Computing Local Thickness of 3D Structures with ImageJ

Robert P. Dougherty
OptiNav, Inc. Bellevue, WA, USA

Karl-Heinz Kunzelmann
Poliklinik für Zahnerhaltung und Parodontologie Ludwig-Maximilians-Universität-München, München, Germany

Microscopy & Microanalysis 2007 Meeting
August 5-9, 2007
Broward County Convention Center
Ft. Lauderdale, Florida

Outline

- Introduction
- Definition
- Properties
- Computation
- Examples

Local Thickness Plugin

- Grande Custom ImageJ Plugin from OptiNav, Inc.
- Ordered by Prof. Dr. Kunzelmann Aug. 2006
- He sent with the order:
  - 7 papers
  - Sample data files
  - c-code by David Coeurjolly

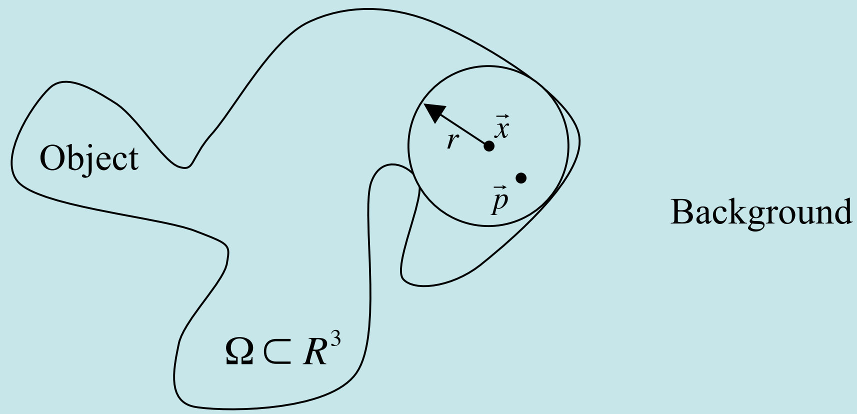
Key Paper

"A new method for the model-independent assessment of thickness in three-dimensional images"
T. Hildebrand and P. Rüesgsegger,
J. of Microscopy, **185** (1996) 67-75.

Another Key Paper

"New algorithms for Euclidean distance transformation on an n-dimensional digitized picture with applications,"
T. Saito and J. Toriwaki,
Pattern Recognition **27** (1994) 1551-1565

Local Thickness Definition



Local Thickness $\quad \tau(\vec{p}) = 2\max\left(\left\{ r \middle| \vec{p} \in sph(\vec{x}, r) \subseteq \Omega, \vec{x} \in \Omega \right\}\right)$

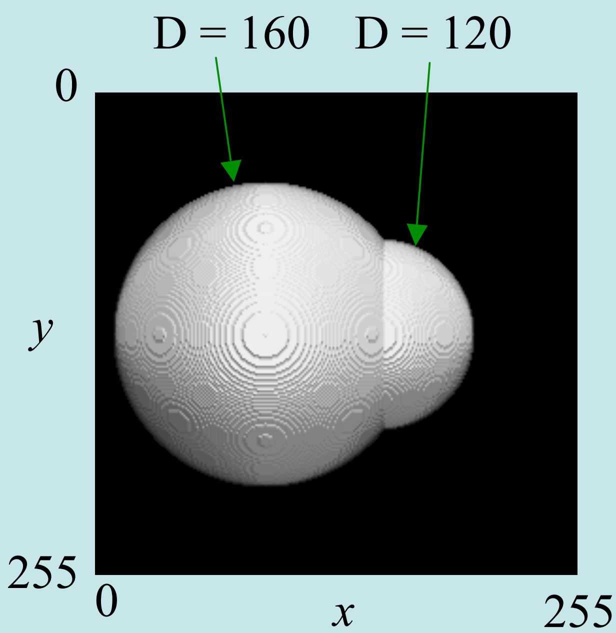
"The diameter of the largest sphere that fits inside the object and contains the point"

Local Thickness Properties

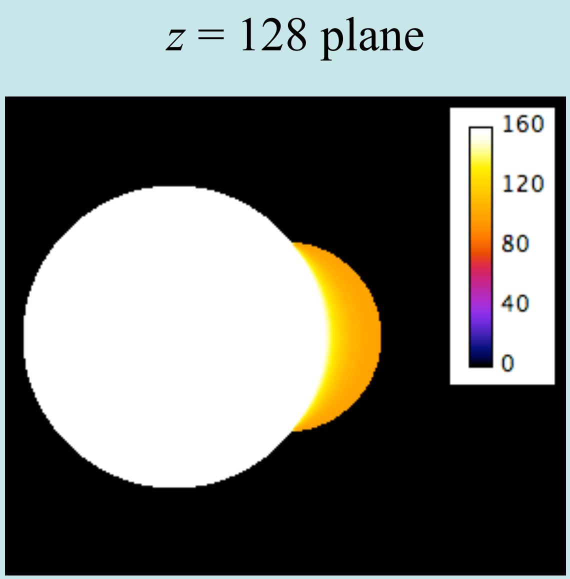
- Inherently 3D
  - Does not depend on structural assumptions
  - Expected results for plates, cylinders, cubes
  - Suited to 3D imaging data
  - Thickness distributions: mean, variance
- Applications
  - Thickness of trabecular bone structures
  - Thickness of paper fibers
  - Planning dental surgery

Local Thickness Properties

Intersecting balls



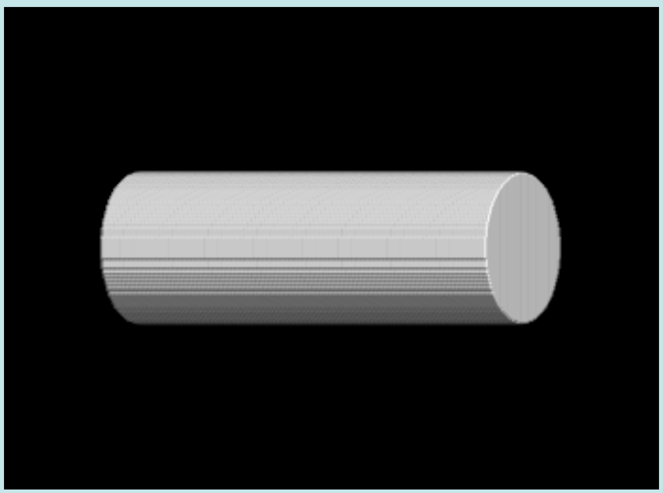
(VolumeJ)

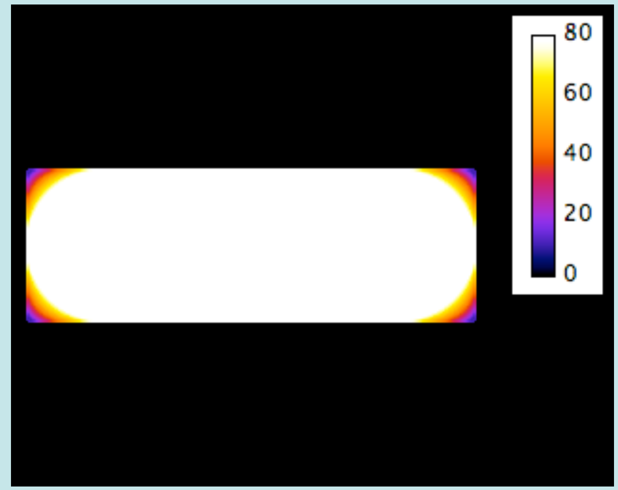


Local Thickness Properties

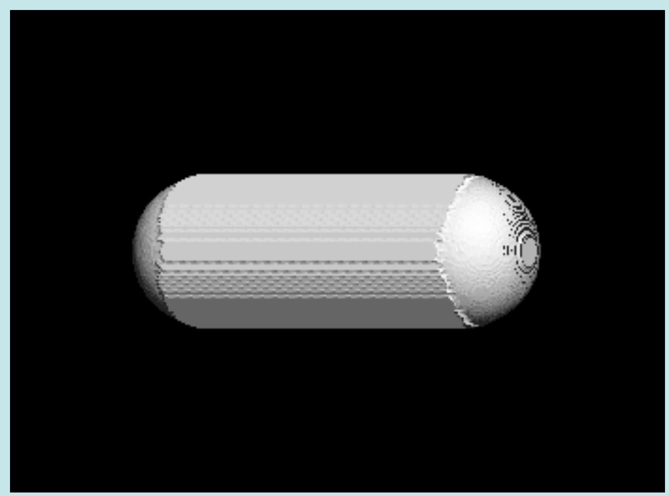
Finite Cylinders

Length 146
Diameter 80





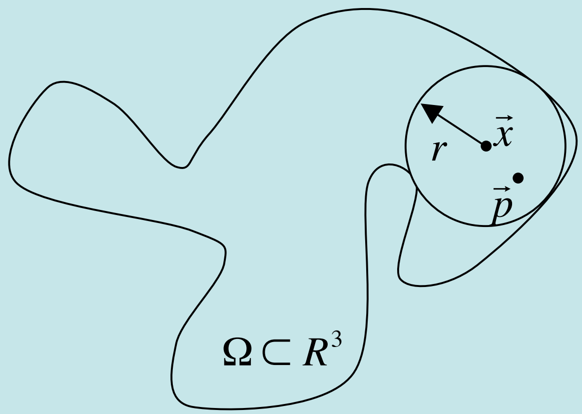
Isosurface 78



Local Thickness Computation

Direct application of the definition would be slow

$$\tau(\vec{p}) = 2\max\left(\left\{r \middle| \vec{p} \in sph(\vec{x}, r) \subseteq \Omega, \vec{x} \in \Omega\right\}\right)$$



$$
\begin{aligned}
&\text{for}(\vec{p} \in \Omega)\{\\
&\quad r_{\max} = 0;\\
&\quad \text{for}(\vec{x} \in \Omega)\{\\
&\qquad \text{for}(r = 0; r \le R; r += \delta r)\{\\
&\qquad\quad \text{Sphere s = new Sphere}(x, r);\\
&\qquad\quad \text{if}(s \subseteq \Omega)\{\\
&\qquad\qquad \text{if}(\vec{p} \in s)\, r_{\max} = r;\\
&\qquad\quad \}\\
&\qquad \}\\
&\quad \}\\
&\quad \tau(\vec{p}) = r_{\max};\\
&\}
\end{aligned}
$$

$O(n^9)$?

