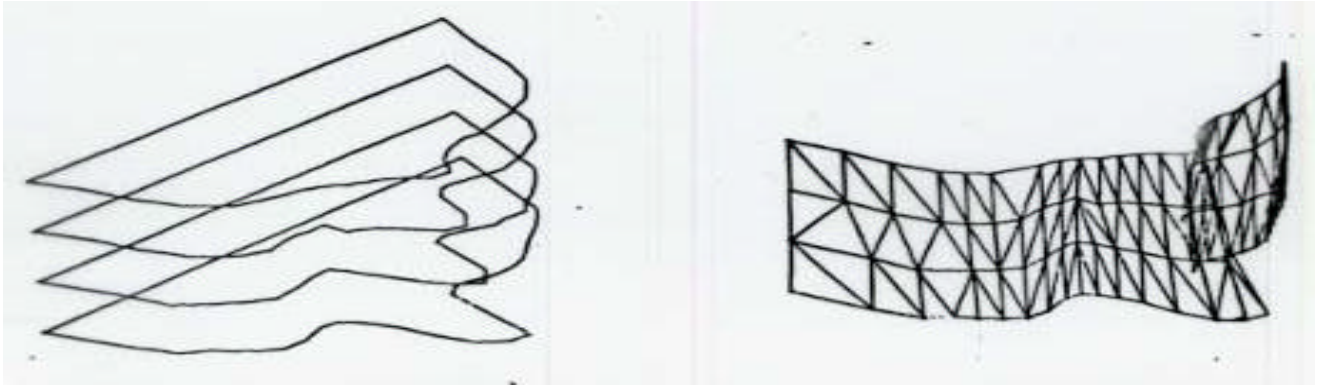


SURFACE CONSTRUCTION ALGORITHMS

2D-based algorithms

1. 2D contour extraction
2. tiling of counours

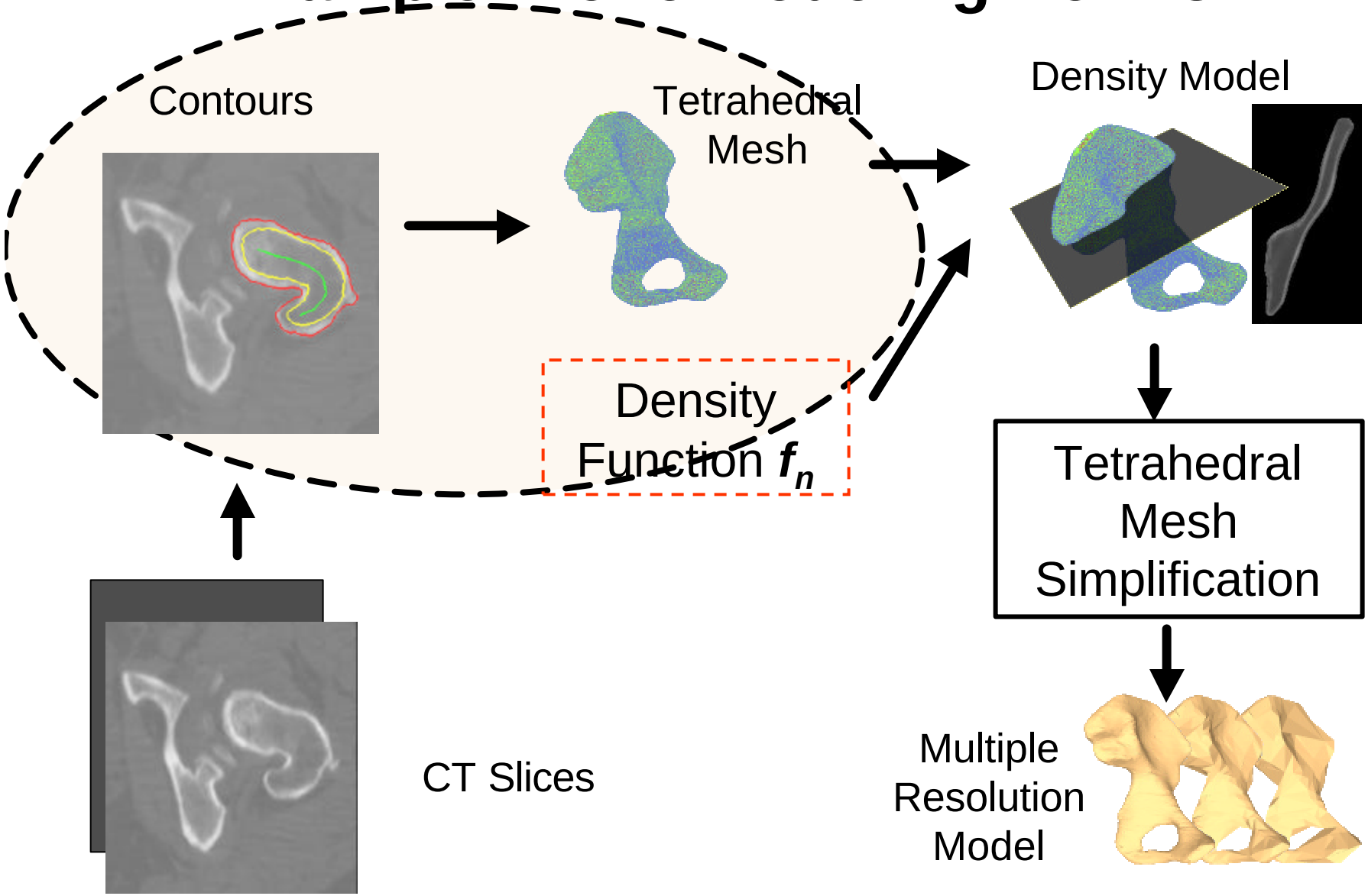
Keppel (1975), Fuchs (1978), Christiansen (1981), Shantz (1981), Ganapathy (1982), Cook (1983), Zyda (1987), Boissonnat (1988), Schwartz (1988)



Contour extraction

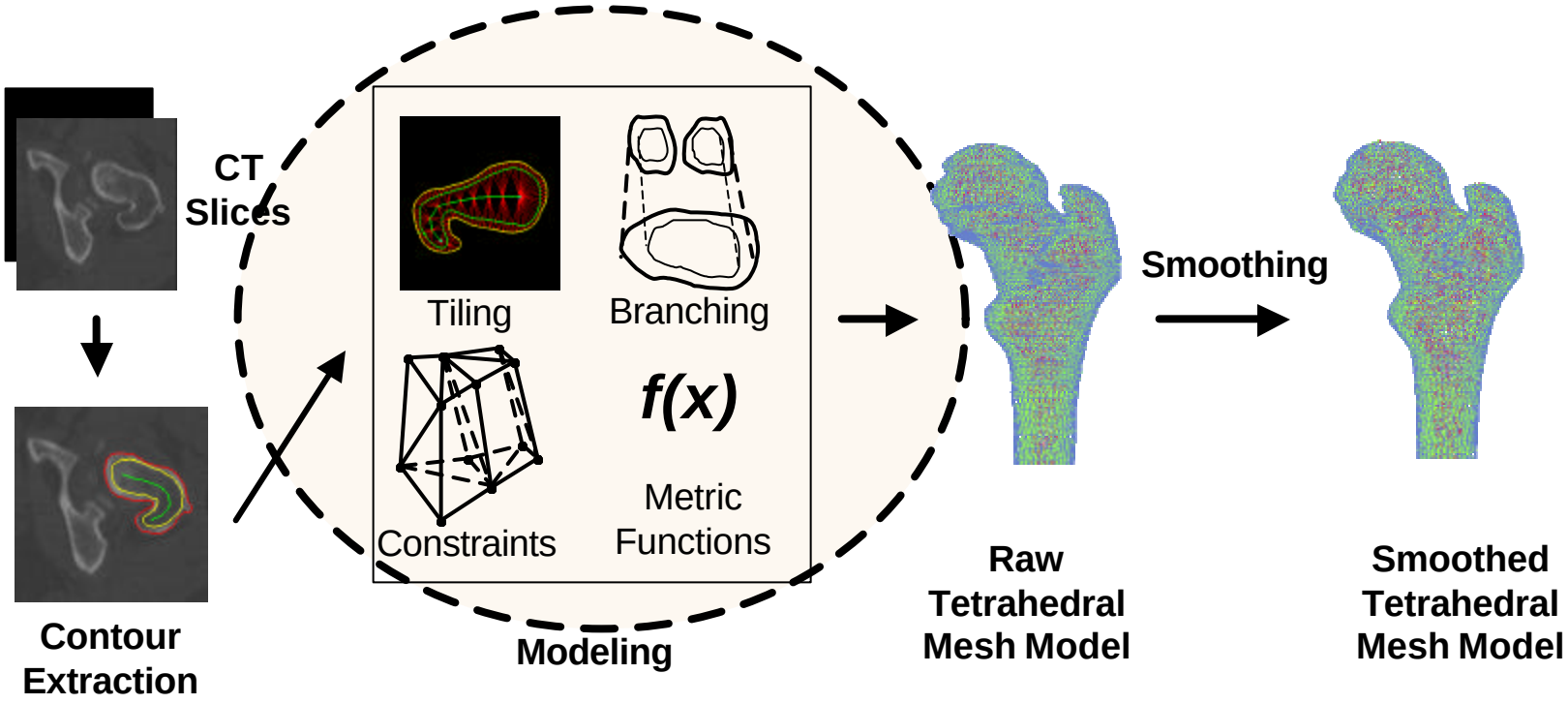
- Sequential scanning
- boundary following (random access to pixels)

