Naval Force Health Protection Program Review 2025

SAND2025-07529C

Automated workflow for the human digital twin – medical image to mesh

PI: Chad Hovey

Collaborators: Michael Buche, Rika Carlsen, Emma Lejeune, Anu Tripathi

Sandia National Laboratories

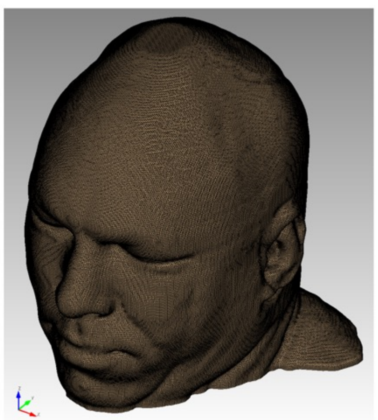
chovey@sandia.gov

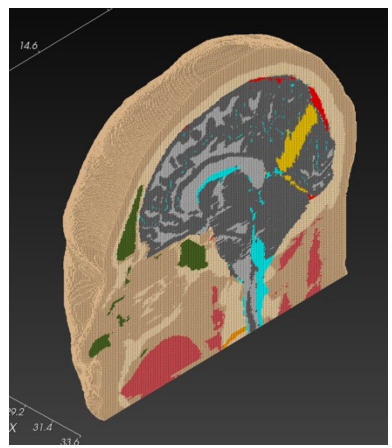
Award Period 10/1/24 to 9/30/25

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Background

- While high-fidelity (quality), the human digital twin made manually is **slow**, **expensive,** and **does not scale.**

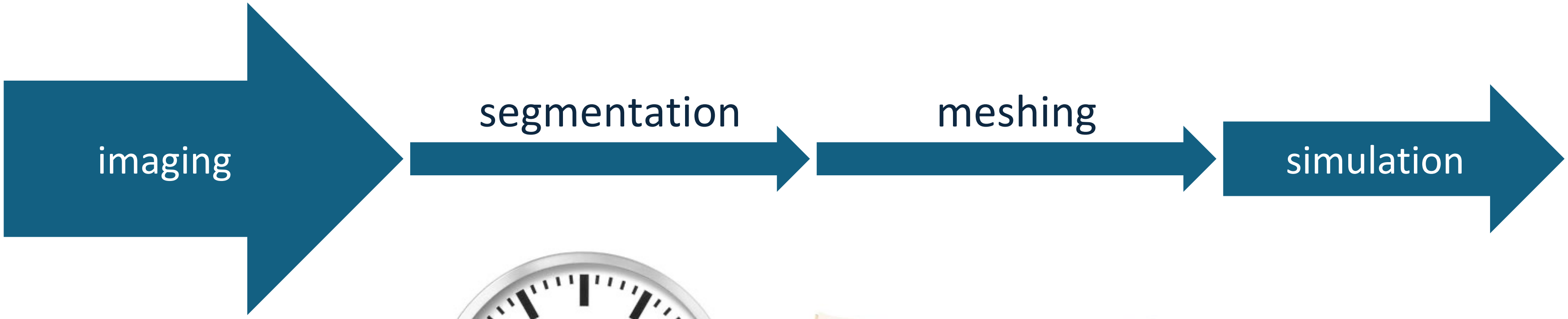




Background

SAND2025-07529C

- For a singleton, what specific part(s) of the workflow are **bottlenecks**?



Background

SAND2025-07529C

Two **bottlenecks**

Class “human” → Segmentation → Mesh → Injury Risk

Class
M/F, S, M, L, XL

… now 8x the bottlenecks as previously,
…this approach <u>does not scale</u>.

Photo credit: US Marine Corps

… <u>intractable</u>.

SAND2025-07529C

Objective

Automate the creation of
a **human digital twin** based on
patient-specific medical imaging.

patient → analysis → injury risk → policy

autotwin

Automated workflow
from Image to Injury Risk
for the Human Digital Twin

Objective

SAND2025-07529C

patient → analysis → injury risk → policy

imaging → segmentation → meshing → simulation

Objective

SAND2025-07529C

- What makes this effort original and exciting?
  - *Empowers stakeholders to perform personalized injury risk assessment, **at scale**.*
- Concisely state your testable hypothesis.
  - *Creation of the human digital twin from medical images can be fully automated.*
  - *The injury risk assessment of automated approach will be of sufficient quality that results therefrom are **indistinguishable** from the results of manual incumbent approaches.*

Objective

- What is the significance and potential scientific impact of the project?

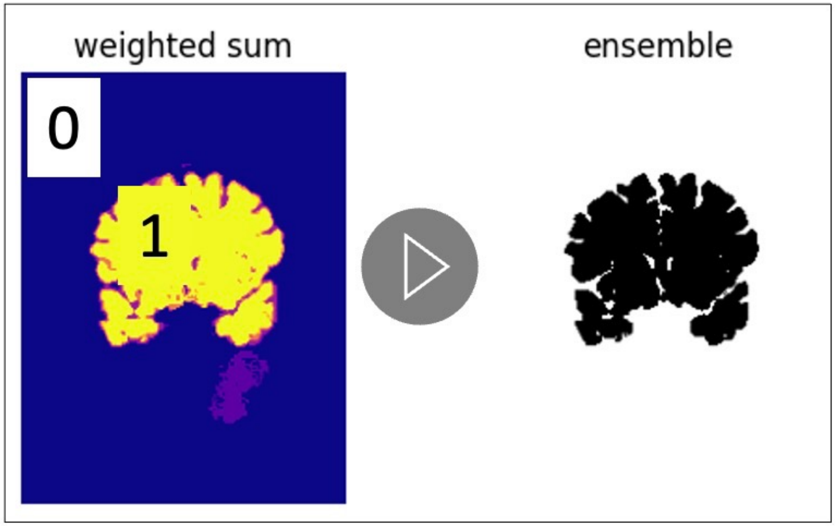
  - *Automation allows injury risk analysis to be performed on a* ***patient-specific*** *basis, and at a significant* ***scale****.*

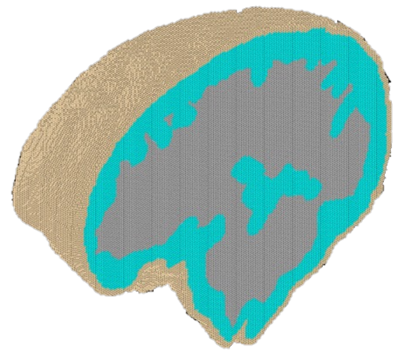
  - *Anticipated impact: drastically improved injury risk assessment.*

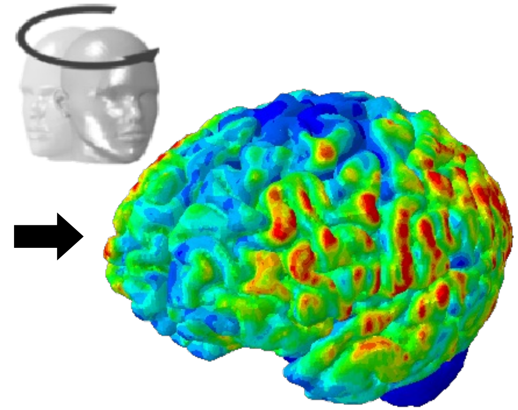
Research Approach

SAND2025-07529C

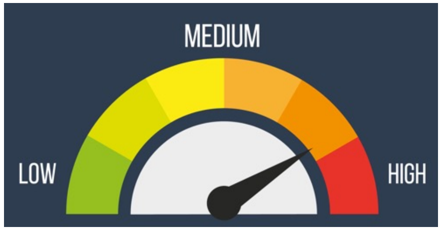
imaging → segmentation → meshing → simulation



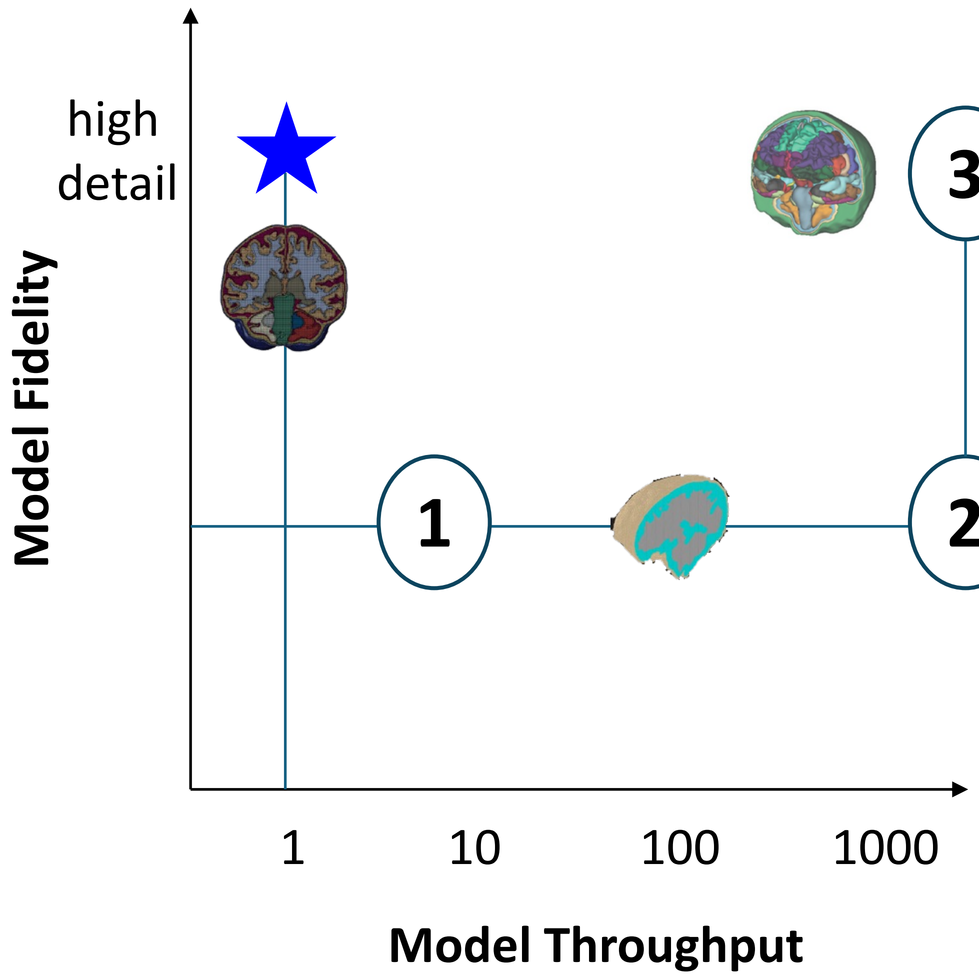




injury risk



Research Approach



Segmentation Start Point

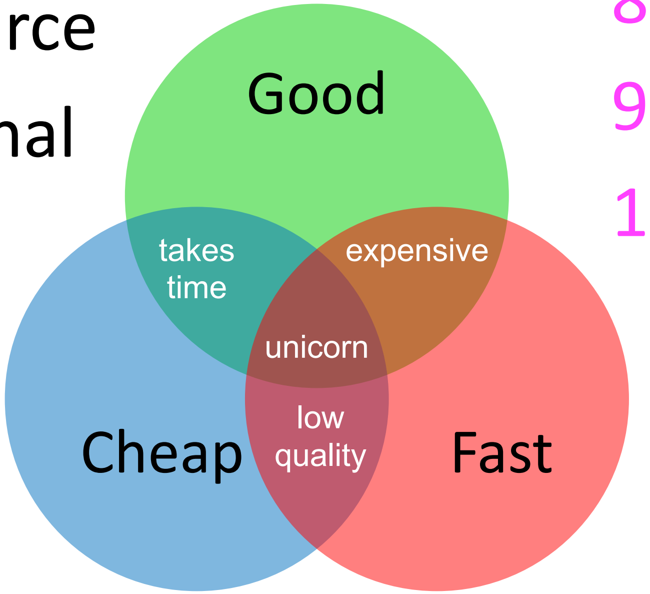
SAND2025-07529C

- **Medical** – imaging, diagnosis, surgical planning
- **Dental** – jaw surgery and dental implants
- **Biomechanical** – orthopedic implants, prosthetics, digital twin, crash testing
- **Materials** – as-built geometries with voids, inclusions, defects
- **Aerospace** – inspect critical components, e.g., turbine blades
- **Civil** – composite structures, voids, rebar, anomalies
- **Geology** – subsurface exploration, seismic
- **Petroleum** – reservoir rocks and fluids, porosity and permeability

