Using 3D Alignment to Describe Wear Between Two Shoe Scans

Eryn Blagg

ISU CSAFE

csafe

Center for Statistics and Applications in Forensic Evidence

Outline

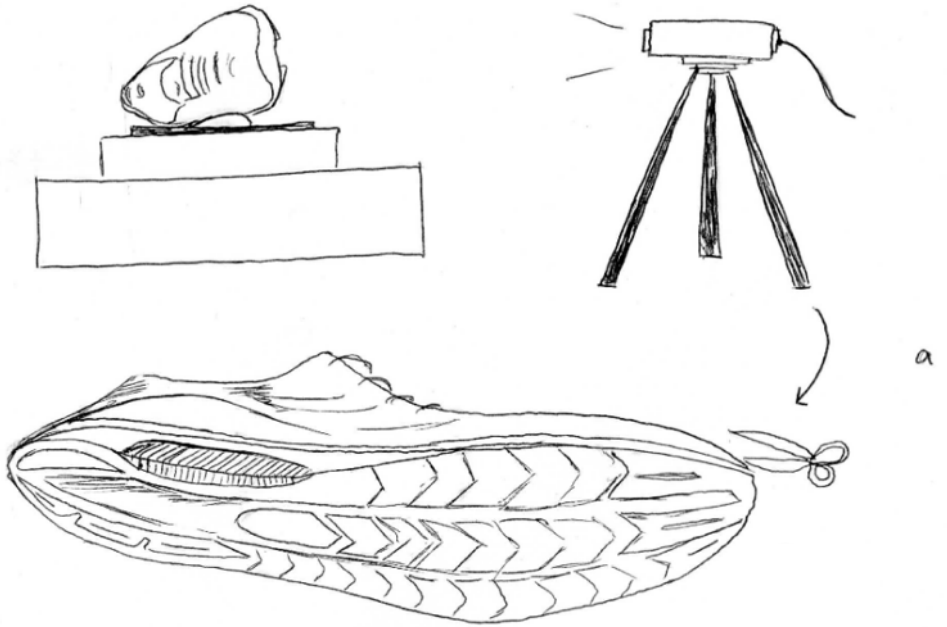
- Background
- Data Collection
- Preprocessing
- Automated Alignment Process
  - Barycenter alignment
  - Point cloud alignment
  - Iterative closest point alignment
- Distance
- Alignment Overtime
- Discussion

Background

- Forensics
  - Describing wear :When time elapses between the crime and recovery of the shoe, the alignment process may be more difficult: identifying marks may have worn away, or new marks may have been acquired due to additional wear
  - First need to Align the shoes
- 3D and 2D Alignment
  - 2D
    - Edge Detection
    - Landmark Alignment
  - 3D
    - Similar areas of research- paleontology, anthropology and medicine