$n = 256 \Rightarrow 4.7\times10^{21}$ operations

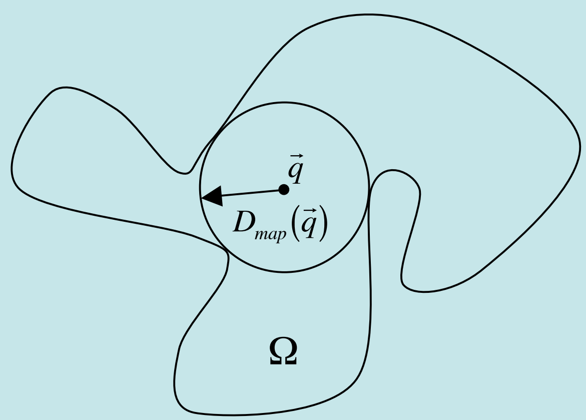
At least 37,000 years on a PC

Local Thickness Computation

First compute the distance map by the “Distance Transformation”

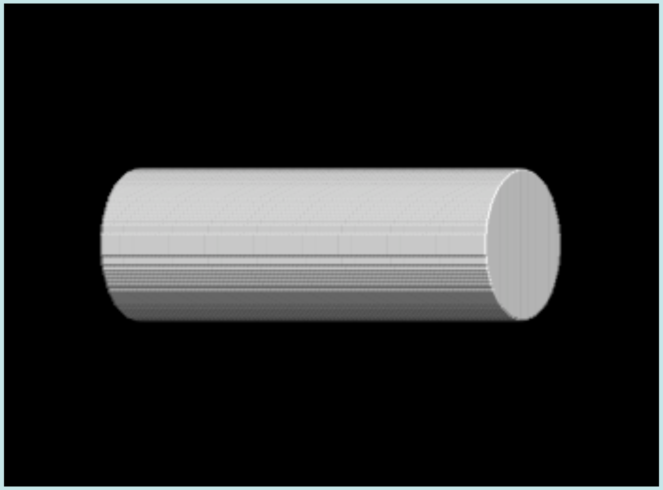
Distance Map $D_{map}(\vec{q}) = \max\left(\{r > 0 \mid sph(\vec{q}, r) \subseteq \Omega\}\right)$

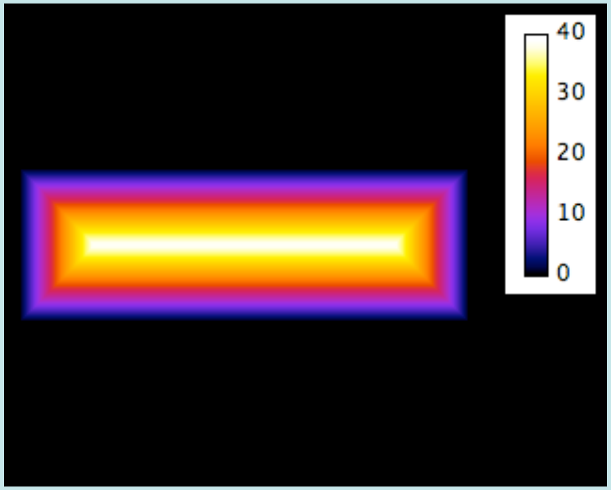
“The radius of the largest sphere centered at $\vec{q}$ that fits inside the object”

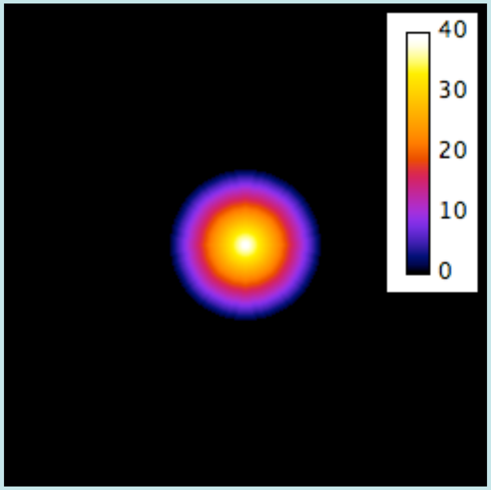


Distance map of a finite cylinder

Length 146
Diameter 80







Local Thickness Computation

Express local thickness in terms of distance map

Local Thickness $$\tau(\vec{p}) = 2 \max_{\vec{q} \in X(\vec{p})} \left(D_{map}(\vec{q}) \right)$$

$$X(\vec{p}) = \{\vec{x} \in \Omega \,|\, \vec{p} \in sph\left(\vec{x}, D_{map}(\vec{x})\right)\}$$

$X(\vec{p}) =$ the set of $\vec{q}$ s whose distance map spheres contain $\vec{p}$

$X(\vec{p}) =$ the set of $\vec{q}$ s that own $\vec{p}$

Local Thickness Computation

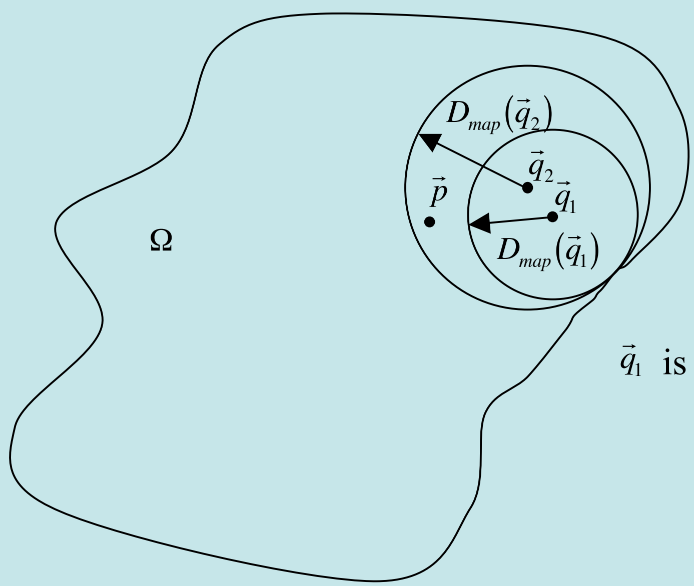
Simplify the search set to the Distance Ridge

$$X_R(\vec{p}) = \{\vec{x} \in \Omega_R \,|\, \vec{p} \in sph(\vec{x}, D_{map}(\vec{x}))\}$$

where

$$\Omega_R = \{\vec{p} \in \Omega \,|\, sph(\vec{p}, D_{map}(\vec{p})) \not\subseteq sph(\vec{x}, D_{map}(\vec{x})), \vec{p} \neq \vec{x}, \vec{x} \in \Omega\}$$

"The center points of all nonredundant spheres"



$$\tau(\vec{p}) = 2 \max_{\vec{q} \in X_R(\vec{p})} \left(D_{map}(\vec{q})\right)$$

$\vec{q}_1$ is redundant because it does not own any points that $\vec{q}_2$ doesn't

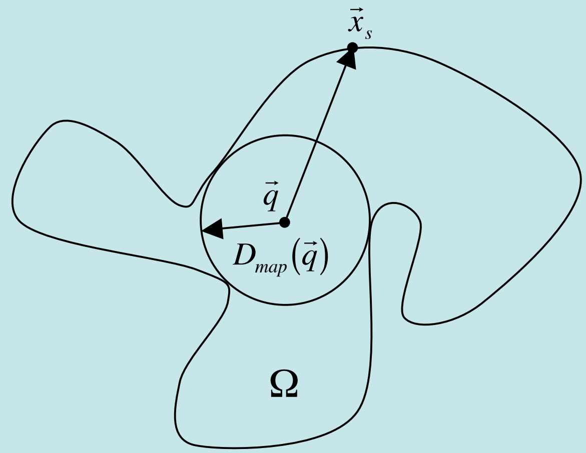
Local Thickness Computation

1. Perform the distance transformation to produce the distance map

2. Remove redundant points to produce the distance ridge

3. Compute the local thickness at each point $\vec{p}$ by scanning the distance ridge to find the “largest” $\vec{q}$ that “owns” $\vec{p}$

Distance Transformation (Step 1 in LT)



Naïve algorithm:
For each $\vec{q}$, scan the surface points, $\vec{x}_s$, to minimize $\|\vec{x}_s - \vec{q}\|$

