Computer Assisted Neural Probe Insertion

Shizra Tariq
04/10/2025

Introduction

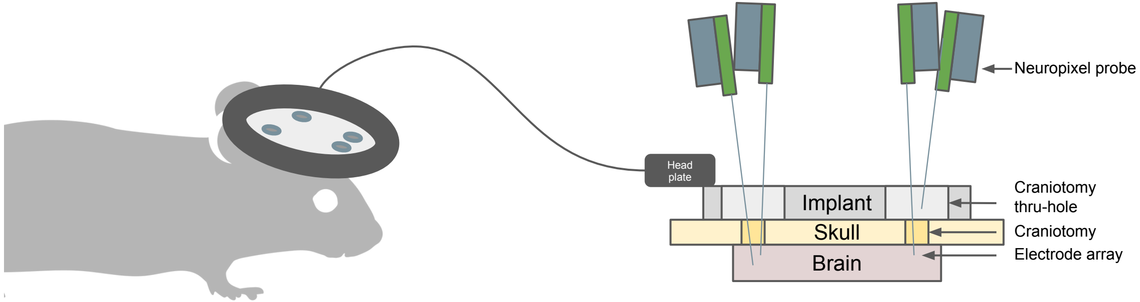
Goal

- Automate probe insertion into mouse brain
- Use computer vision for real-time guidance
- Increase accuracy and safety
- Reduce human error

Why we need to solve this problem???

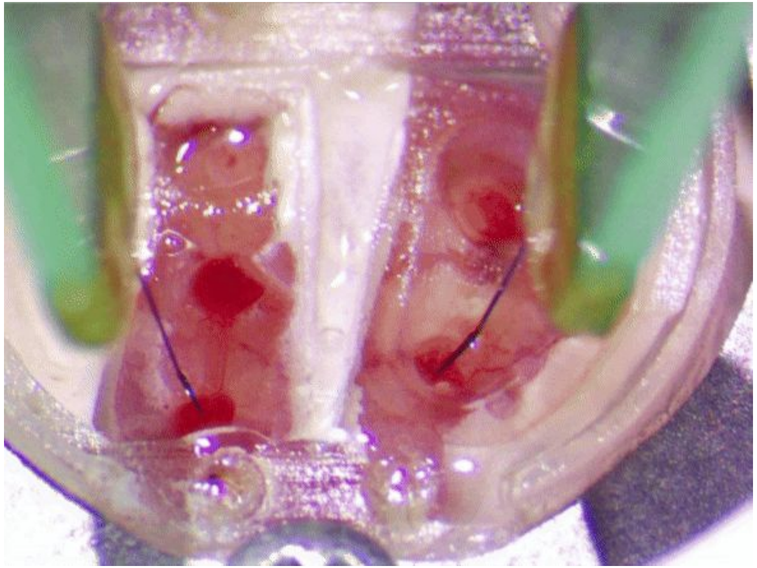
Challenges

- Manual insertion is slow and skill-dependent
- Small mistakes → probe bending/breaking or tissue damage
- Difficult to visualize and control probe position
- Need for consistent and precise insertions



Available Datasets

Two kinds of datasets that were available when I joined the project:



Dataset 1: zoomed-in visual setup

Probes are more visible but hard to detect the insertion point due to lack of surrounding reference

Dataset 2:Wider Camera View

probes are comparatively less visible but provides partial view of the headplate

Probe Detection

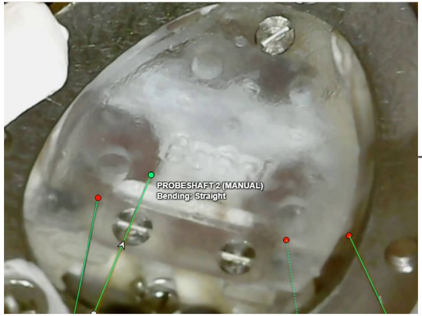
- Our first goal was to detect the probe during insertion.
- We focused on identifying both the probe shaft and probe tip using camera views.
- Once the probe was detected, the next step was to classify whether it was straight or bent.
- Detecting bending is important because it means the probe is not aligned correctly or hitting resistance.
- Early detection of bending helps prevent probe damage or breakage.