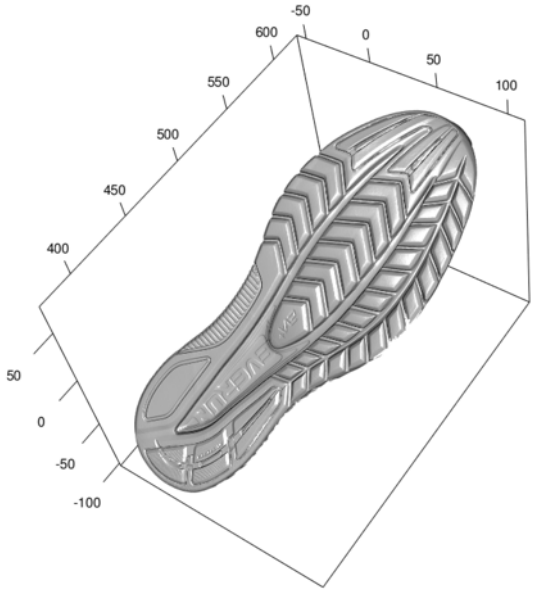
Data Collection

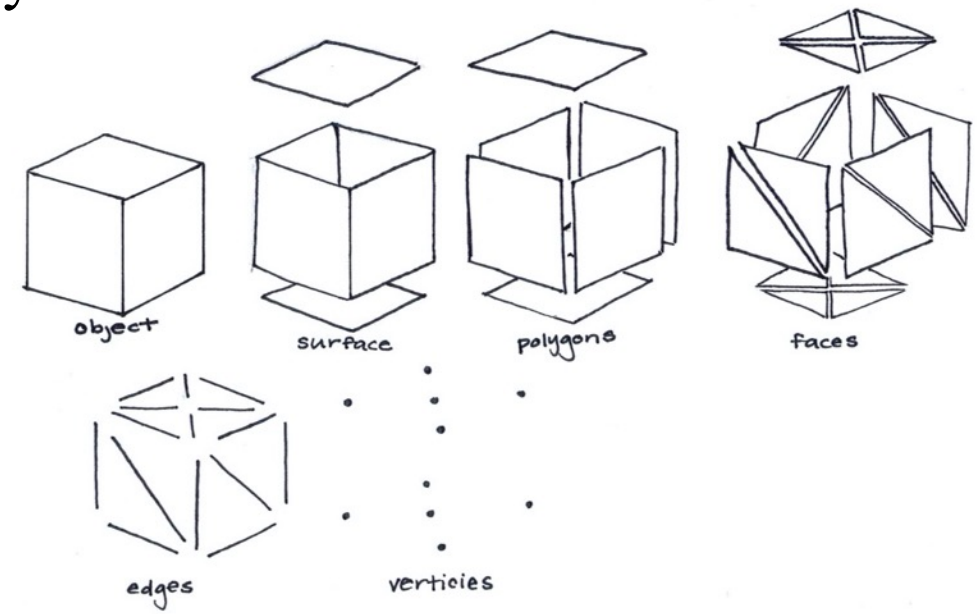
- EinScan Pro Plus scanner with Turntable
- size 10 men's Saucony Kinvara athletic shoe was
- scanned 6 times over the course of a 9-month period (June 2019 to March 2020)



Preprocessing

- Transform the STL file to a mesh object
- The surface of the object as a set of triangles connected by common edges and shared vertices



