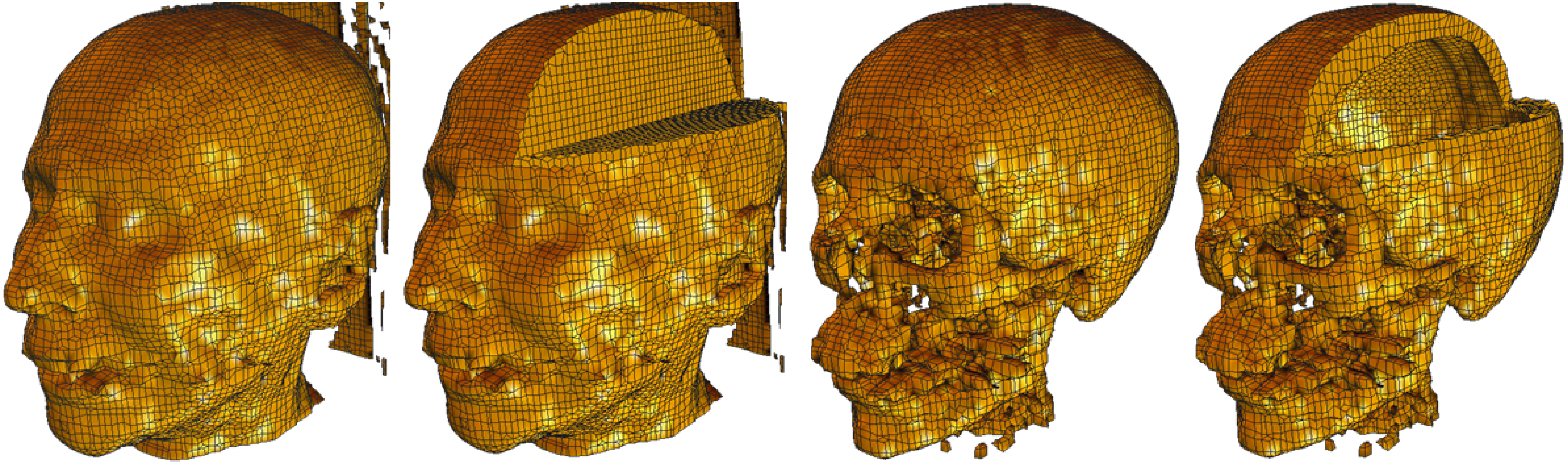


(a) (b) (c) (d) *Zhang et al. 2004*

Geometry From Imaging Data

Part II: Geometry Reconstruction

Computational Biomechanics

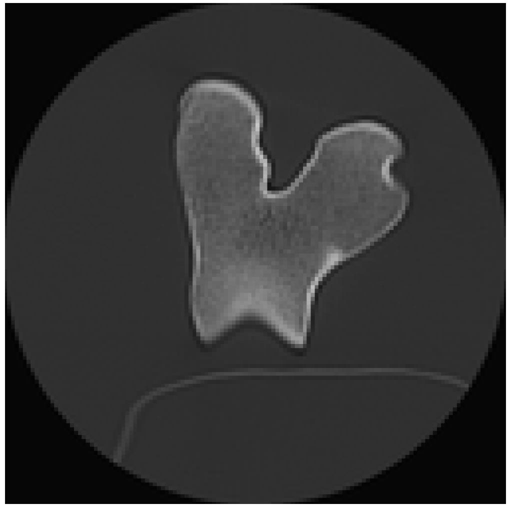
Summer Term 2016

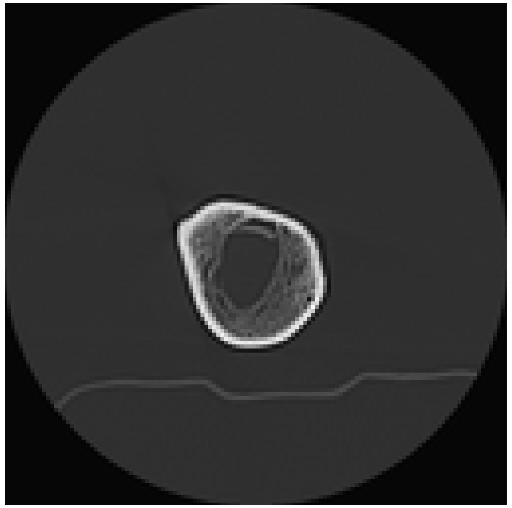
Lecture 5/12

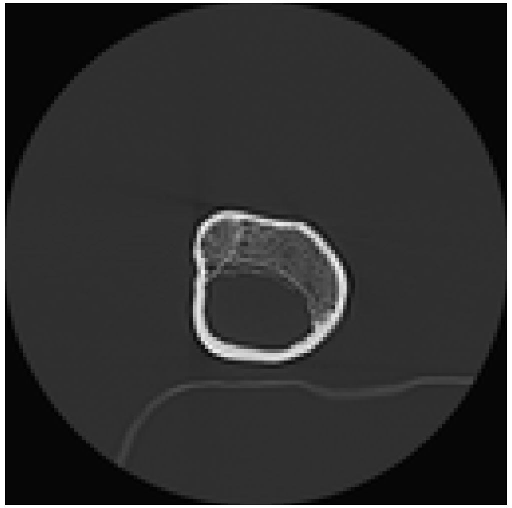
Frank Niemeyer

Starting Point

Note: intentionally reduced resolution

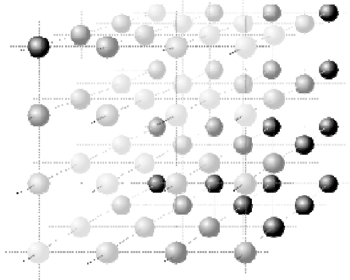




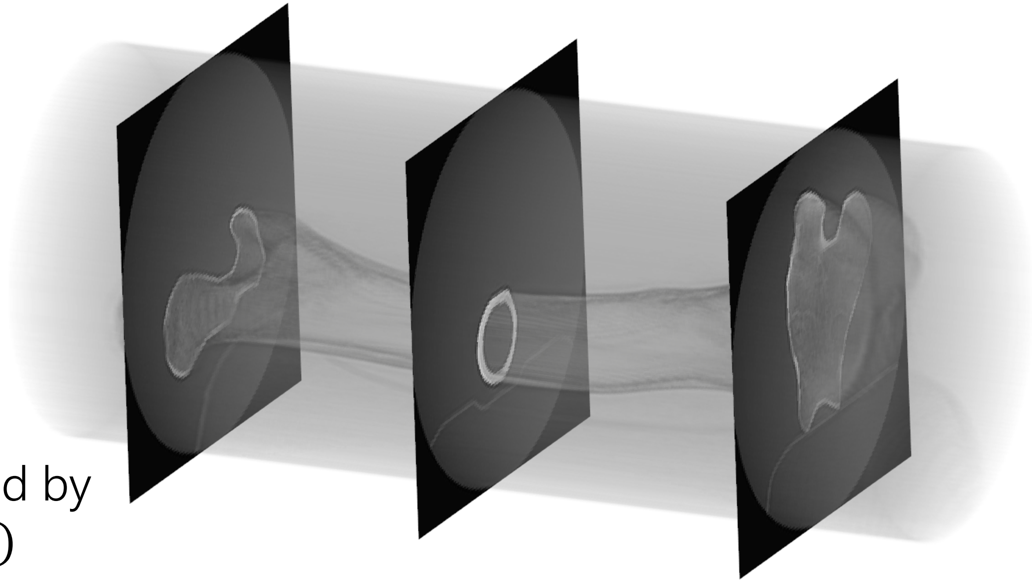


Images made of pixels (DICOM, PNG, …)

Stack



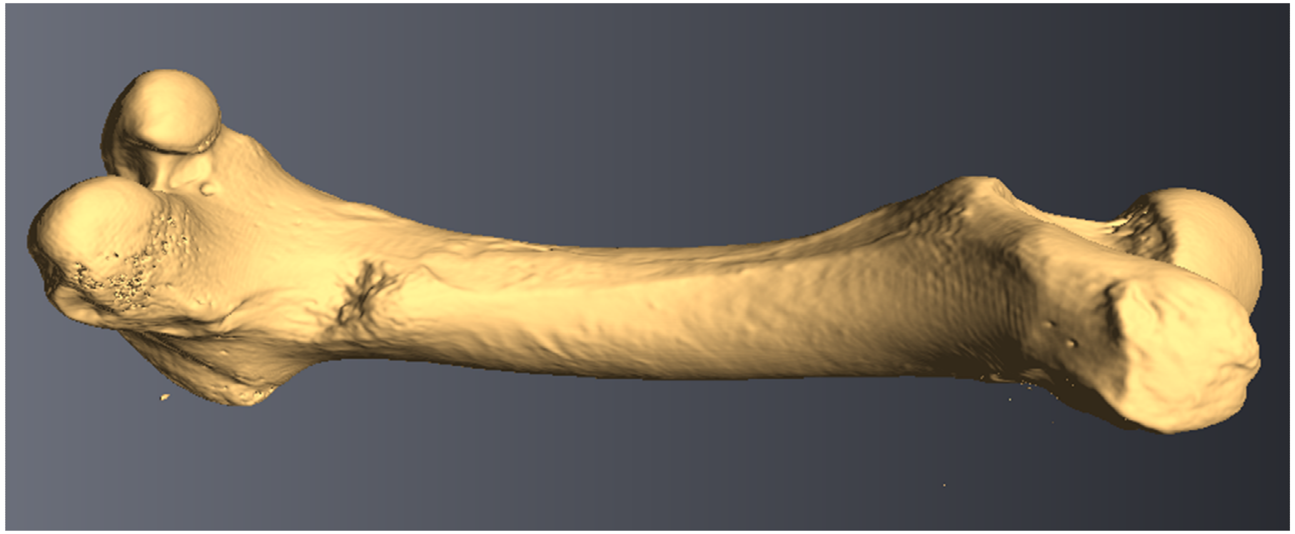
wikimedia.org



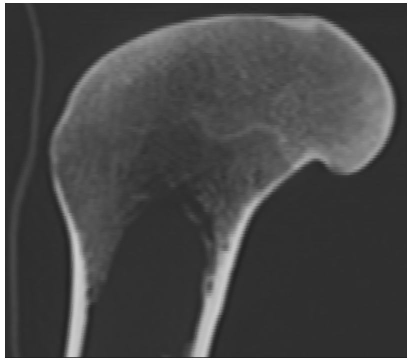
Volume defined by voxels $I(x, y, z)$

Image Processing

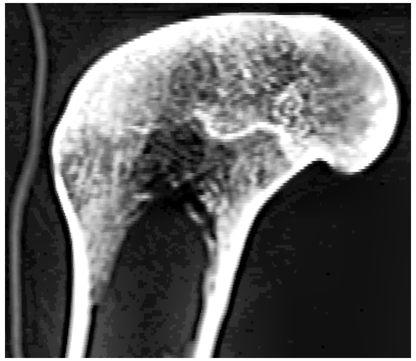
- Visualization
  - Multi-planar
  - Volume rendering
  - Iso-surface
- Cropping (ROI)
- Windowing
- De-noising
- Artifact removal
- Segmentation, labelling, classification:
  - Identify connected components
  - Locate object boundaries, interfaces
  - Manually or (semi-)automated, machine learning