Requirements

SAND2025-07529C

1. Segmentation start point
2. Fast
3. Automated
4. Open-source
5. Professional



6. SPN/NumPy and ABAQUS/EXODUS Format
7. All-hex mesh
8. Multi-material
9. Multi-scale
10. Quality

Closed-Source

SAND2025-07529C

- Altair Hypermesh
- ANSYS Meshing
- COMSOL Multiphysics
- Cubit/Sculpt
- Gridgen
- Hexotic
- Siemens Simcenter
- TrueGrid

Open-Source

SAND2025-07529C

- All are fast (e.g., compiled), have varying professional production attributes, metric measures, limited multi-material and limited multi-scale support.
- None are fully automated for all-hex generation.
- None support all our I/O requirements.

Library	Segmentation Start Point	All-Hex Mesh
cinolib	No	Yes
FEniCS	Yes	No
FreeFEM	Yes	No
GetFEM++	Yes	No
GMSH	Limited	Limited
MeshLab	Yes	No
NETGEN	No	No
OpenFOAM	No	No
TetGen	No	No

Limitations

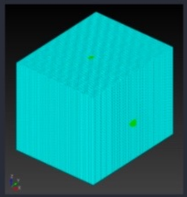
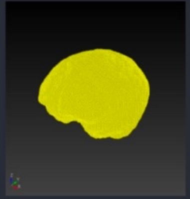
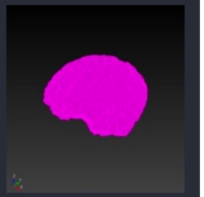
SAND2025-07529C

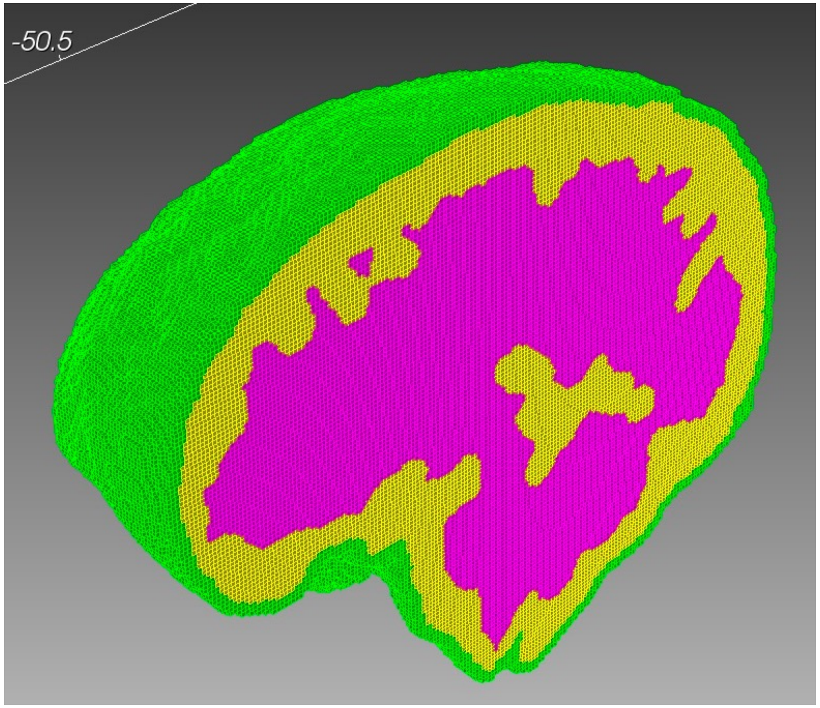
```
 SSSSS     CCCCC    UU   UU   LL         PPPPPP    TTTTTT
SS   SS   CC   CC   UU   UU   LL         PP   PP     TT
SS        CC        UU   UU   LL         PP   PP     TT
 SSSSS    CC        UU   UU   LL         PPPPPP      TT
     SS   CC        UU   UU   LL         PP          TT
SS   SS   CC   CC   UU   UU   LL         PP          TT
 SSSSS     CCCCC     UUUUU    LLLLLLL    PP          TT

                  PARALLEL HEX MESHING
                         FROM
                  VOLUME FRACTION DATA

          SCULPT Version 16.14.7 Build bf6ed33e6b
          Copyright 2015 Sandia Corporation
  Revised Fri Dec 15 08:36:16 2023 -0700
  User Support and Bug Reports: cubit-help@sandia.gov

 SCULPT includes CAMAL by Sandia National Laboratories
SCULPT includes CTH Diatoms by Sandia National Laboratories
SCULPT is a companion application to the CUBIT Geometry and
     Meshing Toolkit by Sandia National Laboratories
```

element count	mesh	all blocks *predominantly showing block 4 (void)*	block 4 removed *predominantly showing block 1 (bone)*	block 1 removed *predominantly showing block 2 (csf)*	block 2 removed *predominantly showing block 3 (brain)*
4,329,925	IXI012-HH-1211-T1_large.e.1.0 (220 MB)	 2,389,783	 240,895	 448,654	 1,250,593



- Closed-source
  - Inhibits collaboration
- Requires void inclusion
  - Bloats file size
- Only Laplace smoothing
  - Shrinks volume to zero
- ~0.1 million voxels/s

automesh

SAND2025-07529C

automesh is an open-source Rust software program that uses a **segmentation**, typically generated from a 3D image stack, to create a finite element **mesh**, composed either of hexahedral (volumetric) or triangular (isosurface) elements.

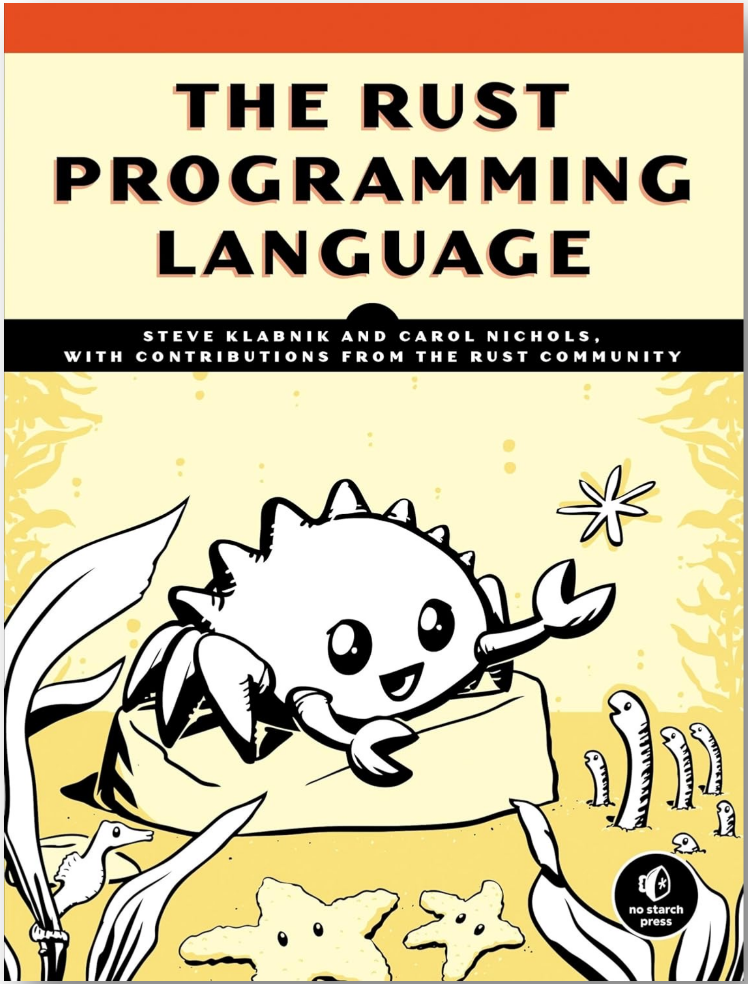
github.com/autotwin

automesh converts between segmentation formats (`.npy`, `.spn`) and mesh formats (`.exo`, `.inp`, `.mesh`, `.stl`, `.vtk`).

automesh can **defeature** voxel domains, apply Laplacian and Taubin **smoothing**, with hierarchical option, and output mesh quality **metrics**.

automesh uses an internal **octree** for fast performance.

- Rust is a modern programming language known for its **performance** and **memory safety**.
- Rust's **ownership model** prevents common programming errors such as null pointer dereferencing and buffer overflows, ensuring that automesh operates reliably and securely.
- Rust's **zero-cost abstractions** allow high performance without sacrificing code readability or maintainability.
- Rust's **strong type system** and **concurrency** features also enhance the robustness, making it well-suited for handling complex computational tasks efficiently.



Klabnik *et al.* 2023.

Most Loved Language

SAND2025-07529C

Technology
Most loved, dreaded, and wanted

Rust is on its seventh year as the most loved language with 87% of developers saying they want to continue using it.

Rust also ties with Python as the most wanted technology with TypeScript running a close second.

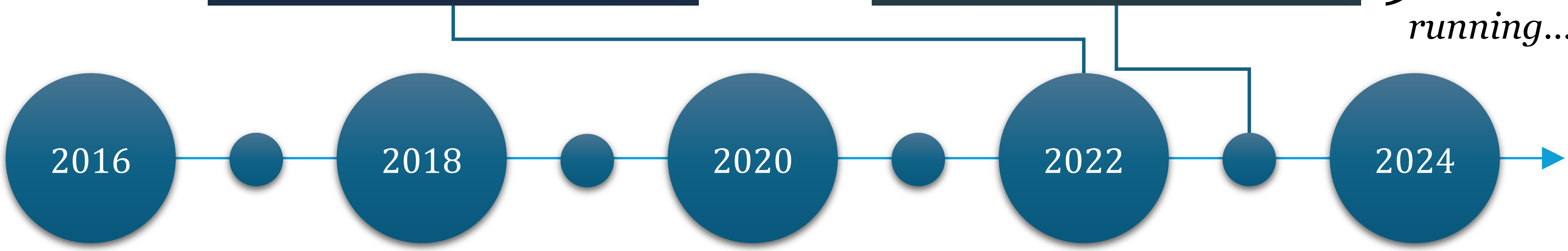
Programming, scripting, and markup languages →

Technology
Admired and Desired

Rust is the most admired language, more than 80% of developers that use it want to use it again next year. Compare this to the least admired language: MATLAB. Less than 20% of developers who used this language want to use it again next year.