Example: Bone Modeling from CT



Credit: Yao and Taylor

Construct Tetrahedral Mesh from Contours



Tetrahedral Mesh Reconstruction from Contours

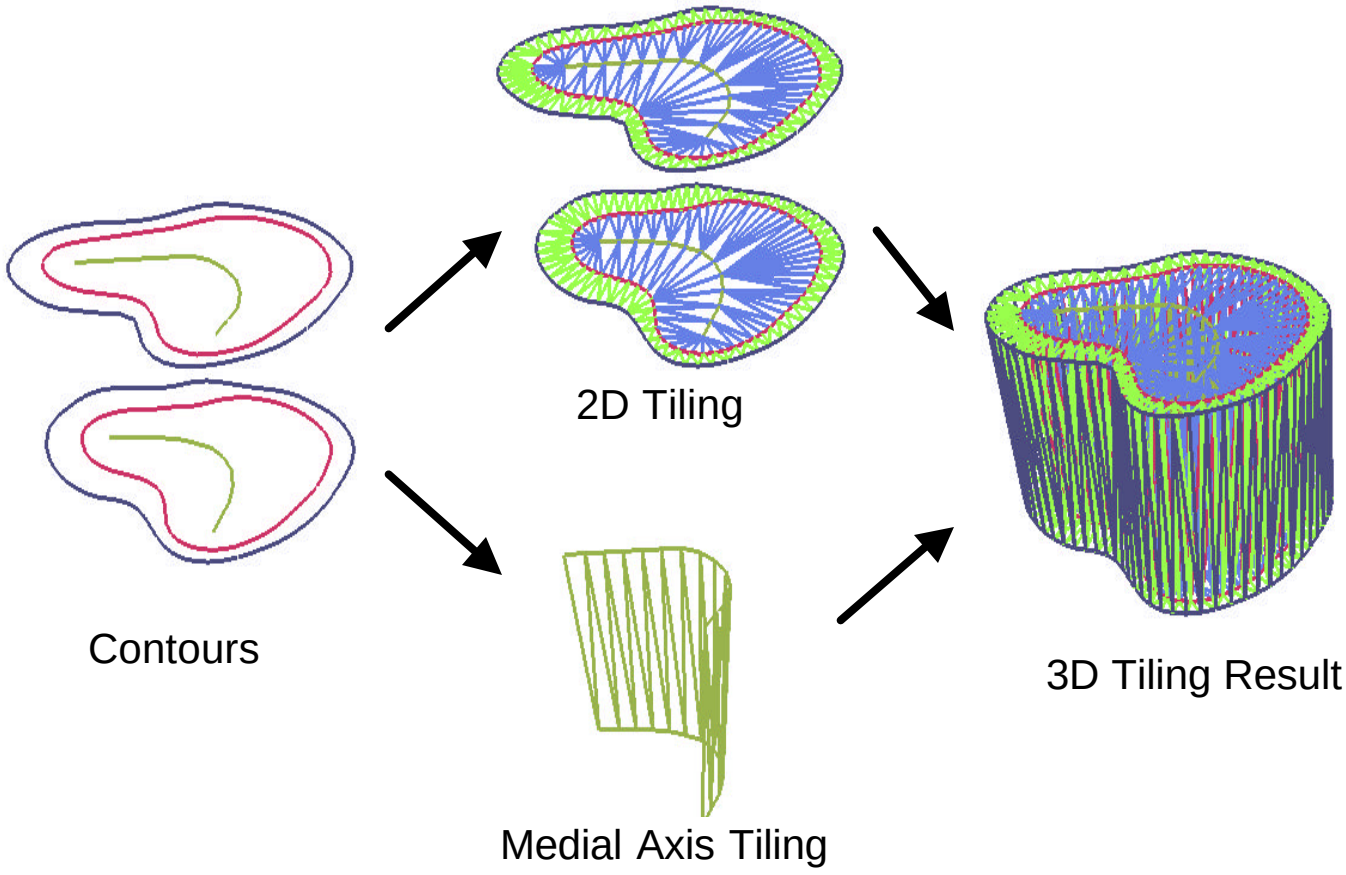
Credit: Yao and Taylor

Tetrahedral Mesh Tiling

- Objectives
  - Subdivide the space between adjacent slices into tetrahedra, slice by slice
- Method
  - Two-steps tiling strategy
    - 2D tiling and medial axis tiling
    - 3D tiling

Credit: Yao and Taylor

Tiling Strategy



Credit: Yao and Taylor

Metric Functions

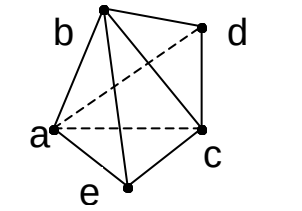
- Maximize Volume, f_v
- Minimize Area, f_a
- Minimize Density Deviation, f_d
- Minimize Span Length, f_s

Current Metric Function:

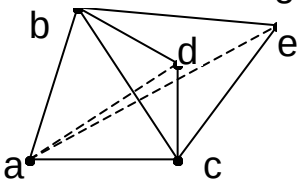
- Combination of minimizing density deviation and span length
- Minimize $F = w_1 * f_d + w_2 * f_s$

Tiling Constraints

- Non-intersection between tetrahedra
- Continuity between slices
- Continuity between layers

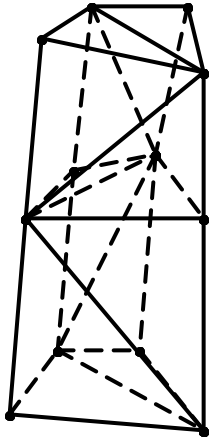


Non-intercrossing

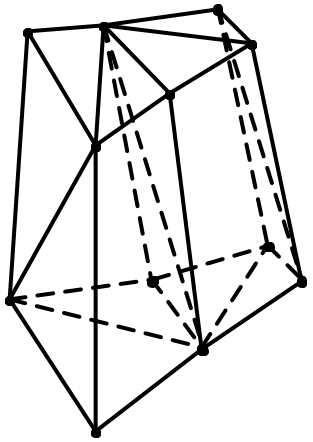


Intercrossing

Intercrossing between tetrahedra



Continuity constraint between slices

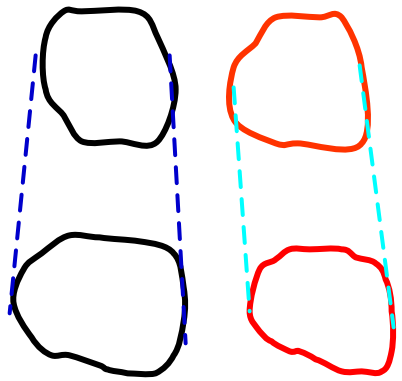


Continuity constraint between layers

Credit: Yao and Taylor

Correspondence Problem

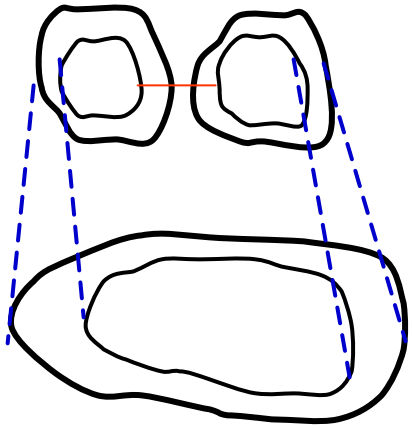
- Examining the overlap and distance between contours on adjacent slices
- Graph based method



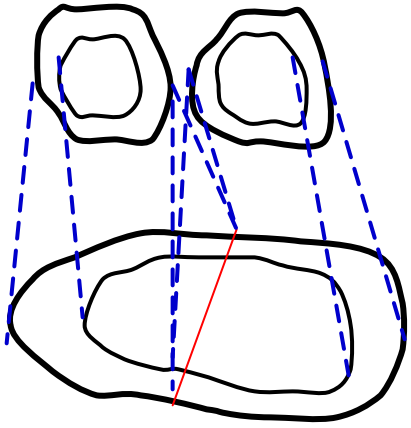
Contour Correspondence

Branching Problem

- Branching Between layers
  - Convert to tiling of 3 contours
- Branching Between contours
  - Composite contour
  - Split contour