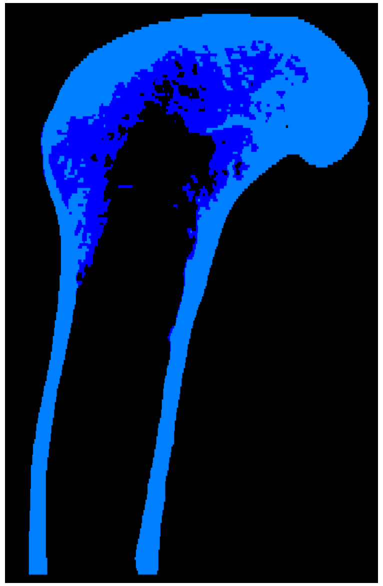
Iso-surface rendering of sheep femur



Original



Windowed, equalized

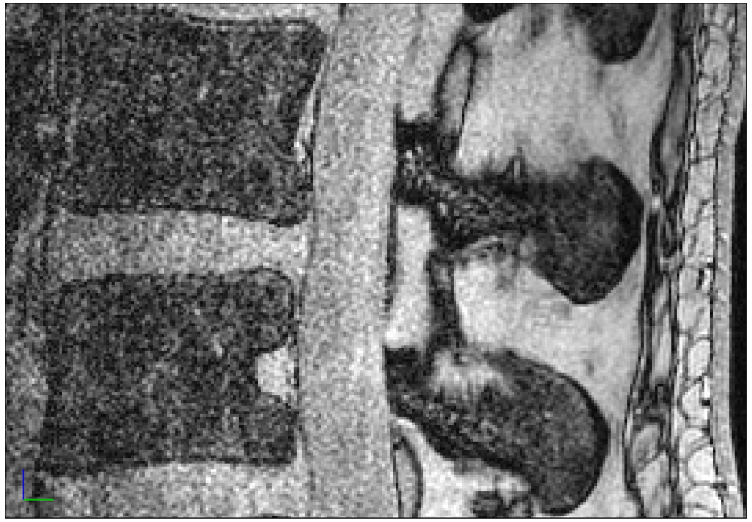


Segmented

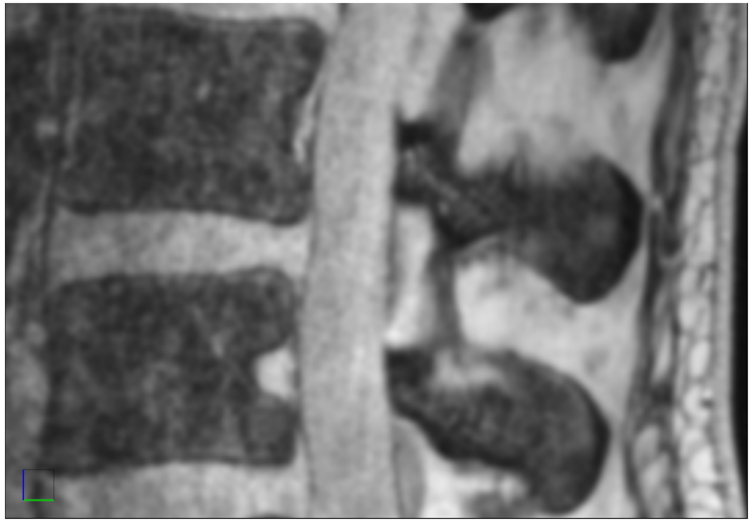
Linear (Low-Pass) Filtering

- Replace each pixel with weighted average of surrounding pixels/voxels (2D or 3D)
- Equivalent to suppressing high-frequency components as $I * K = \mathcal{F}(I) \cdot \mathcal{F}(K)$
- K: kernel (weights), e.g. Gaussian, Lanczos…
- Simple & fast
- Removes noise …
- … but also blurs edges

Original



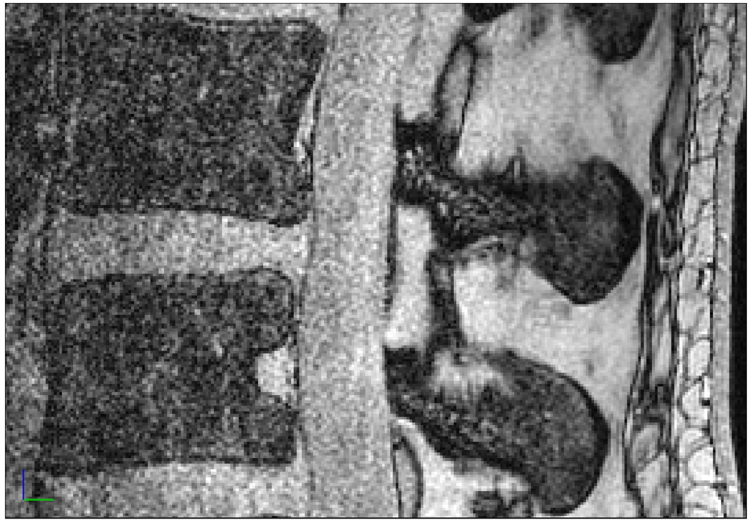
Gaussian filter



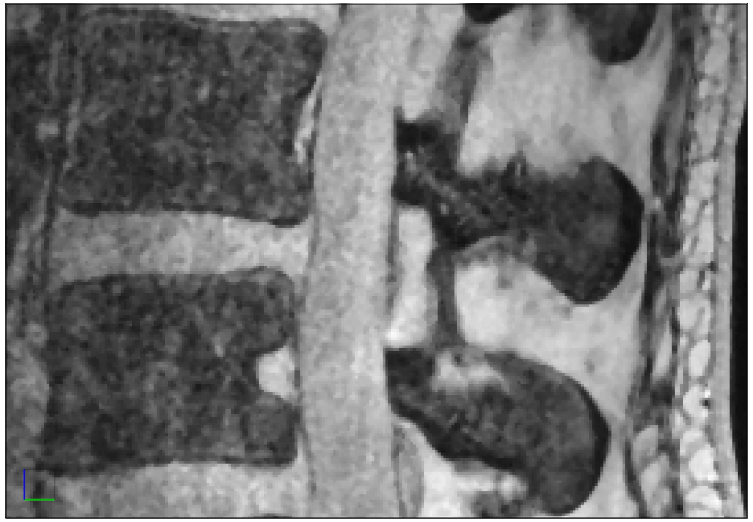
Rank Filters

- Replace each pixel with min/max/median (or any other “rank”) of adjacent pixels/voxels
- Arbitrary adjacency (box, diamond, cross, …)
- Median: “robust” estimator → particularly useful against impulse-like noise (speckle, salt & pepper)
- Better at preserving edges than Gaussian
- Relatively expensive (sorting!)

Original



Median filter



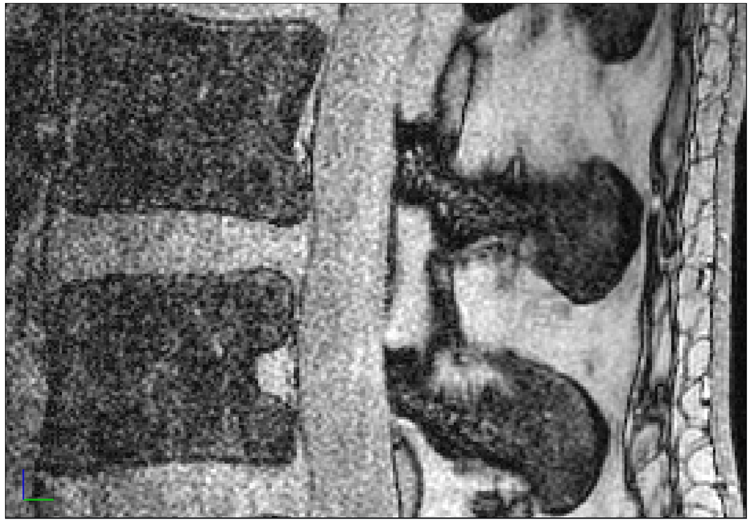
Bilateral Filter

- Replace each pixel with weighted average of surrounding pixels/voxels (2D or 3D)
  - Weights depend on distance (c.f. linear filter)
  - … but also on *similarity* to current pixel

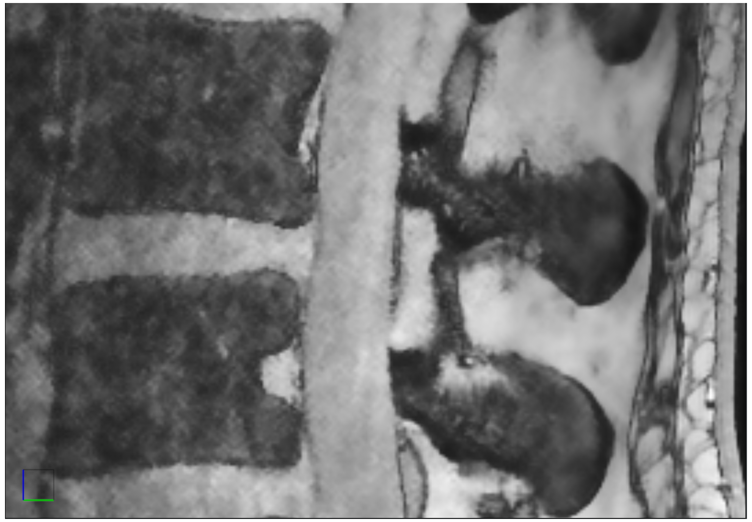
$$\mathcal{B}(I)(\boldsymbol{x}) \propto \sum_{\boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x})} I(\boldsymbol{x}_i)\, K_{\mathrm{r}}(|I(\boldsymbol{x}_i) - I(\boldsymbol{x})|)\, K_s(\|\boldsymbol{x}_i - \boldsymbol{x}\|)$$

  - K_{r}: range kernel, K_{s}: spatial kernel, $\mathcal{N}(\boldsymbol{x})$: neighborhood
- Preserves edges …
- … but not gradients ("gradient reversal")
- Introduces "staircase effect" (cartoon-look)

Original



Bilateral filter



Anisotropic Diffusion

- Observe: Convolution with Gaussian kernel G = solution of an isotropic, homogeneous diffusion equation

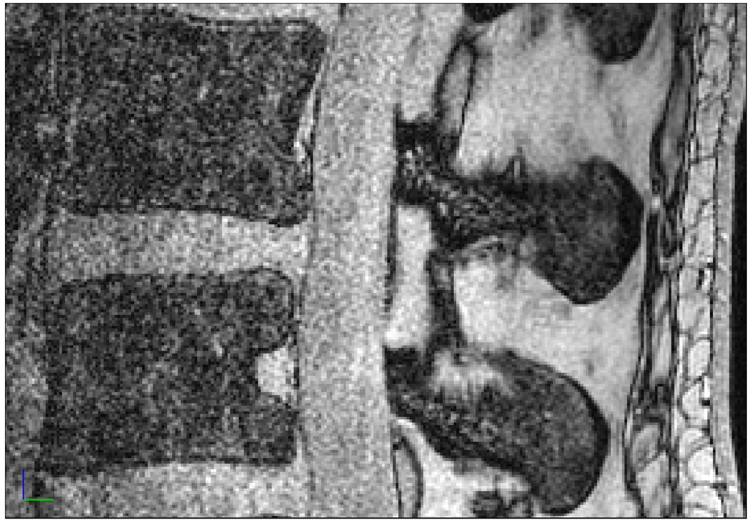
$$\partial_t I = D\nabla^2 I \Rightarrow I(t+\Delta t) = I(t) * G(D, \Delta t)$$

- Preserve edges by making the diffusion tensor D depend on the image

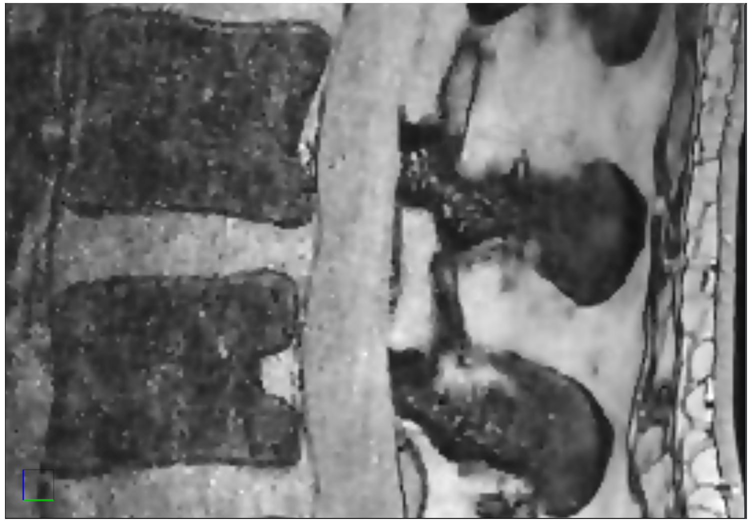
$$\partial_t I = \nabla \cdot (D(I)\nabla I)$$

- Expensive (non-linear PDE)
- Can even be edge-*enhancing* (a.k.a. coherence enhancing diffusion)

Original



Anisotropic diffusion
(actually probably Perona-Malik, i.e. inhomogeneous but isotropic)



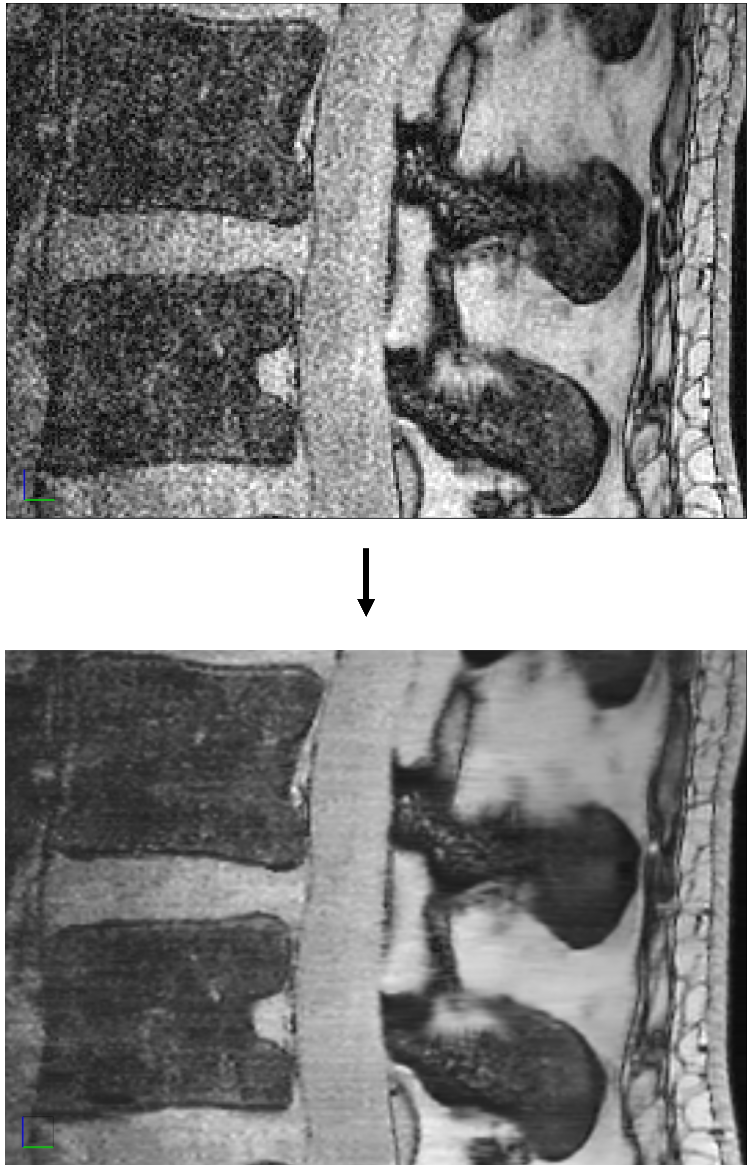
Non-Local Means

- Idea: Exploit self-similarity of images
- Replace each pixel with average of *all other pixels*, weighted by similarity of their neighborhoods

$$\mathcal{M}(I)(i) \propto \sum_j G(d_{ij}) I(j)$$

  - with $d_{ij} = \|\boldsymbol{n}_i - \boldsymbol{n}_j\|^2$
  - $\boldsymbol{n}_i$: gray values of neighborhood of pixel i
  - G: Gaussian (e.g.)
- Less loss of detail than local methods
- Incredibly expensive → GPGPU
- Assumes white noise