Programming, scripting, and markup languages →

9 years running…



Stack Overflow 2022 and 2023.

Stop C/C++. Start Rust.

SAND2025-07529C

"Speaking of languages, it's time to halt starting any new projects in C/C++ and use Rust for those scenarios where a non-garbage-collected (GC) language is required. For the sake of security and reliability. the industry should declare those languages as deprecated," Russinovich said in a tweet.

Microsoft recently said that 70 per cent of its security patches in the last 12 years were fixes for memory safety **bugs** "due largely to **Windows** being written mostly in C and C++".

Google Chrome also said that 70 per cent of all serious security vulnerabilities in the Chrome codebase were memory management and safety bugs written mostly in C++.

Economic Times 2019.

Industry Adoption

MIT Technology Review

How Rust went from a side project to the world's most-loved programming language

For decades, coders wrote critical systems in C and C++. Now they turn to Rust.

By Clive Thompson

February 14, 2023

Mozilla

LINUX

Google

Microsoft

Discord

Meta

amazon

SAMSUNG

Dropbox

MIT Technology Review 2025.

US Government

SAND2025-07529C

White House Recommends Memory-Safe Programming Languages and Security-by-Design

Published March 4, 2024 |  Written by Megan Crouse

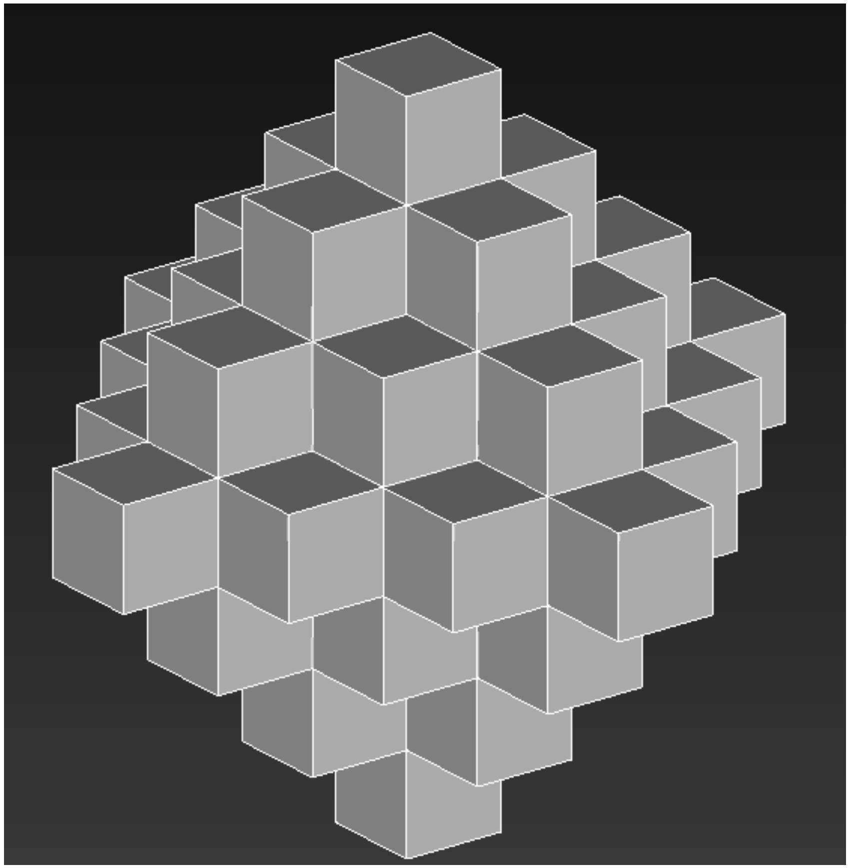
The government would prefer it if you stopped programming tools in C or C++. In a new report, the White House Office of the National Cyber Director (ONCD) has called on developers to use "memory-safe programming languages," a category which excludes the popular languages. The advice is part of U.S. President Biden's Cybersecurity strategy and is a move to "secure the building blocks of cyberspace."

NSA suggested memory-safe programming languages:
*Ada, C#, Delphi/Object Pascal, Go, Java, JavaScript, **Python**, Ruby, **Rust**, Swift.*

TechRepublic 2024.

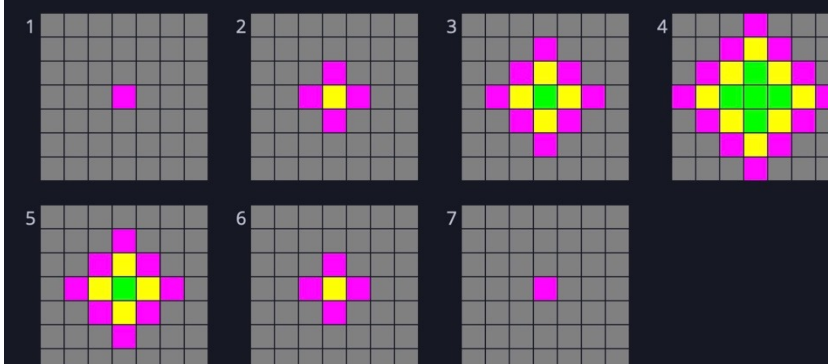
Segmentation Concept

SAND2025-07529C

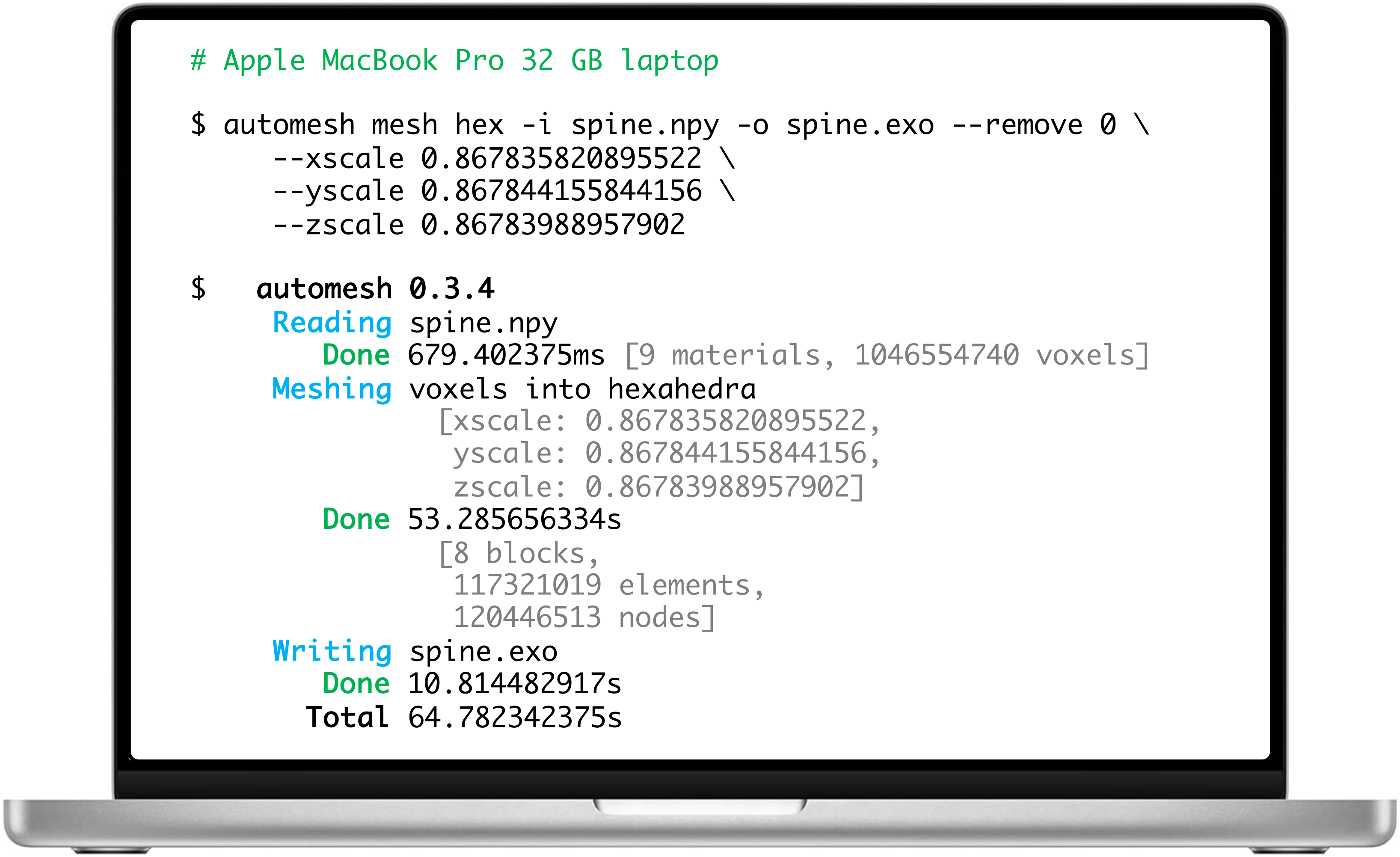


0	0	0	3	0	0	0
0	0	3	2	3	0	0
0	3	2	1	2	3	0
3	2	1	1	1	2	3
0	3	2	1	2	3	0
0	0	3	2	3	0	0
0	0	0	3	0	0	0

Consider each slice, `1` to `7`, in succession:



Command Line Interface

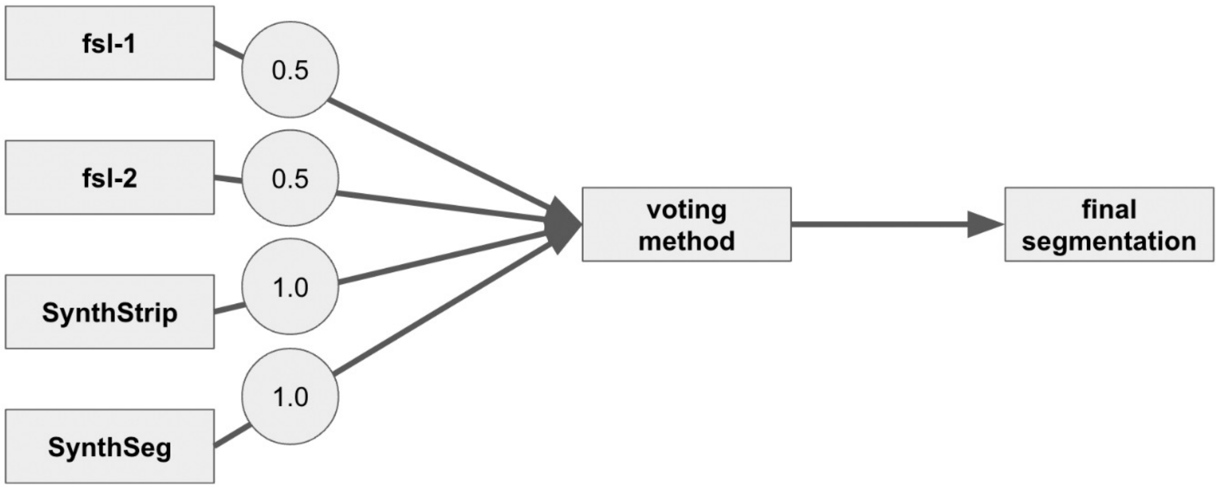


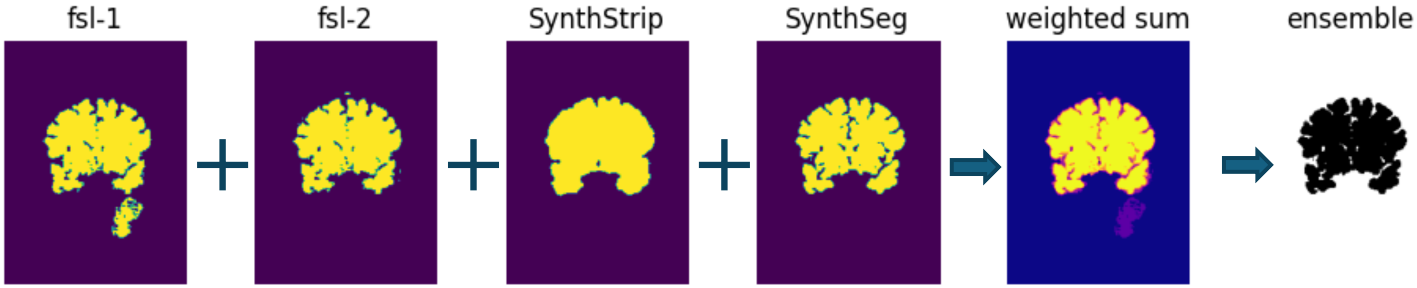
~1 billion voxels, 117 million elements, ~1 minute