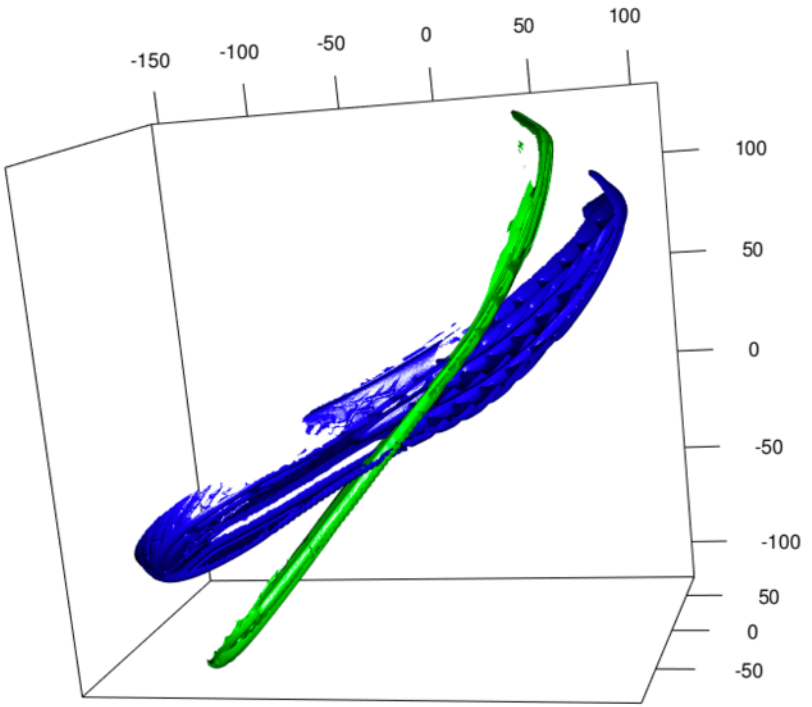


Automated Alignment

Barycenter alignment

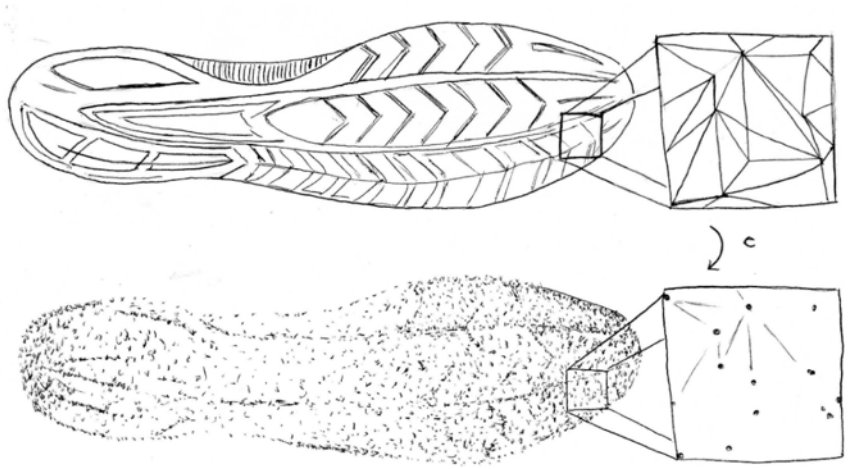
- The first step in our alignment process was to make the original shoe and worn shoes share a similar point in 3D space.

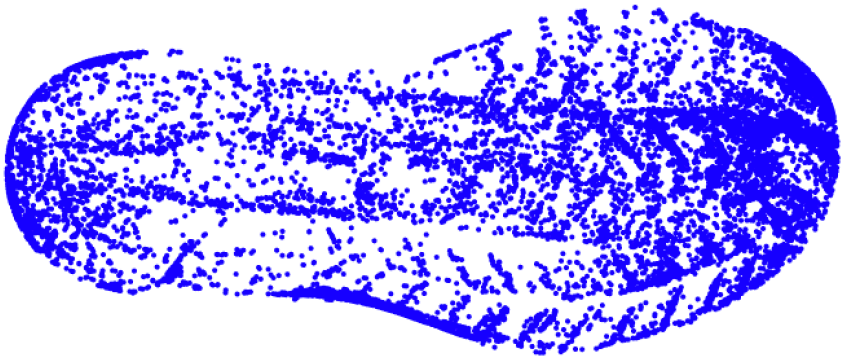
$$\text{Barycenter} := \frac{\sum_{i=1}^{n} x_i m_i}{\sum_{i=1}^{n} m_i},$$



Point Cloud Alignment

- Most statistical alignment tools are designed for a matrix of points
- Extract the vertices from the mesh object