Original



Windowed
non-local means
(only in transversal plane)

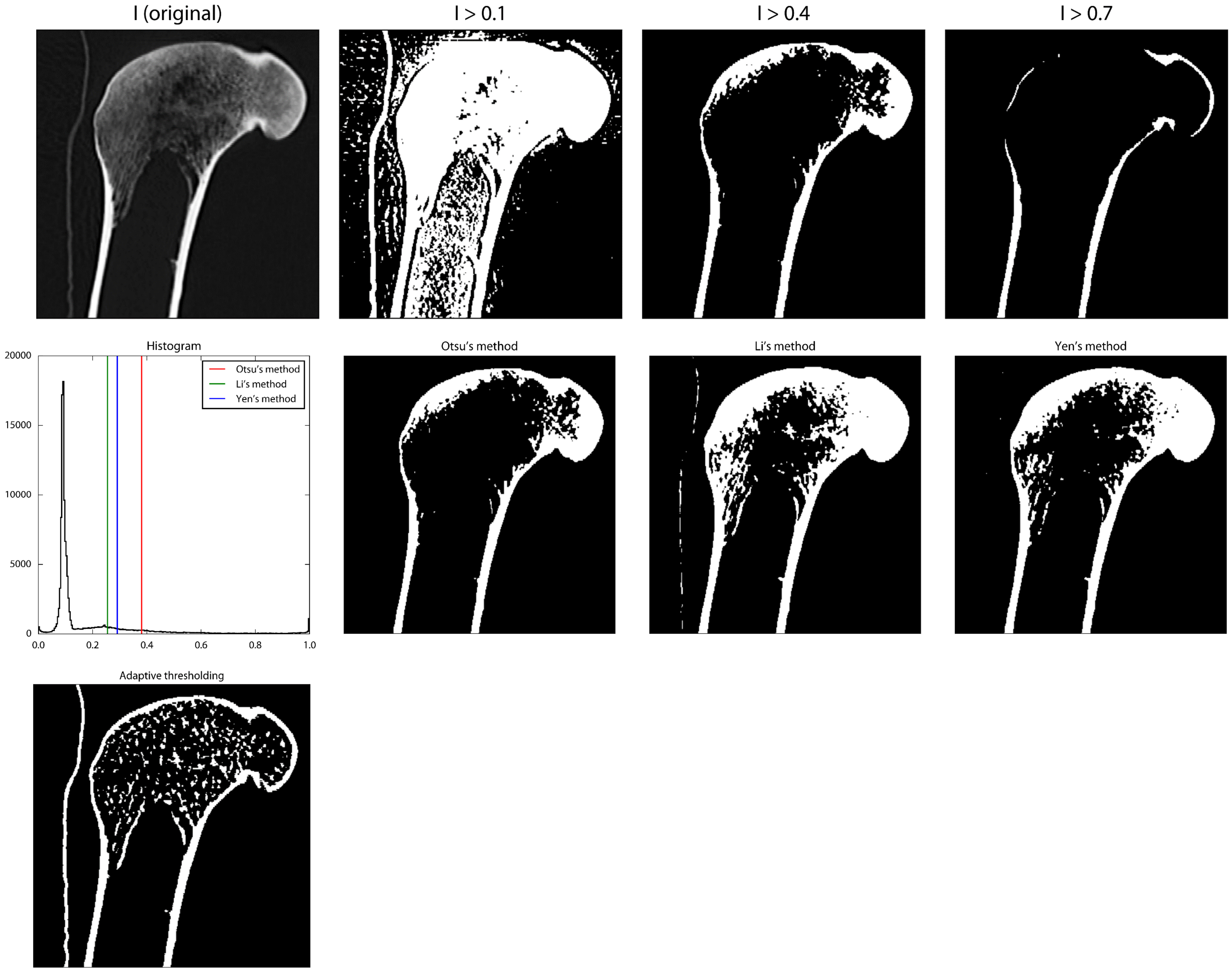
Segmentation Techniques

- Goal: Partition image according to some *similarity measure* into connected subsets of pixels (segments, regions) I_R. Examples for such measures:
  - $\max I_R - \min I_R$
  - $\max(|\bar{I}_R - \max I_R|, |\bar{I}_R - \min I_R|)$
  - $\max\|\nabla I_R\| - \min\|\nabla I_R\|$
  - Others: Texture variance, entropy, energy, statistics of derivatives …

- Pixel-based: Thresholding, clustering
- Region-based: Region growing, split & merge, watershed transform, texture segmentation
- Edge-based: Edge detection and linking
- Model-based: Template matching, active contours, level set, ANN …

Thresholding

- Manually pick a global gray value threshold
- Histogram-based, automatic (Otsu's, Li's, Yen's methods)
- Adaptive (local) thresholding based on local neighborhood

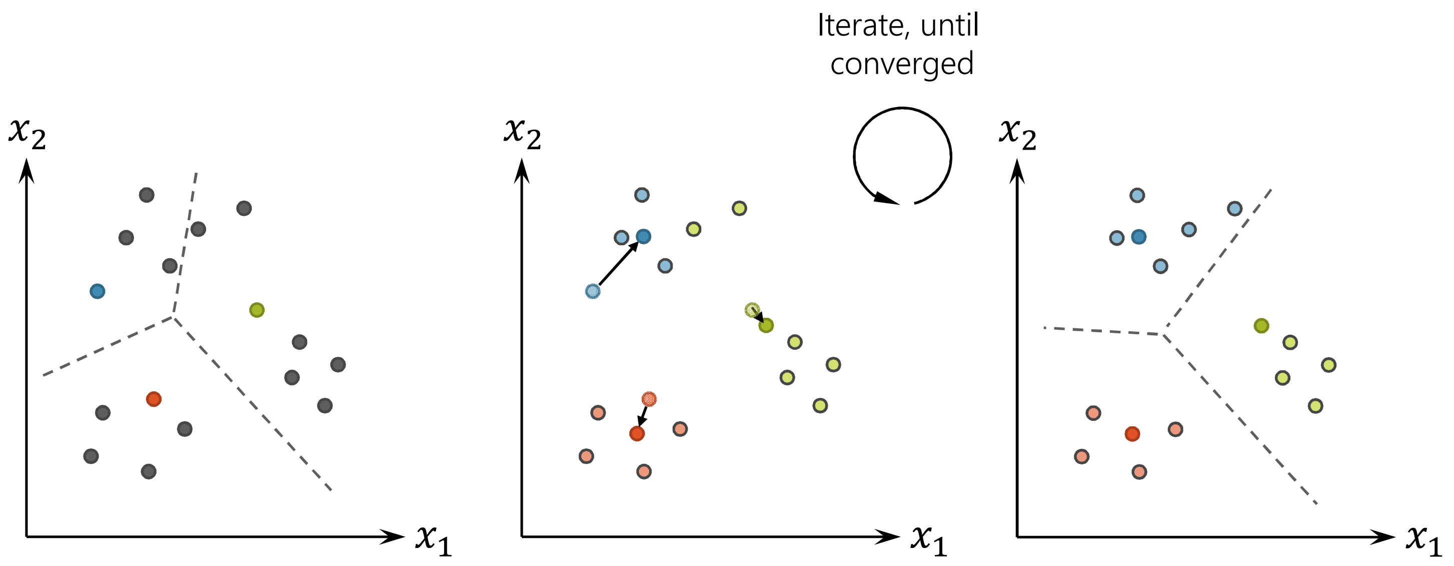


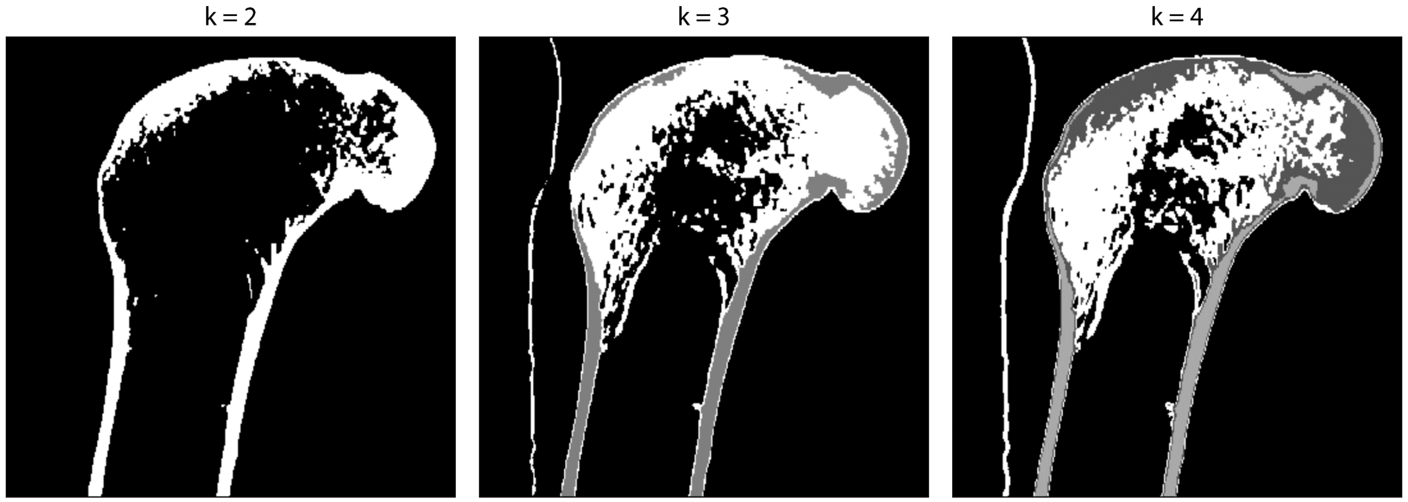
Clustering

- Assumptions: Segments share similar features and are thus clustered in “feature space” → cluster analysis
- k-Means: partition pixels into a set of k Clusters C with means μ_i, such that

$$\sum_{i=1}^{k} \sum_{x \in C_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2 \to \min$$

  - Assumes convex & isotropic clusters of similar size
  - Number of clusters?
  - Sensible metric?