O(n^5)

Dramatically better algorithm: Saito-Toriwaki

Saito-Toriwaki Euclidean Distance Transformation Algorithm

Notation:

Digitized w×h×d picture:

$$F = \{f_{ijk}\}, 0 \le i < w, 0 \le j < h, 0 \le k < d$$

$$f_{ijk} = 1 \text{ if } (i,j,k) \in \Omega, \ 0 \text{ else}$$

$$D_{map}(i,j,k) \equiv \{d_{ijk}\}$$

$$S(i,j,k) \equiv \{s_{ijk}\}, s_{ijk} = d_{ijk}^2, 0 \le i < w, 0 \le j < h, 0 \le k < d$$

$$s_{ijk} = \min_{pqr} \{(i-p)^2 + (j-q)^2 + (k-r)^2 \mid f_{pqr} = 0\}$$

Algorithm:

Transformation 1. Derive picture G from F according to

$$g_{ijk} = \min_{x}\{(i-x)^2 \mid f_{xjk} = 0, 0 \le x < w\}$$

Transformation 2. Derive picture H from G according to

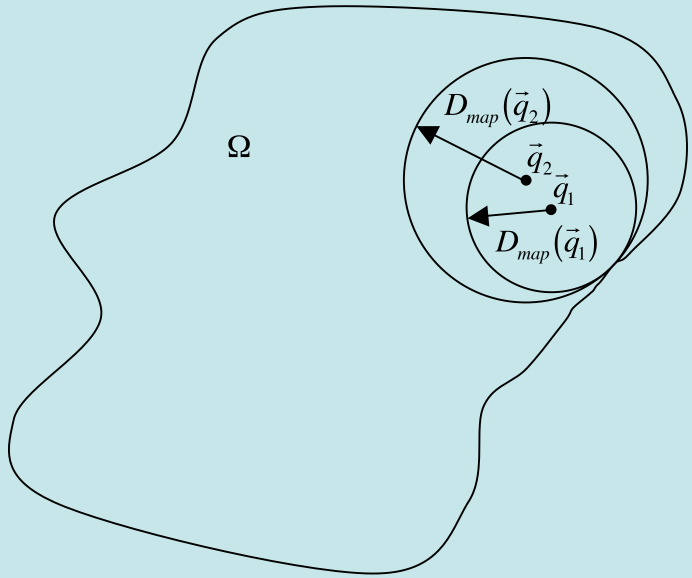
$$h_{ijk} = \min_{y}\{g_{iyk} + (j-y)^2, 0 \le y < h\}$$

Transformation 3. Derive picture S from H according to

$$s_{ijk} = \min_{z}\{h_{ijz} + (k-z)^2, 0 \le z < d\}$$

$O(n^4)$, easy parallel processing, no 3D work arrays needed.

Distance Ridge Computation (Step 2 in LT)



Objective: remove redundant points, such as $\vec{q}_1$, from the distance map to improve the speed of the subsequent local thickness search.

- Difficult
- Complete removal not required

Distance Ridge Computation

Brute force algorithm adapted from Saito & Toriwaki:*

Scan everything in sight

Seems to be $O(n^6)$ and require 4 3D work arrays

*"Reverse Distance Transformation and Skeletons Based upon the Euclidean Metric For n-Dimensional Binary Pictures," T. Saito and J. Toriwaki, IEICE Trans. Inf. & Syst., Vol E77-D, No. 9, Sept. 1994

The reference also gives a more-efficient algorithm for a different skeleton.

Distance Ridge Computation

Template approach implemented in the plugin.
Loosely inspired by Remy & Thiel*

Input: distance map

Output: distance map with some redundant points removed

Algorithm:

- Scan points in the distance map. For each point,
  - Scan neighboring points
    - Use a template to evaluate the point and the neighboring point
      - If the neighboring point does not "own" any more points than the scan point, based on a template, then delete the neighbor point

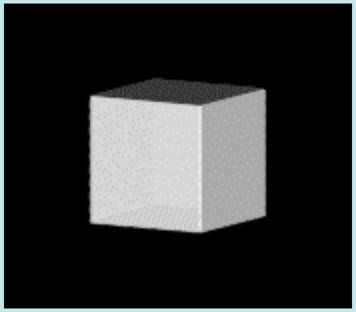
Limitation:

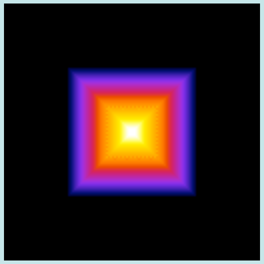
- Does not remove all redundant points

*"Exact medial axis with euclidean distance," E. Remy and E. Thiel, Image and Vision Computing **23** (2005) 167-175.