Composite Contour

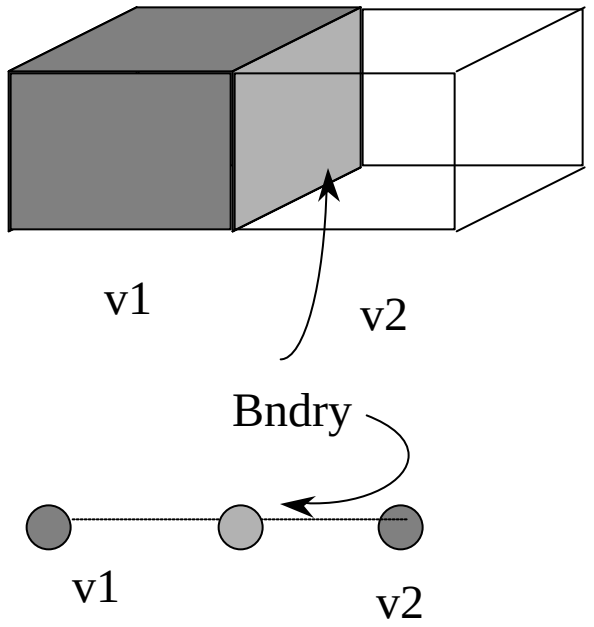


Split Contour

Credit: Yao and Taylor

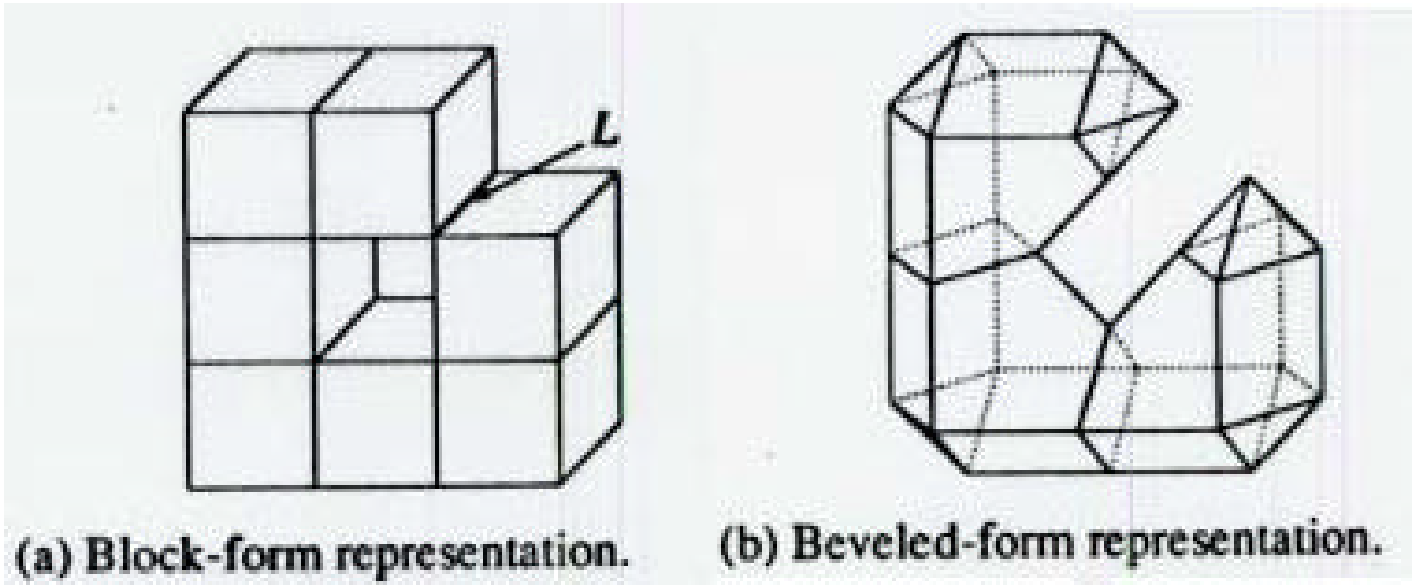
3D-based methods

- Segment image into labeled voxels
- Define surface and connectivity structure
- Can treat boundary element between voxels as a face or a vertex



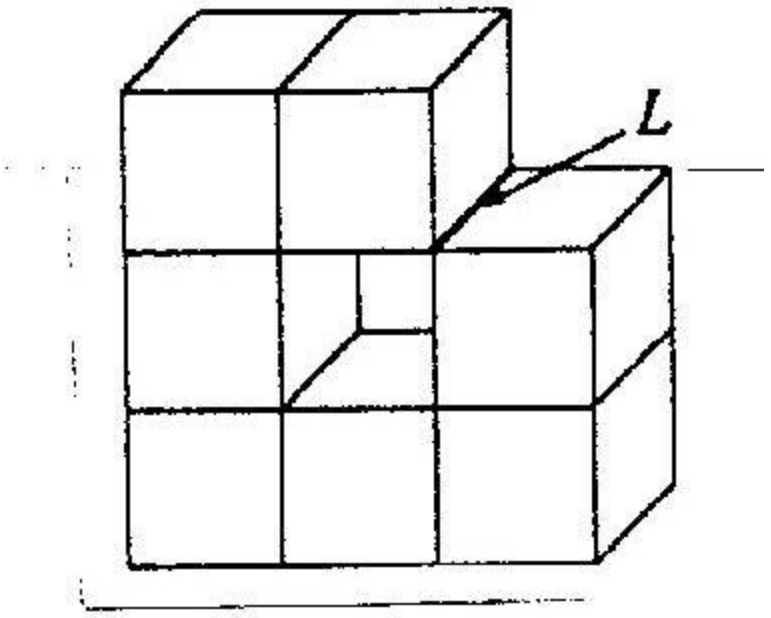
3D-BASED ALGORITHMS

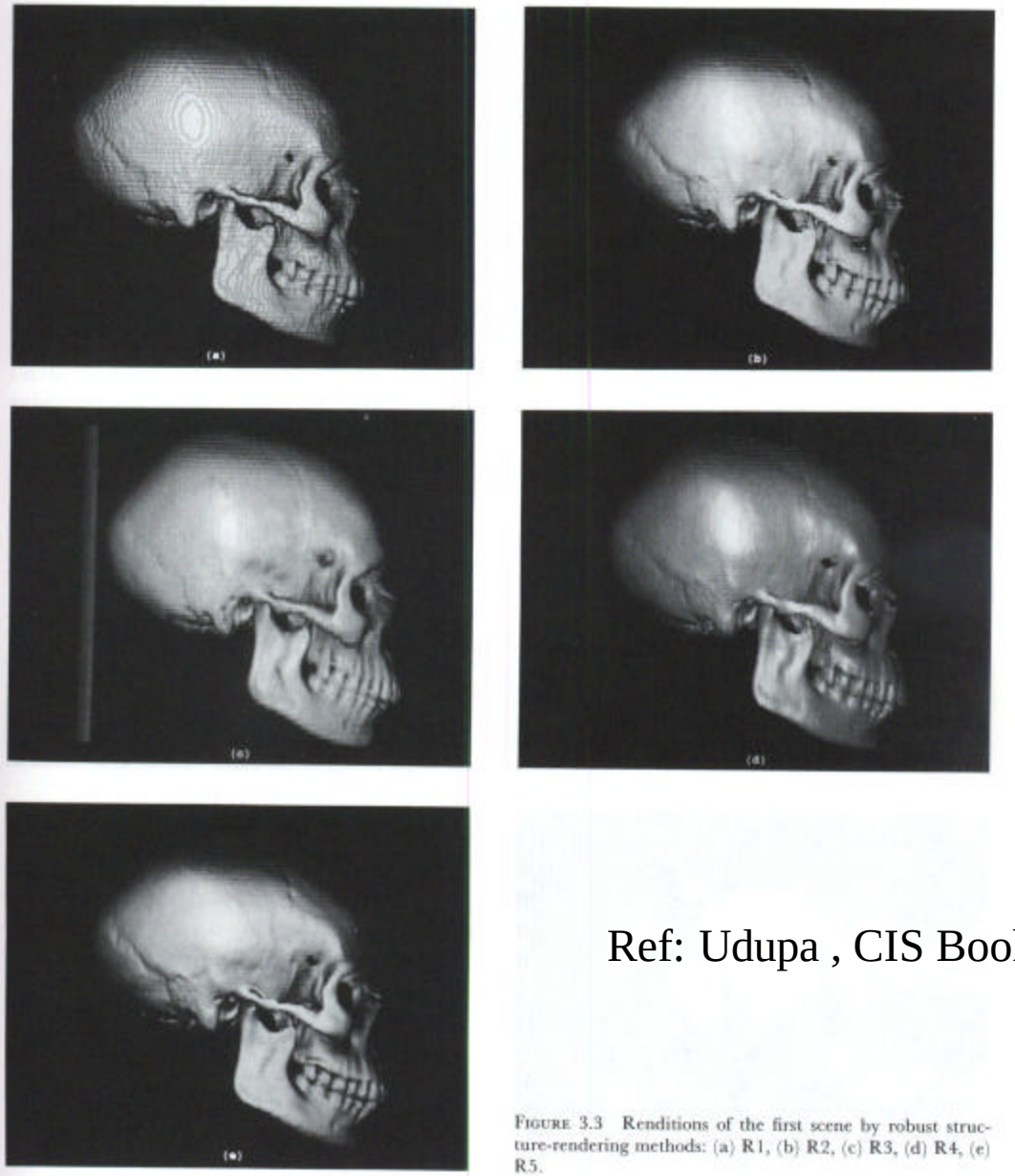
Block-form and Beveled-form representations of surface:



Block form methods

- “Cuberille”-type methods
- Treat voxels as little cubes
- May produce self-intersecting volumes
- E.g., Herman, Udupa



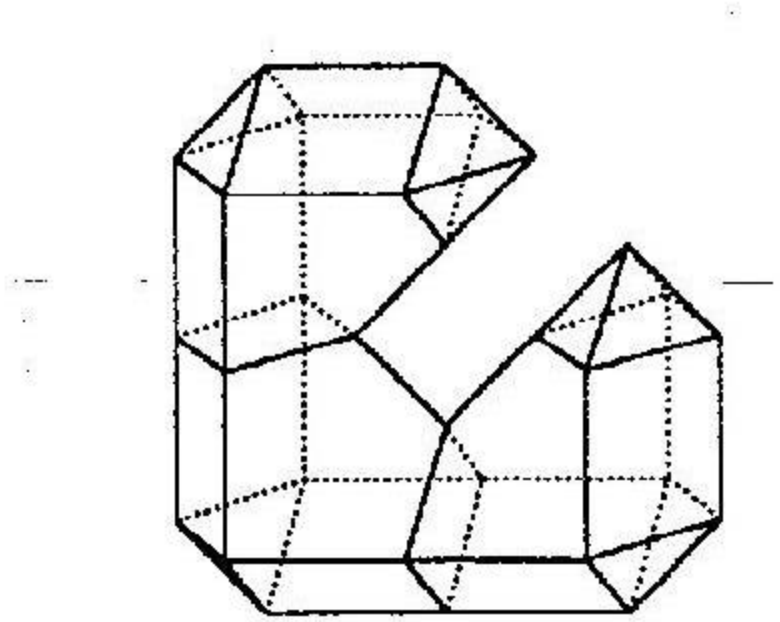


Ref: Udupa , CIS Book, p47

FIGURE 3.3 Renditions of the first scene by robust structure-rendering methods: (a) R1, (b) R2, (c) R3, (d) R4, (e) R5.