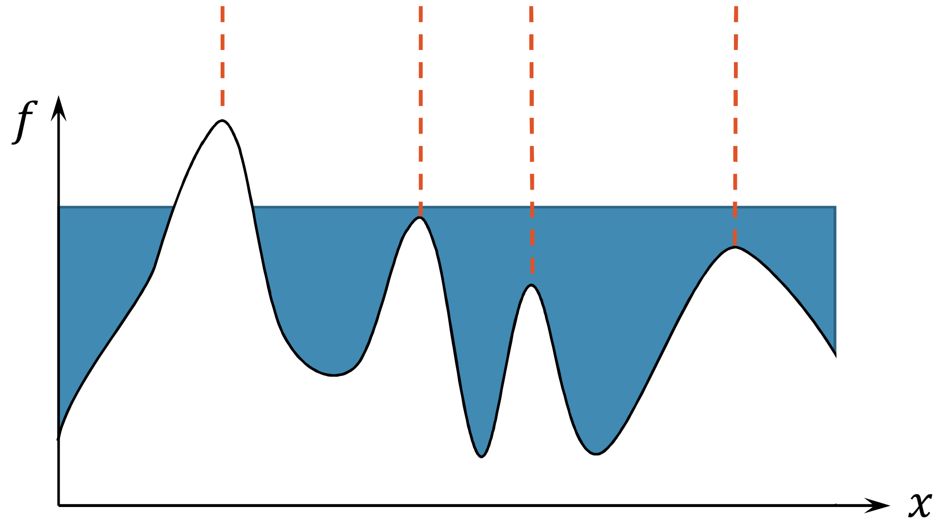




Using scikit-learn, `sklearn.clusters.KMeans`, 1D feature space (gray value)

Watershed Transform

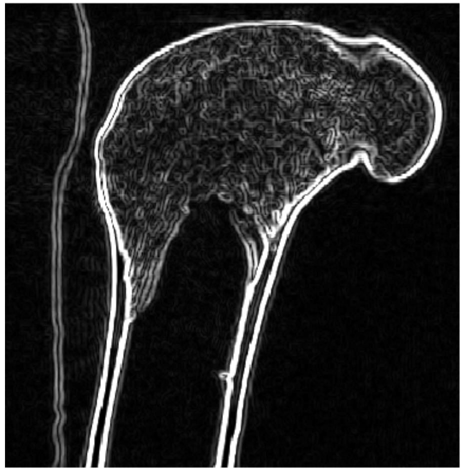
- Intuition:
  - Interpret f (some function of the image I, e.g. $\|\nabla I\|_2$) as height map
  - “Hole” at each local minimum of f
  - Immerse landscape in water → “water reservoirs” (regions)
  - To avoid merging of adjacent regions → build “dam” (contours)
- f must be constructed such that “valleys” = objects to segment
- Pros: intuitive, always creates closed contours
- Cons: susceptible to noise, over-segmentation

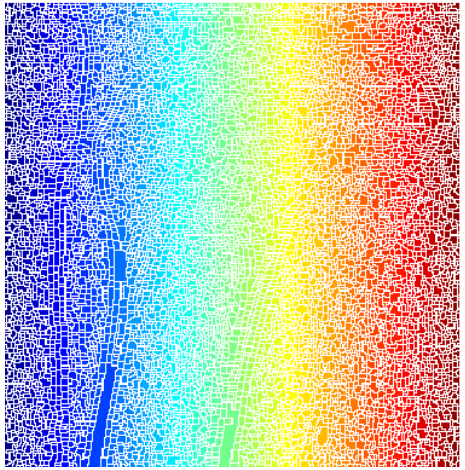


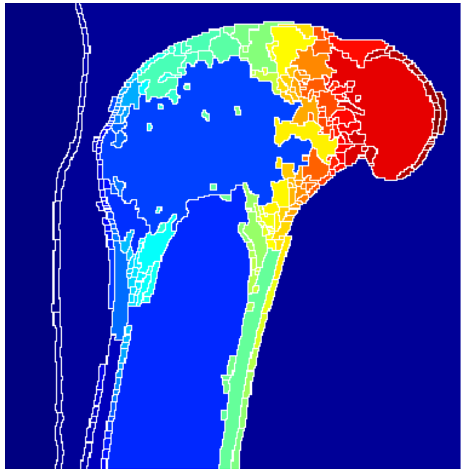
Watershed Transform

Classic

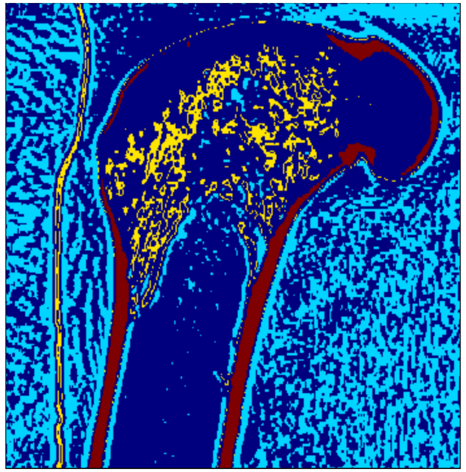

Original image


Gradient magnitude (Sobel)


Over-segmented image


H-min transformed segmented image

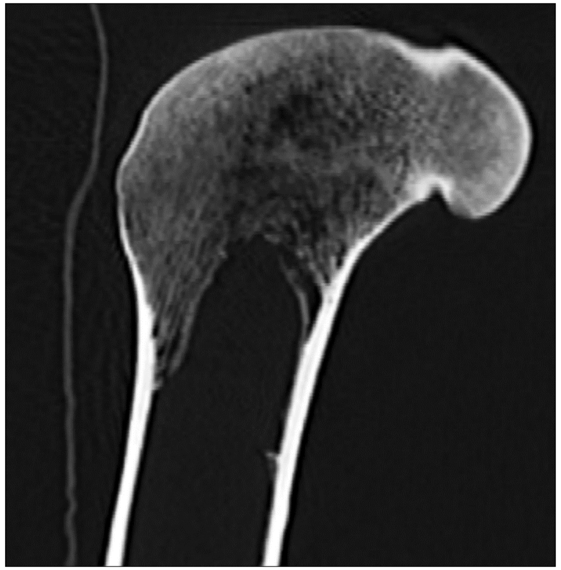
With markers


Marker pixels

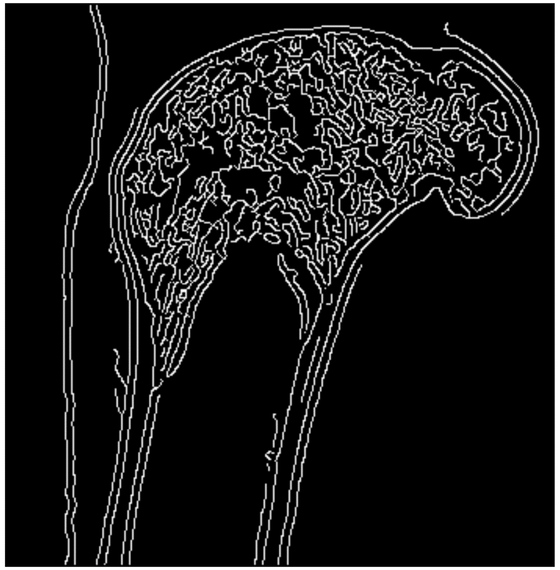

Segmented image

Canny Edge Detection

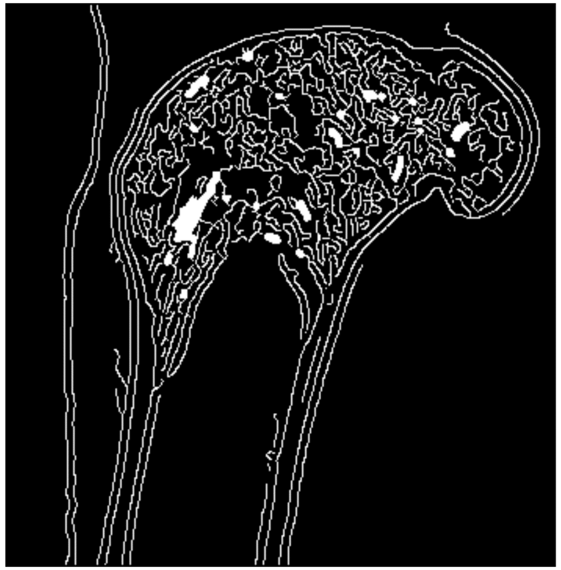
- Basic algorithm:
  - Smooth image with Gaussian
  - Compute image gradient (Sobel)
  - Edge thinning via non-maximum suppression
  - Hysteresis thresholding to suppress weak/unconnected edge pixels



Original image



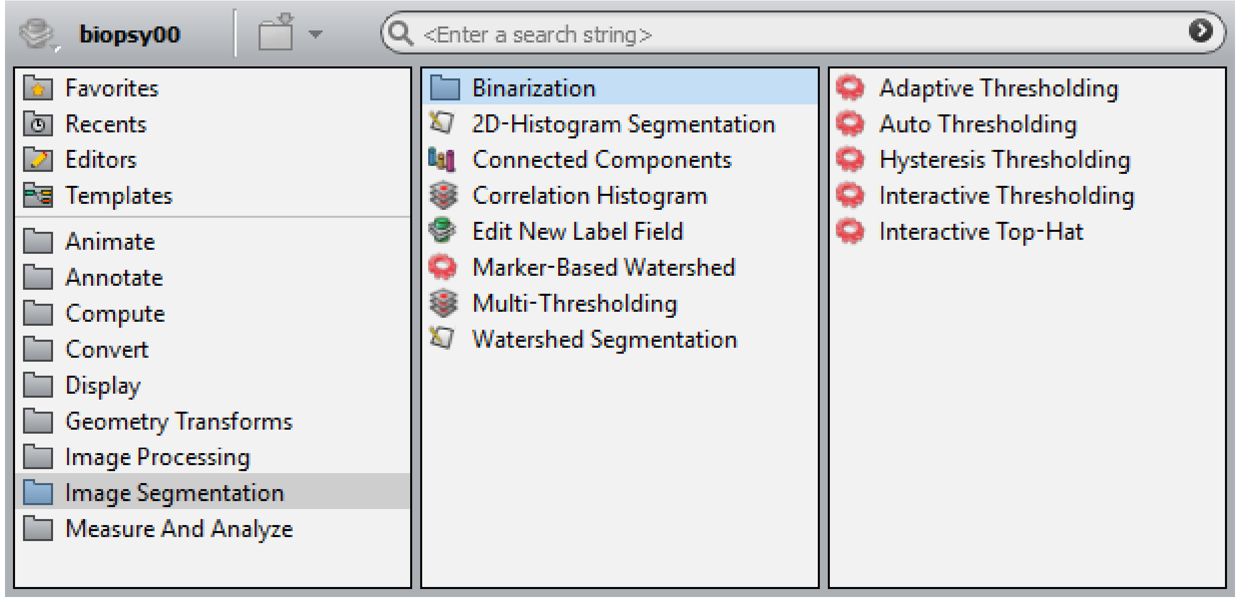
Detected edges



Filled closed regions

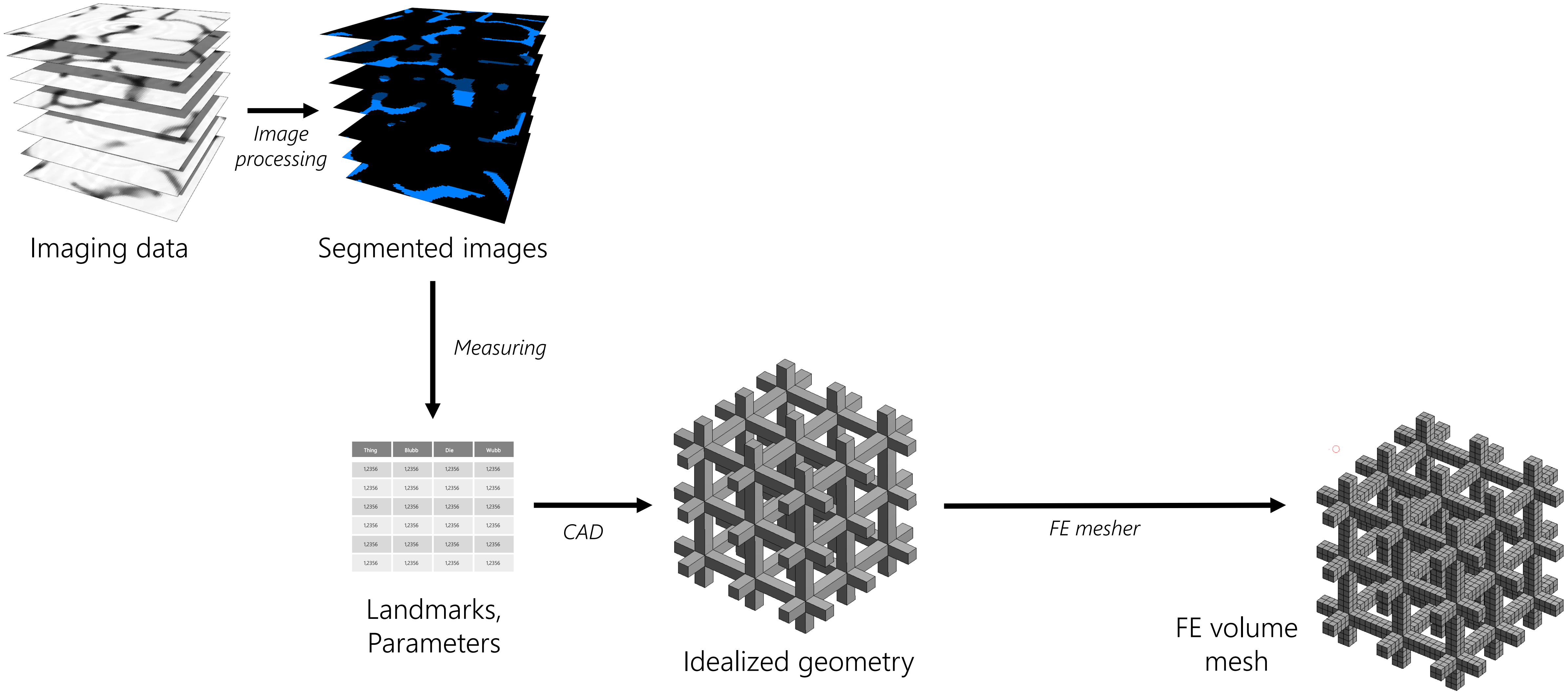
Avizo Segmentation Tools

- Automatic, e.g.
    - (Multi-)Thresholding
    - Joint histogram
    - (Marker-based) watershed
- Interactive, semi-automatic ("Segmentation Editor")
    - Brush
    - Magic Wand (region growing)
    - Propagating Contour (active contour)
    - Watershed