Proposed Solution

Designed a two-stage pipeline:

- Segmentation
- Classification

Methodology



Data labelling and preparation

YOLOv8m-seg

segmentation

ResNet18/ Measuring angle

Classification/ Bending Angle Measurement

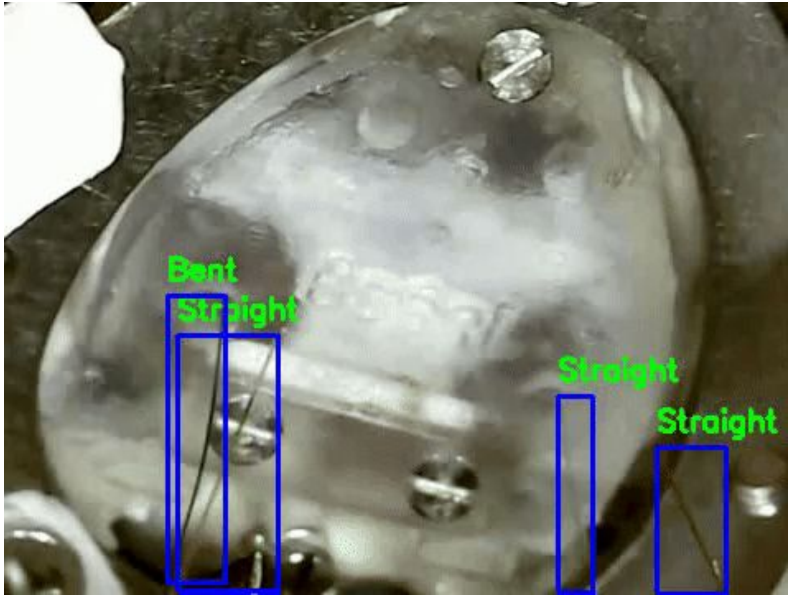
-Capture frame
-Run detection model
-Draw bounding boxes/ keypoints
-Display probe bent/straight status

Real-Time Inference

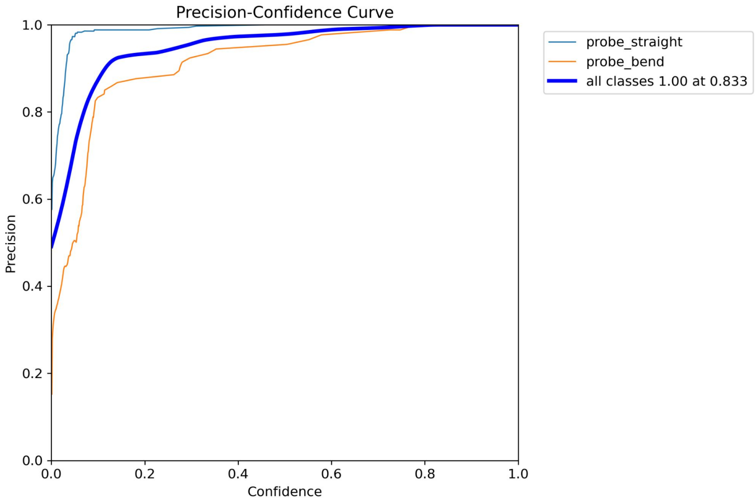
bending beyond threshold → flagging/trigger stop

