New Tools for Automating Arcjet Sample Recession Tracking and Analysis

Ablation Workshop, November 2024

Alexandre Quintart
Flying Squirrel
Magnus Haw
NASA Ames Research Center
Sebastian Colom
NASA Ames Research Center

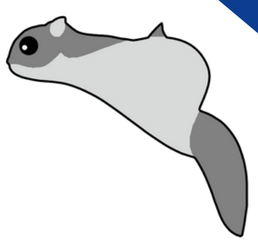


Table of contents

1. Introduction

- NASA heatshields **require** extensive ground testing and modeling.

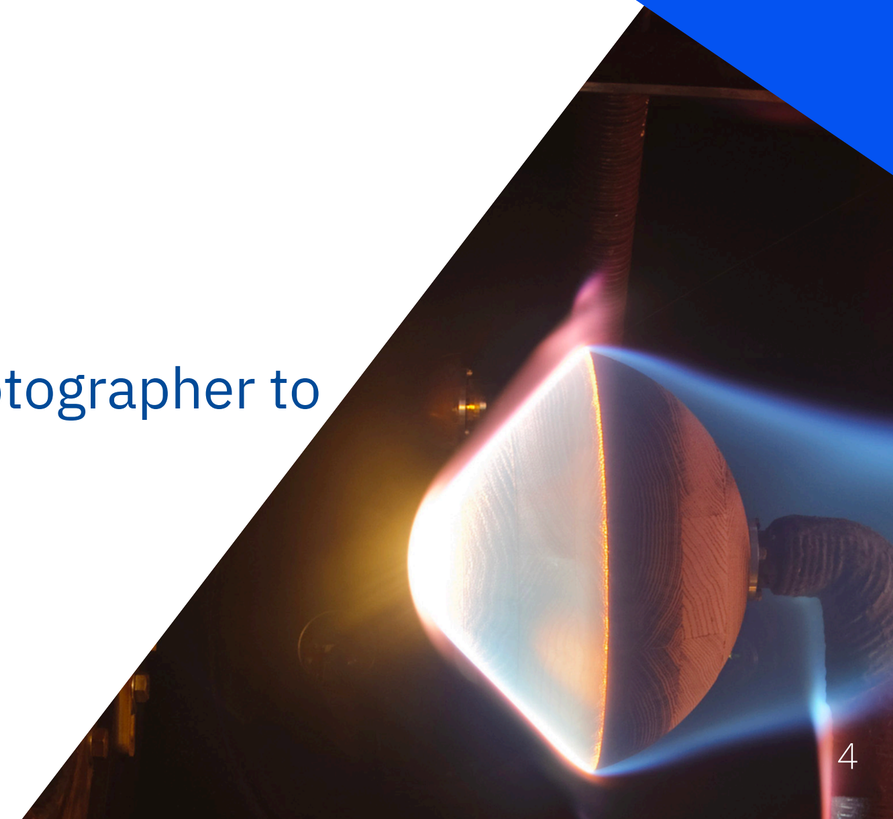
- Ground tests are **expensive** and complex but necessary to evaluate material performance.

→ **Maximizing Data** Extraction in Each Arcjet Test is important !

2. Motivation

Current capabilities:

- **arcjetCV:**
  - Automated recession analysis from video.
  - 2D model validation.
  - Time resolved validation.
- **Manual Photography**
  - 4 views per model, requires lead time for facility photographer to process.
- **Laser scanner:**
  - Pre/post test 3D scans (10 minutes/model).



2. Motivation

Goals:

- **arcjetCV improvement:**
  - Improve machine learning model for recession tracking from videos.
  - Simplify software distribution.

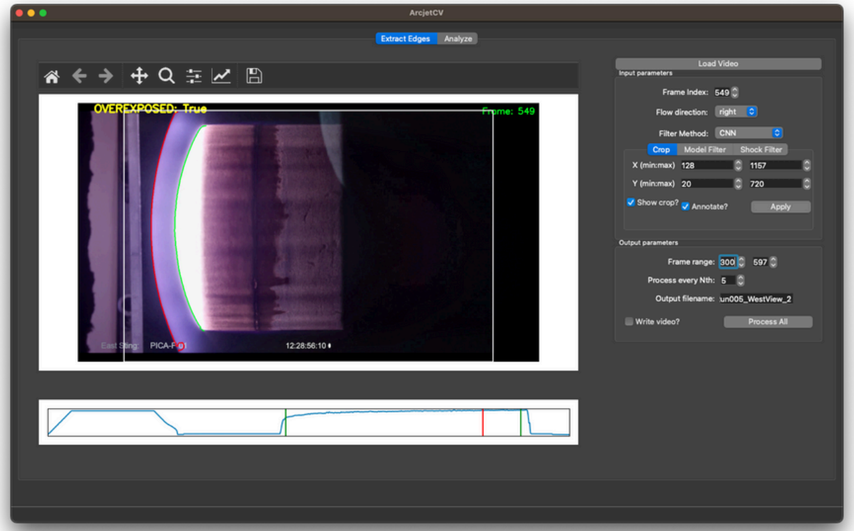
- **STARscan development:**
  - Automate 3D reconstruction, imaging, alignment and analysis scanner.
  - Eliminate pretest imaging and 3D scanning bottleneck.