- "Note that even with the advanced tools provided in Avizo, image segmentation can be a time-consuming process! Due to limited main-memory and for performance reasons, there is only a limited undo space for 2D and 3D interaction. Therefore it is highly recommended to *frequently save the label field* during the process of segmentation." (Avizo Manual)

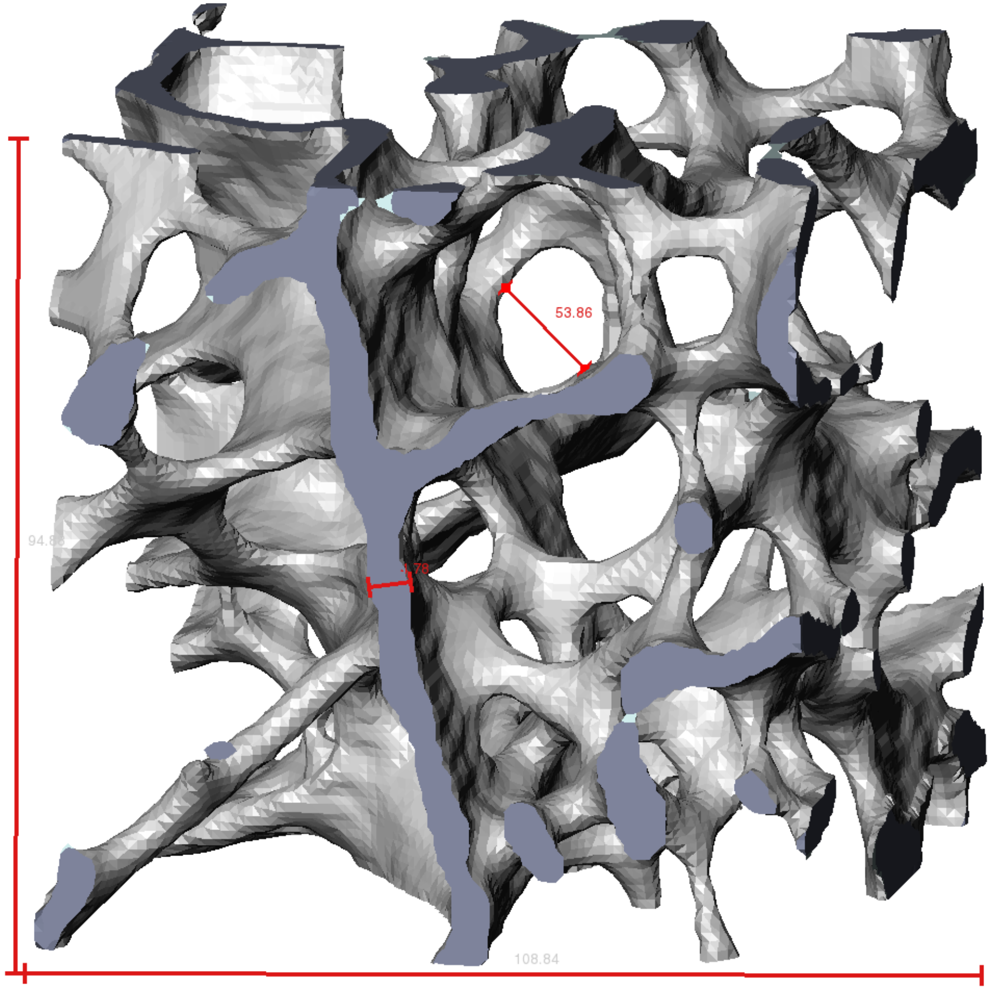
Approaches


Imaging data
Image processing
Segmented images
Measuring
Thing Blubb Die Wubb
1,2356 1,2356 1,2356 1,2356
1,2356 1,2356 1,2356 1,2356
1,2356 1,2356 1,2356 1,2356
1,2356 1,2356 1,2356 1,2356
1,2356 1,2356 1,2356 1,2356
1,2356 1,2356 1,2356 1,2356
Landmarks, Parameters
CAD
Idealized geometry
FE mesher
FE volume mesh

"Bottom-Up" (Solid Modeling)

1. Identify landmarks
2. Derive geometry parameters
3. Generate solid geometry
   - Points → Lines → Faces → Volumes
   - Boolean operations (CSG)
4. Mesh geometry

- Pros & Cons
  - Flexible: parametric, generic
  - Doesn't necessarily require full volumetric data (but prior knowledge)
  - Simplified, idealized anatomy
  - May be tedious to create (depending on level of detail)

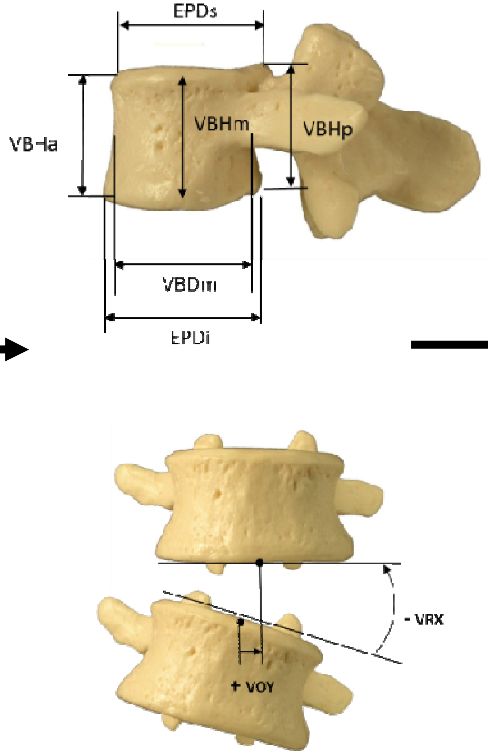


Geometry Reconstruction | Parametric

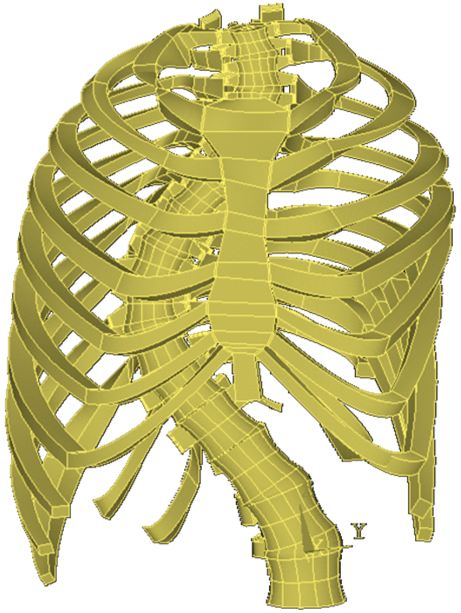
(Semi-Automated) Bottom-Up

Imaging data

Landmarks



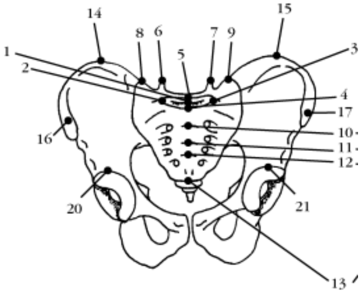
Parameters

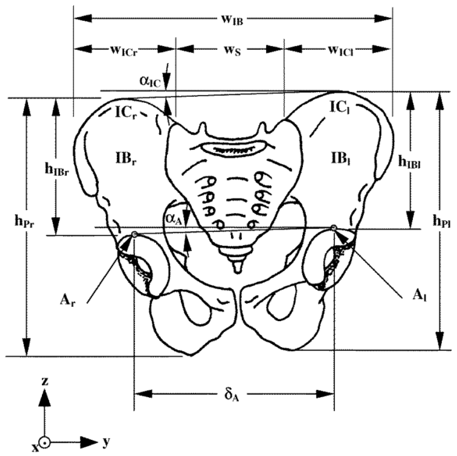


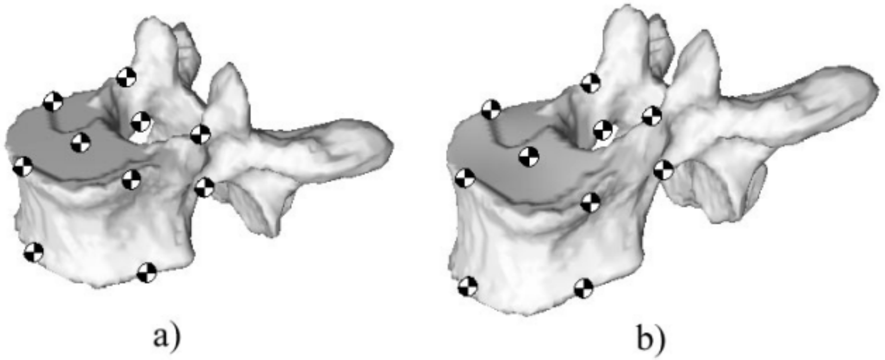
Solid geometry

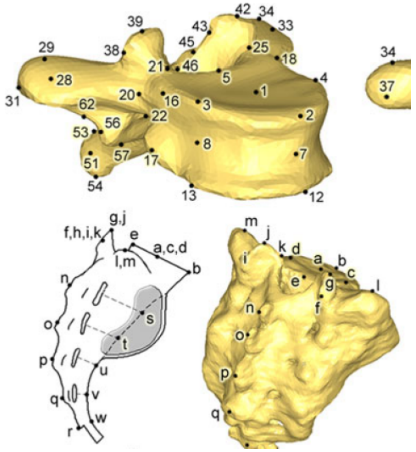
Thanks to Benedikt Schlager

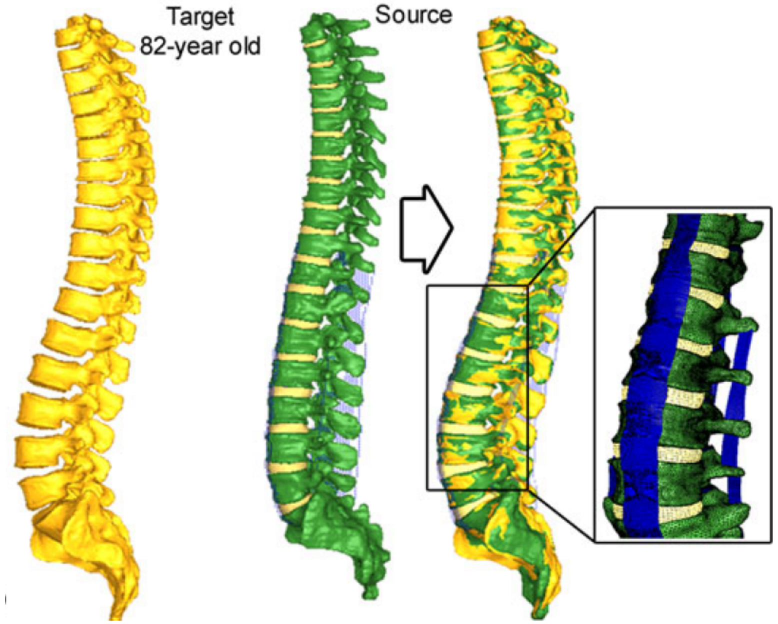
Template Morphing









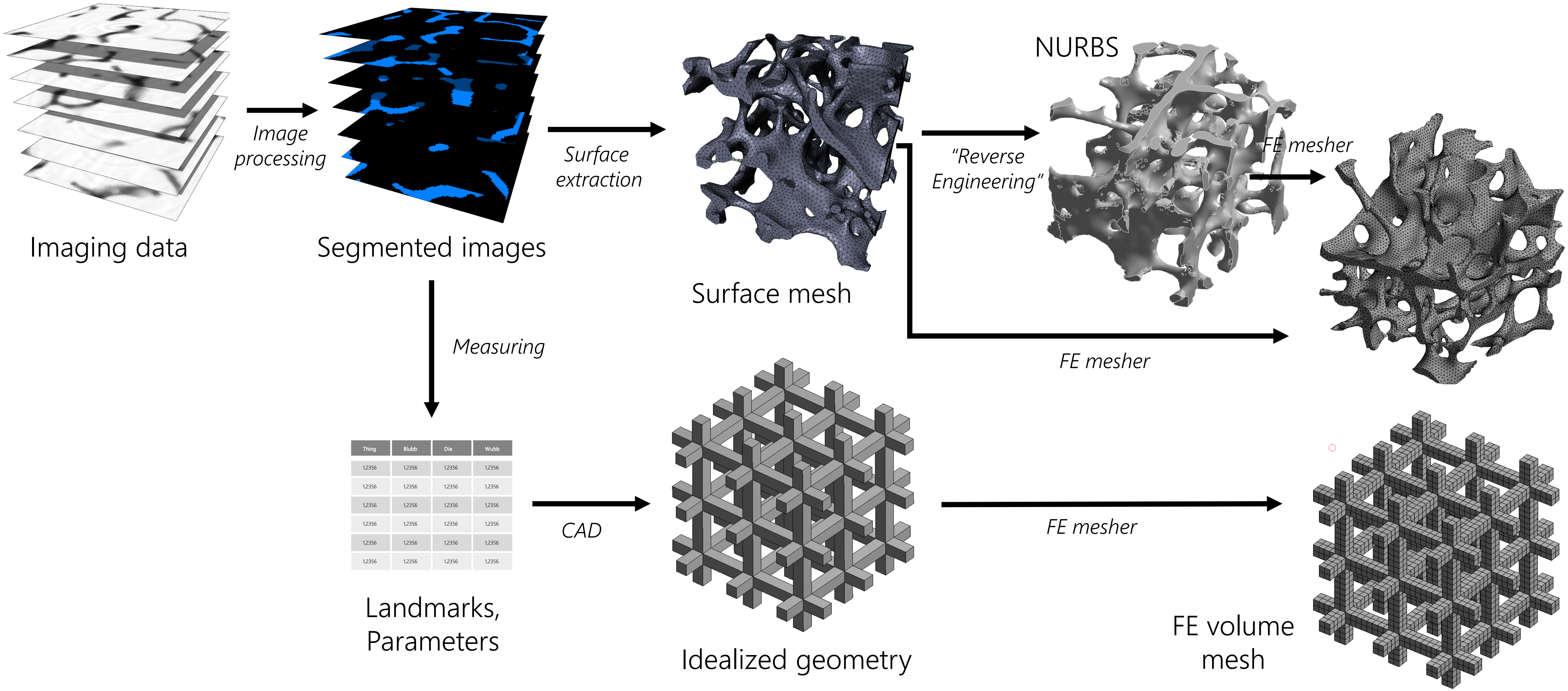


Delorme et al. 2003

Lalonde et al. 2013.