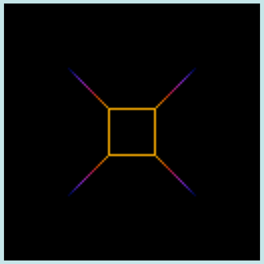
Distance Ridge Examples

Cube

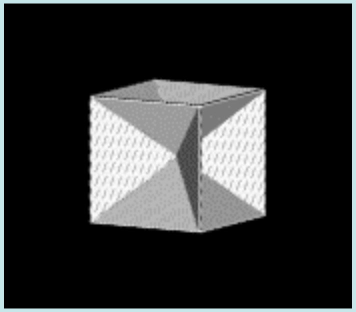




Distance map



One slice of distance ridge

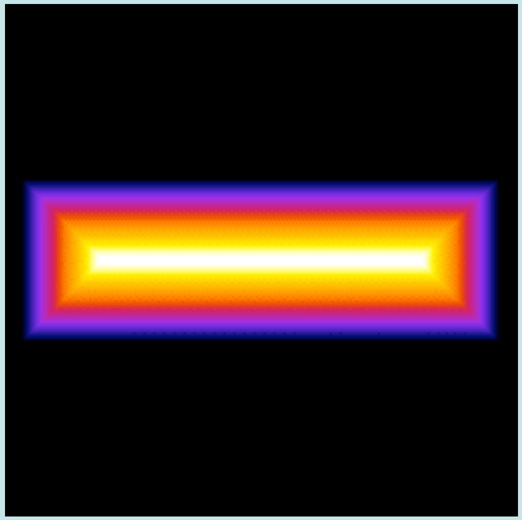


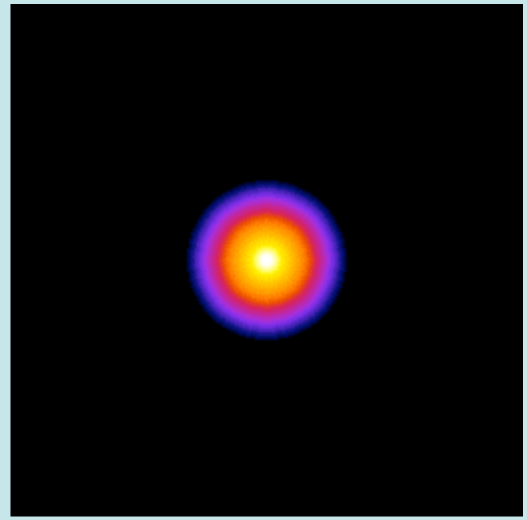
3D rendering of distance ridge

Distance Ridge Examples

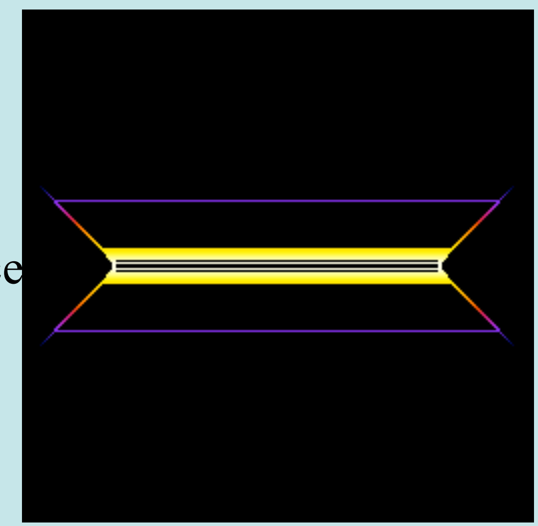
Finite Cylinder

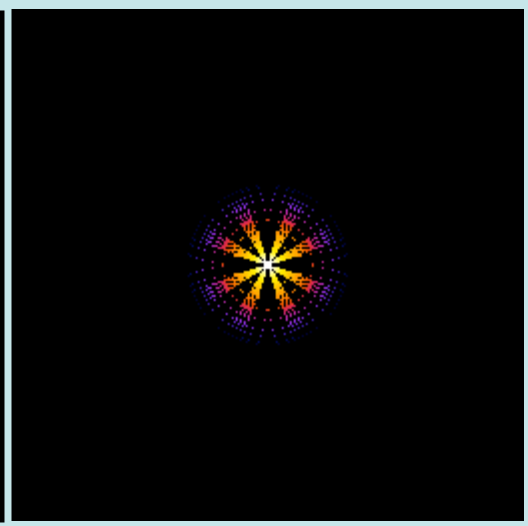
Distance map

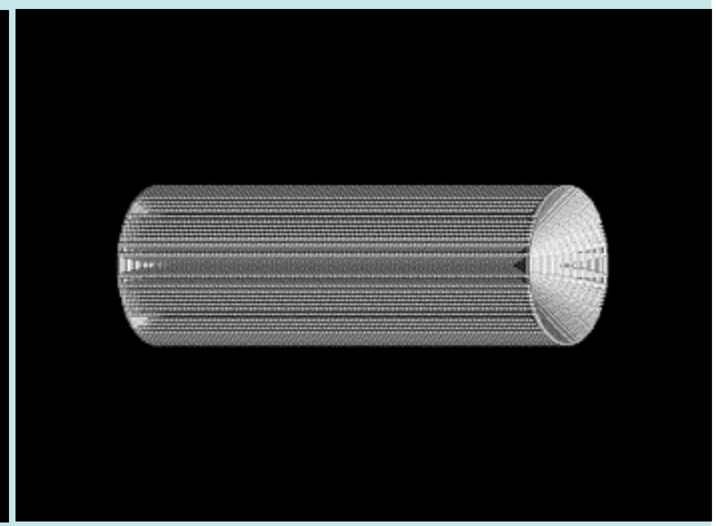




Distance ridge

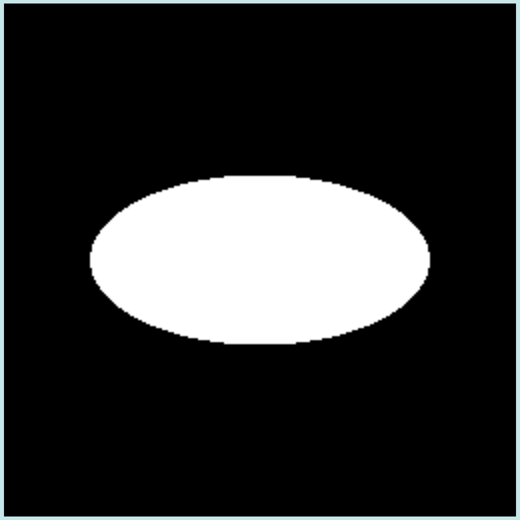




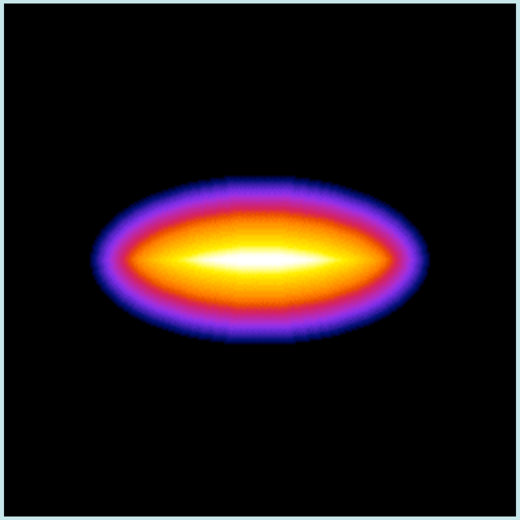


Distance Ridge Examples

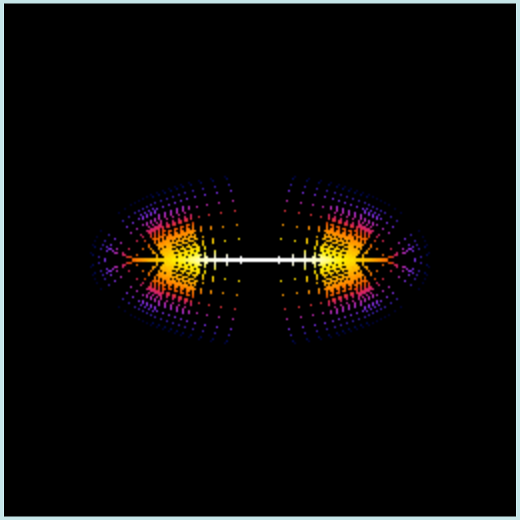
Elliptical Cylinder



Geometry



Distance map



Distance ridge

Local Thickness from Distance Ridge (Step 3)

$$\tau(\vec{p}) = 2 \max_{\vec{q} \in X_R(\vec{p})} \left(D_{map}(\vec{q}) \right)$$

Input: distance ridge: $H = \{h_{ijk}\}$

Output: local thickness: $\mathrm{T} = \{\tau_{ijk}\}$

Let t_{ijk} be the square of half of the local thickness

Make a list of non-zero distance ridge points: $(i_l, j_l, k_l, h_l), l = 0,\ldots, L-1$

Initialize $t_{ijk} = 0 \;\forall\; ijk$

for points (u, v, w) {

for ridge points, l {

$if\ ((i_l - u)^2 + (j_l - v)^2 + (k_l - w)^2 < h_l)$ {

if $(h_l > t_{uvw})$ update $t_{uvw} = h_l$;

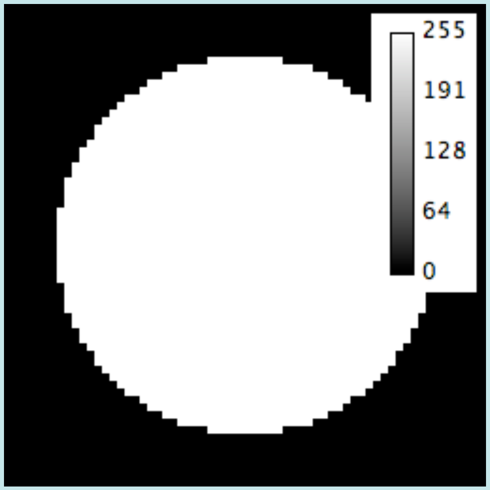
}

}

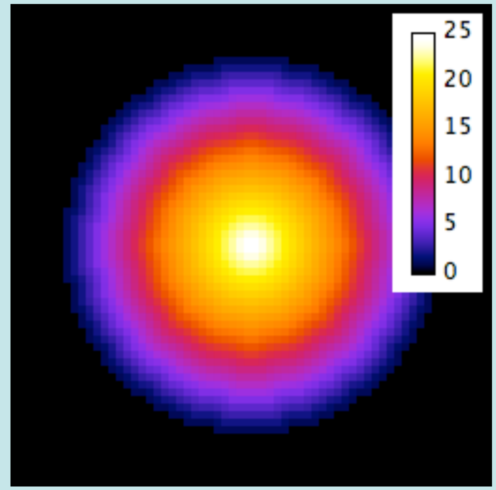
}

Output $\tau_{ijk} = 2\sqrt{t_{ijk}} \;\forall\; ijk$

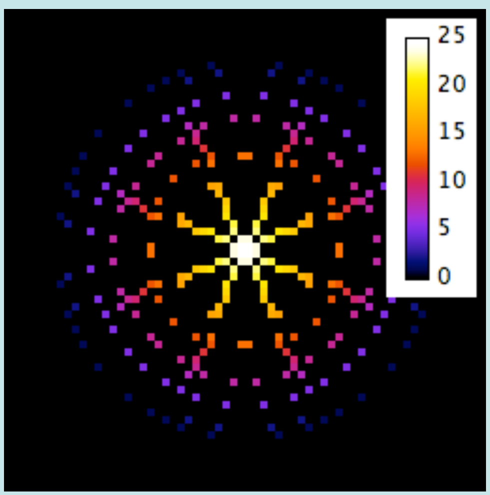
Local Thickness Detail: Small Surface Values Due to Voxel Effects



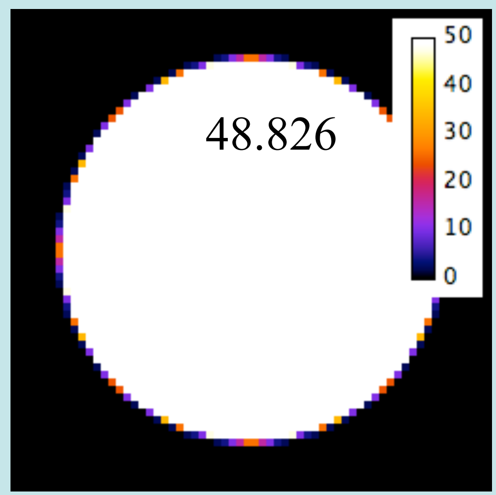
Infinite cylinder, nominal diameter = 50



Distance map

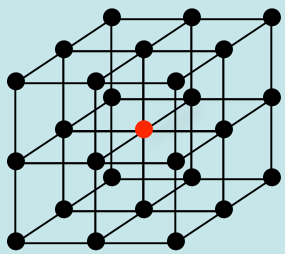


Distance ridge



Local Thickness

Local Thickness Detail: Surface Cleanup Algorithm

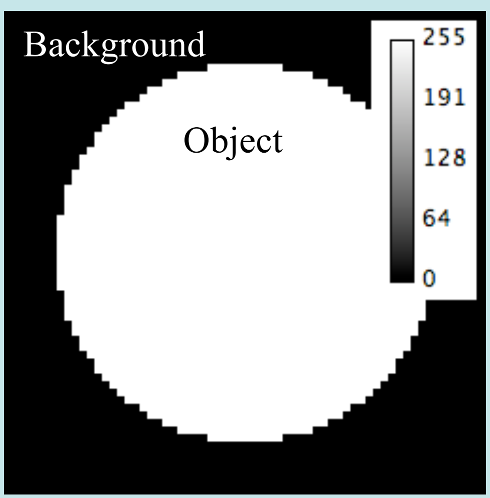


Definitions

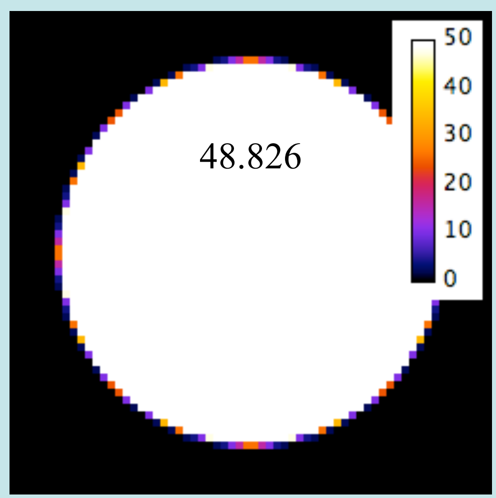
- A voxel has 26 <u>neighbor</u> voxels
- A <u>surface voxel</u> is an object voxel with at least one background neighbor