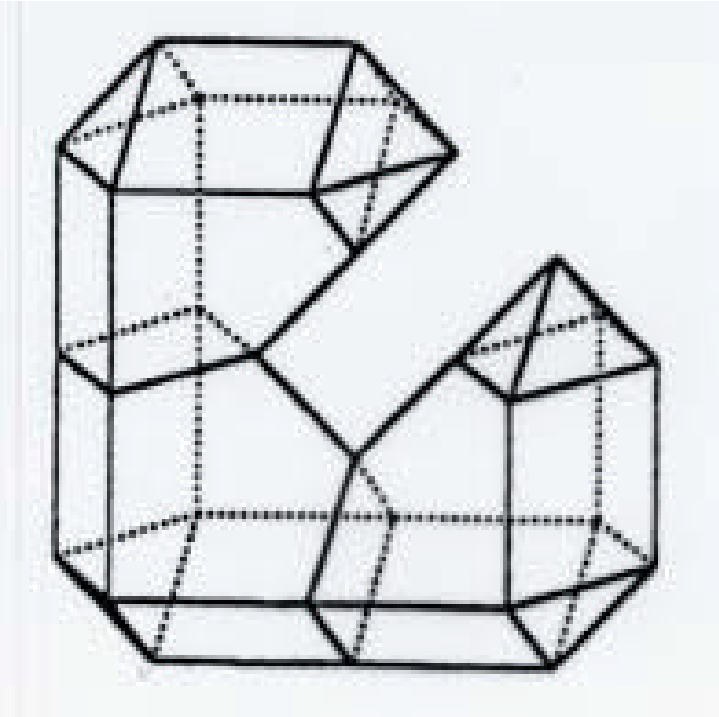
Beveled form methods

- “Marching cubes” type
- Voxels viewed as 3D grid points
- Vertices are points on line between adjacent grid points
- E.g. Lorensen&Cline, Baker, Kalvin, many others



Beveled form methods

- “Marching cubes” type
- Voxels viewed as 3D grid points
- Vertices are points on line between adjacent grid points
- E.g. Lorensen&Cline, Barker Kalvin, many others

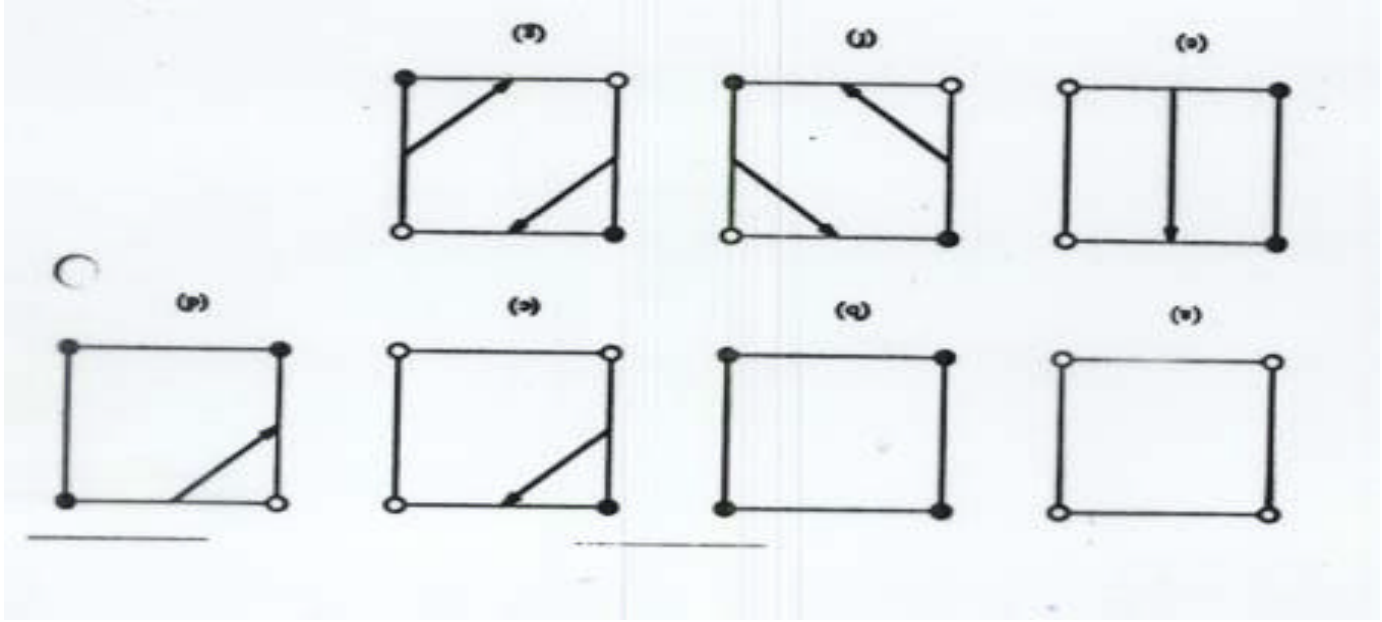


Beveled-form Algorithms and medical Imaging

Classification by definition of *vertex adjacency* (boundary element adjacency).

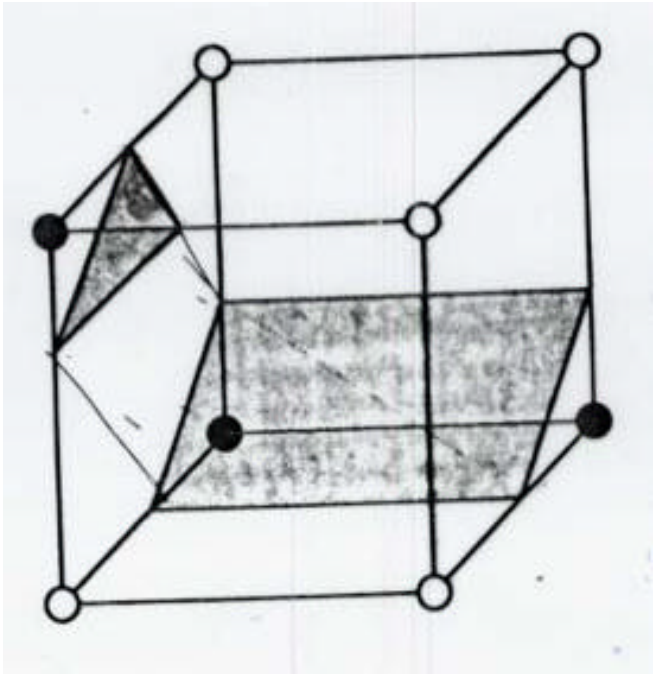
Vertex adjacency can be calculated:

1. Inconsistently
2. Tetrahedral tessellation
3. Supersampling
4. Voxel topology best for 3D medical applications.



Beveled form basic approach

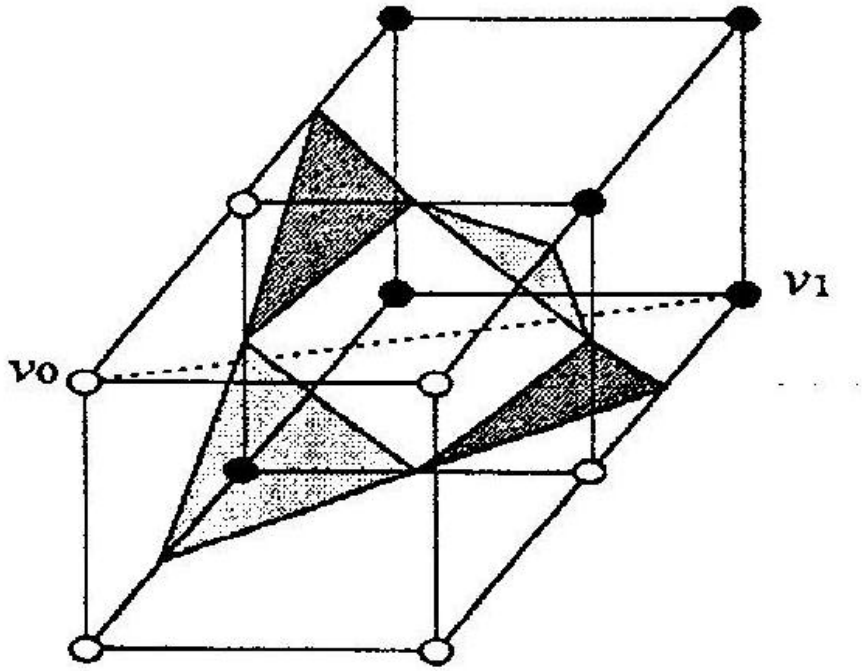
- Segment the 3D volume
- Scan 3D volume to process “8-cells” sequentially
- Use labels of 8 cells as index in (256 element) lookup table to determine where surfaces pass thru cell
- Connect up topology
- Use various methods to resolve ambiguities



Source: Kalvin survey

Marching Cubes

- Lorensen & Kline
- Probably best known
- Used symmetries to reduce number of cases to consider from 256 to 15
- BUT there is an ambiguity



