DXTER

Translating Pennington Biomedical Research Center's 3D Avatar Universal Software from MATLAB to Python

Berend Grandt, Kim Nguyen,
Shelby Primeaux, & Grishma Shrestha

December 2, 2025

LSU®

Pennington Biomedical Research Center

Outline

Introduction

Project Introduction: DEXA and PBRC

- The 3D Avatar Universal Software provided by Dr. Steven B. Heymsfield from Pennington Biomedical Research Center was created to replace DEXA scans. It acted as a pipeline that could derive biometrics from meshes using MATLAB.
- While MATLAB is powerful, it requires high-cost licenses that are expensive to maintain and are subject to frequent changes and version updates.

Project Introduction: DEXA and PBRC

- The 3D Avatar Universal Software provided by Dr. Steven B. Heymsfield from Pennington Biomedical Research Center was created to replace DEXA scans. It acted as a pipeline that could derive biometrics from meshes using MATLAB.
- While MATLAB is powerful, it requires high-cost licenses that are expensive to maintain and are subject to frequent changes and version updates.

Our Goal

- Our role is to translate the MATLAB code into Python.
- Python is a free, open source alternative that includes high-quality libraries (**NumPy**, **Trimesh**, etc.) which contain many functionalities useful for this research.
- This replicates PBRC's cost effective alternative to expensive DEXA scans.

3D Universal Software (MATLAB)

- Pennington's MATLAB code is able to calculate the volume and surface area of body parts such as the arms, thighs, and neck.
- To do this, the body is sectioned by triangularization, giving (x, y, z) points. Then this plane is intersected and the intersection points create the convex hull that is used to measure the circumference.

What's been done

What's been done

Previous Work

- Pennington's MATLAB code was analyzed and functions were organized into an Excel file that documents what they did and how to calculate values.
- This information was given to us by Clint and has helped us build a new approach.

What's been done

Previous Work

- Pennington's MATLAB code was analyzed and functions were organized into an Excel file that documents what they did and how to calculate values.
- This information was given to us by Clint and has helped us build a new approach.

Spring 2025

- The students added documentation and research papers that would help with calculations and approximations.
- They focused on removing redundant code and adding Python logic to fix the body orientation, visualize the mesh object, and locate various regions on the mesh file.

Approaches

Problems

MATLAB Documentation

- Initially, we received a lot of documentation with little context for how the code is intended to run.
- We were unable to get the MATLAB code to run on our machines, so we have focused on breaking it down into sections and getting smaller portions to work in Python.

Problems

MATLAB Documentation

- Initially, we received a lot of documentation with little context for how the code is intended to run.
- We were unable to get the MATLAB code to run on our machines, so we have focused on breaking it down into sections and getting smaller portions to work in Python.

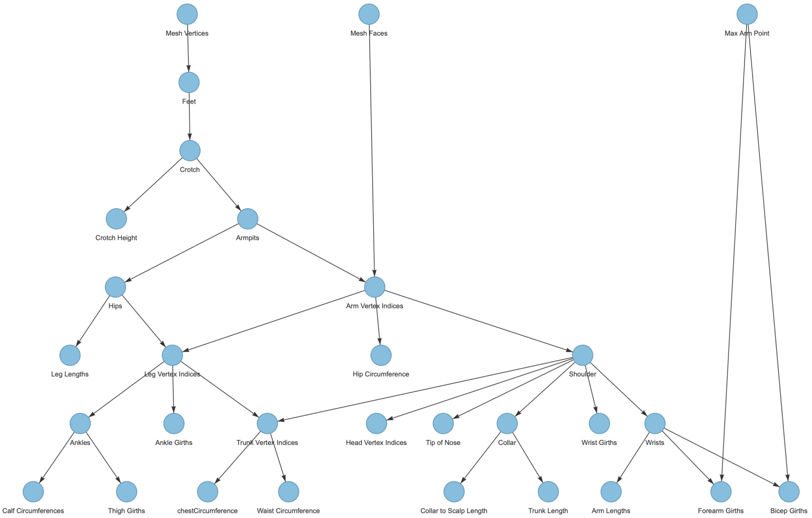
Understanding the System

- We started by focusing on improving the orientation, cleaning the obj files, and translating the arm. After receiving more information from Clint and gaining access to code from previous semesters, we decided to pivot to orienting the crotch first, since everything else in the code derives from the crotch.
- The main challenge comes from the multitude of functions that are called throughout the Avatar classification to locate each body part.