Algorithm

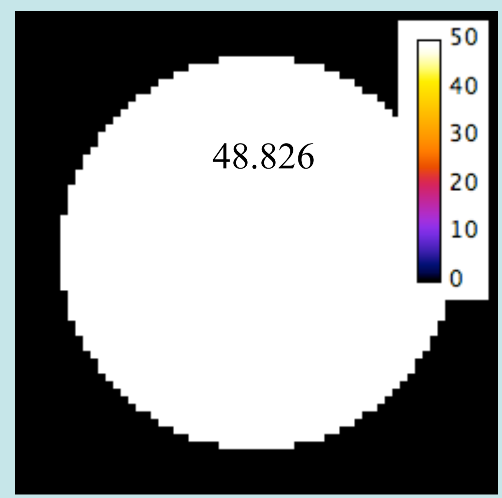
Replace the local thickness value of every surface voxel by the average of the local thicknesses of its neighboring, non-surface, object voxels.



Infinite cylinder
nominal diameter = 50

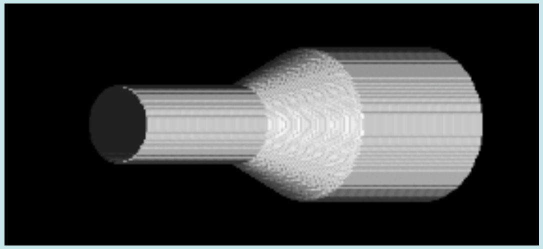


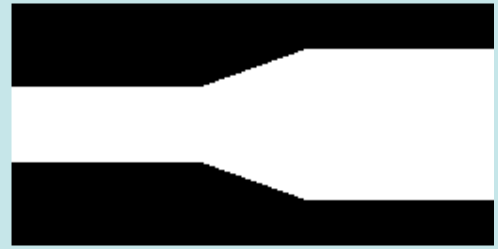
Original Local Thickness



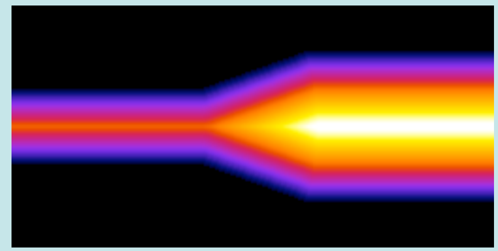
Local Thickness after Cleanup

Example: Cylinder with Diameter Change

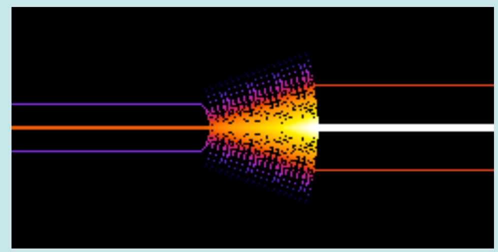




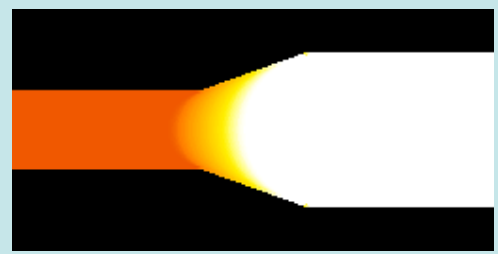
Distance map

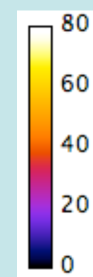


Distance ridge

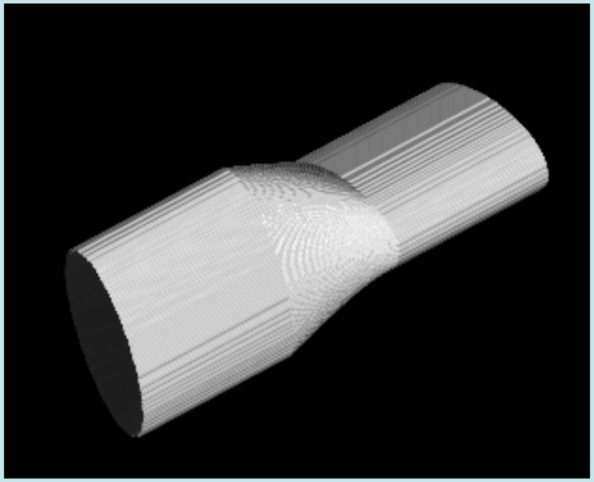


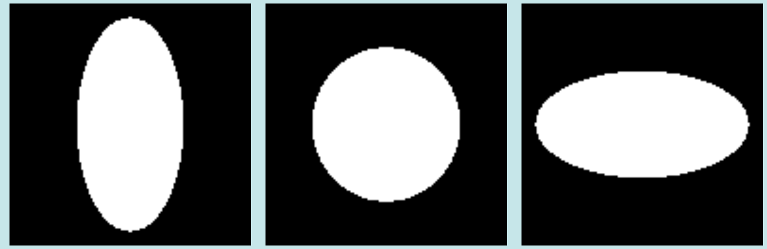
Local thickness



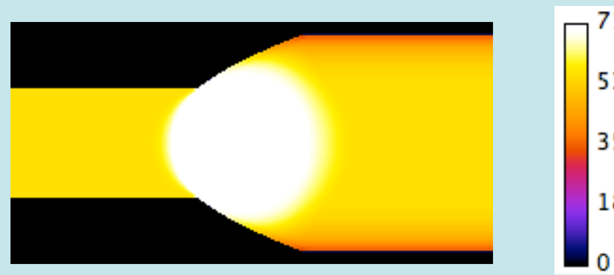


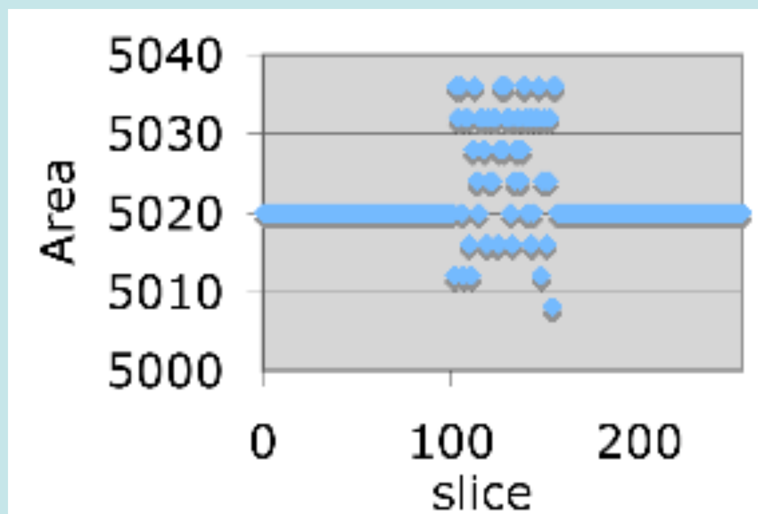
Example: Elliptical Cylinder with Transition