Geometry Reconstruction

Approaches

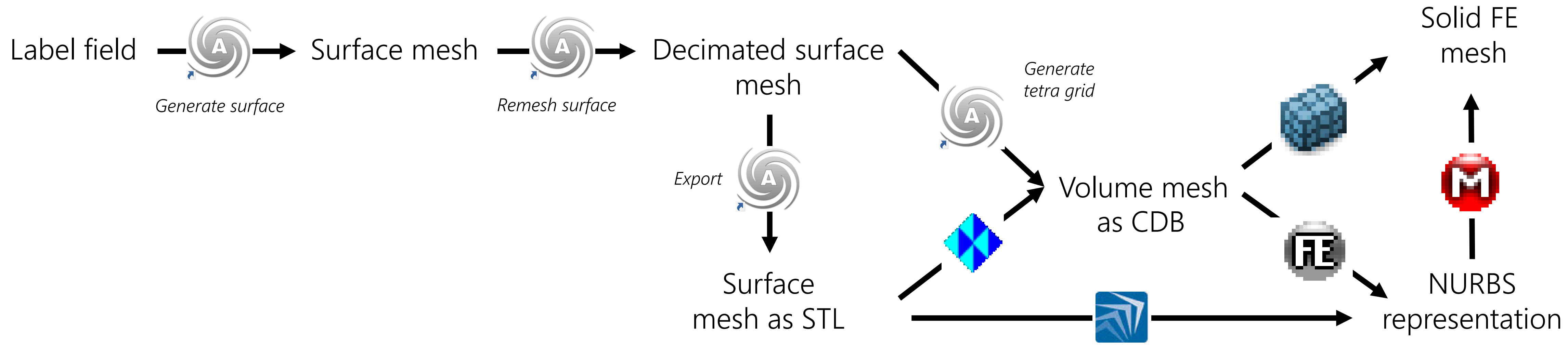


Geometry Reconstruction | Non-Parametric | Surface-based

"Avizo-to-ANSYS" Workflow(s)

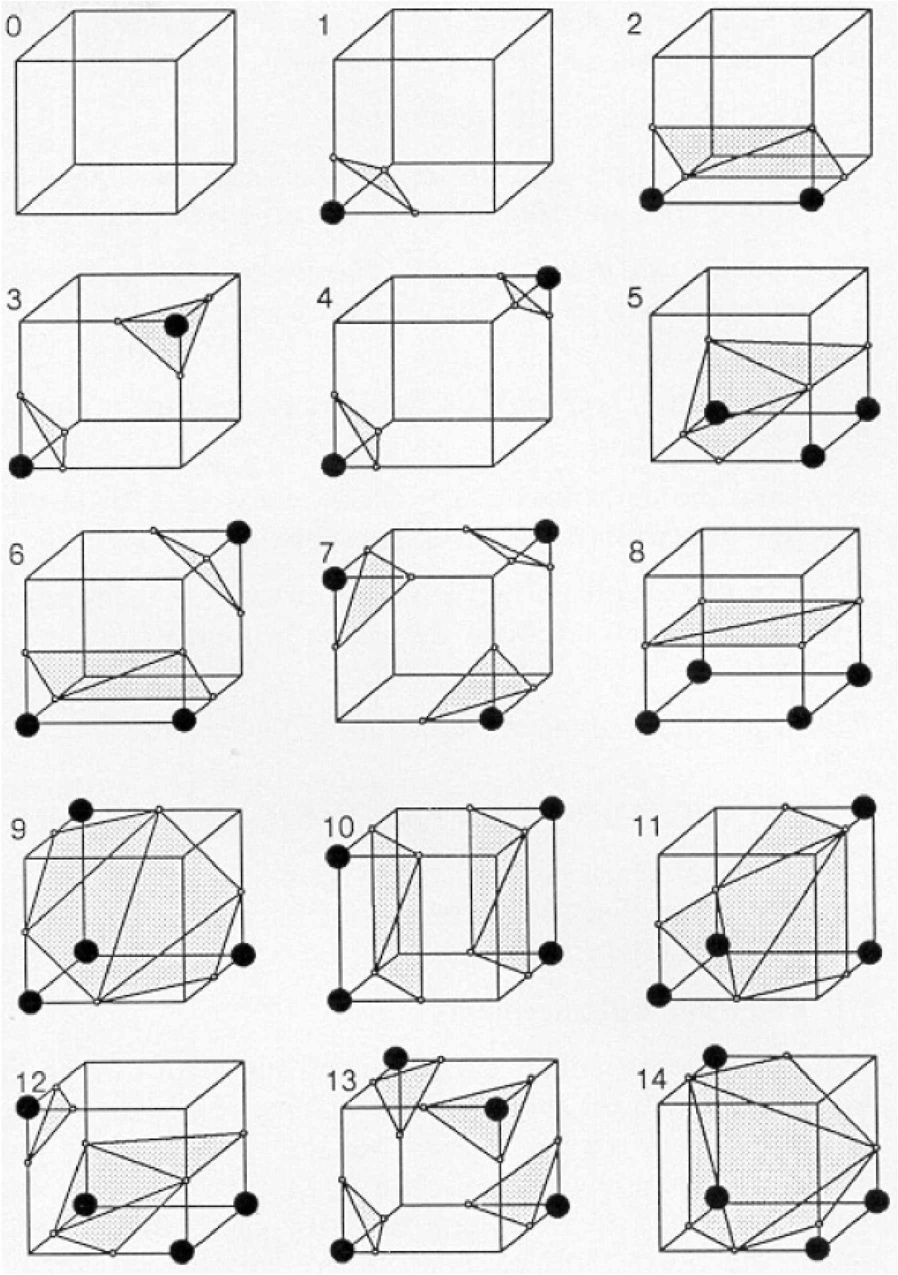
- Segmentation → label field
- Generate surface mesh → triangulated surface
- Remesh surface → decimated mesh
- Create volume mesh → tetrahedral grid
- Either use that directly ("External model"), *or*
- Try to reverse engineer NURBS representation

Avizo
ICEM CFD
Finite Element Modeler
Mechanical
External model
SpaceClaim



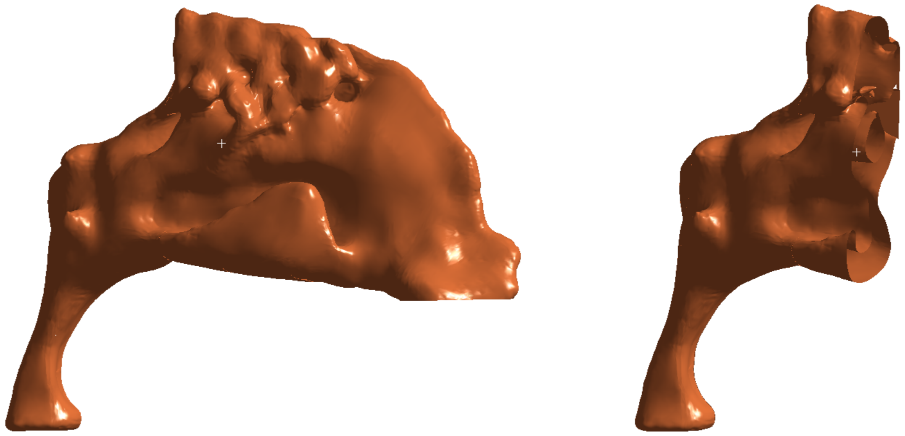
Marching Cubes

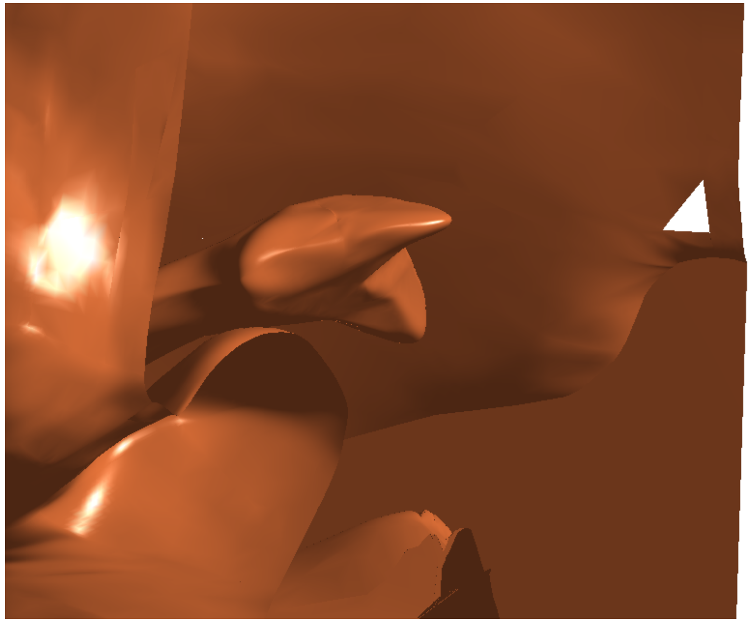
- Iterate over voxel grid à cubes of 8 voxels each
- Choose triangulation depending on voxel values:
  - All values above/below threshold: no surface
  - Otherwise: look-up table → triangulation
- Interpolate between voxels to estimate exact surface-edge-intersection
- Results in triangle mesh
  - Facetted, piecewise-linear, non-smooth (C^0 cont.)
- Sensitive to noise
- Often requires repairing if used in non-visualization contexts