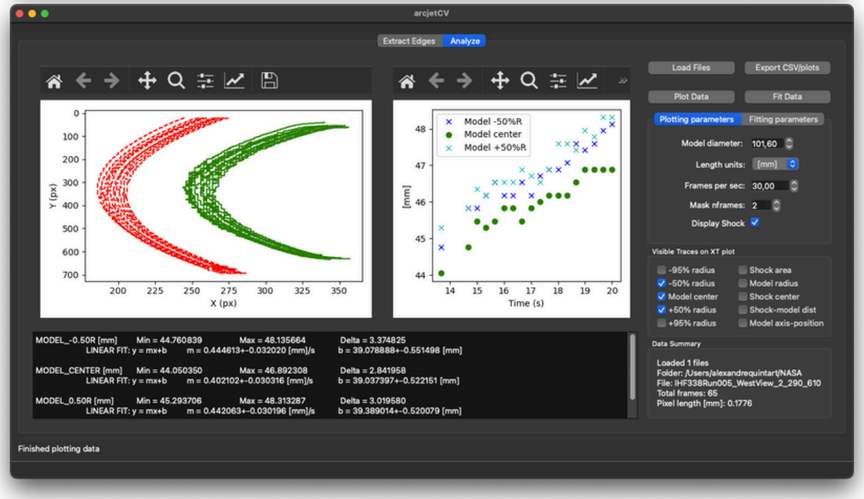
Recession from videos

arcjetCV

Ablation analysis during test

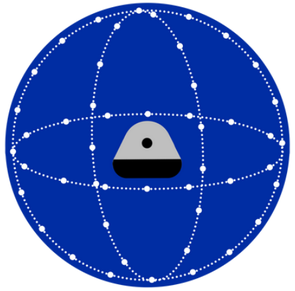
- **open-source software** tool for analyzing heatshield material performance during arcjet tests.
- **Machine learning** (CNN) models for automated video segmentation.
- Tracks **time-dependent** material **recession** and shock standoff distance.
- GUI for **user-friendly** operation; API for batch processing.