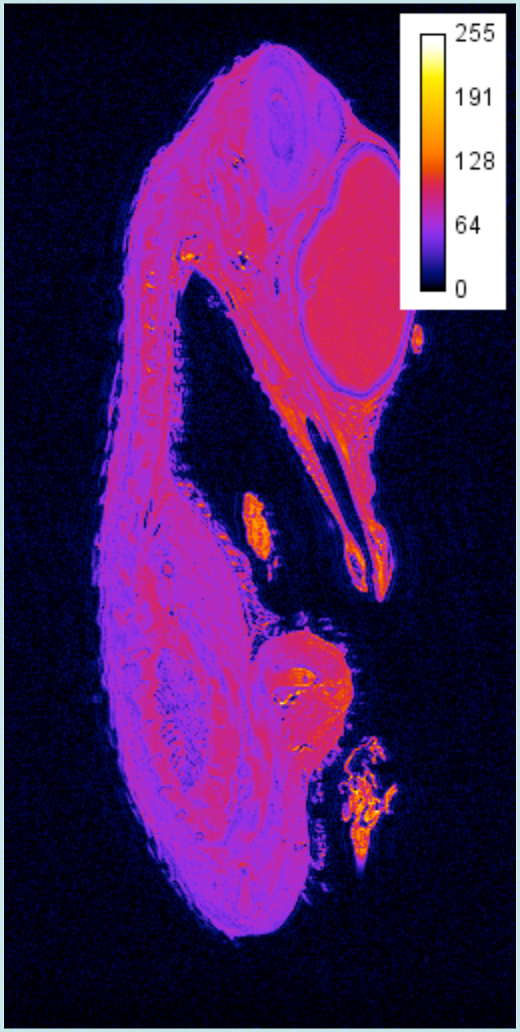


Local thickness



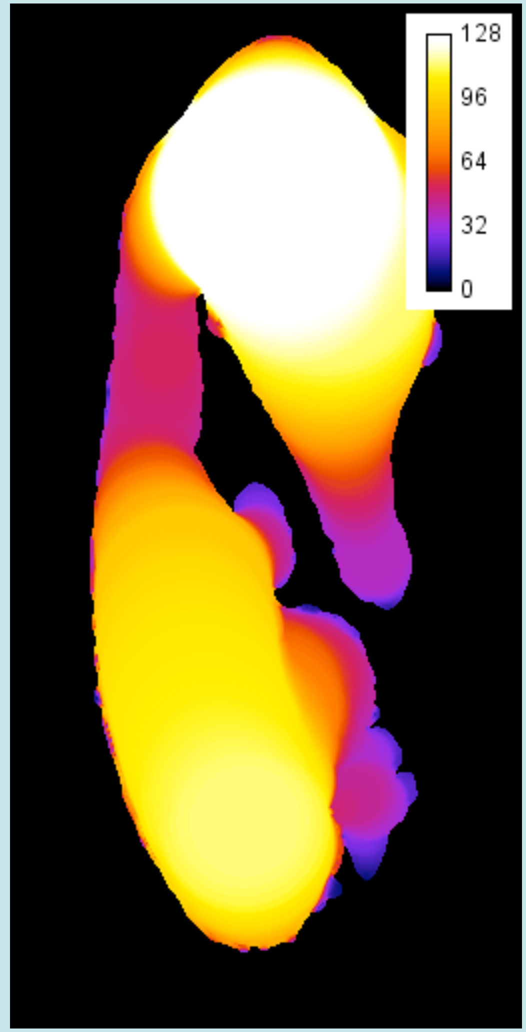


Example: μMRI Quail Embryo



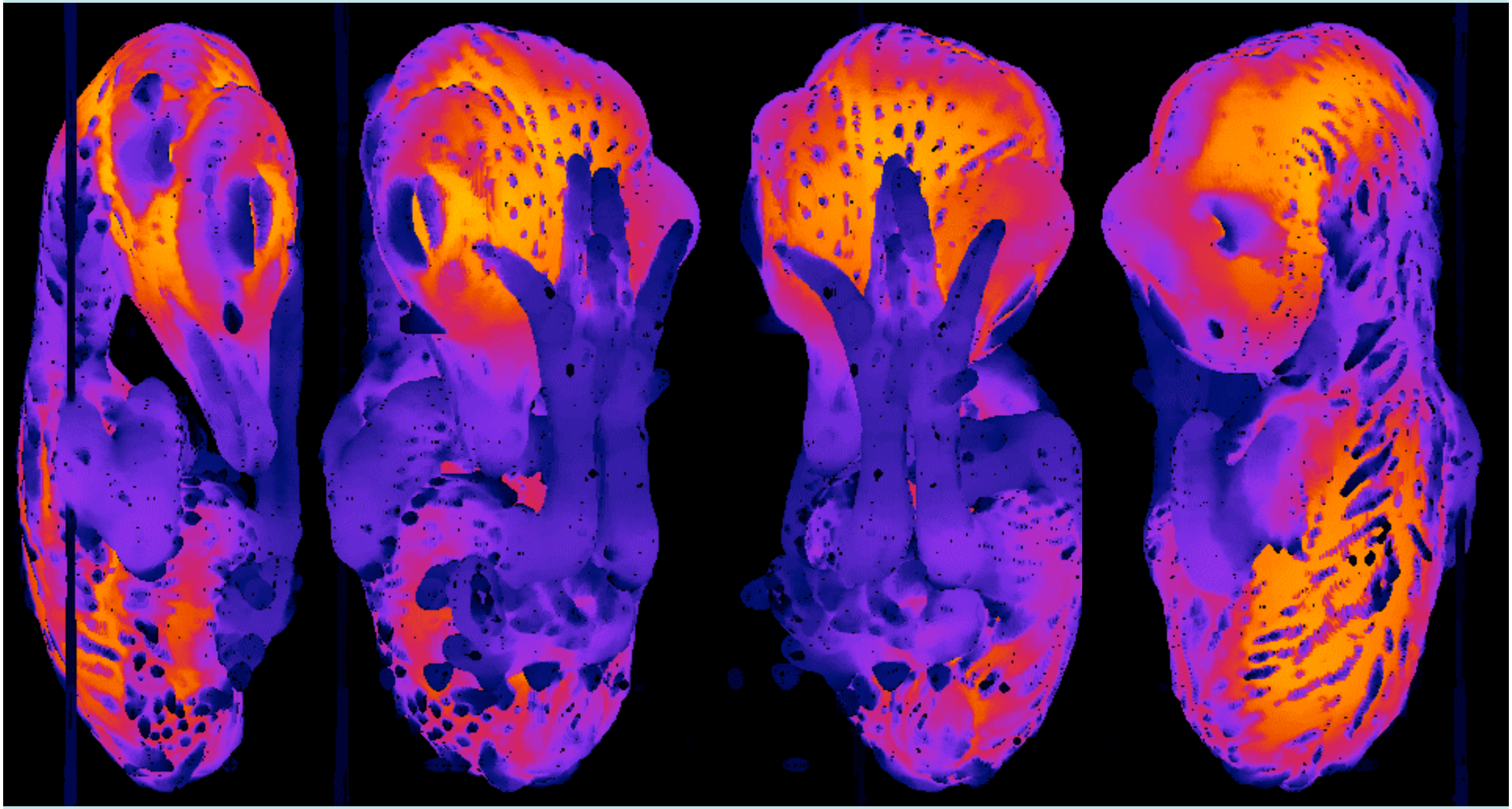
Caltech MRI Atlases
http://atlasserv.caltech.edu/

Binary image after 3D blur and threshold of 3



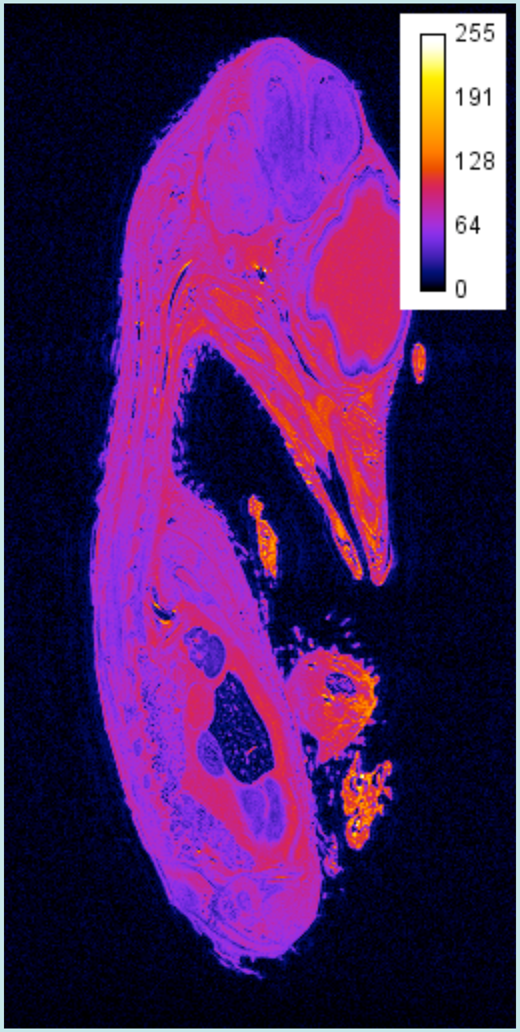
Local thickness (z=88 slice)