Research Outcomes

Image to Segmentation

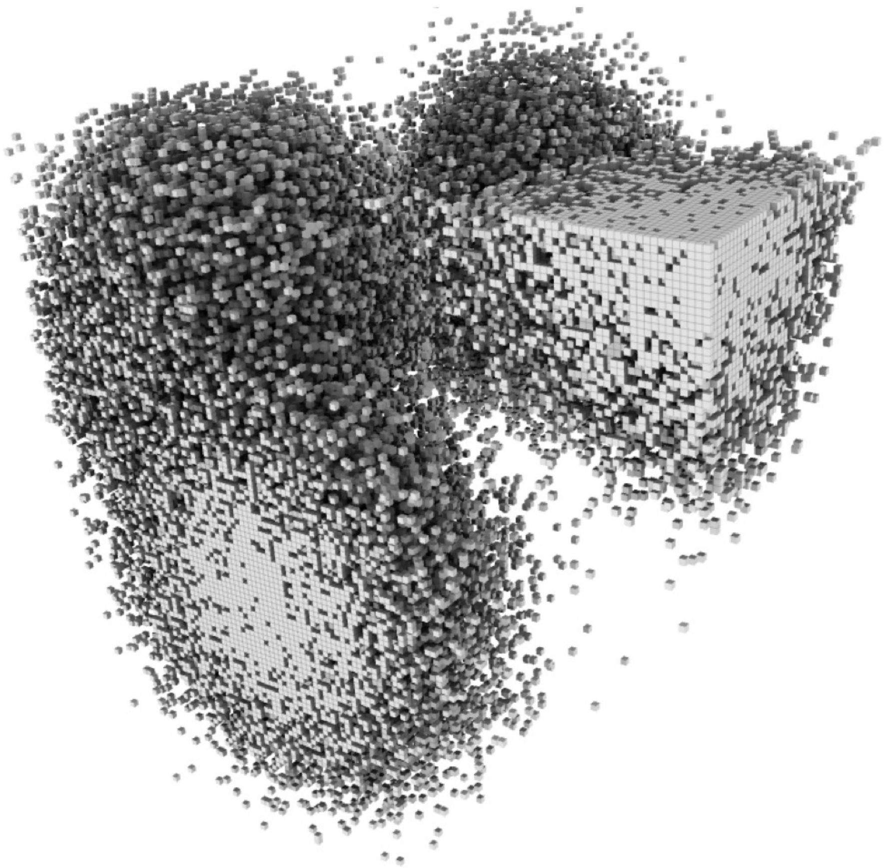
SAND2025-07529C



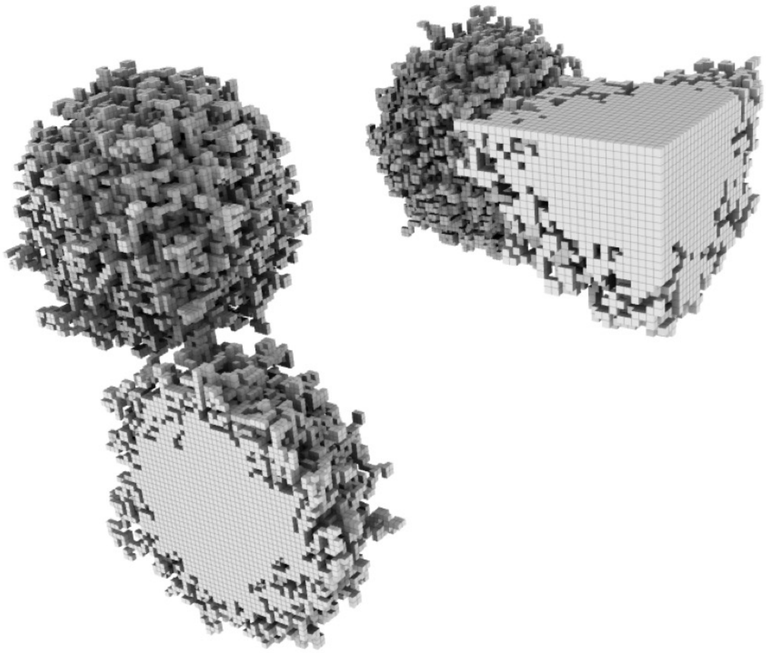


Defeaturing

Before Defeaturing

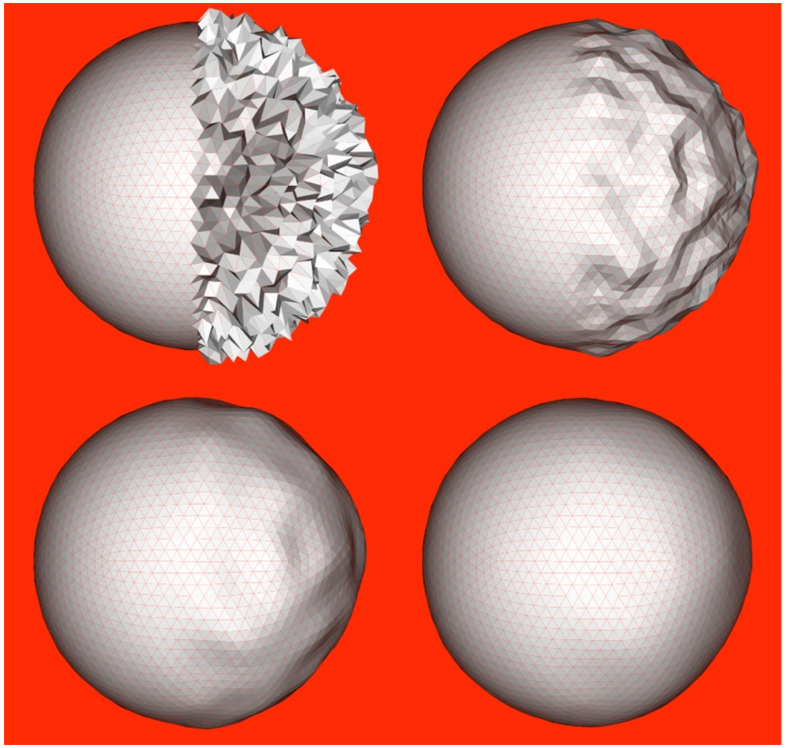


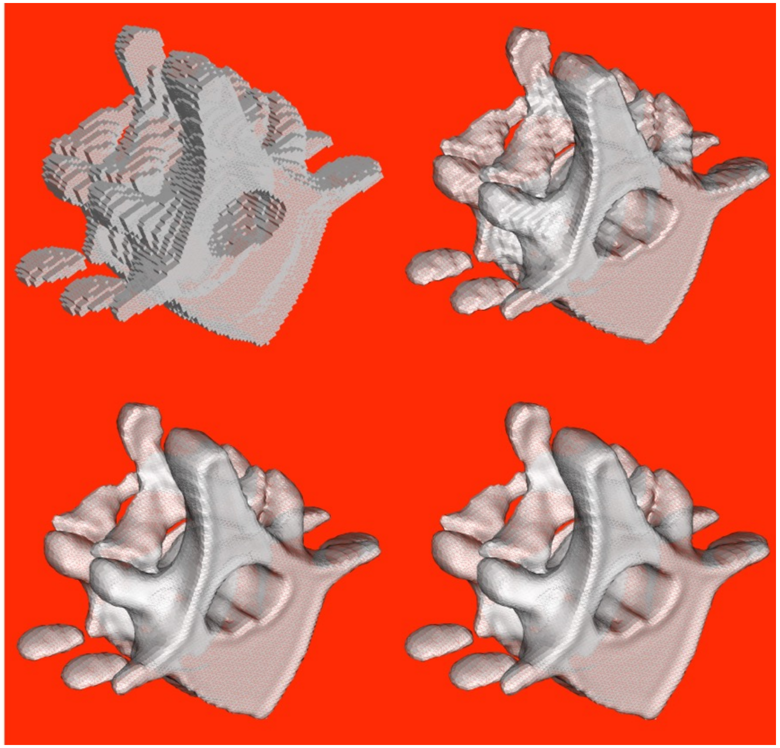
After Defeaturing



Smoothing

SAND2025-07529C

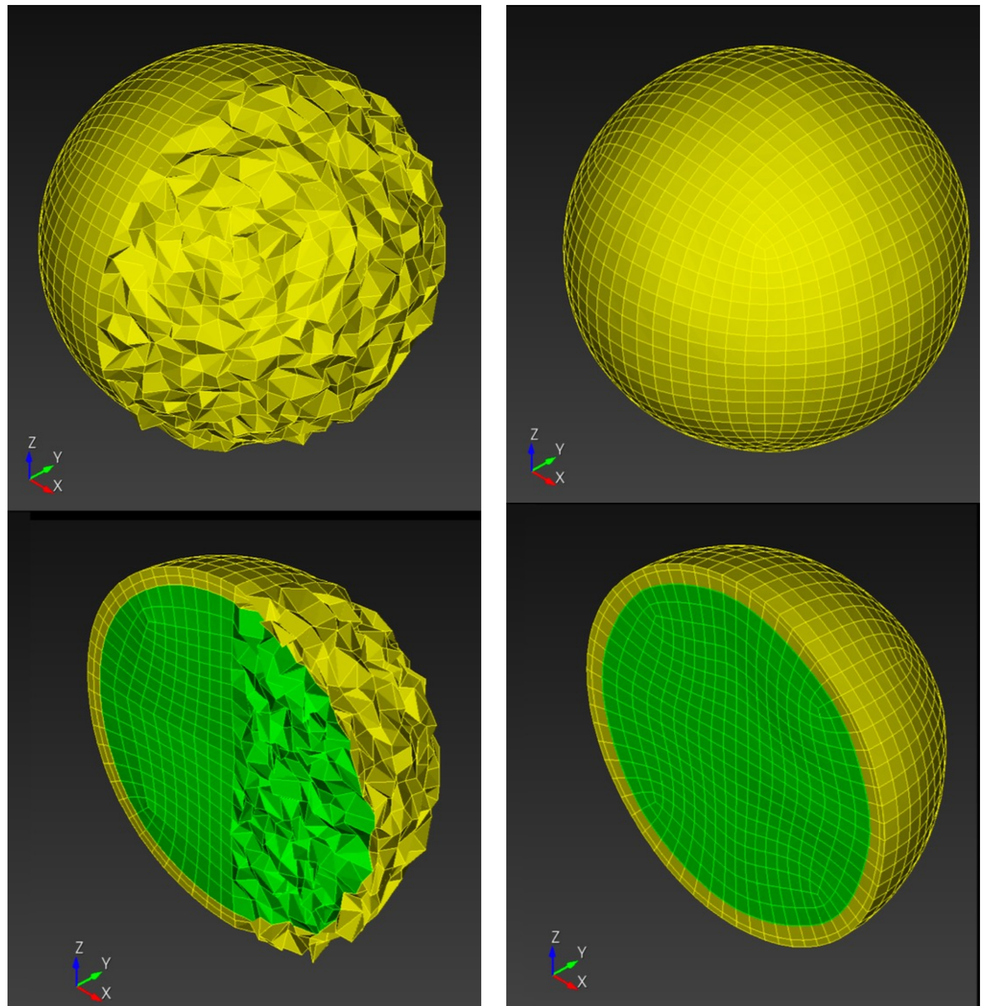




Taubin 1995.

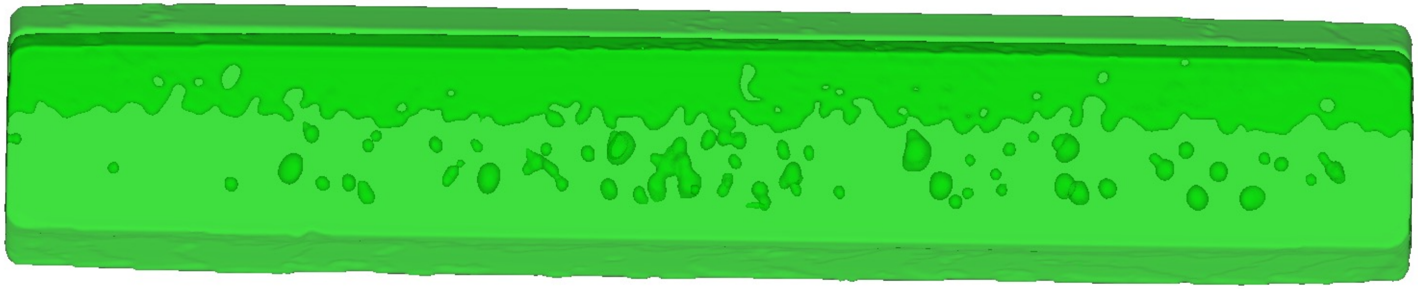
Smoothing

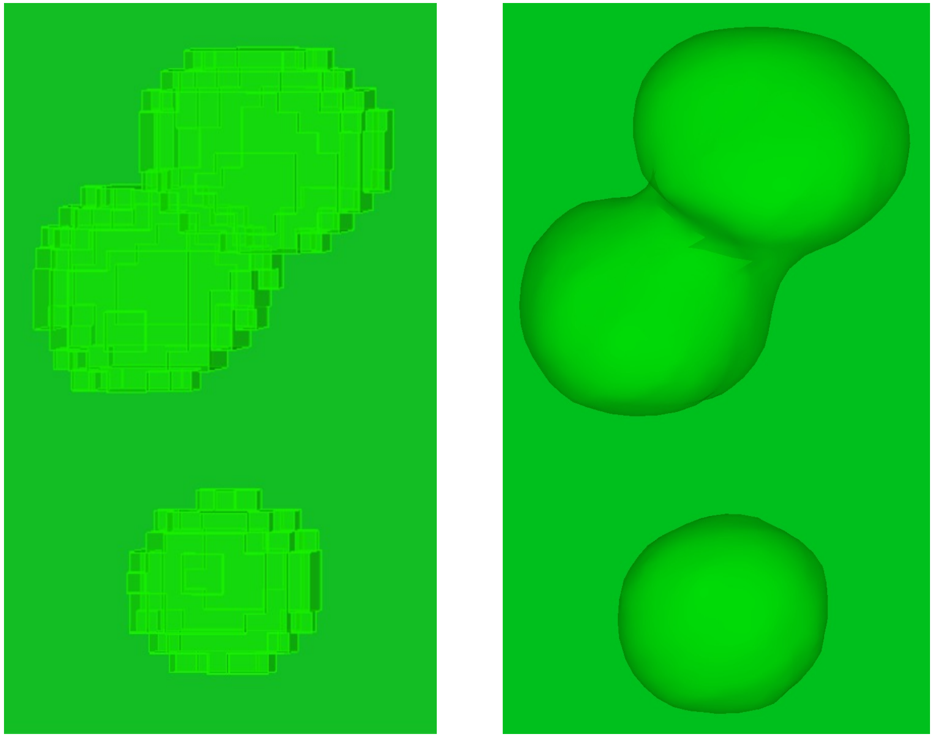
Before Smoothing | After Smoothing



Surface Reconstruction

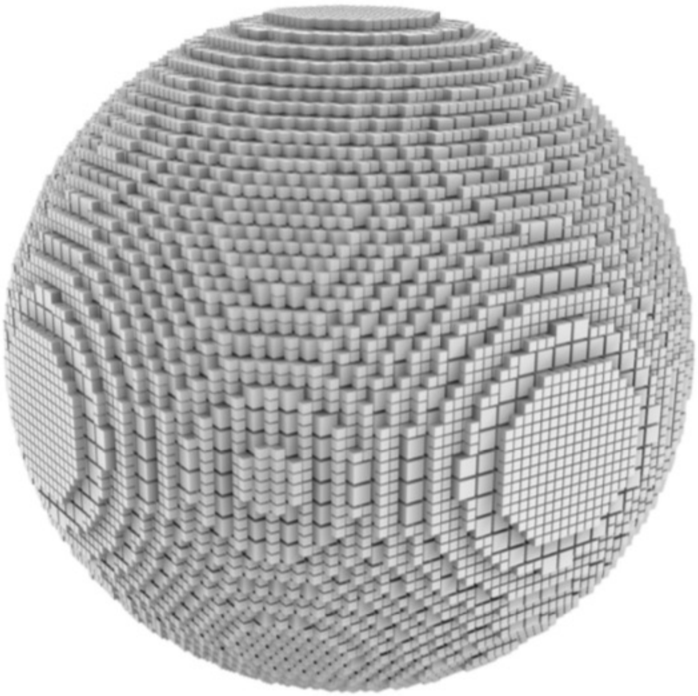
SAND2025-07529C

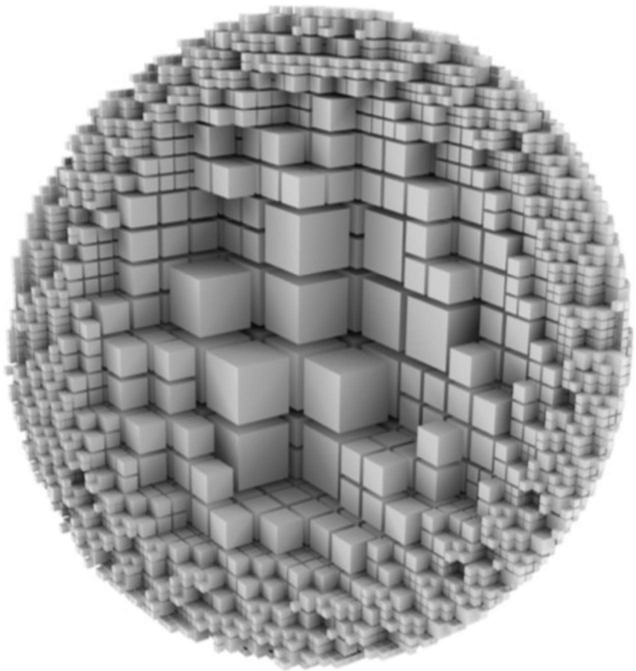




Octree

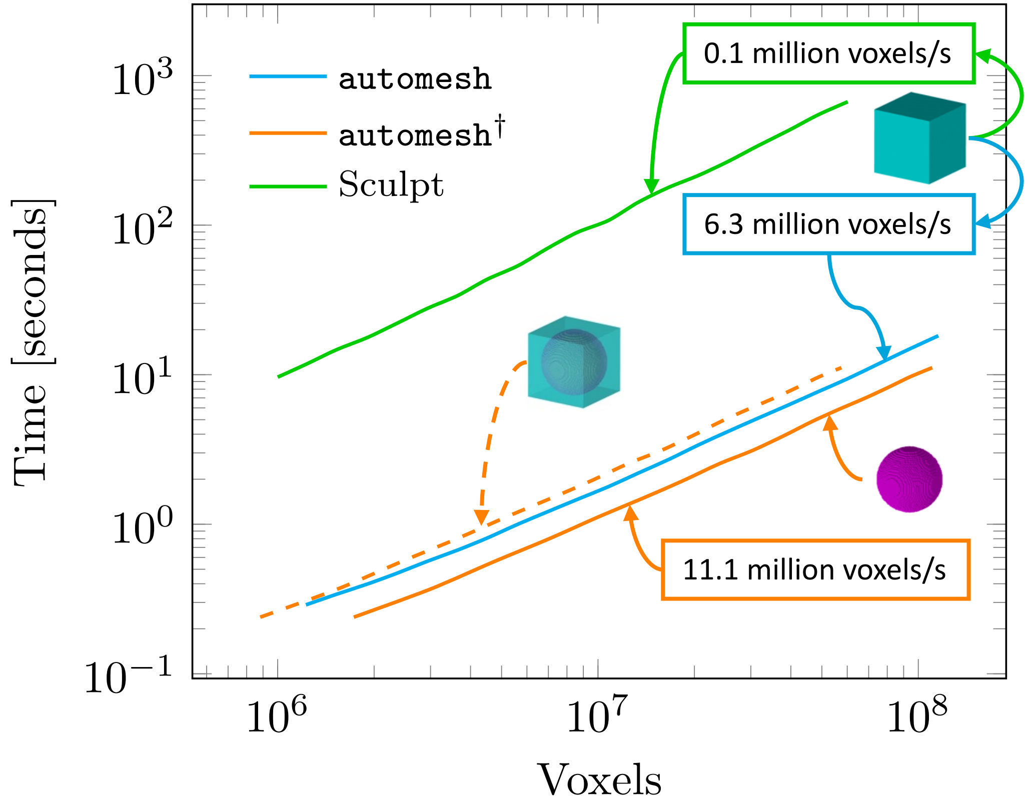
SAND2025-07529C





Performance

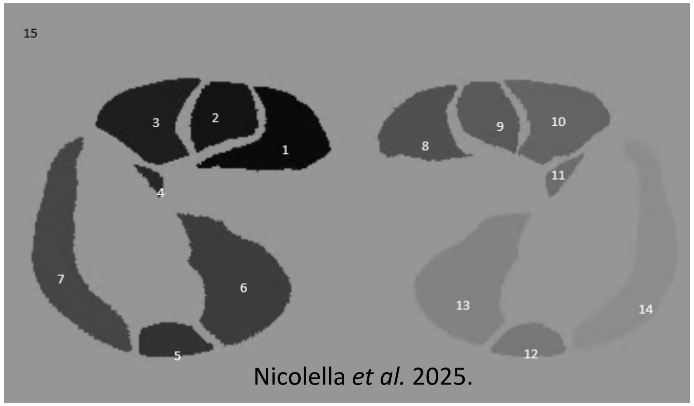
- Time for direct voxel-to-hex mesh
  - Cube of single material
  - Embedded sphere†
- Ideal *O(N)* scaling, single thread rates:
  - 6.3 million voxels/s
  - 11.1 million voxels/s†
  - *Sculpt 0.1 million voxels/s*
- Memory of 125 GB
  - 1 billion voxels (~1 min)
  - *Sculpt max: ~100 million voxels*