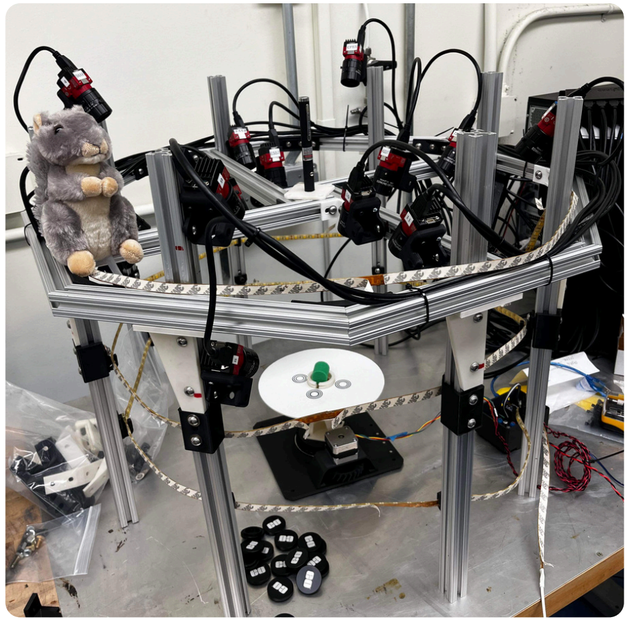


New photogrammetry scanner

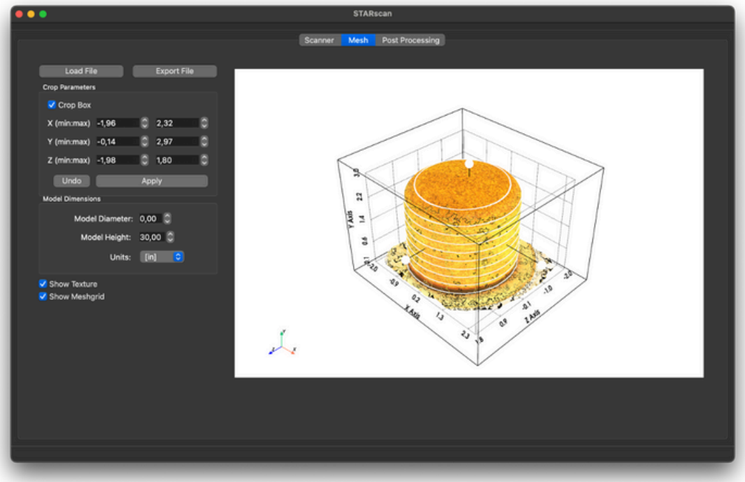
STARscan



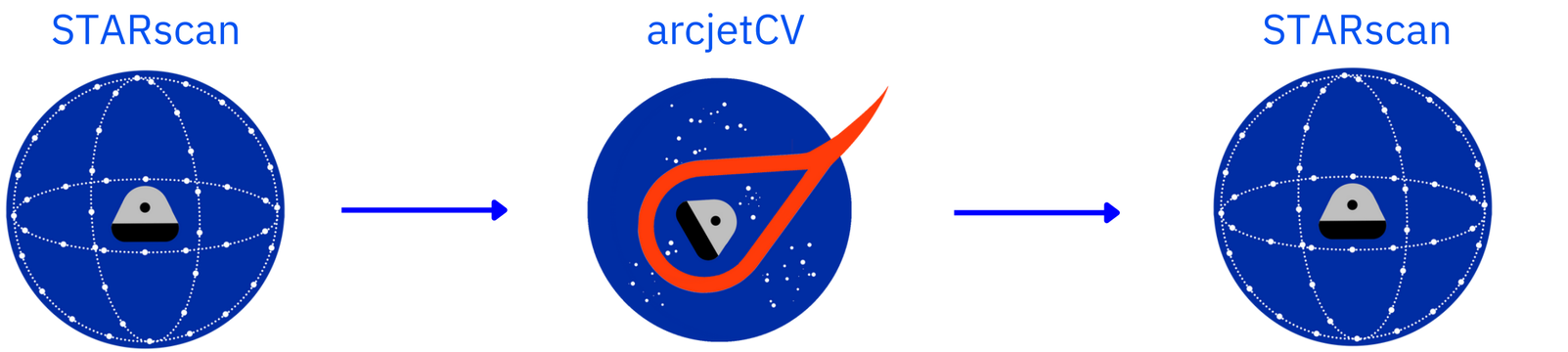
Scan sample before and after test



- 2 custom **3D scanning systems** built to accelerate and streamline scanning and analysis of arcjet test articles.
- **Software** that provides analysis of the shoulder **curvature**, **roughness** and **recession**.
- **Principle:** Photogrammetry
- **Accuracy:** ~ 0.35mm
- **Scan time:** 2 min



New Arcjet Test Analysis Process



Benefits:

- Cross-validation.
- In house 3D analysis pipeline.
- Standard imaging of models.
- Faster analysis and scanning.

arcjetCV improvement

Goals

- Increase the accuracy of the convolutional neural network detection.
- Make it easier to distribute the software around.

Improvements:

- New edge class
- New architecture
- Expanded training dataset
- Distribution on pypi (Python Package Index)
- Available as an executable for Windows (.exe) and macOS (.app)
- Distribution on conda

arcjetCV: Edge class

Problem:

- Standard CNN architectures may **struggle** with accurate **edge detection** in samples where boundary details are critical.
- Misclassification or blurring of edge regions can lead to **errors** in measuring **recession**.

Solution:

- Introduce a **dedicated class** for sample edges.

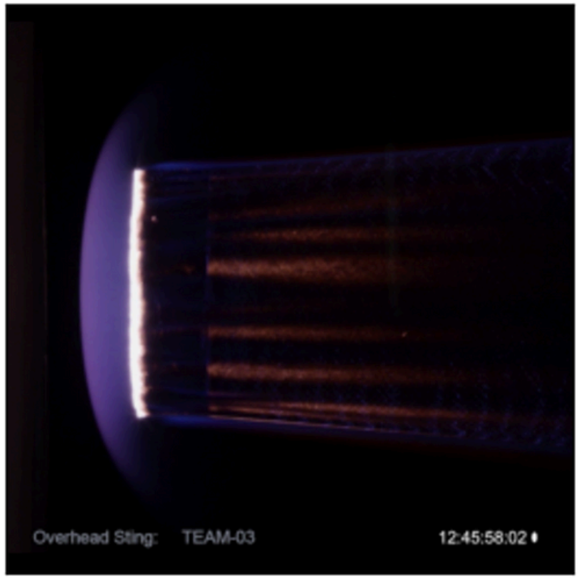
→ This class specifically labels boundary areas where material recession occurs, allowing the model to **focus** on fine **edge** details.

Advantages:

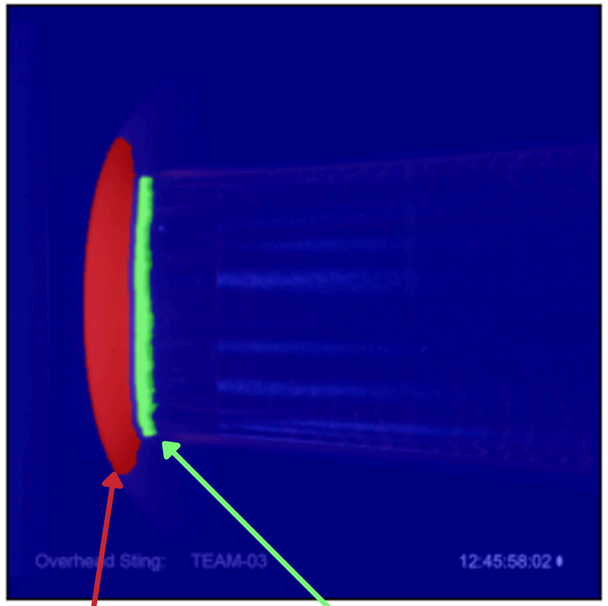
- **Reduced misclassification**:by distinguishing edge from non-edge pixels more effectively, minimizing errors in recession measurement.”