Wyvill, McPheters, Wyvill

Step 1: determine edges on each face of 8 cube

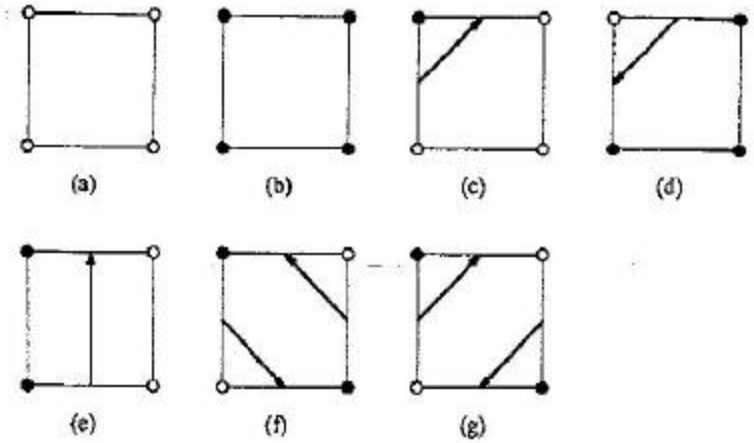
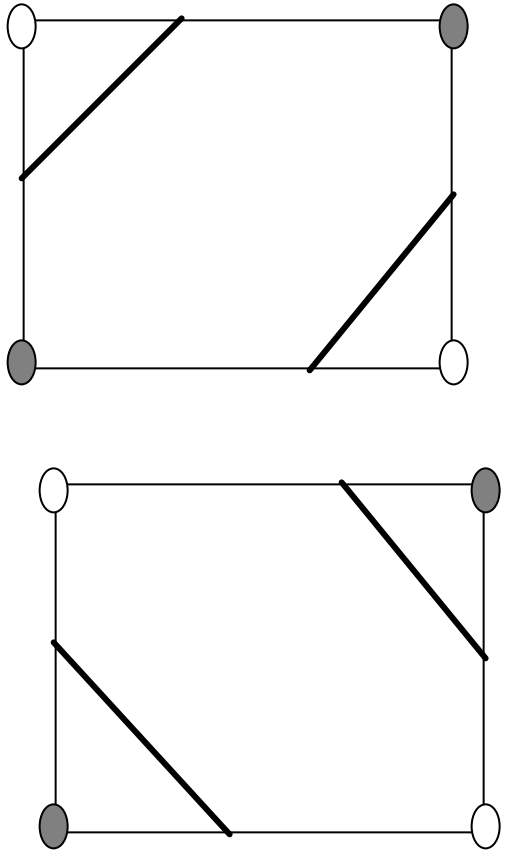


Figure 6: The seven cases for calculating vertices and ec

Step 2: Connect the edges up to make surfaces

Ambiguities

- Arise when alternate corners of a 4-face have different labels
- Ways to resolve:
  - supersampling
  - look at adjacencent cells
  - tetrahedral tessallation



Tetrahedral Tessalation

- Many Authors
- Divide each 8-cube into tetrahedra
- Connect tetrahedra
- No ambiguities

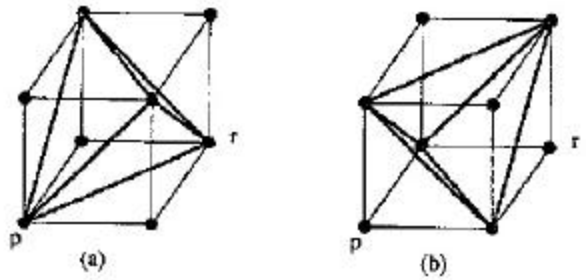


Figure 8: The two tetrahedral partitionings of an 8-cell.

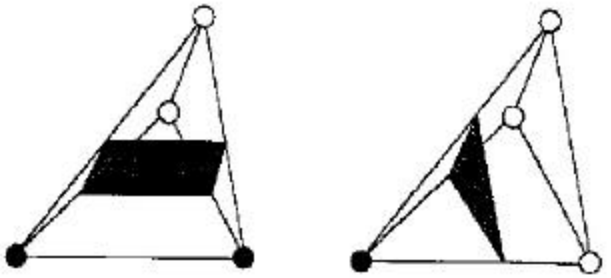


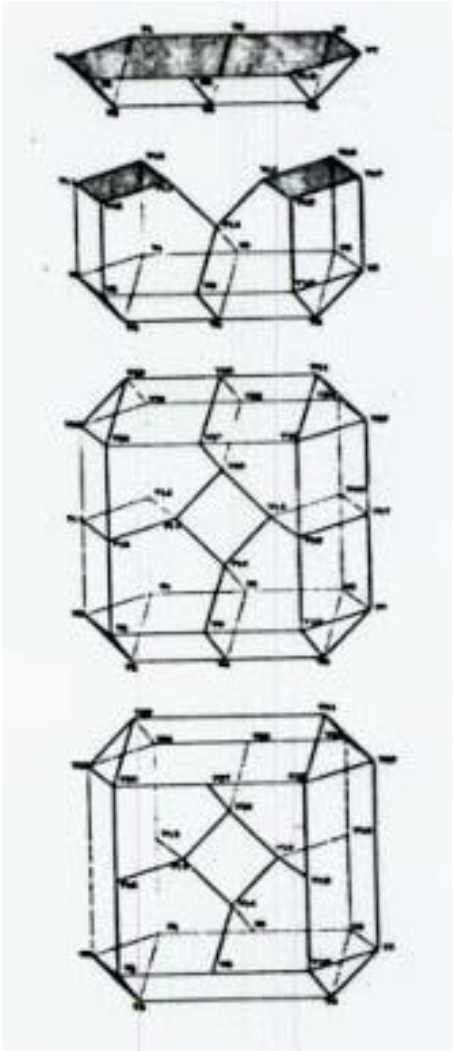
Figure 9: The two cases used for surface construction.

Beveled-form algorithms based on the tetrahedral decomposition of the 3D volume have been developed Payne and Toga [34], Hall and Warren [21], and Nielson *et al.* [29].

While this approach does provide a neat resolution to the ambiguous 8-cell problem, it

Alligator Algorithm

- Phase 1: Initial Construction
- Phase 2: Adaptive Merging

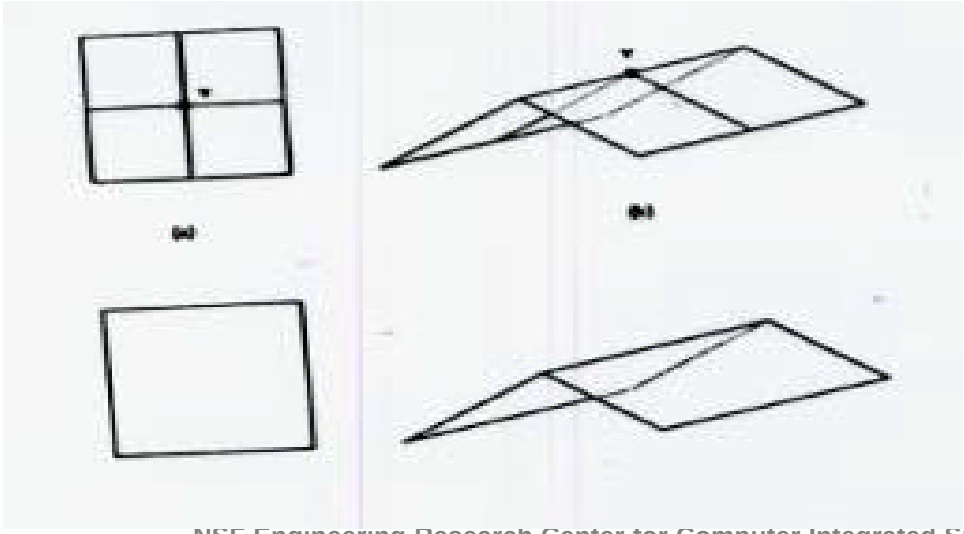


ALLIGATOR ALGORITHMS

Phase 2 - Adaptive face merging

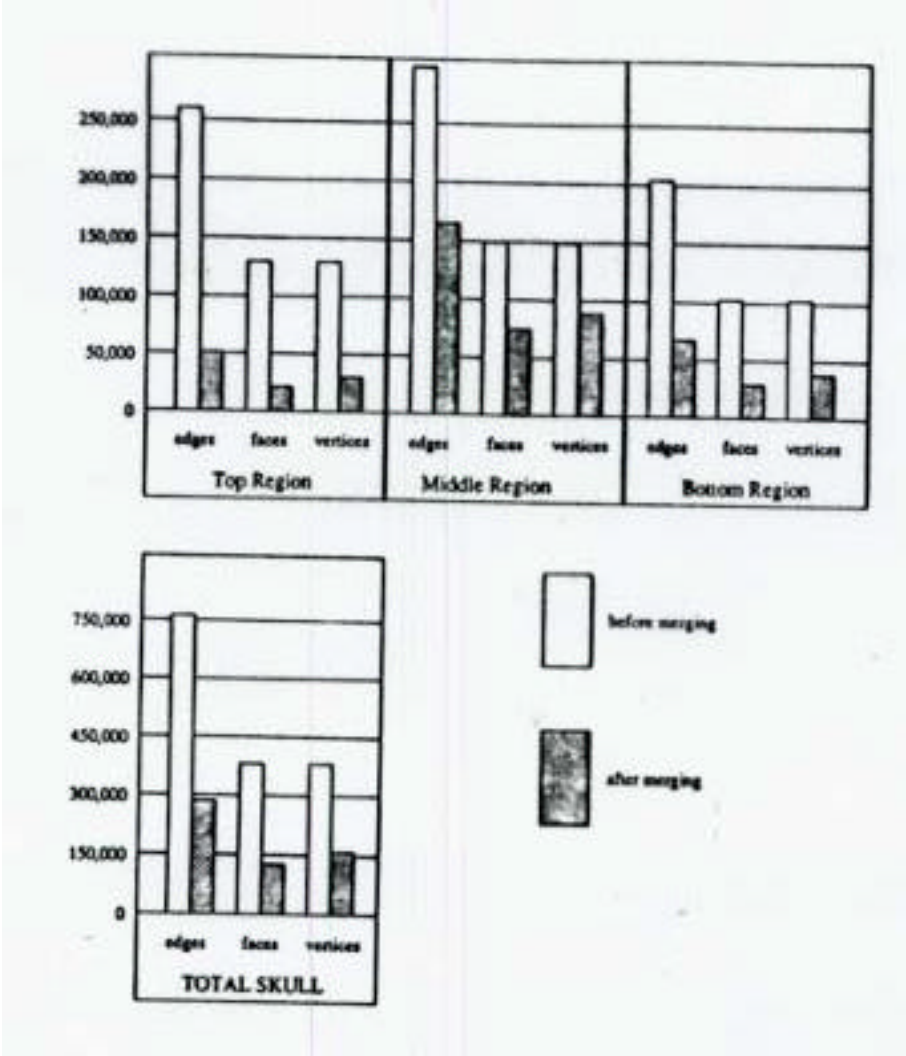
Algorithm exploits the following:

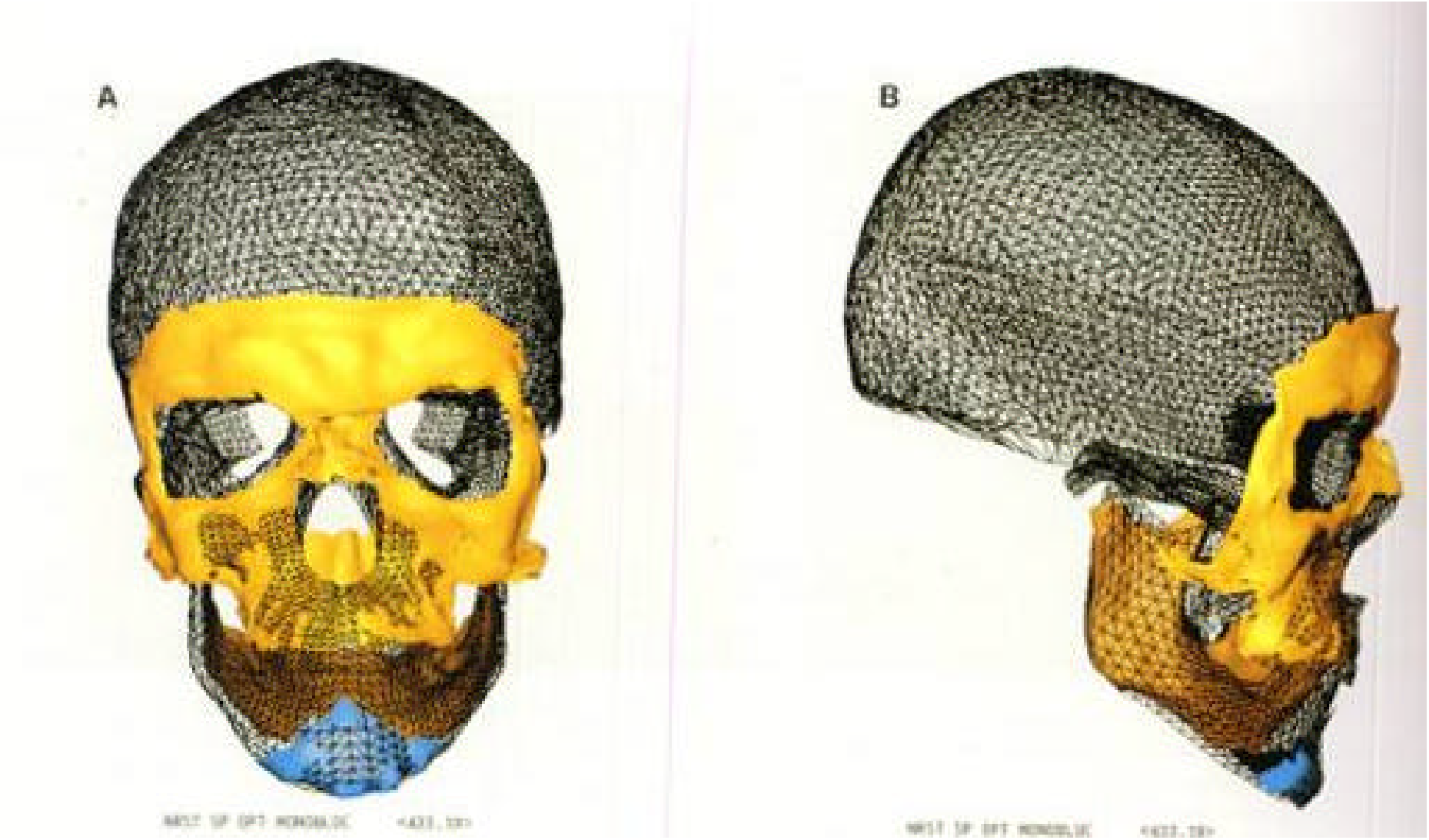
1. beveled-form property:
   - each vertex lies on 4 faces
   - only 2 possible ways for a vertex to lie on 4 regular faces.
2. Euler operators
   - simple, high-level operations
   - efficient
   - simplifies proof of correctness (e.g. topological genus)



ALLIGATOR ALGORITHMS

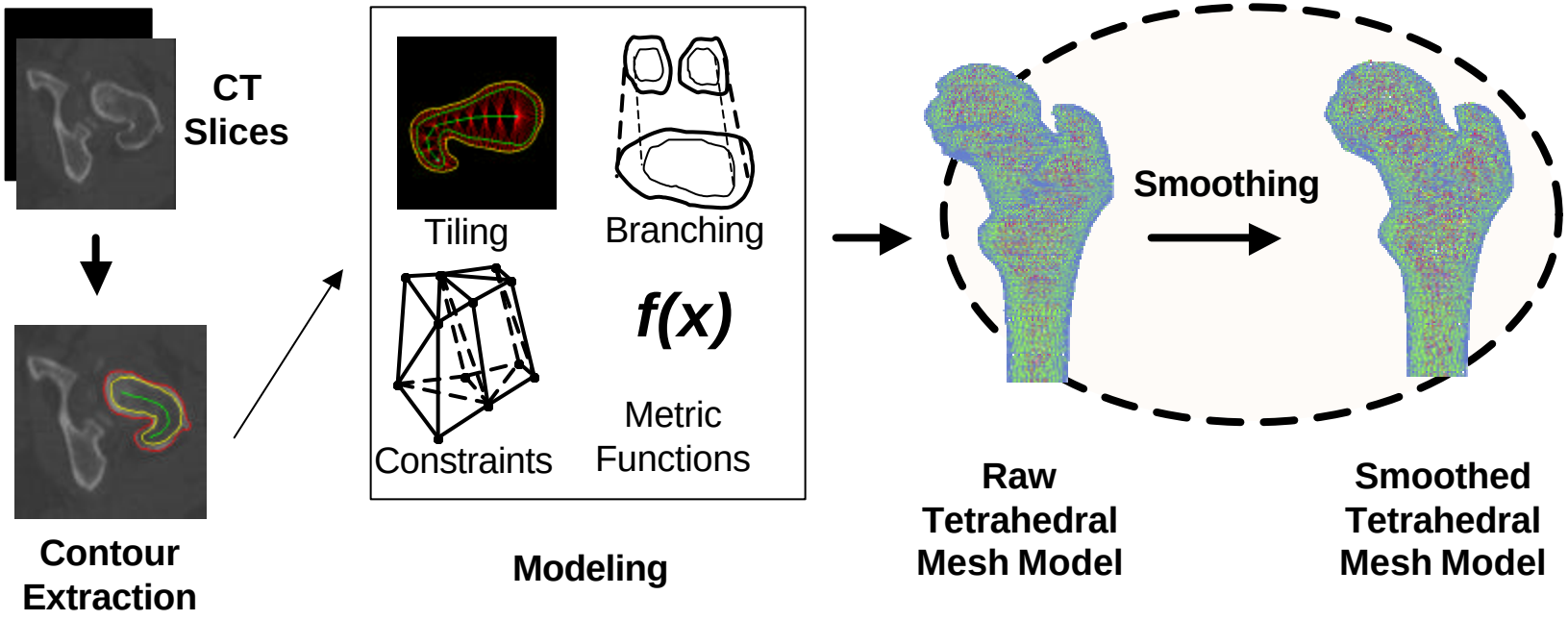
Phase 2 - Adaptive face merging





Source: C. Cutting, CIS Book

Tetrahedral Mesh Smoothing



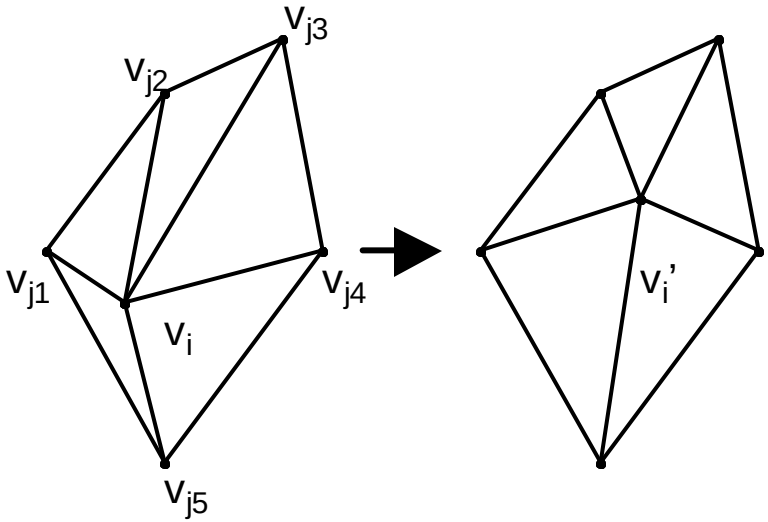
Tetrahedral Mesh Smoothing

- Motivations
  - Noises in CT data set
  - Artifacts during segmentation
- Methods
  - Enhanced Laplacian method