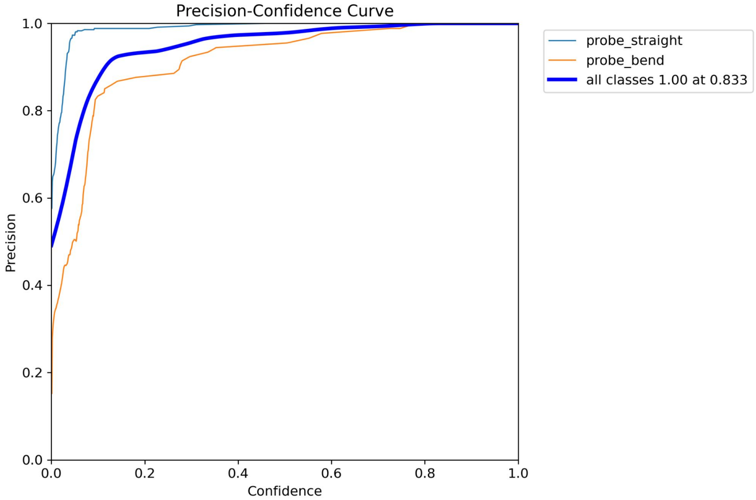
Decision and feedback

Results

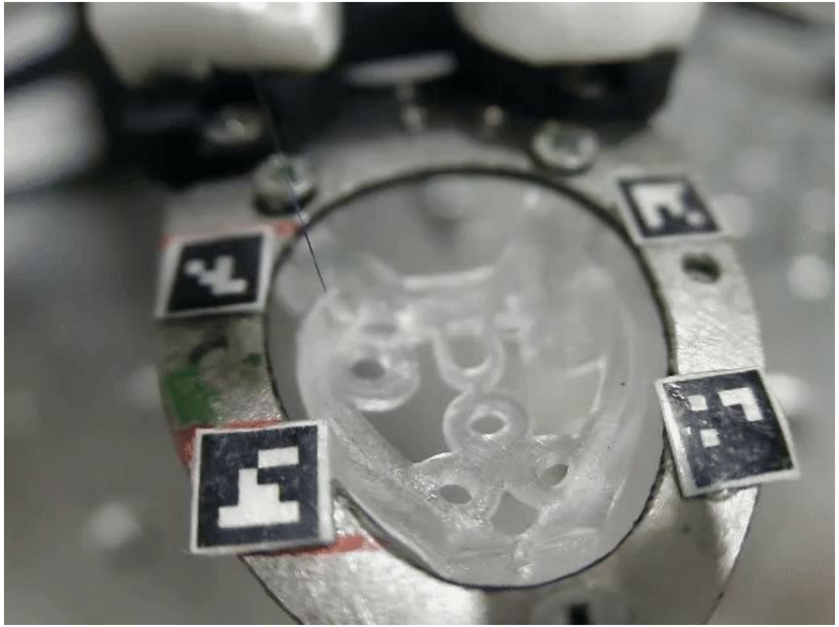


Detection accuracy for bent probes was comparatively lower due to dataset imbalance