arcjetCV: Edge class

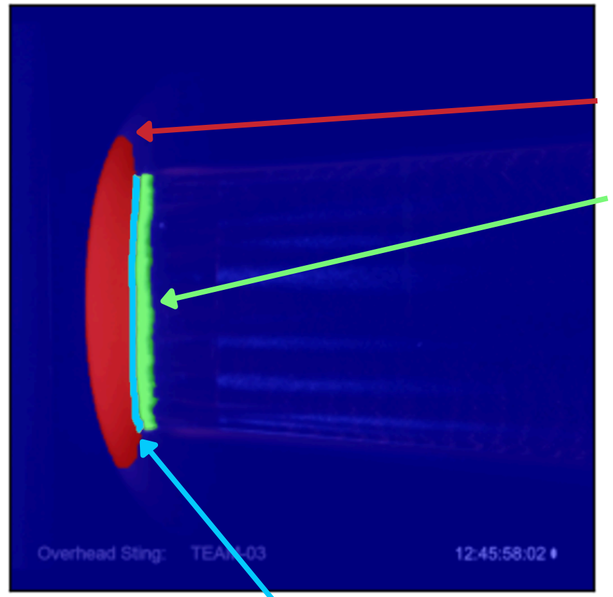
Original frame



Old training mask



New training mask



arcjetCV: New architecture

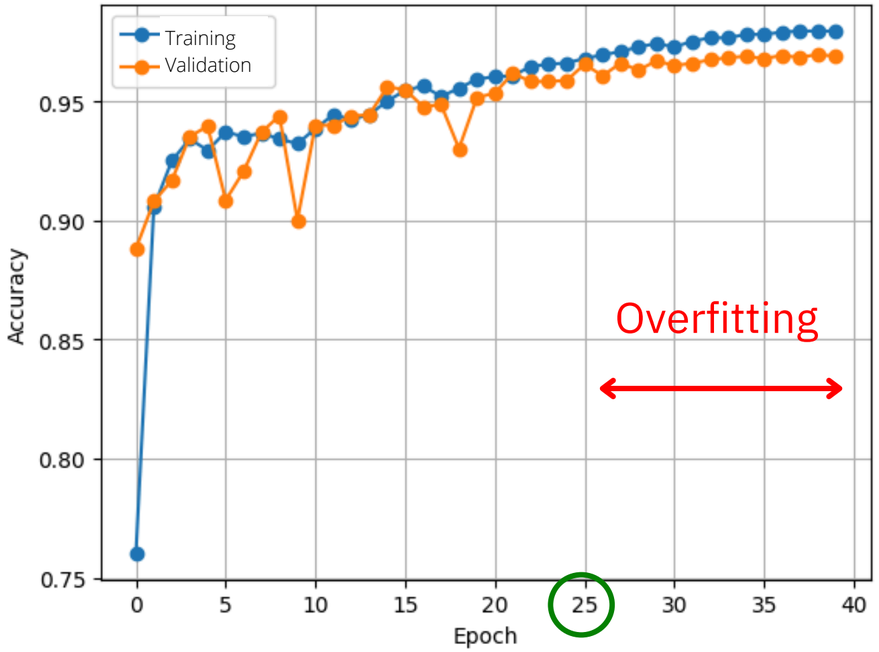
Xception

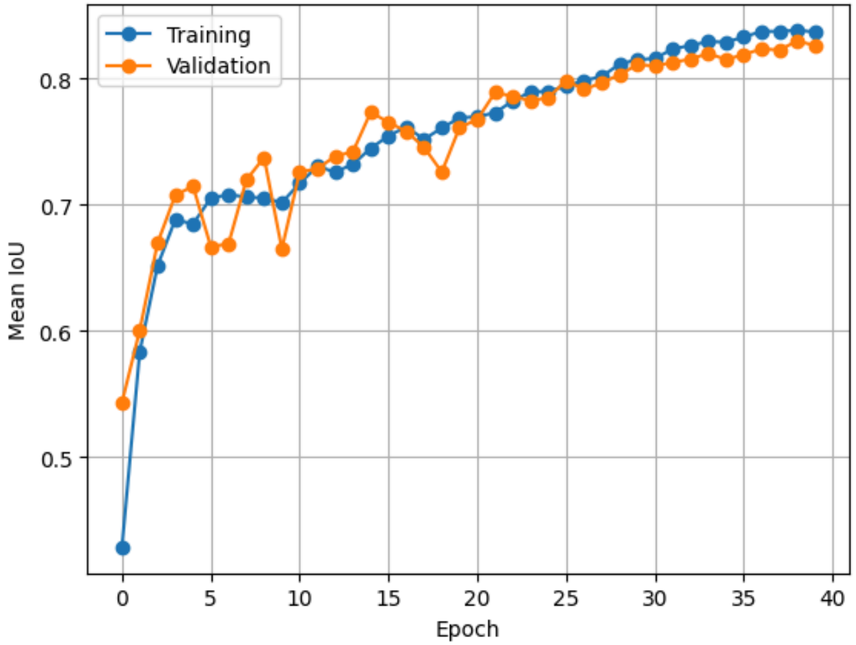
Key Advantages:

- **Improved Edge Detection:** Xception captures fine details, making it ideal for tasks needing precise segmentation.
- **Superior Feature Extraction:** Residual connections in Xception improve feature learning over VGG16’s standard convolutions.
- **Higher Accuracy & Lower Cost:** Xception delivers better accuracy and is computationally more efficient, despite its deeper architecture.

Conclusion: Xception is a more effective model for high-precision tasks like edge segmentation.

ML architecture performance





Model with 25 epochs	Pixel Accuracy	mIoU
VGG16 (Old model)	0.9411	0.7311
ResNet18	0.9413	0.7543
Xception	**0.9687**	**0.8005**

$$\text{Pixel Accuracy} = \frac{\text{Number of Correctly Classified Pixels}}{\text{Total Number of Pixels}}$$

Mean Intersection over Union (mIoU):

$$\text{mIoU} = \frac{1}{N}\sum_{i=1}^{N}\text{IoU}_i$$

$$\text{IoU}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i}$$

→Handles Class Imbalance

where N = Total number of classes.

where:

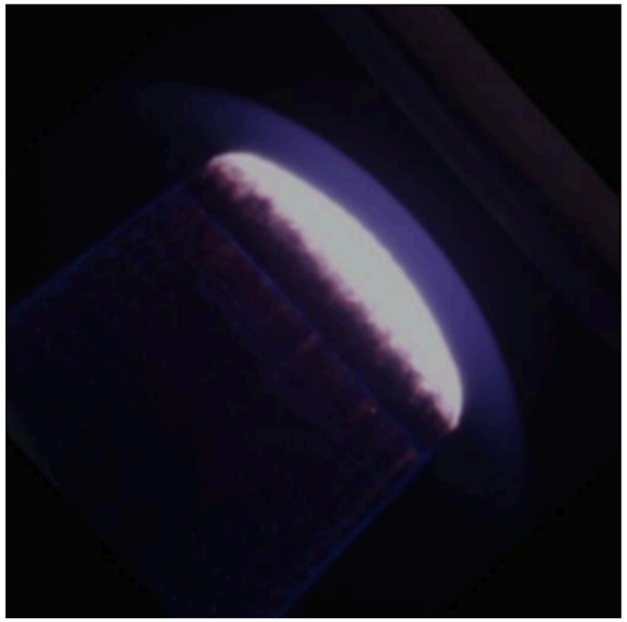
- True Positive (TP) = Pixels correctly predicted for class i,
- False Positive (FP) = Pixels incorrectly predicted as class i,
- False Negative (FN) = Pixels of class i that were missed.

arcjetCV: Augmented training dataset

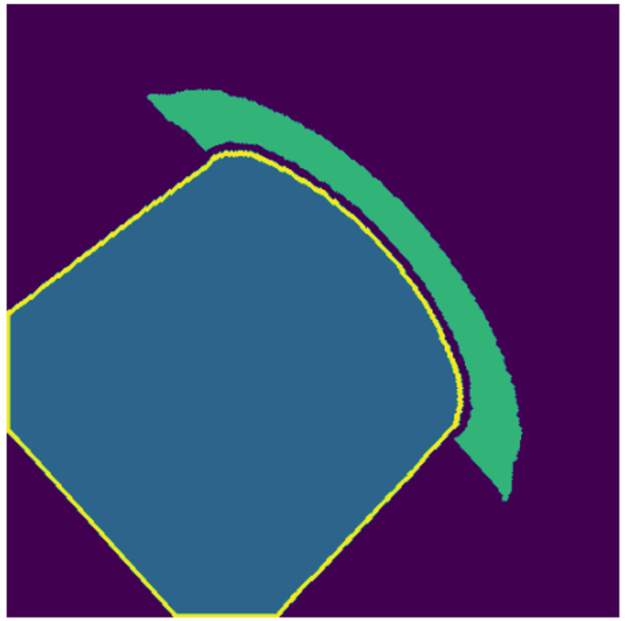
- New frames from different facilities.
    - VKI
    - UIUC
- New frames from new tests.

→ Size of the new dataset: **1778** training frames.

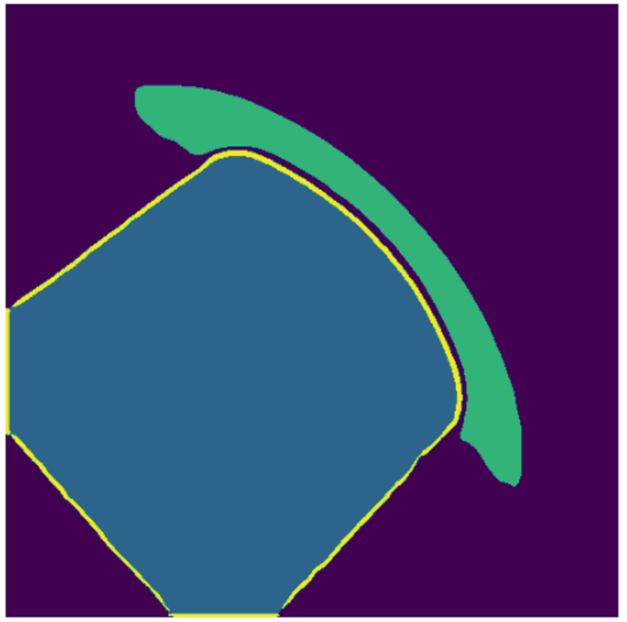
arcjetCV: New model result



Challenging video frame



Ground truth masks



Xception prediction

Recomended

arcjetCV: Pypi installation

1) Open your terminal

2) Install arcjetCV

```
pip install arcjetCV
```

3) Run arcjetCV:

```
arcjetCV
```

Easiest BUT Slower

arcjetCV: Executable installation

Download the excecutable .exe or .app here:

Flexible

arcjetCV: conda installation

1) Open your terminal

2) Install arcjetCV

Prerequisites:

- git-lfs
- Miniconda or Anaconda
- Xcode Command Line Tools (macOS only)

Windows Users and Developers

```
git clone https://github.com/magnus-haw/arcjetCV.git
cd arcjetCV
conda env create -f env/arcjetCV_env_[cpu/gpu].yml
conda activate arcjetCV
python -m pip install -e .
```

Unix Users (macOS and Linux)

```
conda create --name arcjetcv conda-forge::arcjetcv
```

3) Run arcjetCV:

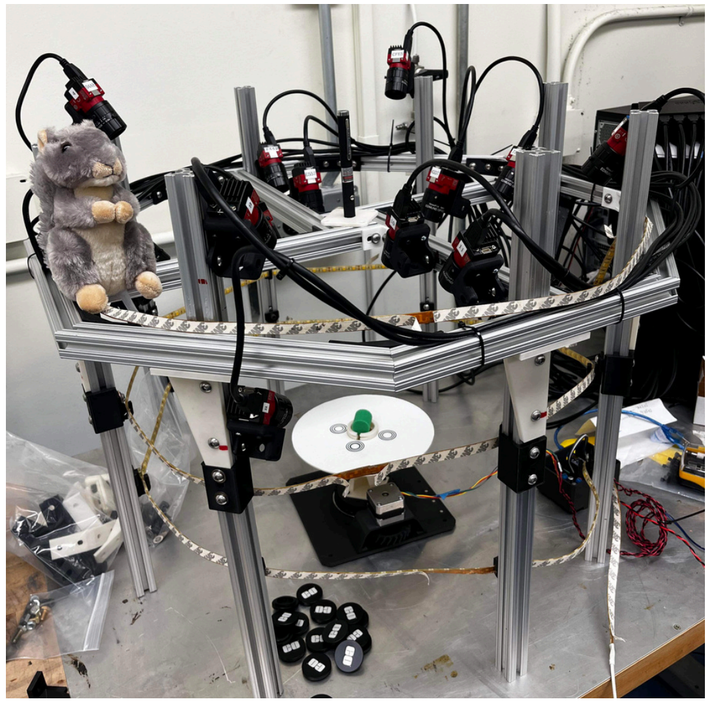
```
conda activate arcjetCV
arcjetCV
```

see the Git Repo

STARscan Scanners

Static Config (16+ cameras)

- Expensive (more cameras)
- Not portable
- Longer calibration
+ Faster scan



Turntable Config (6 cameras)

+ Cheaper (fewer cameras)
+ Portable
+ Fast to calibrate
- Longer to scan

STARscan Scanners Hardware

- **Cameras:**

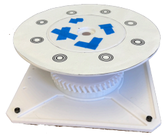
AlliedVision 12MP, VimbaX API, 16mm f/1.6 lens, Min iris: ~F13

- **Rig:**

8020 bars & 3D connectors, flexible T-nut mounts

- **CCTags:**

Circular patterns for scaling and alignment



- **Turntable:**

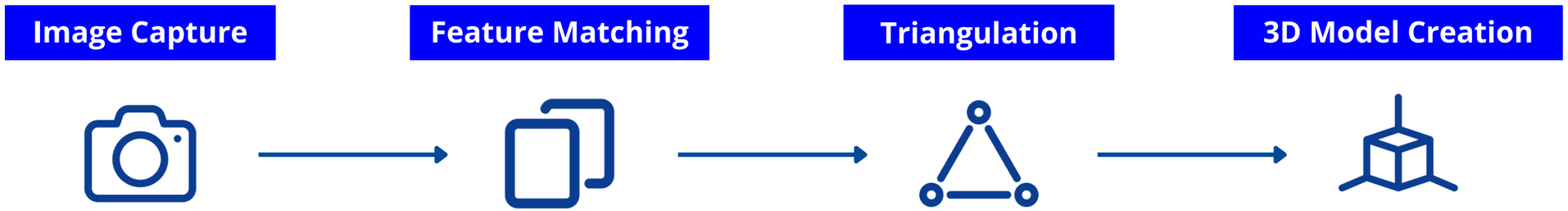
3D printed, stepper motor, Arduino-controlled

- **Lighting:**

LED panels/strips, white sheets for background masking

3D mesh from photos

1. Image Capture: Take multiple overlapping photos of the object or area from various angles.
2. Feature Matching: Software identifies common points or features across different images.
3. Triangulation: Uses geometry to calculate the position of points in 3D space based on camera positions.
4. 3D Model Creation: Combines all points to form a 3D model, adding texture and detail.



STARscan uses Meshroom from AliceVision for photogrammetry-based 3D reconstruction [1].

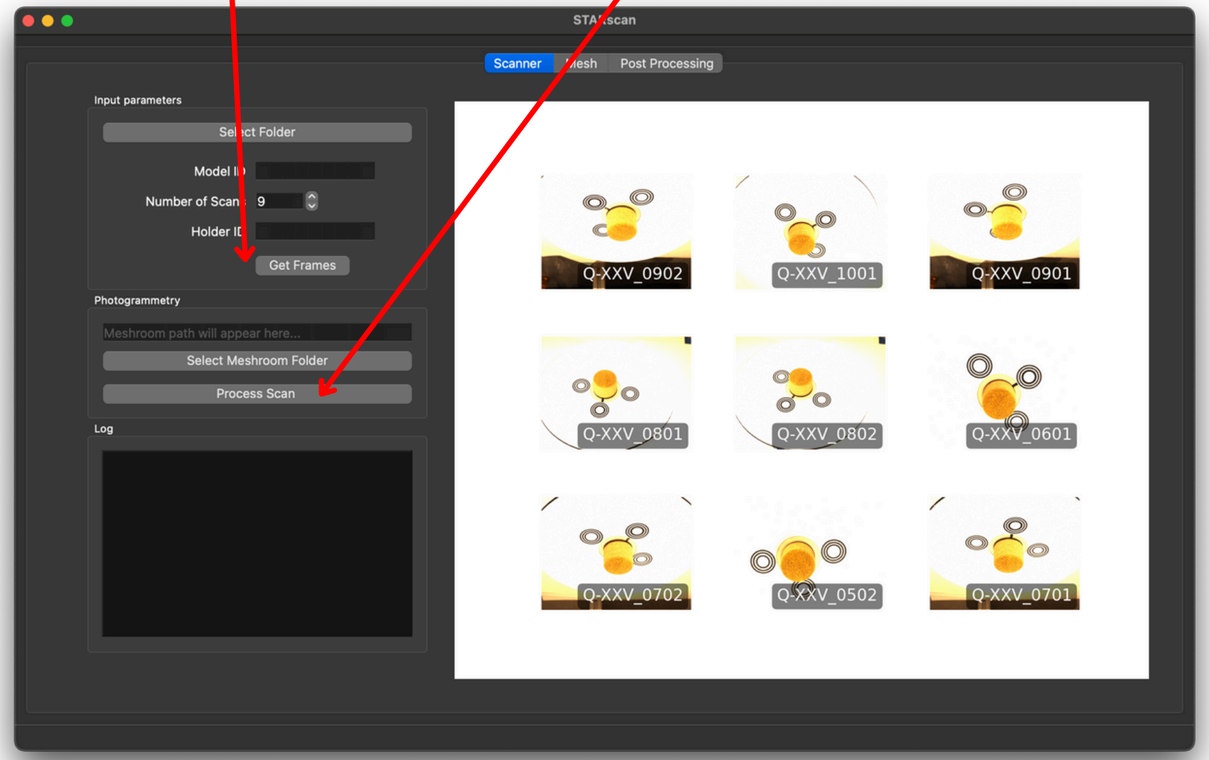
[1] AliceVision Meshroom. Available at: https://alicevision.org/#meshroom