Classic Laplacian Smoothing Method

- Equation

$$v_i' = \frac{1}{|N_i|} \sum_{j \in N_i} v_j$$

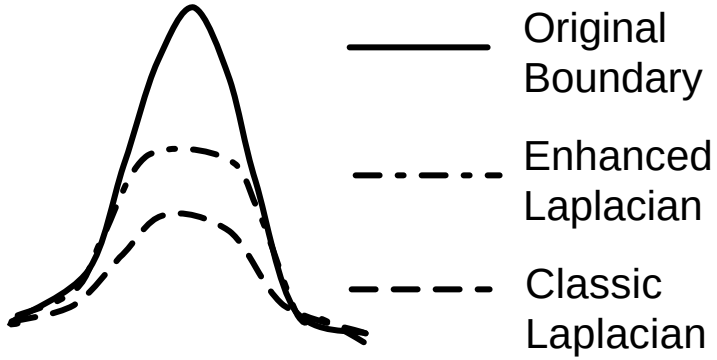


- Drawbacks
  - Shrinkage
  - Invalid elements

Enhanced Laplacian Smoothing Method

- Objective
  - Reduce shrinkage
- Method
  - Project back to boundary

$$v_i' = proj\left(\frac{1}{|N_i|}\sum_{j \in N_i} v_j\right)$$

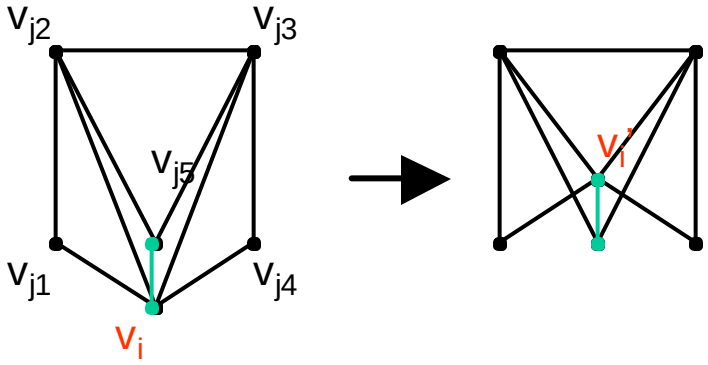


Enhanced Laplacian Smoothing Method

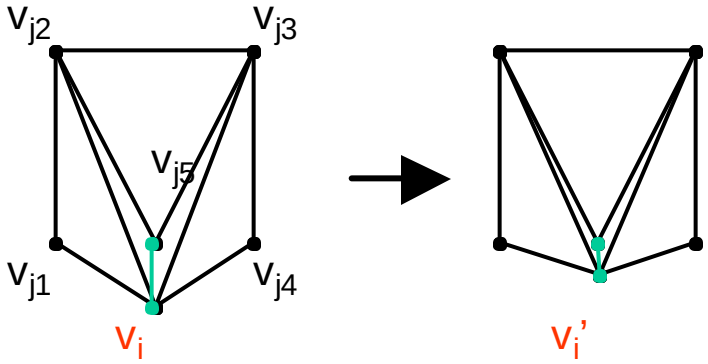
- Objective
  - Prevent invalid element
- Method
  - Iterative assignment