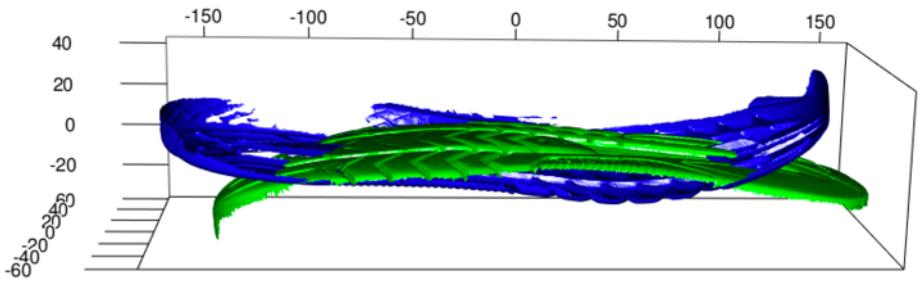




Principal Component Alignment

- General PCA
  - Step 1: Calculate the covariance matrix of data points.
  - Step 2: Calculate eigen vectors and corresponding eigen values.
  - Step 3: Sort the eigen vectors according to their eigen values in decreasing order
- This aligns the shoes by their eigen vectors up to a flip
- The alignment matrix produced by PCA is not unique, and may have a negative determinant (which "flips" the shoe scan inside-out). In this case, we need to multiply the pca-based alignment matrix by a vector $(\pm 1, \pm 1, \pm 1)$

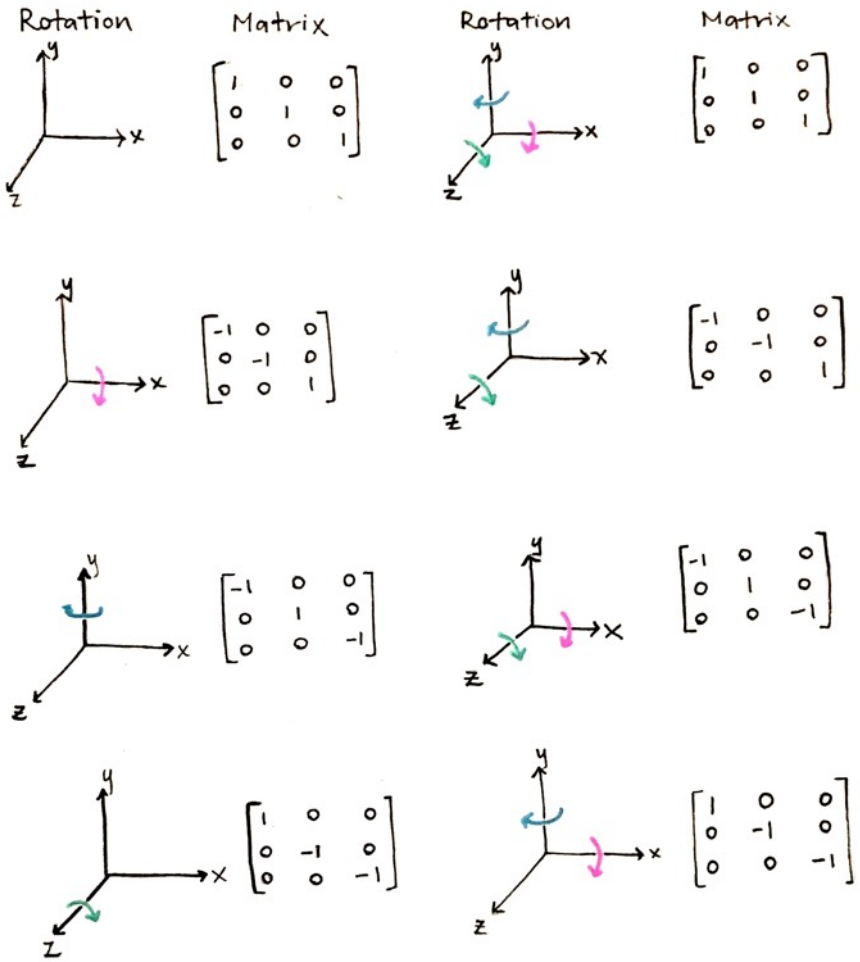
	PC1	PC2	PC3
x	0.560	-0.262	0.786
y	0.304	0.948	0.099
z	0.771	-0.183	-0.610