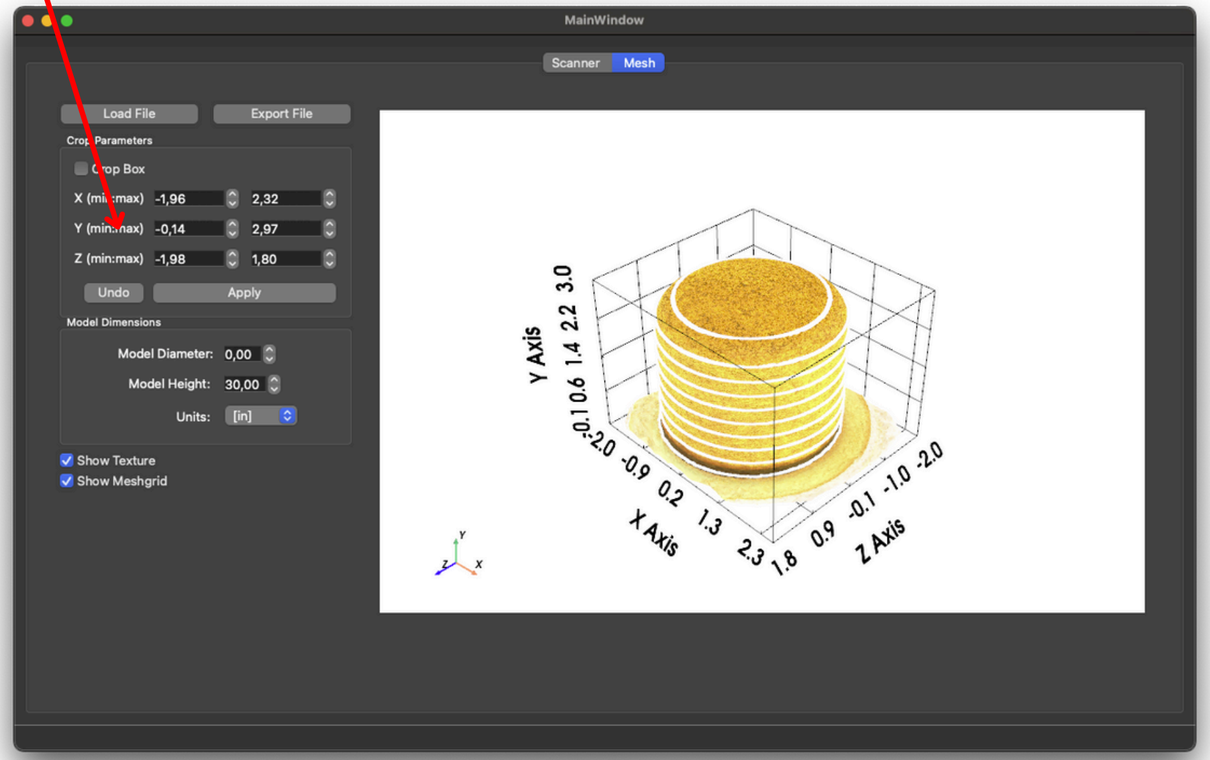
STARscan GUI

1) Get the frames

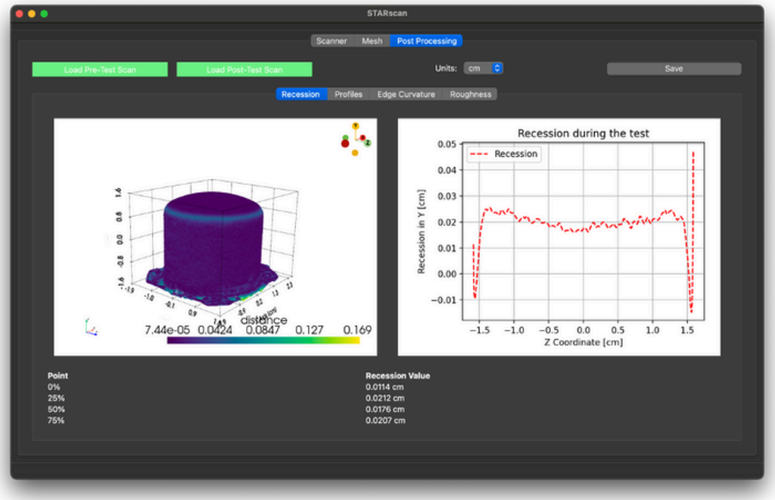
2) Generate Mesh

3) Crop

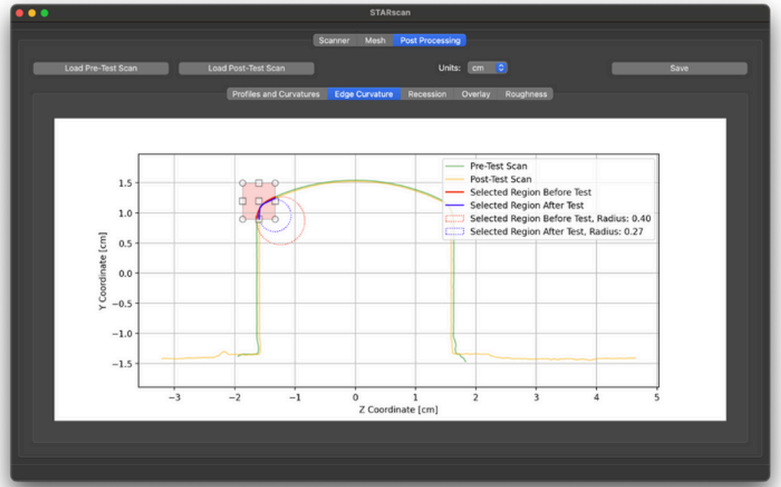




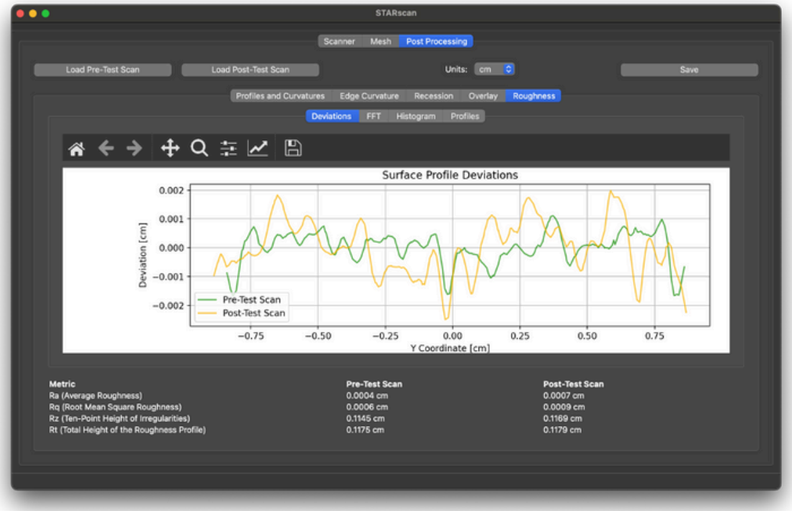
STARscan GUI: Post-processing



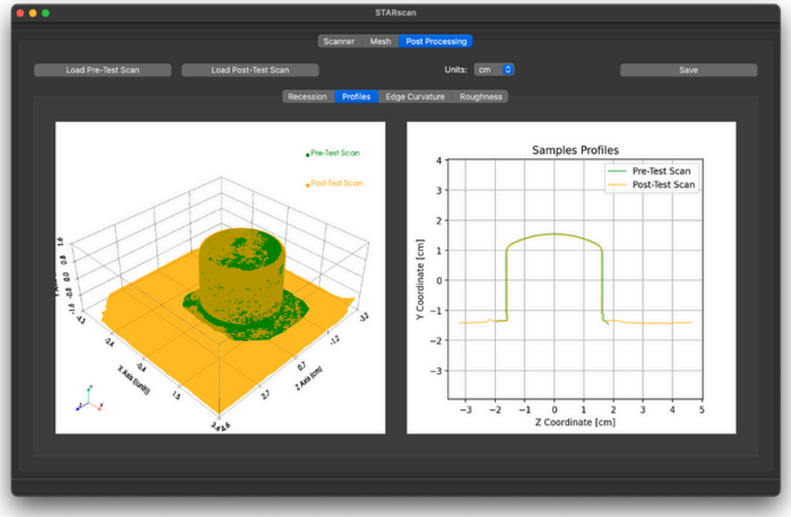
Recession



Shoulder Curvature



Roughness



Profiles

8. Conclusion

arcjetCV:

- New Class
- Expanded Dataset
- New Architecture
- 3 installation ways

ArcjetCV enables:

- Automated recession statistics
- 2D model validation
- Time resolved validation

STARscan:

- 2 scanners
- 1 User interface
- 4 Postprocessing features

STARscan enables:

- Automate 3D reconstruction, imaging, alignment and analysis.
- Eliminate pretest imaging and 3D scanning bottleneck.

→ Better mission planning and risk management

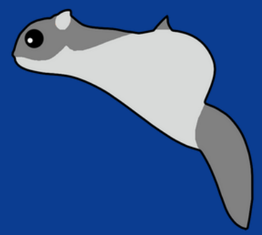
Acknowledgments

- Federico Semeraro – NASA Ames Research Center
- Jeremie Meurisse – NASA Ames Research Center
- Magnus Haw – NASA Ames Research Center
- Sebastian Colom – NASA Ames Research Center
- Margaret Stackpoole – NASA Ames Research Center

Fundings:

- MSR
- ESM
- TSM Branch IRAD

Thank you for your attention !



Get updates of
STARscan and **arcjetCV**