Old Approach

- Our approach started off using a *bottom-up* method which involved "going into the weeds" by tracing a function backward to see all the previous functions it called.
- While thorough for understanding implementation details, this method was time-consuming and led us to shift to a more efficient *top-down* strategy.

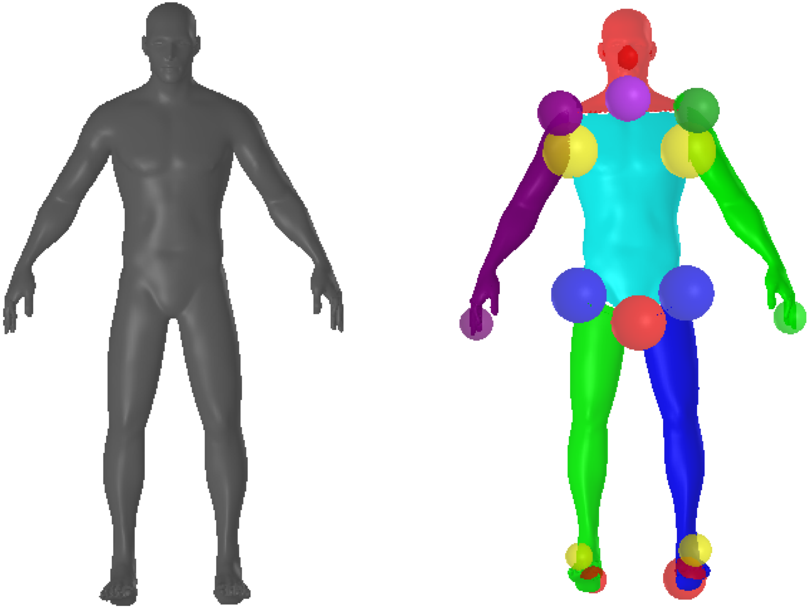
New Approach



,

Progress

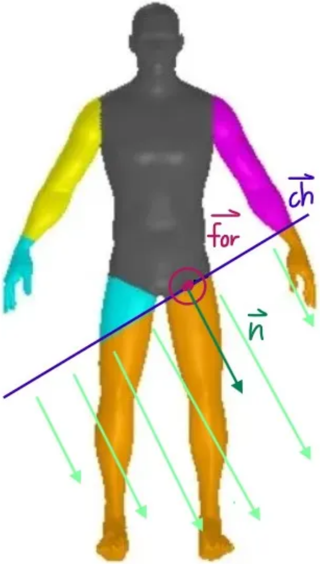
Before vs After



Understanding the Capabilities

- **Trimesh** performs basic repair operations and is useful in computing surface areas and volumes. **Trimesh** also has concavity and convexity applications which could be a future consideration to make the code more efficient.
- **Numpy** provides basic calculations such as Looping, Min, Max, and others.
- **SciPy** is an open source library built on **NumPy** which can perform high-level, efficient algorithms

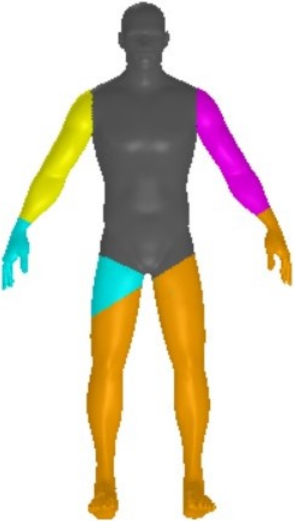
Leg Goals

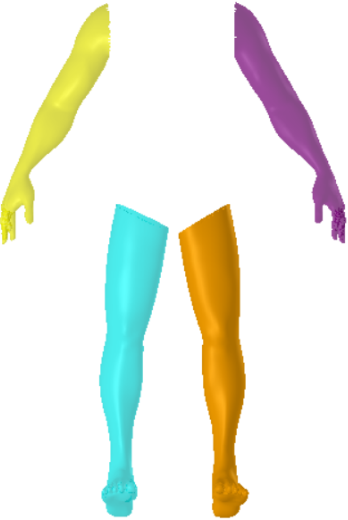


Slicing

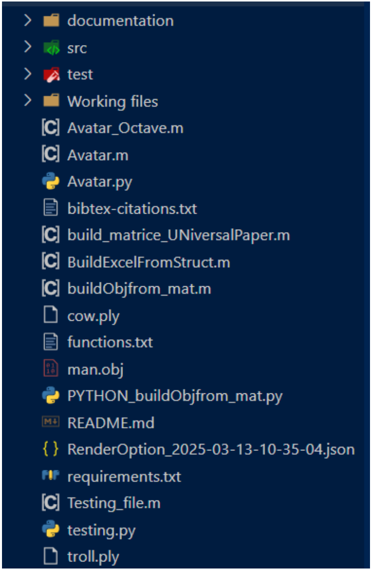
- trimesh.slice_plane(
    plane_origin=crotch,
    plane_normal=n
)
  Returns: The positive normal side of the plane used to slice the mesh. (Hip/Crotch plane)
- Finding the normal vector:
  n = np.cross(v, front)
- **Problem:** How do we isolate to just the left leg?