Example: μMRI Quail Embryo



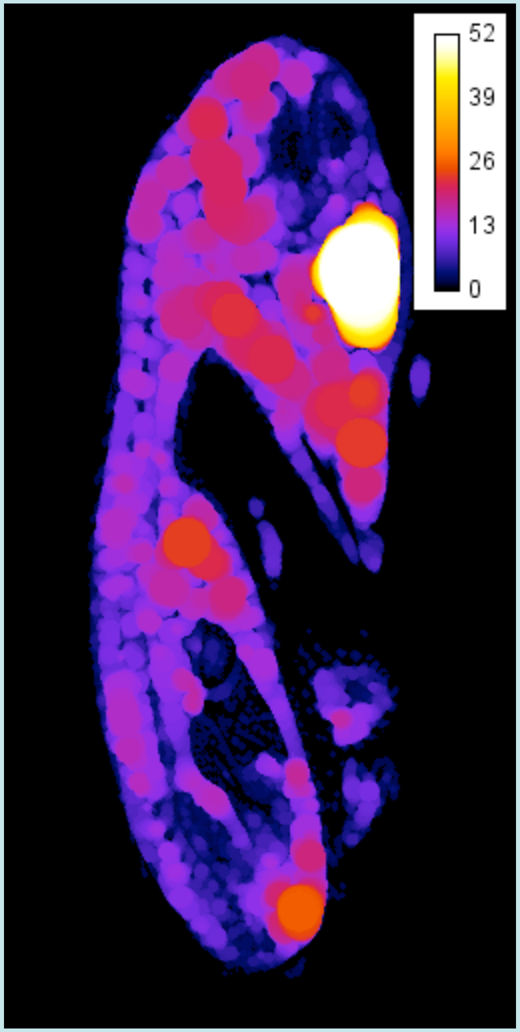
Local thicknes: surface projection

μMRI Quail Embryo: Internal Structure



Caltech MRI Atlases
http://atlasserv.caltech.edu/
z = 109 slice

Binary image with
threshold = 44

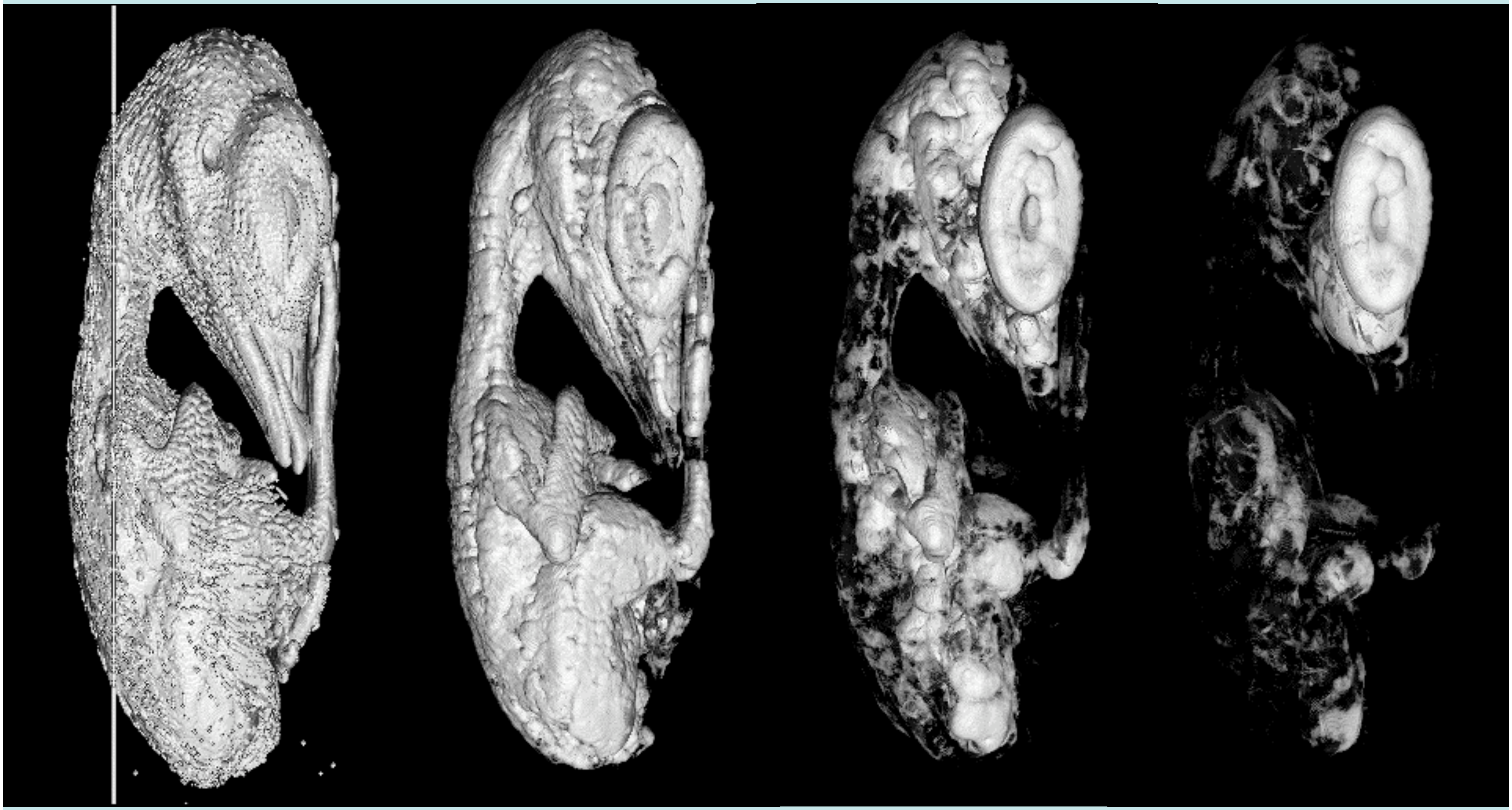


Local thickness (z=109 slice)

Example: μMRI Quail Embryo

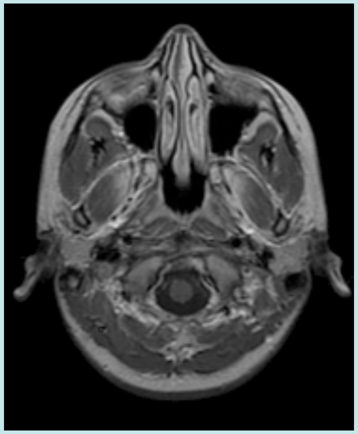
Render the local thickness using VolumeJ with different thickness thresholds

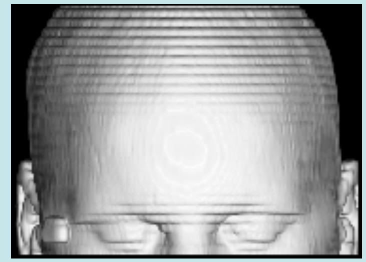
2 10 20 30



Example: MRI Stack

(Expanded by 5 in z)





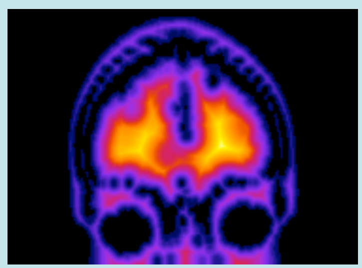
VolumeJ

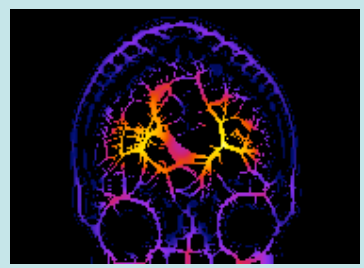
Distance map (threshold=40)