$$v_i^{'(0)} = proj\left(\frac{1}{|N_i|}\sum_{j \in N_i} v_j\right)$$

$$v_i^{'(k)} = a \cdot v_i + (1-a)v_i^{'(k-1)} \quad ,0 \le a \le 1$$

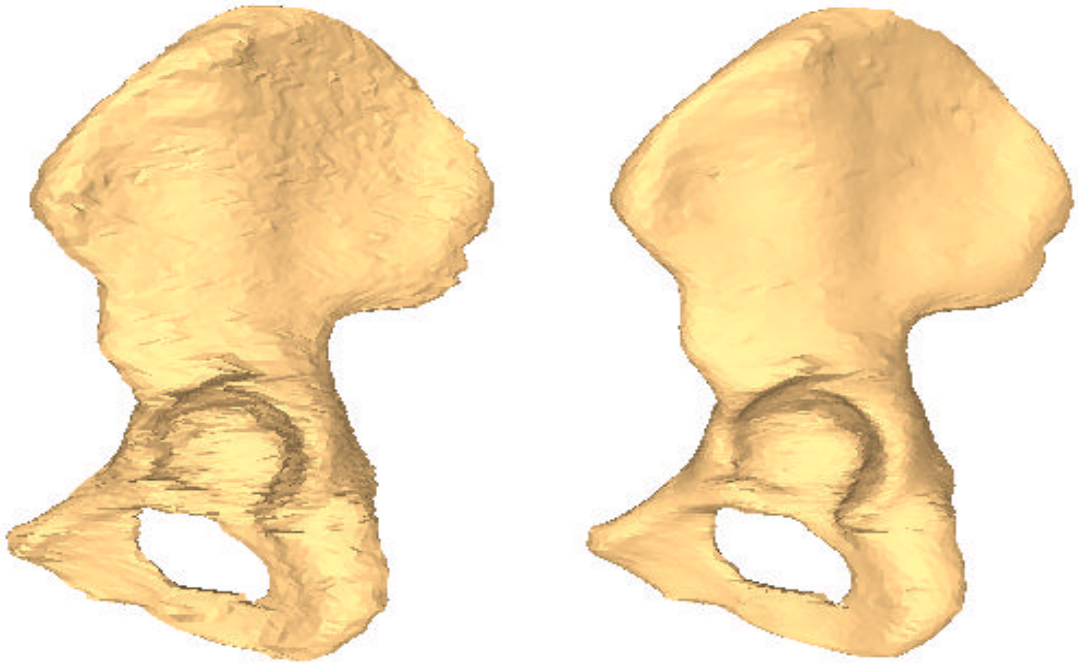


Classic Laplacian



Enhanced Laplacian

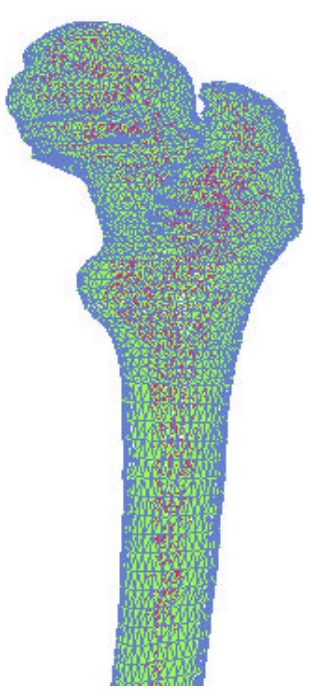
Mesh Smoothing Results

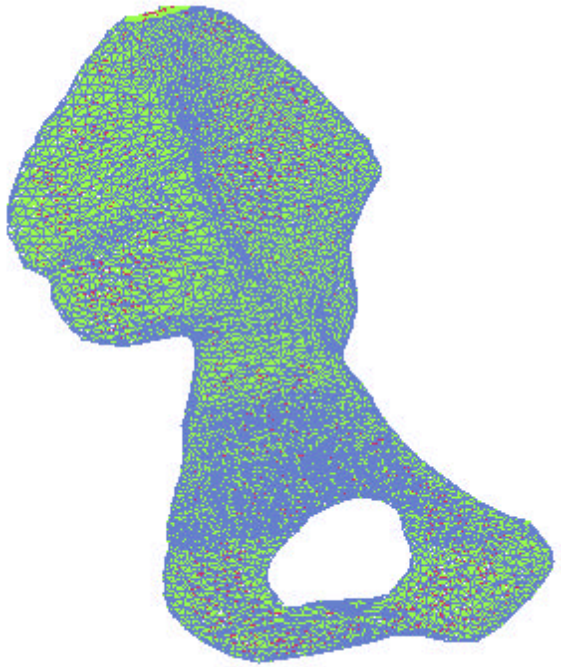


a) Before Smoothing

b) After Smoothing

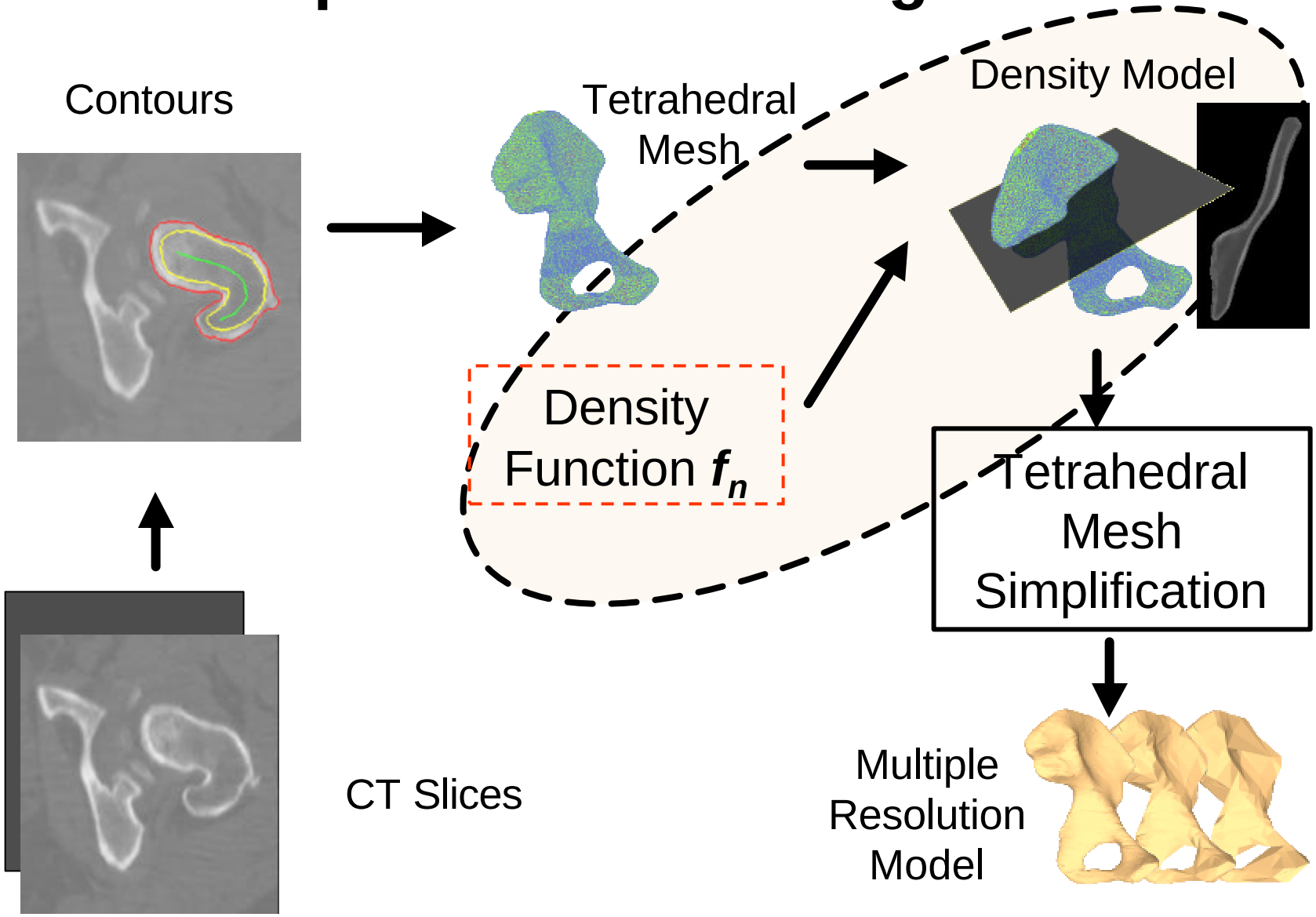
Tetrahedral Mesh Models





Model	Num of Vertices	Num of Tetrahedra	Num of Slices	Total Num of Voxels inside	Avg Num of voxels Per Tetra	Volume (mm^3)	Avg Vol. Per Tetra (mm^3)
Femur	6163	31,537	83	1,802,978	57.1	312,107	9.9
Pelvis	8219	32,741	110	1,941,998	59.3	347,070	10.6

Example: Bone Modeling from CT



Density Functions

- n-degree Bernstein polynomial in barycentric coordinate

$$D(m) = \sum_{i+j+k+l=n}^{n} C_{i,j,k,l} B^{n}_{i,j,k,l}(m)$$

$C_{i,j,k,l}$ polynomial coefficient

$$B^{n}_{i,j,k,l}(m) = \frac{n!}{i!\,j!k!l!} m_x^i m_y^j m_z^k m_w^l$$ barycentric Bernstein basis

Barycentric Coordinate of Tetrahedron

- Local coordinate system
- Symmetric and normalized
- Every 3D position can be defined by an unique coordinate (x, y, z, w)

  $V = x*V_a+y*V_b+z*V_c+w*V_d$

  $x+y+z+w=1$, V_a, V_b, V_c, V_d are coordinate of Tetrahedron vertices

  x,y,z,w within[0,1] if V is inside the tetrahedron

Density Functions

- Advantages
  - Efficient in storage
  - Continuous function
  - Explicit form
  - Convenient to integrate, to differentiate, and to interpolate

Fitting Density Function

- Minimize the density difference between the density function and CT data set

$$\min \sum_{r_i \in \Omega} \left(\left(\sum_{i+j+k+l=n}^{n} C_{i,j,k,l} B^{n}_{i,j,k,l}\left(m_{r_i}\right) \right) - T\left(m_{r_i}\right) \right)^2$$

W is the set of sample voxels, $T(m_{ri})$ is the density value from the CT data set.

$$\begin{bmatrix} B_1(m_{r1}) & B_2(m_{r1}) & \dots & B_m(m_{r1}) \\ B_1(m_{r2}) & B_2(m_{r2}) & \dots & B_m(m_{r2}) \\ \vdots & \vdots & \vdots & \vdots \\ B_1(m_{rs}) & B_2(m_{rs}) & \dots & B_m(m_{rs}) \end{bmatrix} \cdot \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{bmatrix} = \begin{bmatrix} T(m_{r1}) \\ T(m_{r2}) \\ \vdots \\ T(m_{rs}) \end{bmatrix}$$

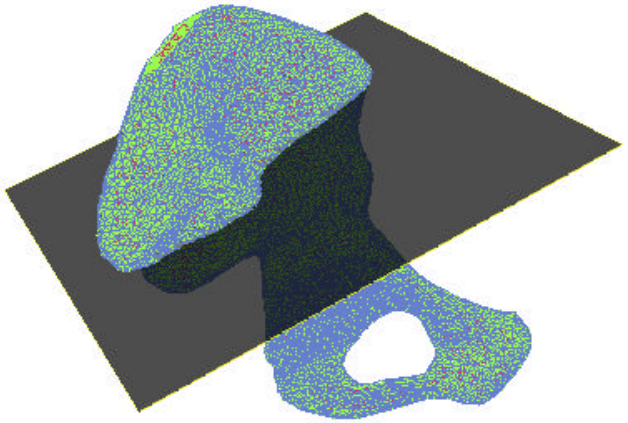
s: number of sample voxels

m: number of density function coefficient,

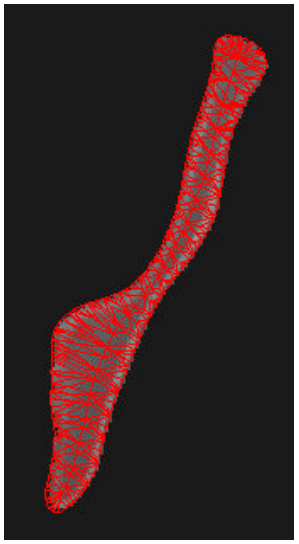
$s>2m$

Accuracy vs Degree of Density Function

- Use CT data set as ground truth
- Cut an arbitrary plane through the model

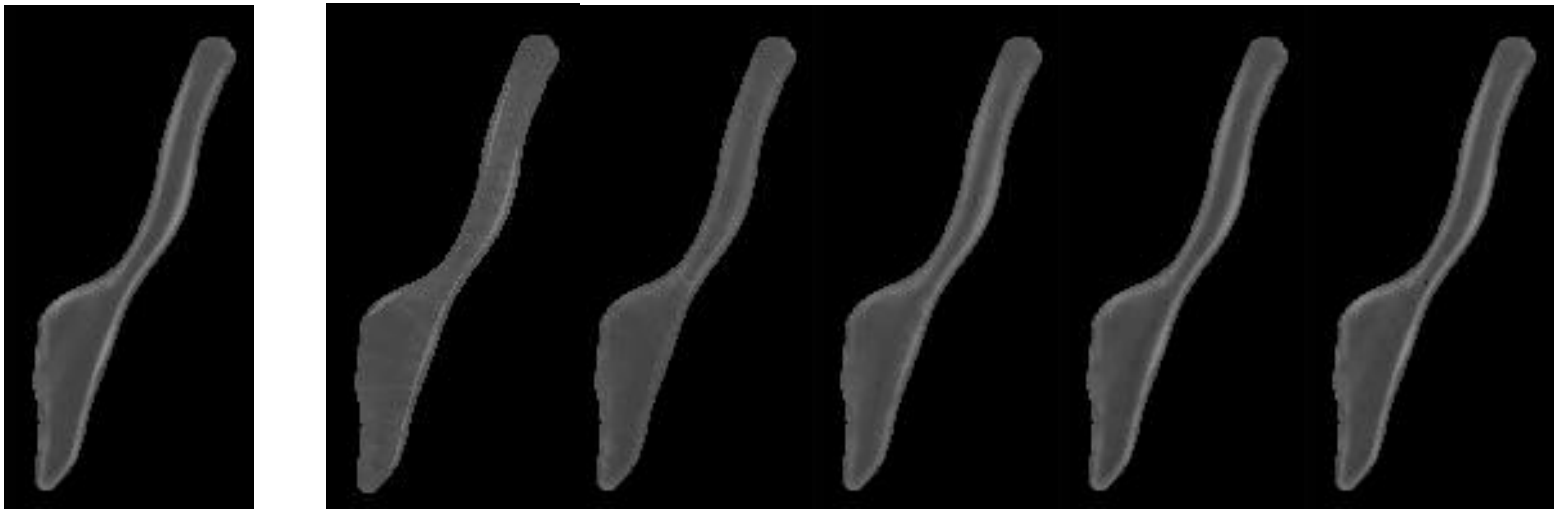


Arbitrary Cutting Plane



Partitions by tetrahedra
on cutting plane

Accuracy vs Degree of Density Function (cont')



Ground Truth n=0 n=1 n=2 n=3 n=4

Degree	0	1	2	3	4	5	6	7	8
Coeff Number	1	4	10	20	35	56	84	120	165
Avg. Density Err (%)	3.291	1.583	0.766	0.442	0.298	0.216	0.167	0.149	0.128

Example: Bone Modeling from CT