Rotation Matrix

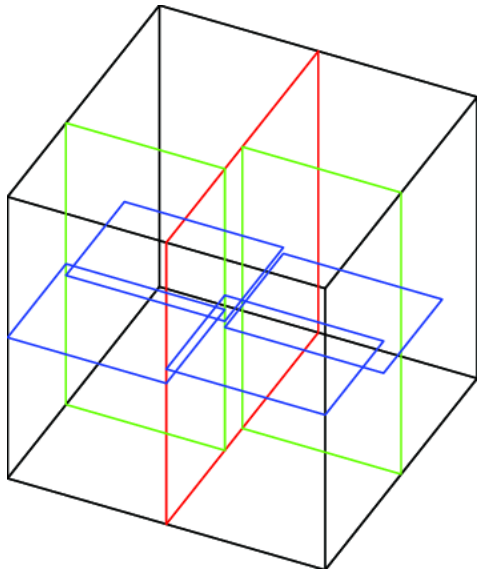
- Need to apply a rotation matrix
    1. rotation around the x-axis 180 degrees
    2. rotation around the y-axis 180 degrees
    3. rotation around the z-axis 180 degrees
    4. rotation around the x-axis 180 degrees and the y-axis 180 degrees
    5. rotation around the x-180 degrees and the z-axis 180 degrees
    6. rotation around the y-axis 180 degrees and the z-axis 180 degrees
    7. rotation around the x-axis 180 degrees and the y-axis 180 degrees and the z-axis 180 degrees
    8. No rotation is needed

In order to choose which one of these matrices produces images which are most closely aligned, the first shoe scan in time is set as the base scan. Then each of the four matrices above was applied to the second scan, and the difference is measured.



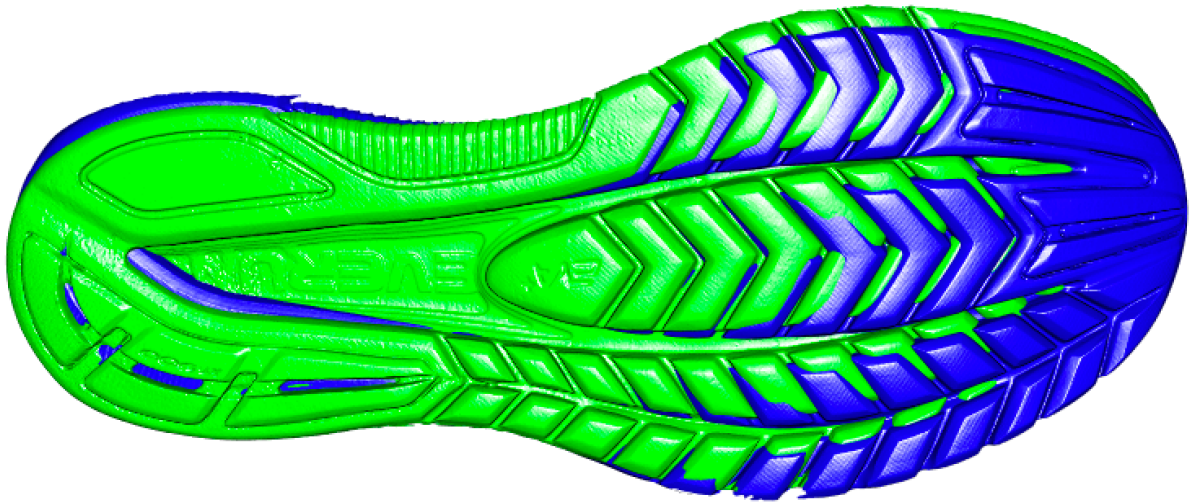
Defining the difference using K-D Trees

- branch-and-bound search is performed to find the nearest neighbor to a query point in a 3-dimensional point cloud.
- difference $(x1 - x2) + (y1 - y2) + (z1 - z2)$ is then reported for each of the points in the base scan.
- The number of differences is determined by the number of query points in the base scan: the number of vertices in the mesh object.
- With n differences, we summarize the total "distance" between the scans using the absolute average



Rotation matrix applied

- scan 1 and scan 3
- the four matrices lead to the following 4 differences: 6.457, 9.577, 9.215 and 2.394
- Thus scan 3 is rotated using diag(−1,−1, 1) .