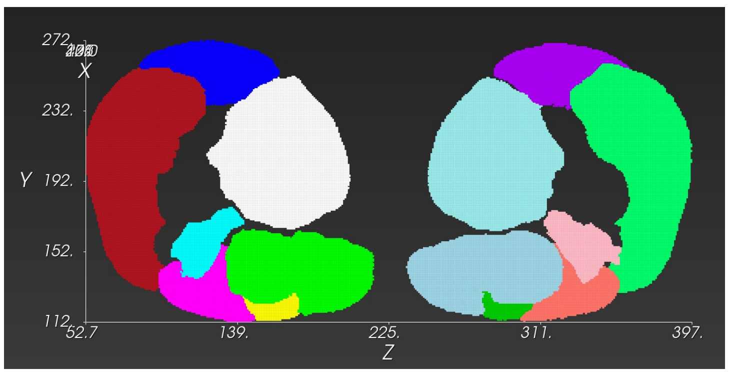
Performance speedup: **100x**

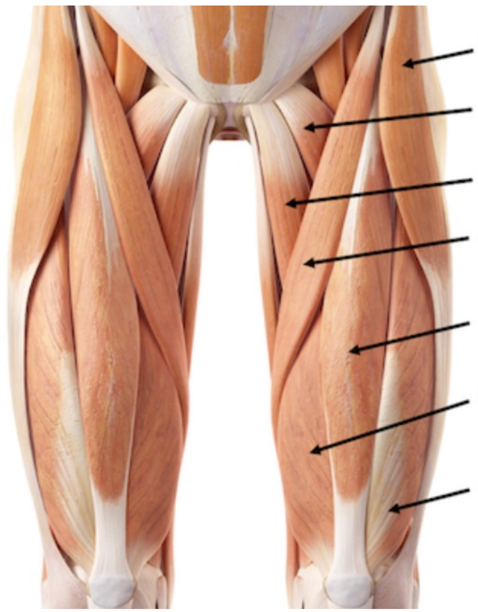
Success Stories

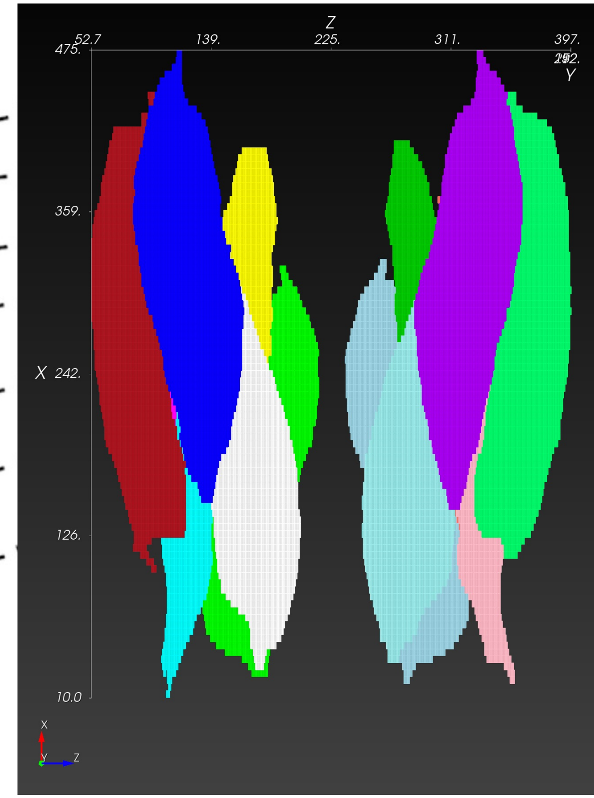
SAND2025-07529C



Nicolella *et al.* 2025.







Segmentation to Mesh: Upper limb muscles

Upper limb leg muscles (muscle_slices.npy) from a clinical MRI. The images are 512 x 512 pixels (450 x 450 mm) with a depth of 110 slices (550 mm or 5mm/slice). There are 14 muscles (values 1 – 14). The background is value 15.

Scientific Impact

Demonstration of workflow capabilities:

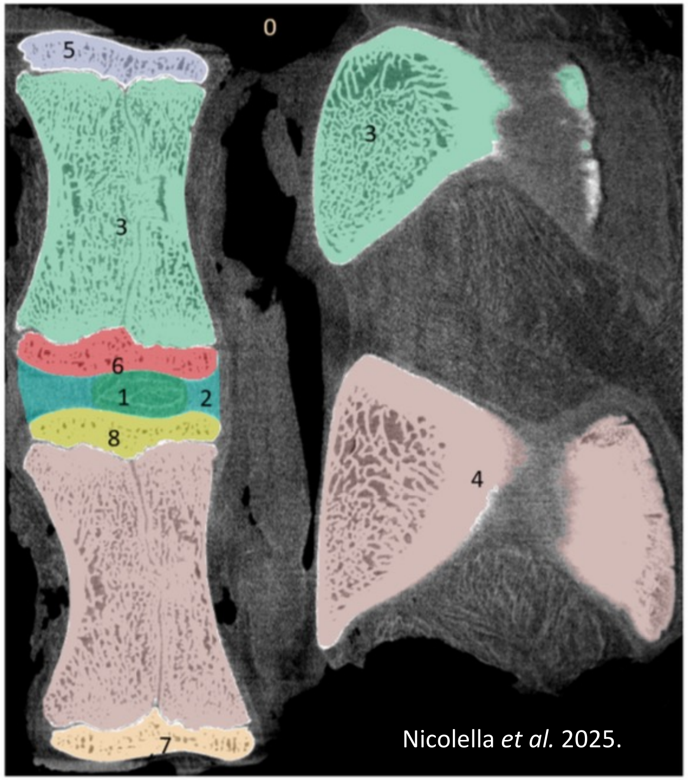
- Segmentation -> Mesh

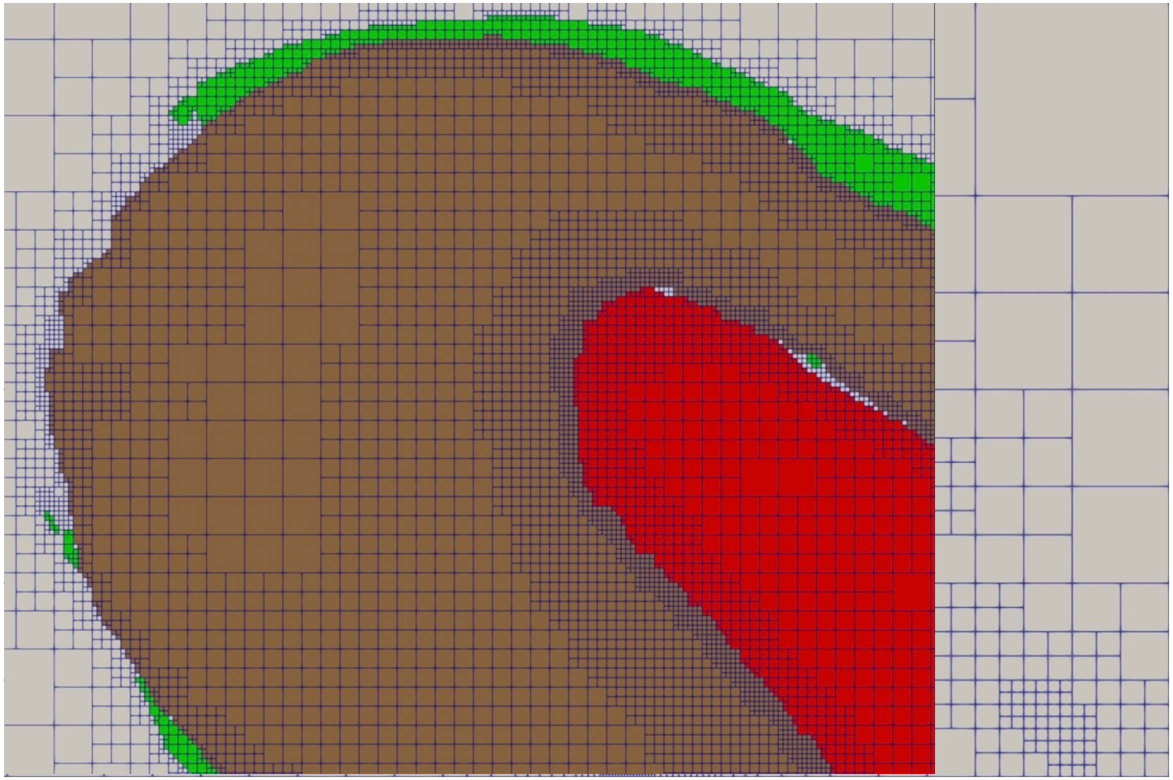
Naval Relevance

- Critical step toward automated human digital twin.
- Empower Naval collaborators with automated tools.

Success Stories

SAND2025-07529C





Segmentation to Mesh: Functional Spinal Unit

Functional spinal unit from a uCT; the images have a width of 1449 pixel (1257.5 mm), height of 770 pixel (668.24 mm) and depth of 938 slices (814.03 mm). There are 8 objects (numbered 1-8) plus the background (numbered 0).