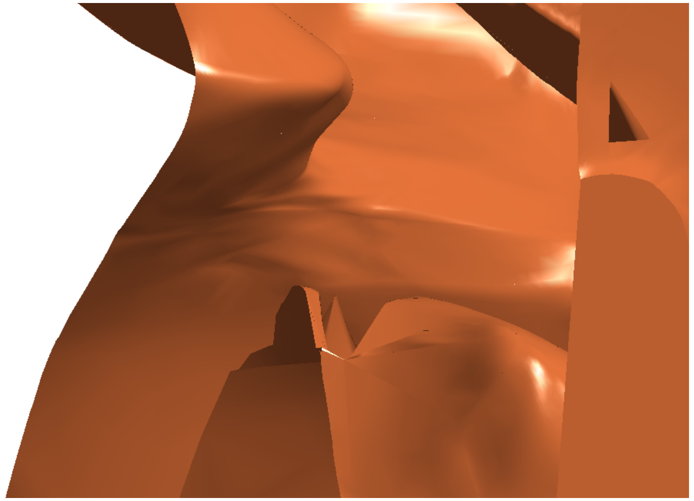


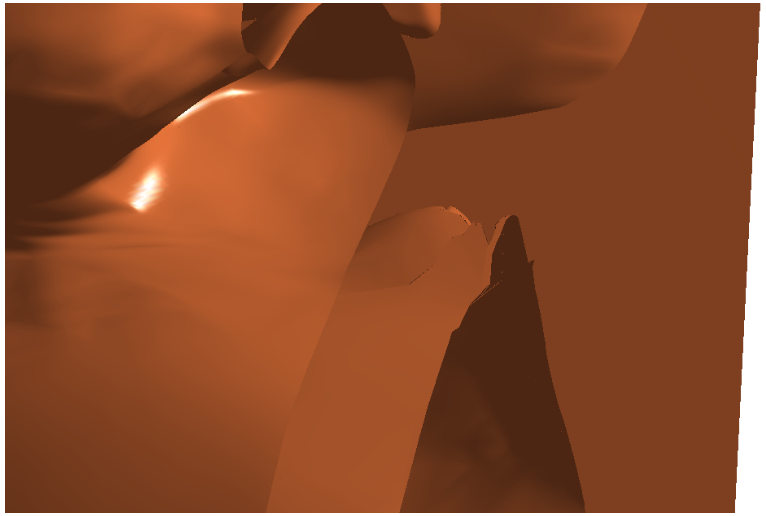
Lorensen & Cline, 1987

Marching Cubes





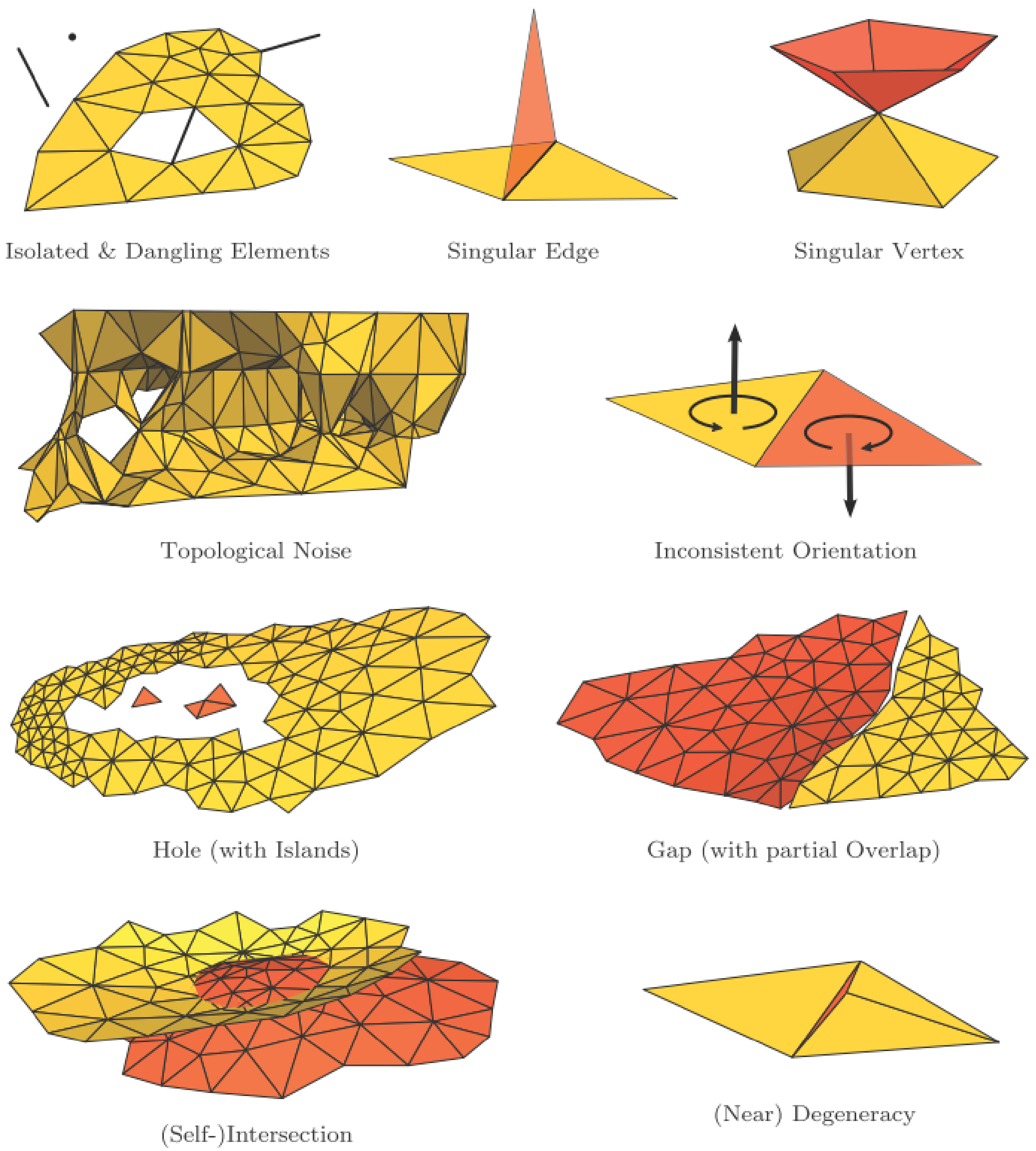




Thanks to Lucas Engelhardt

Mesh Repairing

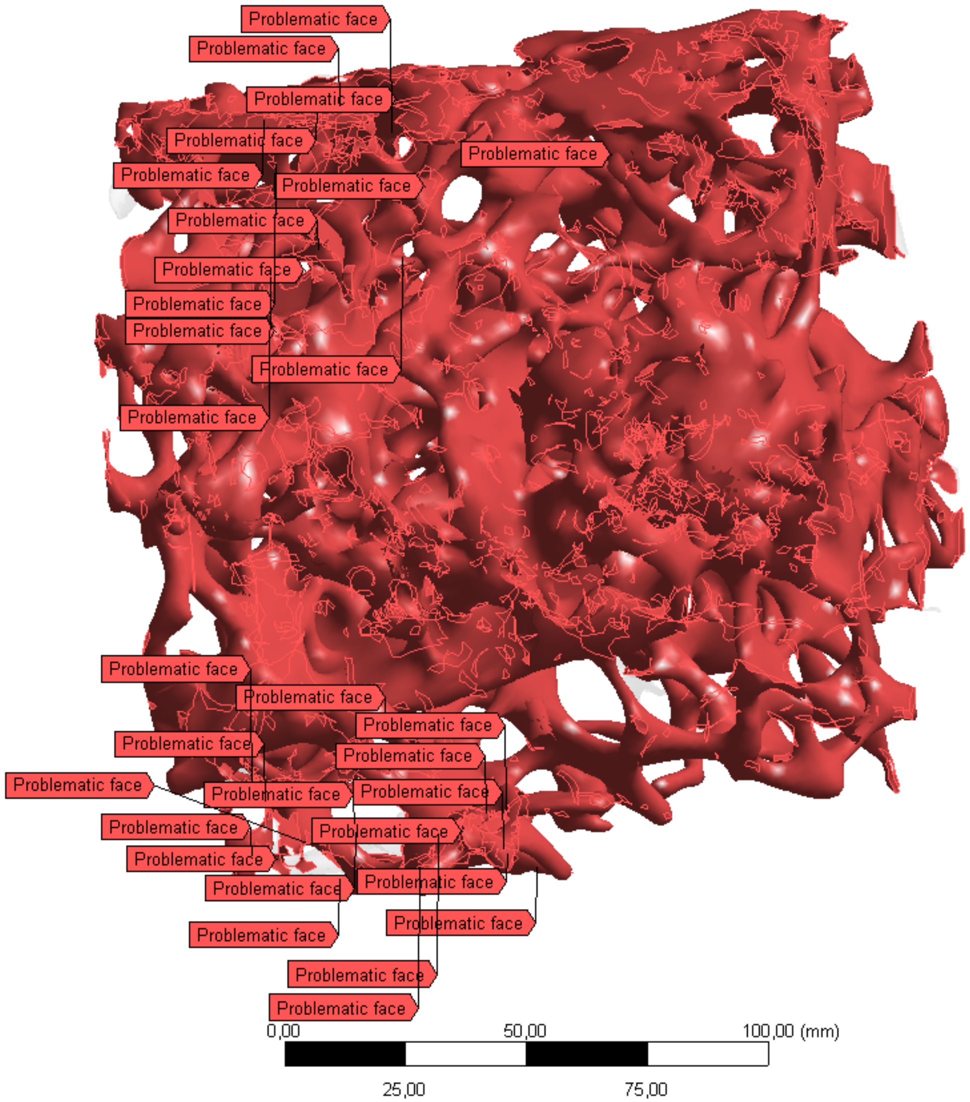
- Fill holes and self-intersections
- Fix singular edges, vertices
- Remove non-manifold edges
- Remove unconnected components
- Fix inconsistent normal direction
- Smoothing
- Software: e.g. Avizo's surface editor, MeshLab, Meshmixer, netfabb, Blender, SpaceClaim…
  - c.f. http://meshrepair.org



from Attene et al., 2013

NURBS Reconstruction

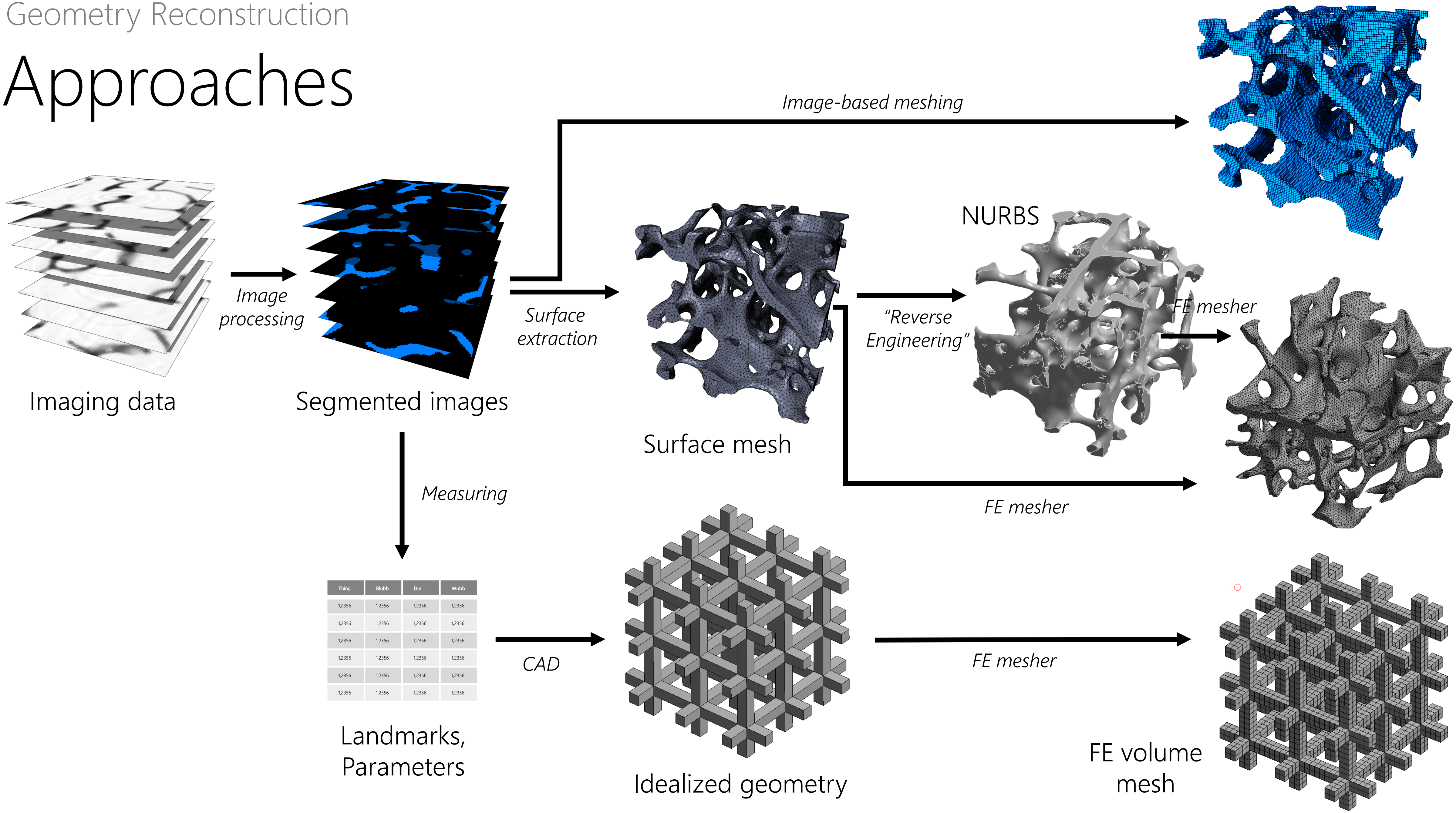
- Further processing often requires "CAD" geometry (analytical, compact description)
- *Either:* Convert facetted surface mesh to NURBS representation, e.g.
  - Finite Element Modeler (ANSYS Workbench)
  - Rhino3D
  - SpaceClaim
  - SolidWorks (treat STL as complicated CAD geometry)
- *Or:* Use special software to work directly with STL, e.g.
  - Materialise Mimics + 3-matic
  - MeshLab, Meshmixer, SpaceClaim …