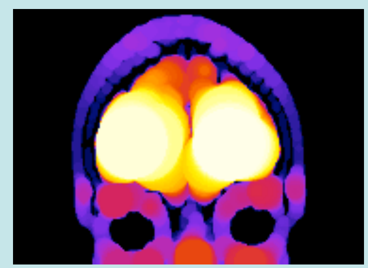
Distance ridge

Local thickness

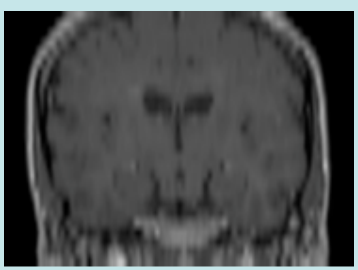
Slice 42

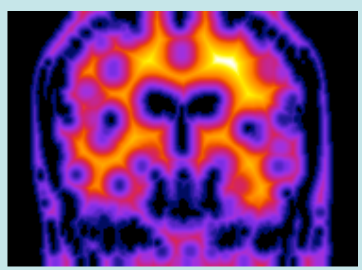


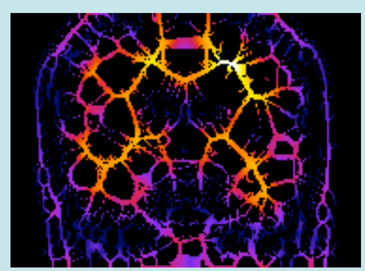


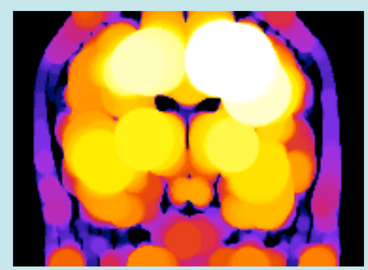


Slice 116







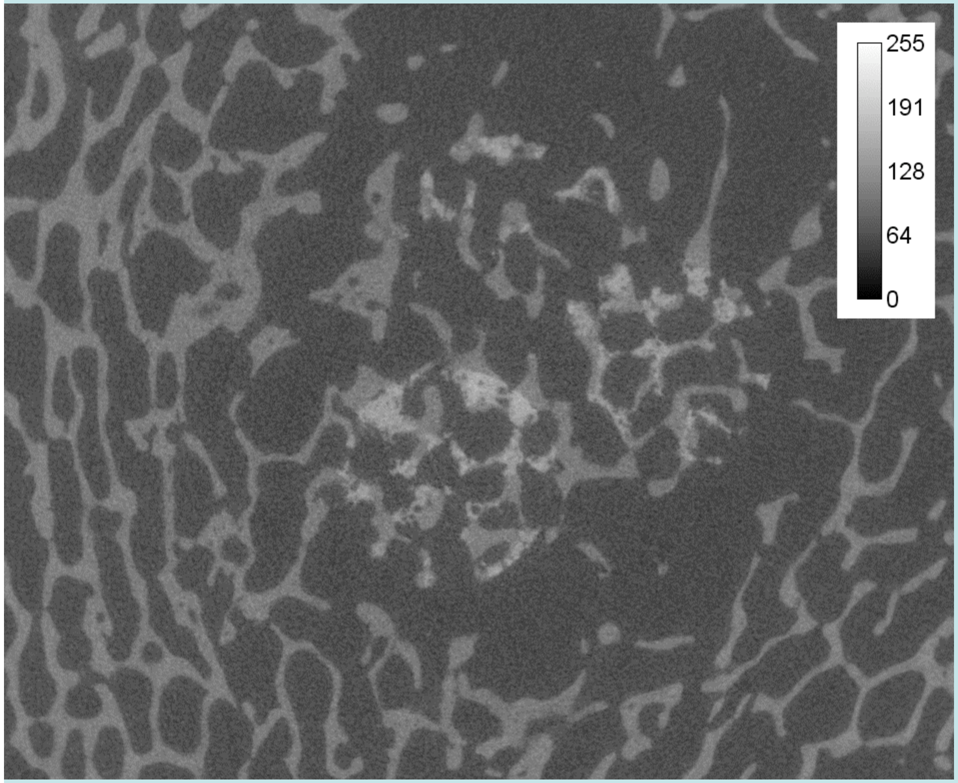


Example: µCT of Trabecular Rabbit Femur

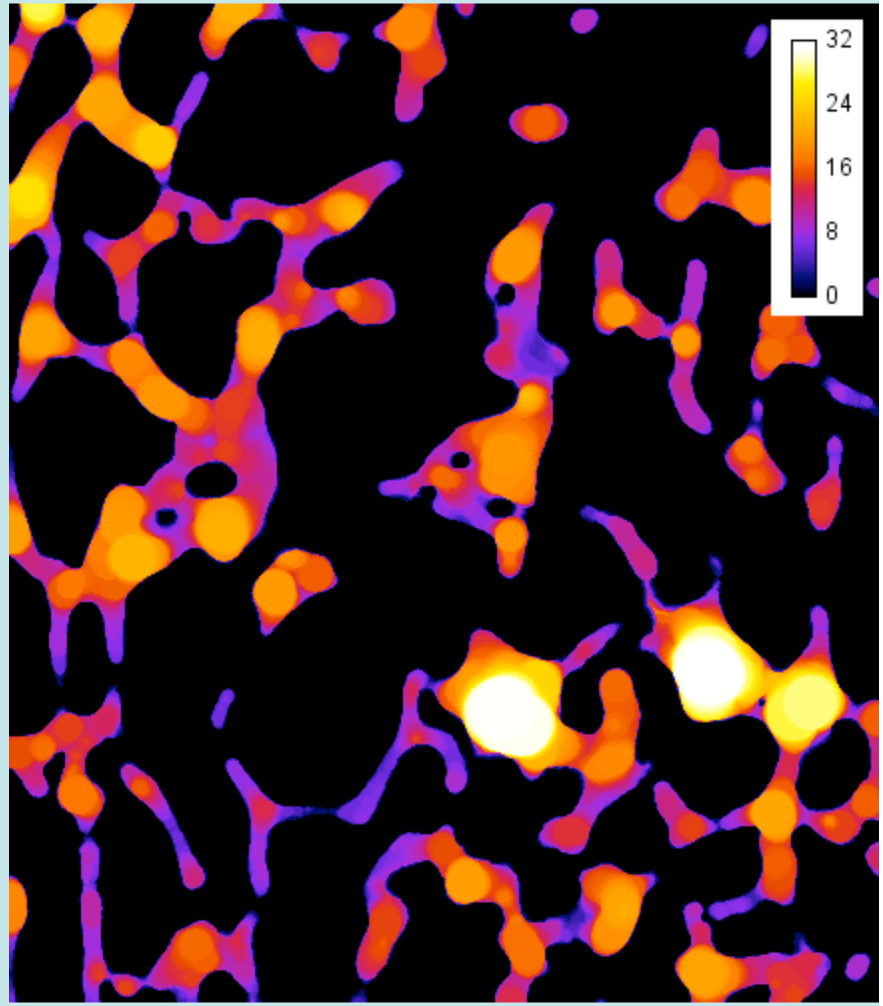
Dr. Miriam Draenert
Poliklinik für Zahnerhaltung und Parodontologie
Ludwig-Maximilians-Universität-München

- Hole drilled in rabbit femur
- Hole filled with hydroxyapatite particles
- Bone morphogenetic proteins (BMPs) to stimulate growth
- Rabbits sacrificed after 90 days
- Femur prepared for evaluations, including µCT
- Mean local thickness evaluated statistically to compare different BMPs

Example: µCT of Trabecular Rabbit Femur

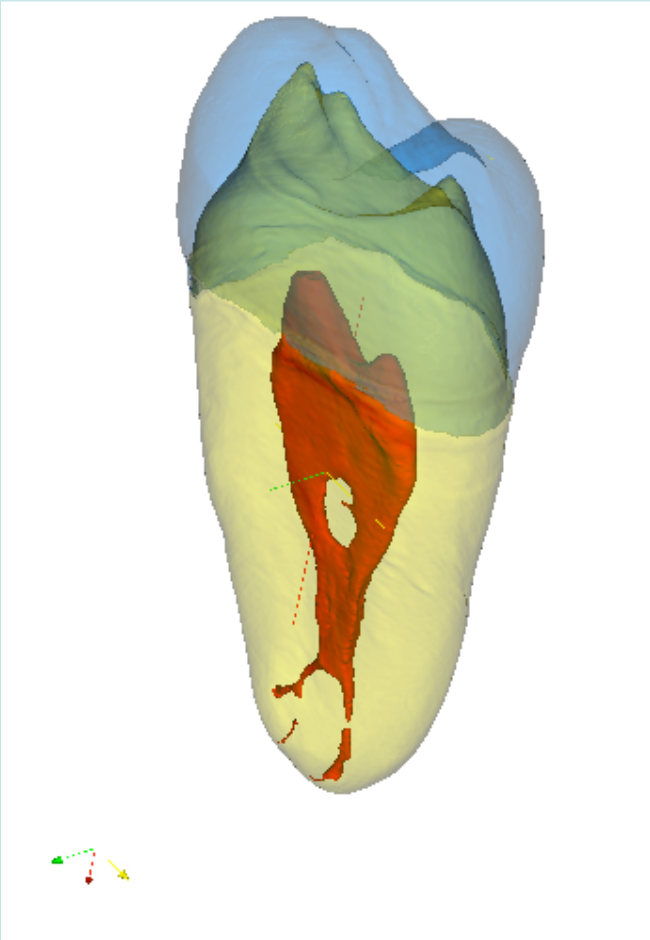


Slice 173 of µCT



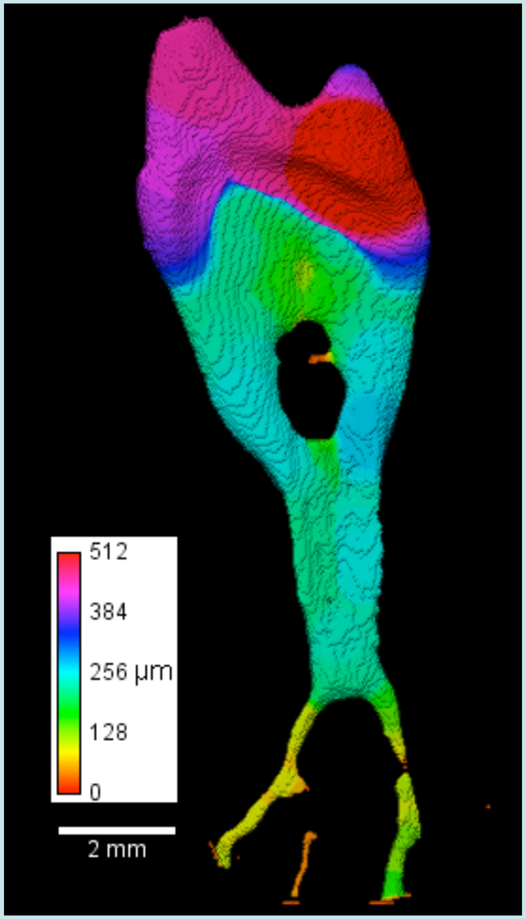
Local thickness (z=173 slice)

Example: Tooth



Enamel, dentin and pulp segmented from µCT

ParaView 2.4 (Kitware, www.kitware.com)



Local thickness of pulp

VolumeJ plugin for ImageJ

(v. 1.7a, Michael Abramoff)

Summary

• New, Quantitative, 3D Tool
• Open Source, Freely available

1. EDT_S1D.java
   - Inputs an 8-bit image stack describing the 3D geometry
   - Distance map: Saito-Toriwaki Euclidean Distance Transformation
   - Parallel processing using all available processors
   - Output is a 32-bit stack
2. Distance_Ridge.java
   - Inputs a 32-bit distance map stack
   - Applies a template algorithm to remove many of the redundant points
   - Overwrites the input stack with the resulting quasi-distance ridge
3. Local_Thickness_Parallel.java
   - Inputs a distance map or distance ridge stack
   - Uses a direct search method to compute the local thickness stack
   - Parallel processing using all available processors
4. Clean_Up_Local_Thickness.java
   - Adjusts surface values of a local thickness stack to compensate for voxel artifacts

Conclusion

Different types of local thickness results can be obtained using different threshold values when the image stacks have a range of density values