Iterative Closest Point Alignment

1. For each point x_k in scan X_A find the corresponding point y_k in scan X_B such that the distance between x_k and all of the points in X_B is minimized, i.e.:

$$y_k(x_k) = \arg\min_{y \in X_B} d(x_k, y)$$

2. Assign each tuple (x_k, y_k) a weight w_k as the inverse distance between the points of the tuple.

$$w_k = d(x_k, y_k)^{-1}$$

3. Compute the transformation consisting of rotation $R \in R^{3x3}$ and translation $t \in R^3$ that minimizes the weighted least squares problem:

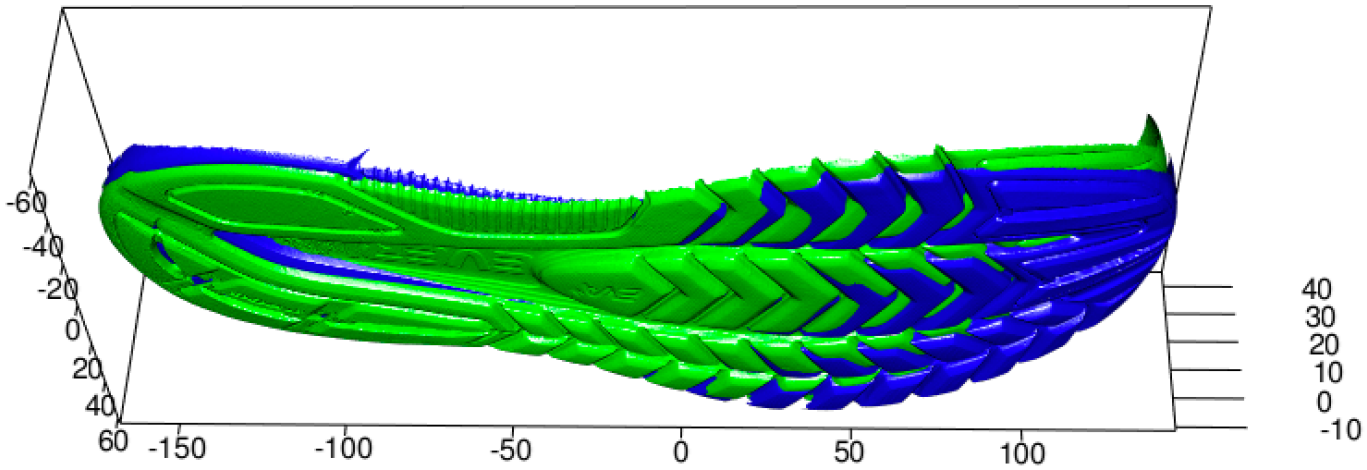
$$e(\mathbf{R}, \mathbf{t}) = \sum_{X_A} w_k ||\mathbf{R}\mathbf{x_k} + \mathbf{t} - \mathbf{y_k}||^2.$$