FE Modeler in action

Geometry Reconstruction

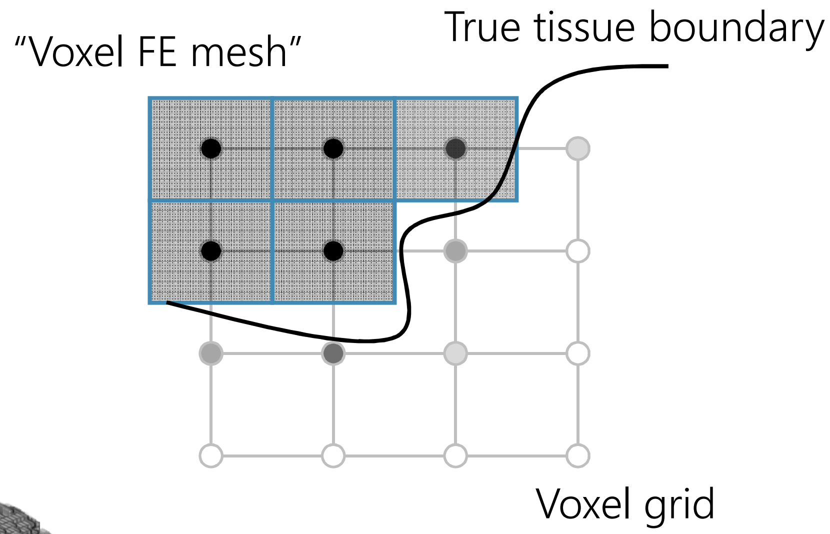
Approaches

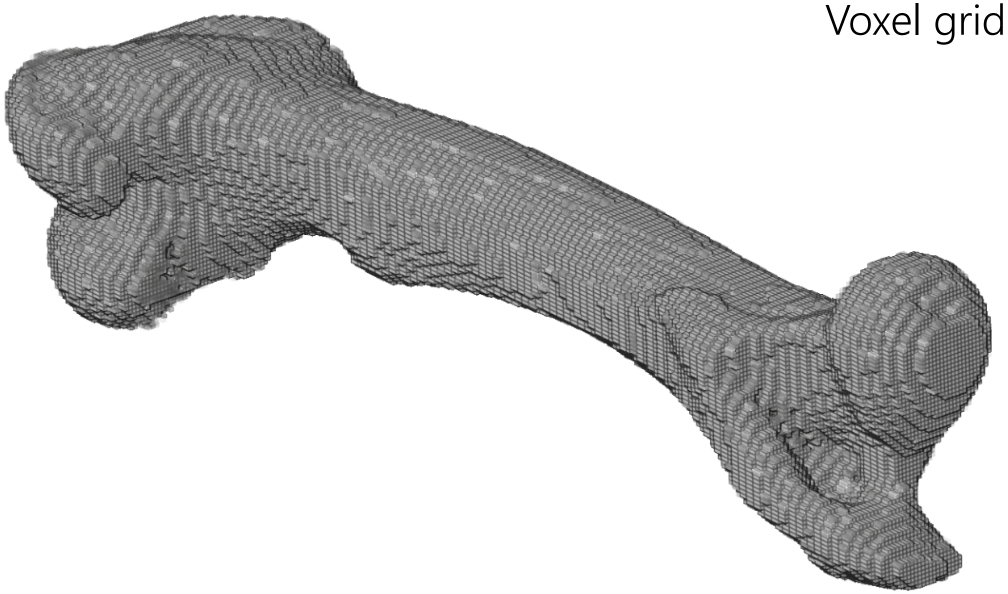


Thing	Blubb	Die	Wubb
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356

"Voxel Method"

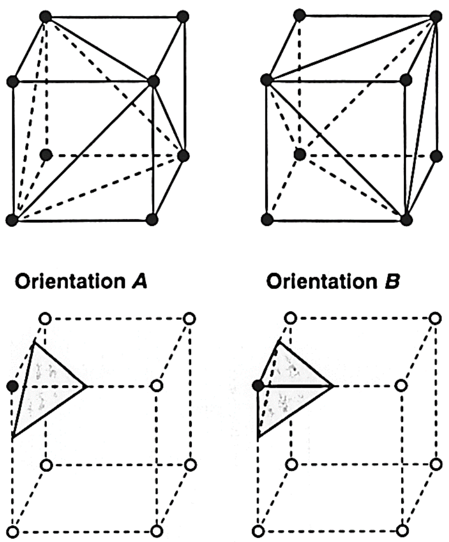
- Trivial approach" (Keyak et al. 1990)
  - Directly convert voxel grid to FE mesh
  - 1 voxel → 1 hexahedron (i.e. nearest neighbor interpolation, uniform spatial discretization)
- No prior (explicit) surface reconstruction
- Robust, arbitrarily complex topologies
- Automatically conforming interfaces of parts
- Optimal element quality
- Many DOFs
- Poor surface reconstruction, singularities
- Neither surface nor volume preserving



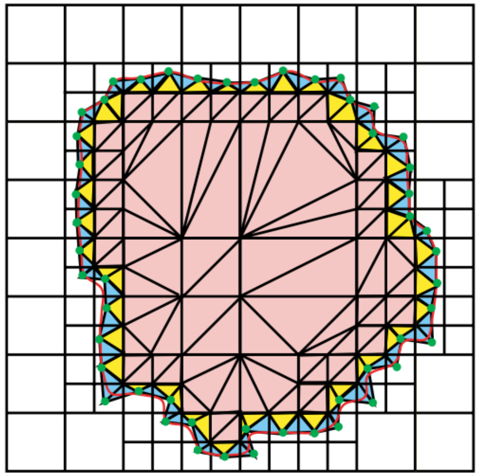


Advanced Algorithms

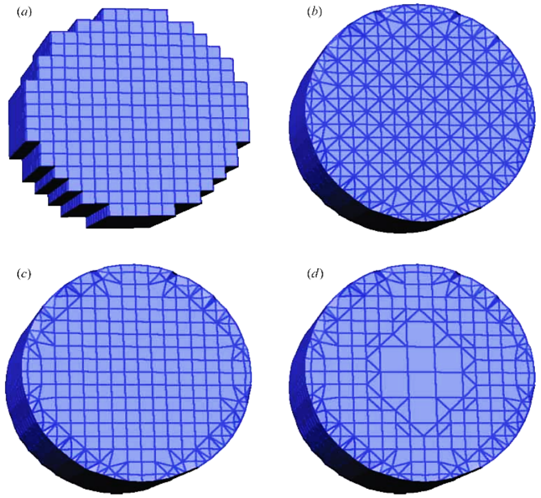
- Try to detect material boundaries and align generated element edges accordingly
- VoMaC (Müller & Rüegsegger 1995):
  - Create tetrahedrons instead of triangles
  - Inner "cubes" → five tetrahedra
- Zhang et al. 2005:
  - Adaptive meshing of inner cells
  - Hexahedral mesh generation
- EVoMaC (Young et al. 2008):
  - Multi-part meshing with conforming interfaces
  - Octree-based mesh decimation
- Software: SimpleWare +ScanFE



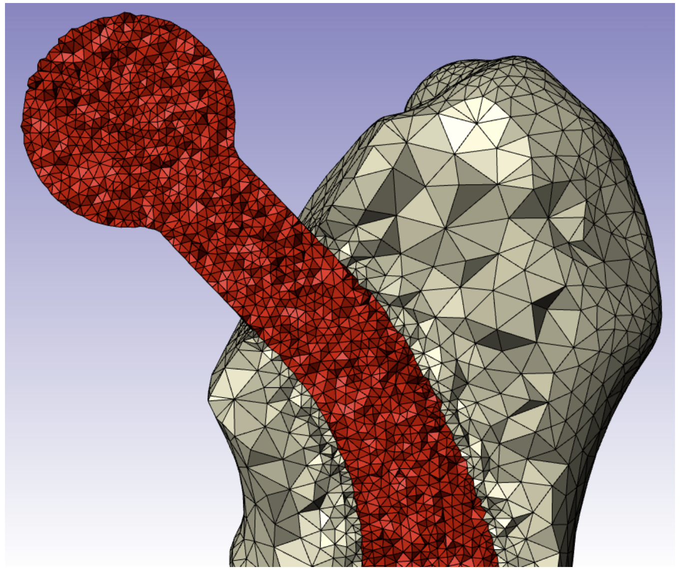
Müller & Rüegsegger, 1995



Zhang et al., 2005

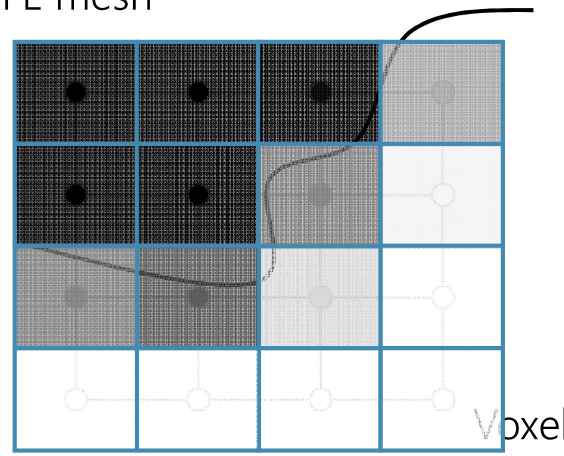


Young et al., 2008



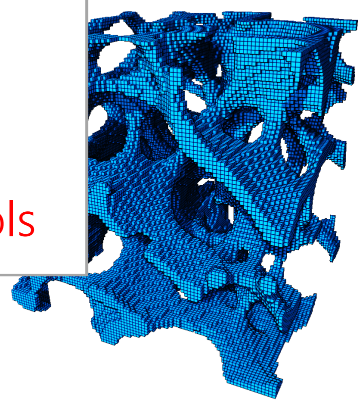
Material Properties from Gray Values

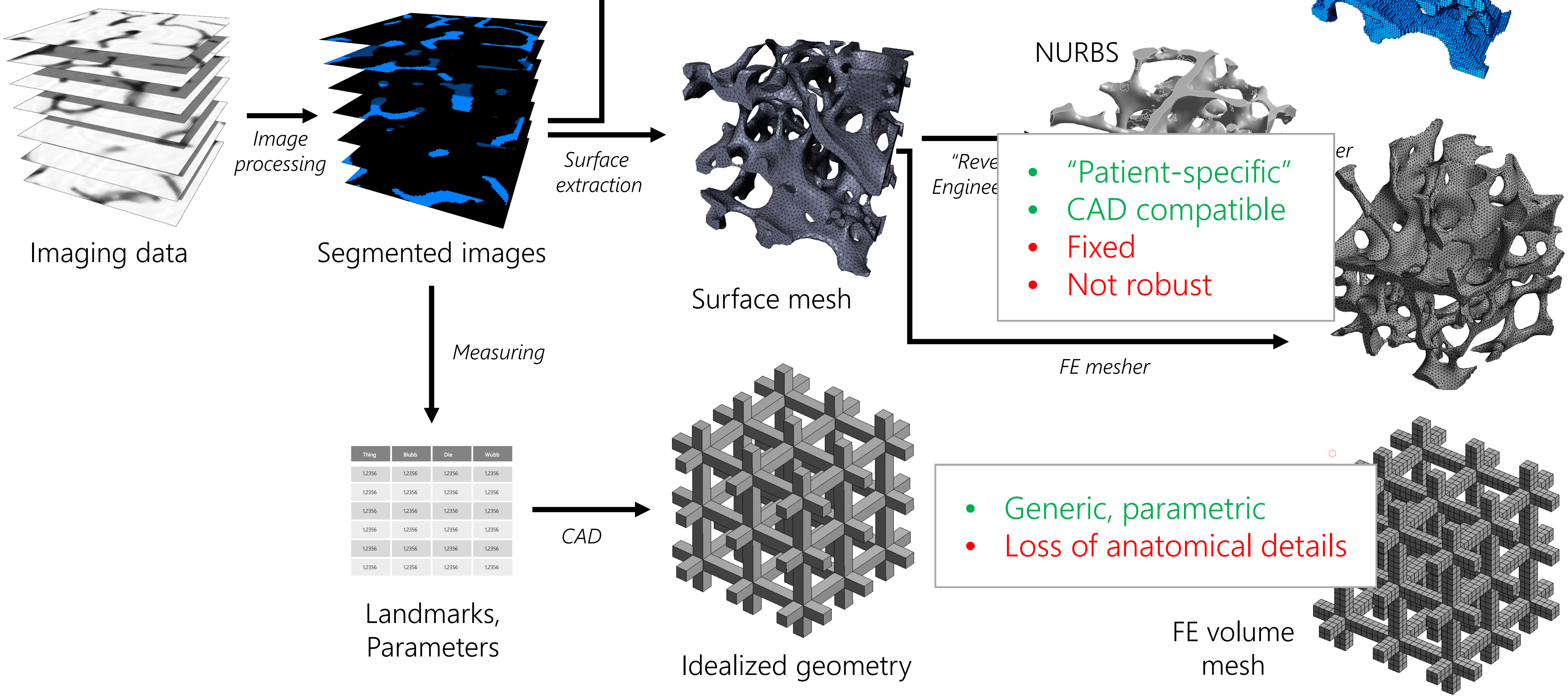
- Image-based meshing: 1-to-1 relation between elements and voxels
- Idea: element-wise (apparent) material properties depending on image intensity
- $I \propto \mu \propto \rho = f^{-1}(E)$
- f: density-stiffness relation for some material
- f depends on material type and scale
- E.g. Carter and Hayes 1977 (homogenized trabecular bone)
  - $E = f(\rho) \propto \rho^3$



Approaches

- "Patient-specific"
- Robust
- Fixed
- Requires special tools





Thing	Blubb	Die	Wubb
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356
1,2356	1,2356	1,2356	1,2356