Scientific Impact

Demonstration of workflow capabilities:

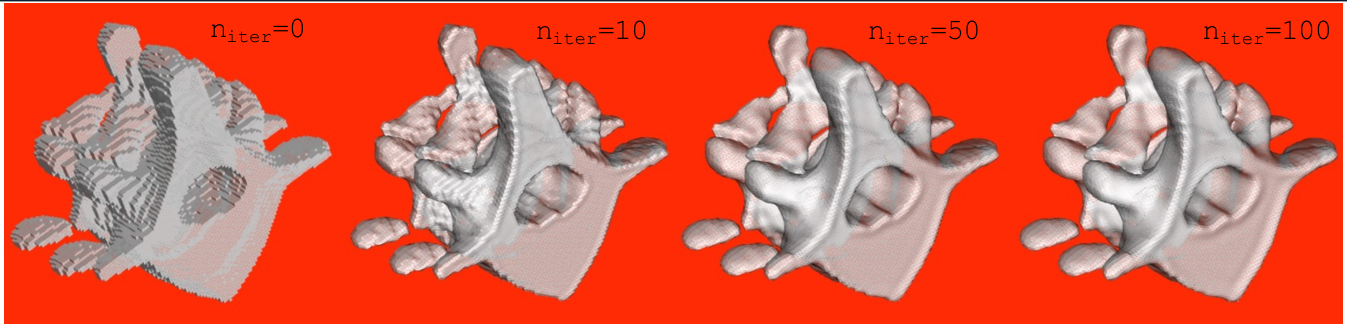
- Segmentation -> Mesh

Naval Relevance

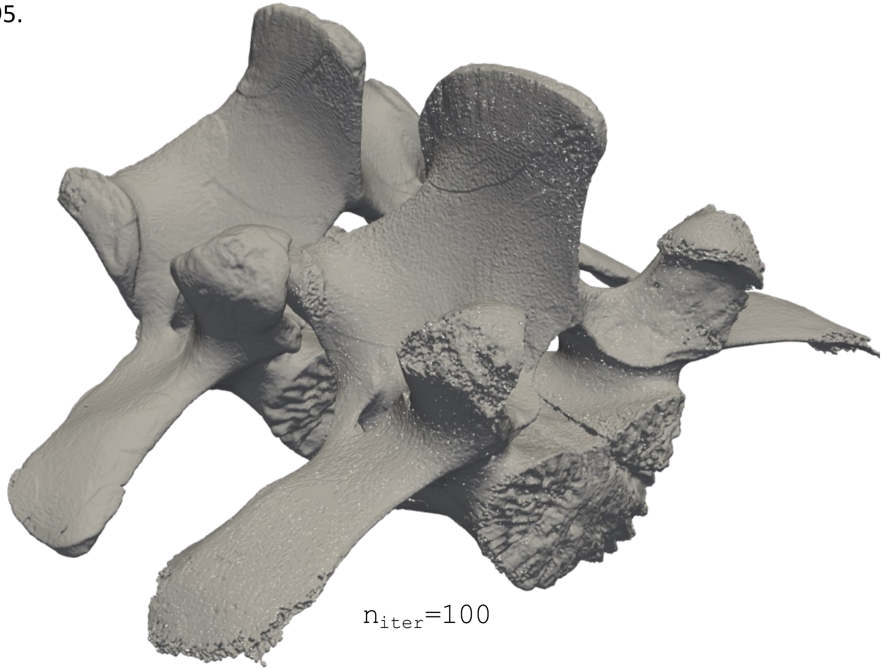
- Progress toward automated human digital twin.
- Empower Naval collaborators with automated tools.

Success Stories

SAND2025-07529C



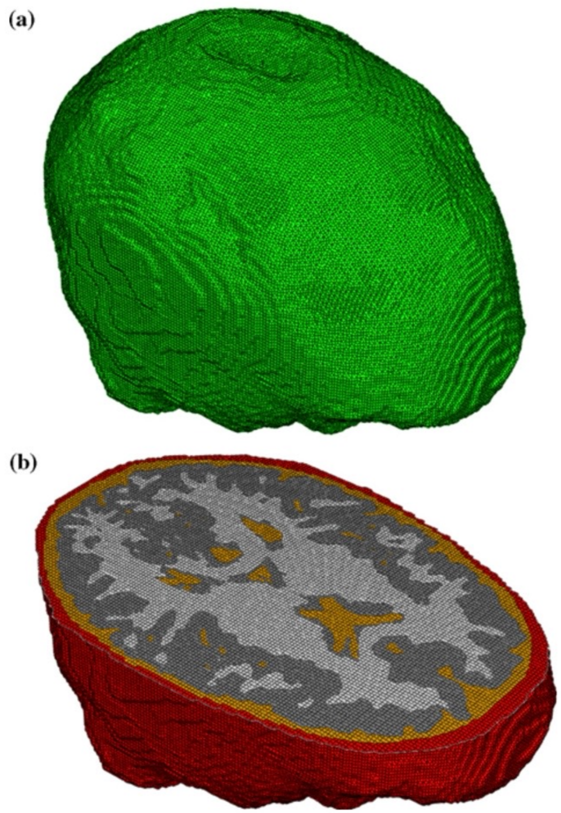
Taubin 1995.




```
$ automesh defeature -i spine.npy -o spine_defeature_20.npy -m 20
$ automesh mesh hex --remove 0 -i spine_defeature_20.npy \
      --xscale 0.867835820895522 --yscale 0.867844155844156 --zscale 0.86783988957902 \
      -o spine_without_void_defeature_20_s100.exo smooth --hierarchical --iterations 100
```

Success Stores

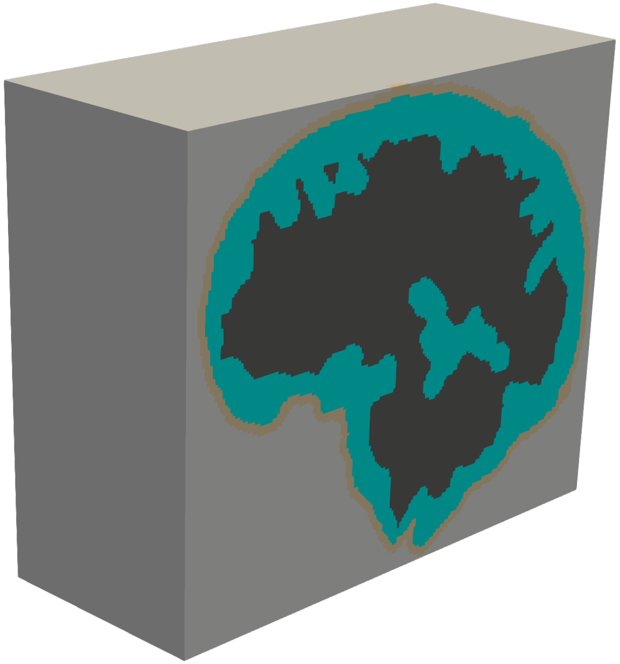
SAND2025-07529C



Chen *et al.* 2010

Automated IXI Data Set

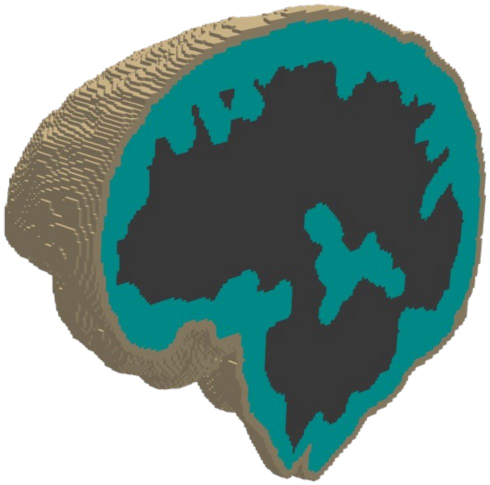
Comparison of Chen singleton to **`autotwin`** segmentation of mesh workflow.



`automesh` with void
IXI012-HH-1211-T1_large_with_void

Scientific Impact

- Open-source workflow, guaranteed reproducibility
- Exclude unnecessary void (arbitrary exclusions)
- Taubin smoothing with hierarchical control



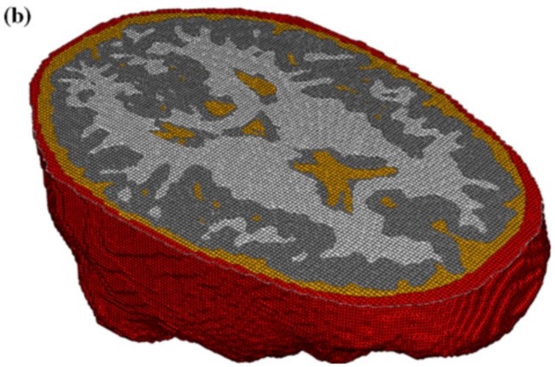
`automesh` without void
IXI012-HH-1211-T1_large_without_void

Naval Relevance

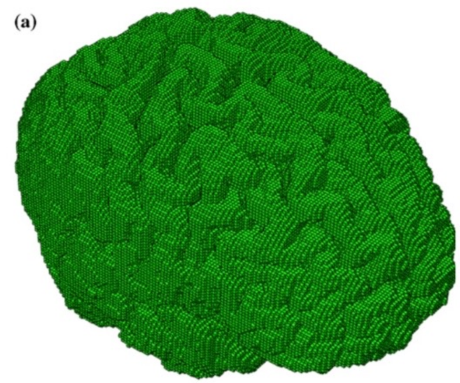
- Critical step toward building the human digital twin.
- Empower Naval collaborators with automated tools.