4. Apply the transformation found in step 3 to X_A:

$$X'_A = RX_A + t$$

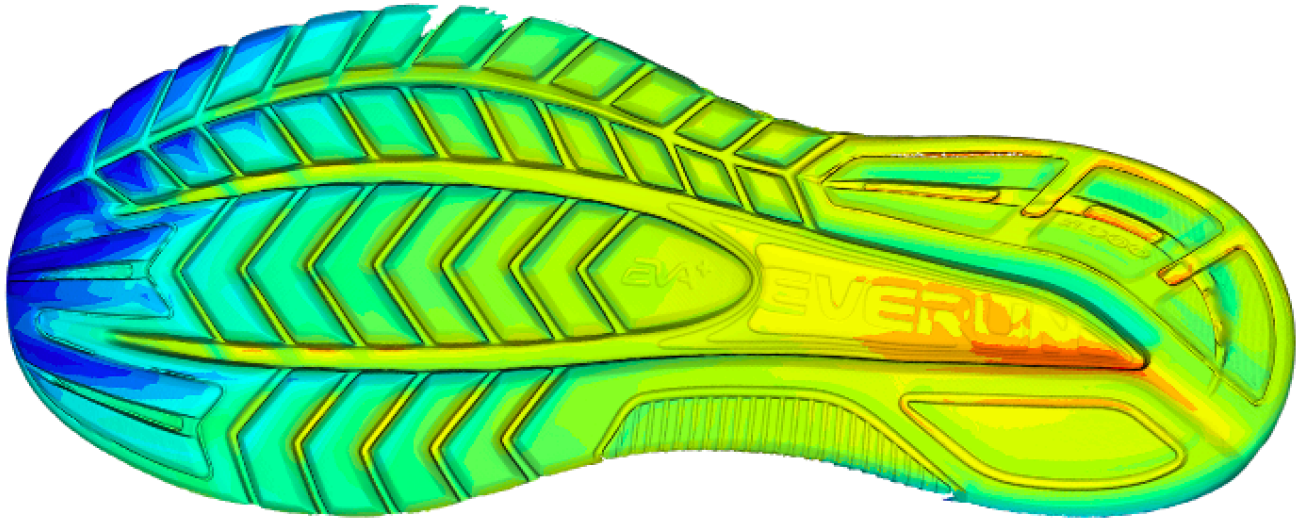
5. Repeat 1-5 until number of iterations is up, or until the error is below a set threshold, replacing X_A with X'_A in step 1.