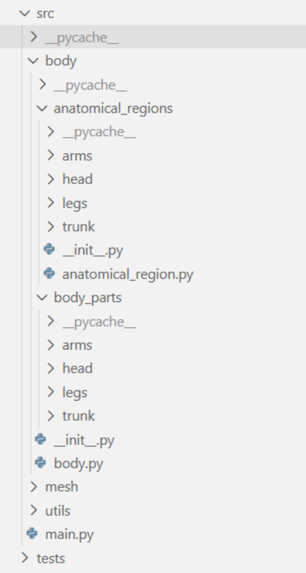
Legs After





Reorganization





Class Structure

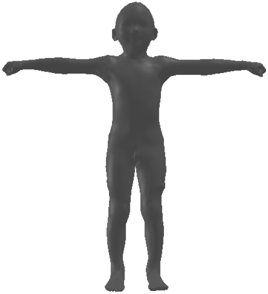
```python
@staticmethod
@cache
def _locate_armpits(mesh: trimesh.Trimesh) -> tuple:
    """Locate both armpits. Returns tuple of (left_armpit, right_armpit) as np.ndarray."""
    print("Called locate_armpits (Trunk)")

    new_mesh = mesh.copy()
    kdtree = cKDTree(new_mesh.vertices)

    # 1) Locate hips
    left_hip, right_hip = Trunk._locate_hips(mesh)

    # 2-4) Trace from each hip to armpit
    left_armpit_point = Trunk._trace_hip_to_armpit(new_mesh, kdtree, left_hip, side='left')
    right_armpit_point = Trunk._trace_hip_to_armpit(new_mesh, kdtree, right_hip, side='right')

    return (left_armpit_point, right_armpit_point)
```

All Together

```python
class Body(Mesh):

    def __init__(self, mesh_file):
        # Mesh cleaning, orientation

        mesh = trimesh.load_mesh(mesh_file)
        super().__init__(mesh)
        self.mesh = self.orient_mesh(mesh)

        # Body parts

        self.parts: dict[ANATOMICAL_REGION, Anatomical_Region] = {
            "head": Head(self.mesh),
            "trunk": Trunk(self.mesh),
            "left arm": Arm(self.mesh, 'left'),
            "right arm": Arm(self.mesh, 'right'),
            "left leg": Leg(self.mesh, "left"),
            "right leg": Leg(self.mesh, "right"),
        }

        # Landmarks & measurements

        self.subregion_meshes = { key: bp.mesh for key, bp in self.parts.items() }
        self.landmarks = { key: bp.landmarks for key, bp in self.parts.items() }
        self.measurements = { key: bp.measurements for key, bp in self.parts.items() }
```

Conclusion

Future Semesters



Acknowledgements

Thank You!

- We would like to thank Prof. Peter Wolenski, Dr. Nadejda Drenska, and Matthew Lemoine for guiding and supporting us.
- We would like to thank Clinten Graham, Muhammed Habibovic, Steven Heymsfield, Sophia Ramirez, and Sima Sobhiyeh for their help with learning the MATLAB code.

References

S. Sobhiyeh, M. Dechenaud, A. Dunkel, M. LaBorde, S. Kennedy, J. Shepherd, S. Heymsfield, and P. Wolenski, "Hole filling in 3d scans for digital anthropometric applications.," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2019.

S. Sobhiyeh, A. Dunkel, S. Heymsfield, S. Kennedy, D. Marcelline, J. Weston, J. Sheperd, and P. Wolenski, "Digital anthropometry for body circumference measurements: Toward the development of universal three-dimensional optical system analysis software.," in *Obes Sci Pract*, 2021.

S. Sobhiyeh, N. Borel, M. Dechenaud, C. Graham, J. Sheperd, M. Wong, P. Wolenski, and S. Heymsfield, "Fully automated pipeline for body composition estimation from 3d optical scans using principal component analysis: A shape up study.," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*, 2020.

S. Sobhiyeh, A. Dunkel, M. Dechenaud, S. Kennedy, J. Shepherd, S. Heymsfield, and P. Wolenski, "Crotch detection on 3d optical scans of human subjects.," 2019.