Mesh of Brain

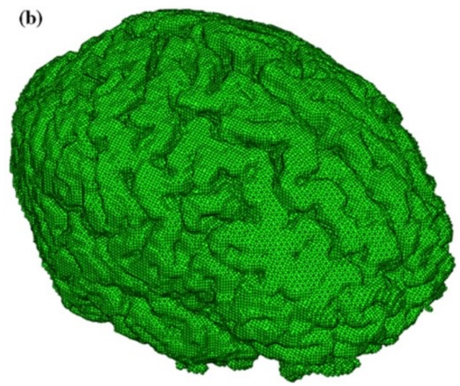
SAND2025-07529C



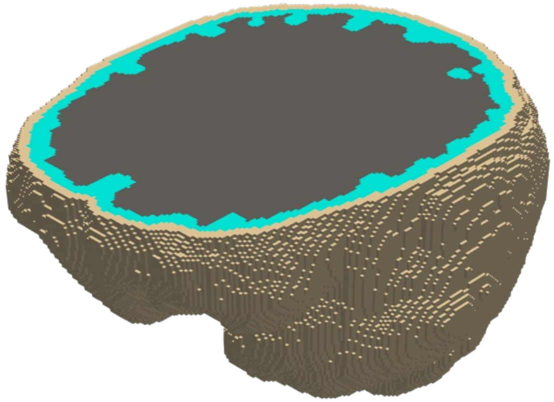
Chen *et al.* 2010



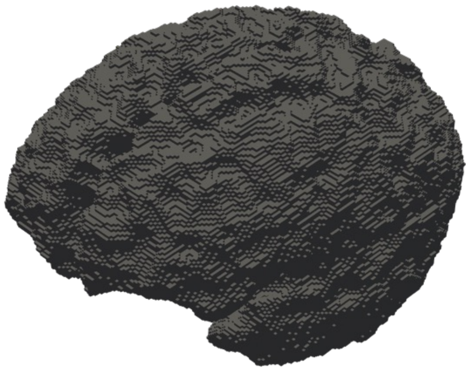
Before smoothing



After hierarchical Taubin smoothing
8 iterations



`automesh`



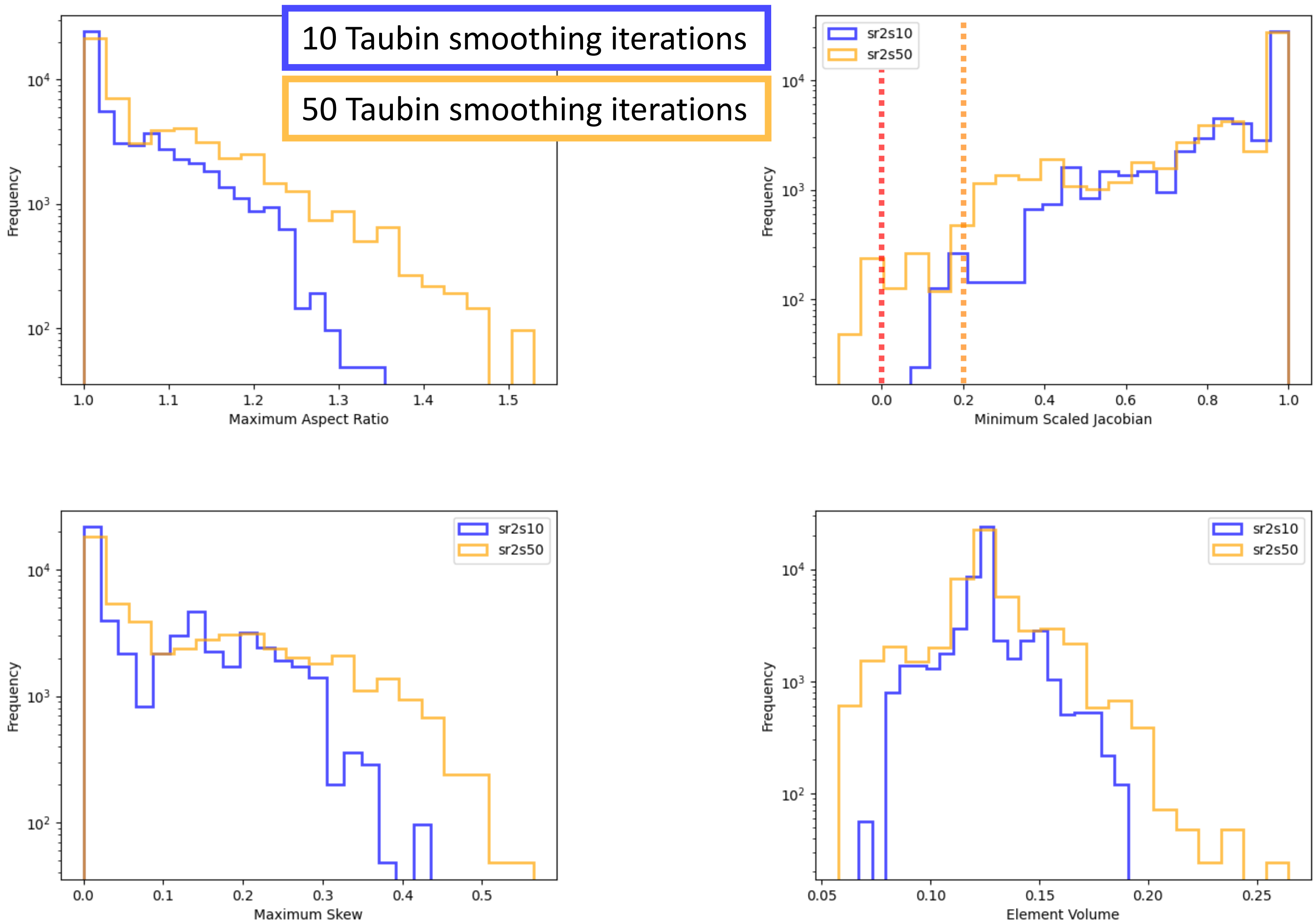
Before smoothing

After hierarchical Taubin smoothing
8 iterations

Issues

Smoothing v. Quality

SAND2025-07529C



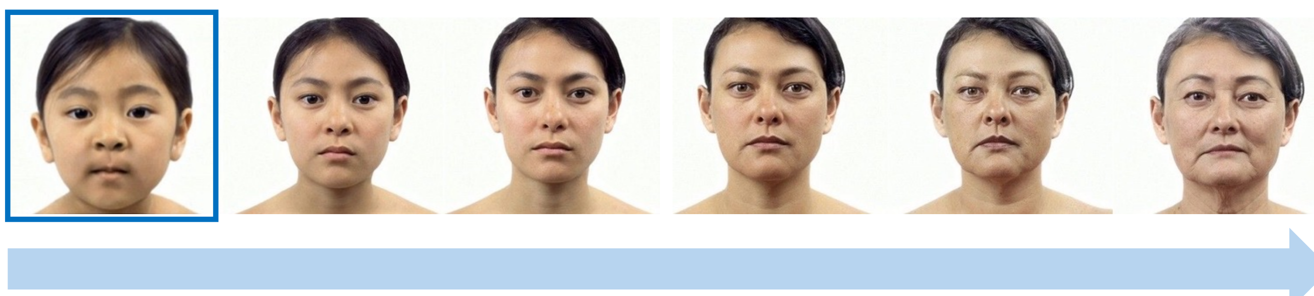
Conclusions

SAND2025-07529C

- Segmentation to mesh workflow is automated.
  - But there is still room for improvement…
- Automated workflows are required to scale.
  - Populations, time-evolutions, length-scale resolution.

Geometry is patient-specific.

Geometry is patient-specific and can be time-specific too.



Cerniello 2015.

Path Forward

- Smoothing
  - Existing smoothing techniques can erode quality.
  - **Energetic smoothing** shows promise as an alternative.
- Tetrahedral meshes
- Adaptive all-hex meshes
- Surface conformity

Path Forward

SAND2025-07529C

Activities Schedule	FY26	FY27	FY28	FY29	FY30
Objective 1					
Task 1: Increase segmentation fidelity.					
Task 2: Solver discussions, alternatives to ABAQUS (open-source?)					
Task 3: Injury risk metric integration.					
Task 4: Enhance meshing capabilities					
Task 5: Integration of Task1 through Task 4.					
Task 6: **Deployment** to broader biomechanics community.					

Start Tests, Demos, & Key Events Milestone S&T Delivery

Conferences & Papers

SAND2025-07529C

1. Tripathi A, Buche M, Carlsen R, Hovey C, Lejeune E, “Effect of Meshing Techniques Used in Subject-Specific Human Digital Twins for Mild Traumatic Brain Injury Risk Assessment,” submitted to Military Health System Research Symposium (MHSRS), Kissimmee, FL, 2025.
2. Polonsky A, Hovey C, Lamb J, Chao P, Echlin M, Lim H, Johnson K, Deitz J, Pollock T, submitted to The 7th International Congress on 3D Materials Science (3DMS 2025), Anaheim, CA, 2025.
3. Buche M, Hovey C, “Fast and automatic mesh generation from segmentation,” 2nd Solid Mechanics Analysis Resource and Technical Exchange, Livermore, CA 2025.
4. Polonsky A, Chao P, Hovey C, Bordas A, Oakley R, Deitz J, Emery J, “Directing Flow: Pipelines for 3D Data,” invited to Minerals, Metals & Materials (TMS), Las Vegas, NV, 2025.
5. Tripathi A, Buche M, Carlsen R, Hovey C, Lejeune E, “Automated, Open-Source Workflow for Creation of Human Digital Twins at Scale,” International Mechanical Engineering Congress & Exposition (IMECE), Portland, OR, 2024.
6. Polonsky A, Chao P, Hovey C, Bordas A, Oakley R, Deitz J, Emery J, “Directing Flow: Pipelines for 3D Data,” invited to Minerals, Metals & Materials (TMS), Las Vegas, NV, 2025.
7. Polonsky A, Hovey C, Lamb J, Chao P, Echlin M, Lim H, Johnson K, Deitz J, Pollock T, “,” submitted to The 7th International Congress on 3D Materials Science (3DMS 2025), Anaheim, CA, 2025.

NEW

8. Hovey CB, Buche MR, automesh: Automatic mesh generation in Rust, submitted to The Journal of Open Source Software (JOSS), 2025.

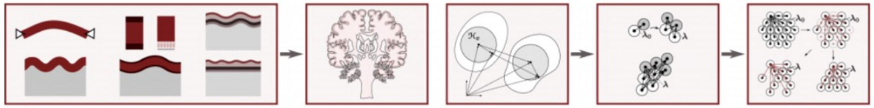
Collaboration & Discussion

SAND2025-07529C

BU College of Engineering
Department of Mechanical Engineering
Lejeune Lab

Lejeune Lab

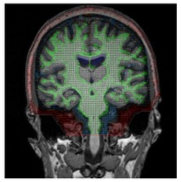
Our goal is to leverage the state of the art in computational mechanics to investigate multiscale emergent behavior in biological and bio-inspired systems, and make predictions about mechanical phenomena. Current areas of research involve (1) integrating data-driven and physics based computational models and (2) predicting the behavior of highly heterogeneous soft materials. As a group, we are trying our best to share the code and datasets associated with our publications under open source licenses.



IBL INJURY BIOMECHANICS LAB

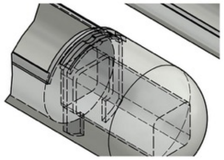
About People Research Teaching & Outreach Open Positions More

Research Projects



Multiscale Computational Modeling of Concussive Injury

We are creating biofidelic computational models of concussive injury that accurately account for the structural and functional damage that occurs across length scales - from the tissue level down to the subcellular level. These models can be applied to study how mechanical loads to the head at the macroscale translate to damage at the cellular and subcellular level, which can ultimately lead to neurodegeneration and loss of cogntiive and motor function.



Experimental Platform for Simultaneous Mechanical Testing and Imaging

We are developing experimental testing platforms that enable tissue samples to be mechanically loaded while being imaged at the same time. These testing platforms will allow us to study the evolution of structural damage that occurs in biological tissue during mechanical loading.

References

Cerniello, Anthony. Filmmaker uses digital wizardry to age face in five minutes. SBS News, February 26, 2015.

Chen Y, Ostoja-Starzewski M. MRI-based finite element modeling of head trauma: spherically focusing shear waves. Acta mechanica. 2010 Aug;213(1):155-67.

Economic Times 2019. https://economictimes.indiatimes.com/tech/technology/it-is-time-to-shun-c-c-languages-for-new-projects-microsoft-azure-cto/articleshow/94453692.cms?from=mdr, citing Microsoft® 2019, "Trends, challenges, and strategic shifts in the software vulnerability mitigation landscape", and Google 2021, "An update on Memory Safety in Chrome".

Klabnik S, Nichols C. The Rust programming language. No Starch Press; 2023 Feb 28.

MIT Technology Review 2025. https://www.technologyreview.com/2023/02/14/1067869/rust-worlds-fastest-growing-programming-language/

Nicolella, D. P., Shaffer, S. K., Frazer, L. L., Pant, A. D., & Kote, V. B. 2025. I-PREDICT. MTEC.

Stack Overflow 2022 and 2023. https://survey.stackoverflow.co/2022, https://survey.stackoverflow.co/2023

Taubin G. A signal processing approach to fair surface design. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques 1995 Sep 15 (pp. 351-358).

TechRepublic 2024. https://www.techrepublic.com/article/white-house-report-memory-safe-programming-languages/, and https://www.tomshardware.com/software/security-software/white-house-urges-developers-to-avoid-c-and-c-use-memory-safe-programming-languages