ICP applied to Scan 1 and 3



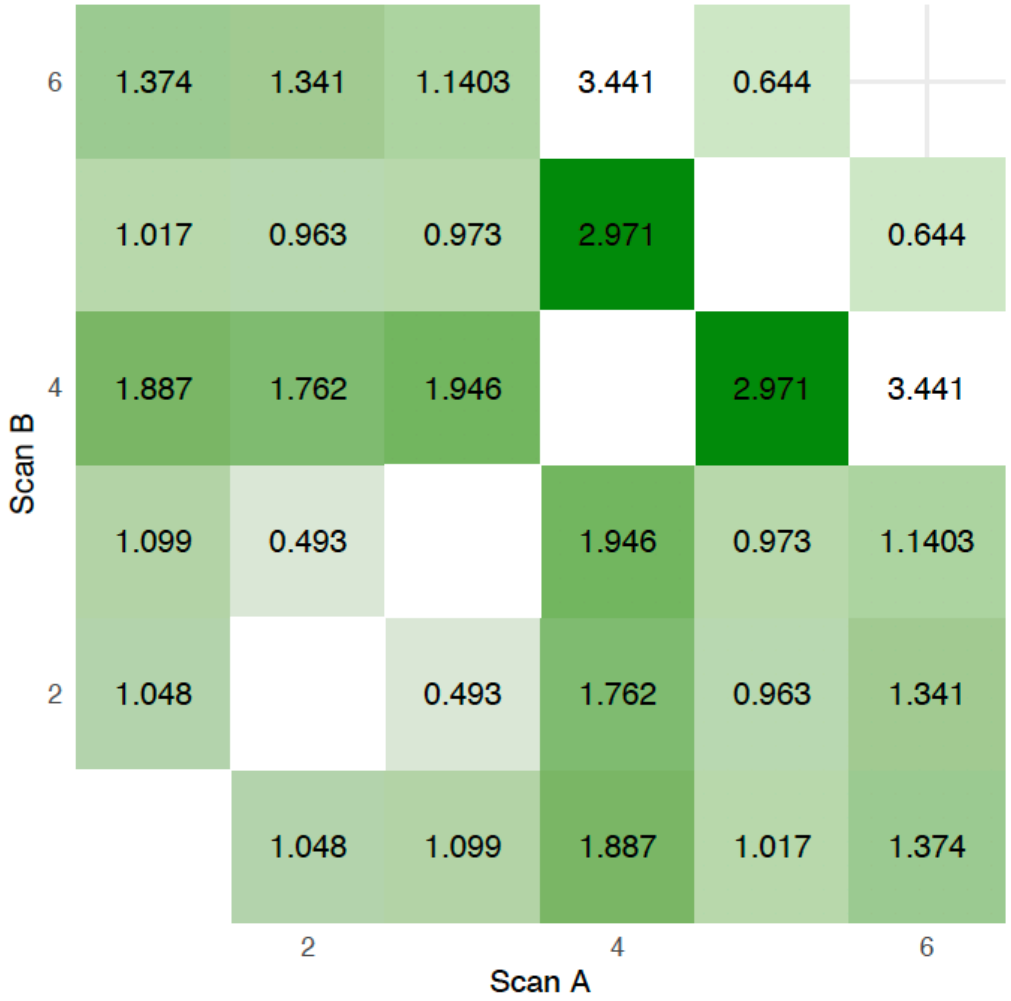
Distance

- In order to get the final distance, the difference vectors are computed from scan A to scan B, and then from scan B to scan A.
- Once the vectors of differences for both scan A and scan B are computed, the absolute values of the differences are taken and the mean of the absolute values is computed give a final distance metric between the two shoes.

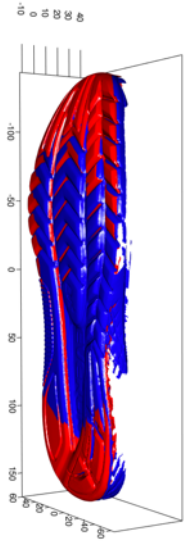


Alignment Overtime

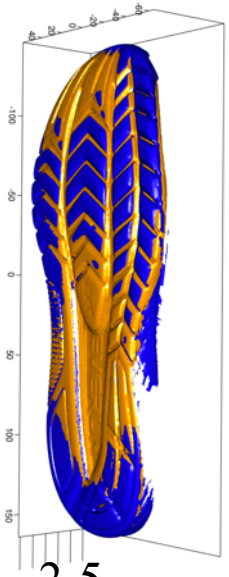
Average Surface Distance



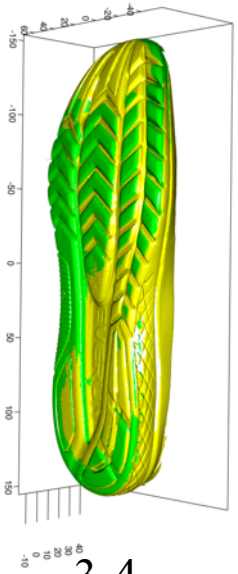
Examples



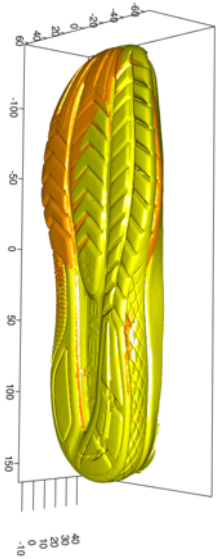
1-2



2-5

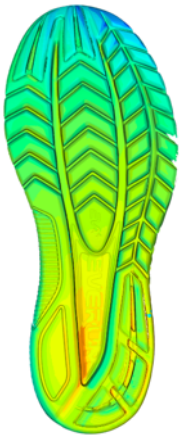


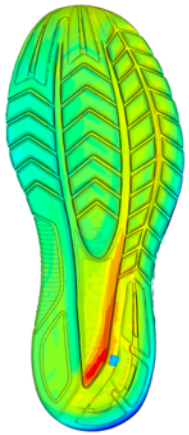
3-4



4-5

5-6





Discussion

- Not consistent cropping of the stl files
- No consistent points that are not changing
- Curvature of shoes