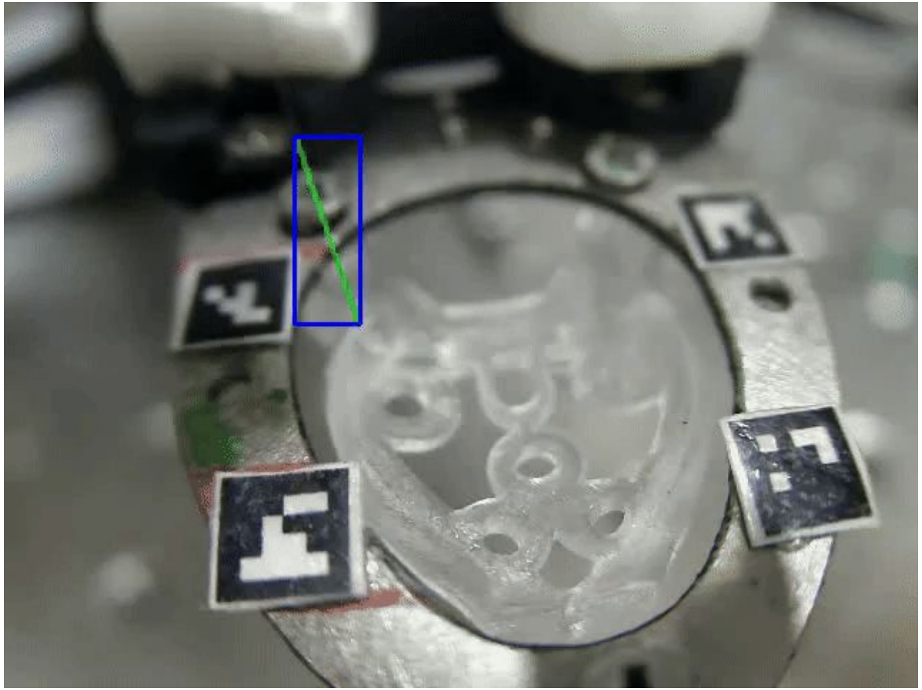


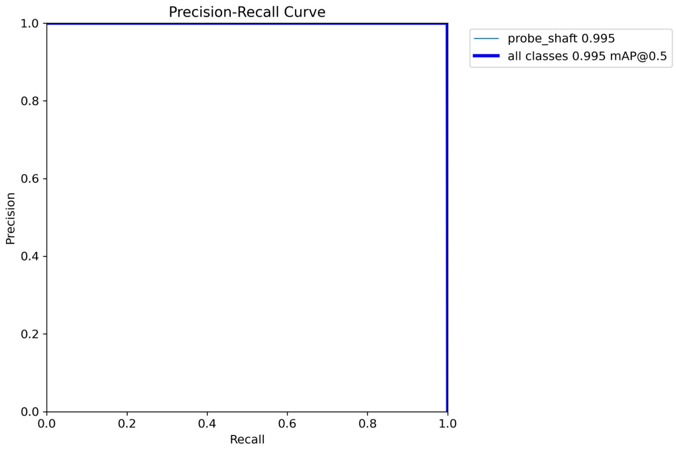
New Dataset

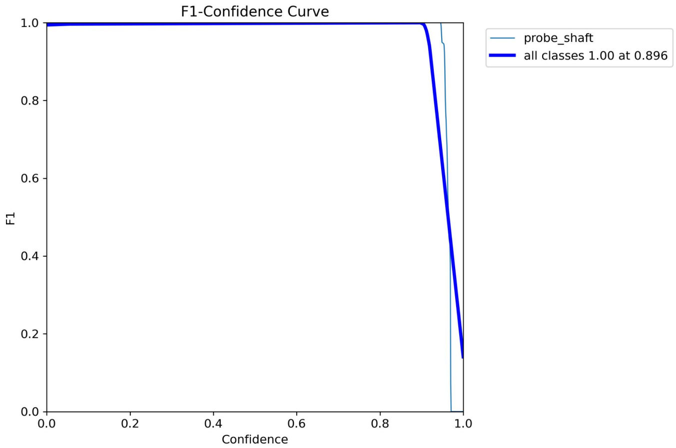


- Collected new dataset after camera change
- Increased noise and background complexity
- Change enabled clear visibility of headplate
- Headplate provides reference for insertion point estimation

Results

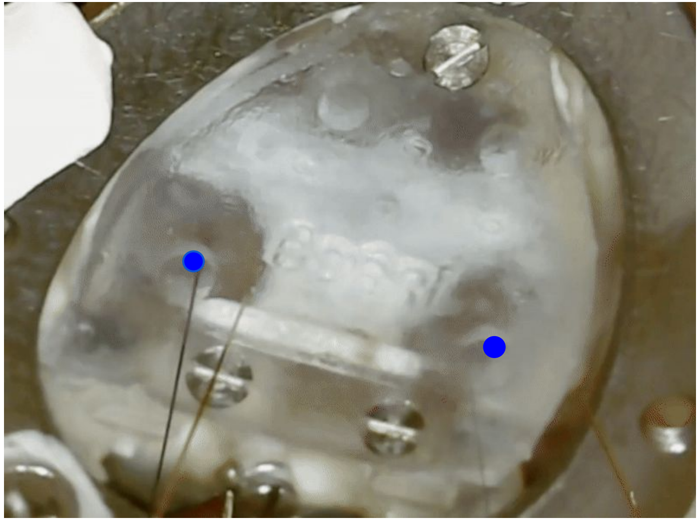



Precision-Recall Curve
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
probe_shaft 0.995
all classes 0.995 mAP@0.5


F1-Confidence Curve
F1
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
probe_shaft
all classes 1.00 at 0.896

Insertion Point Estimation

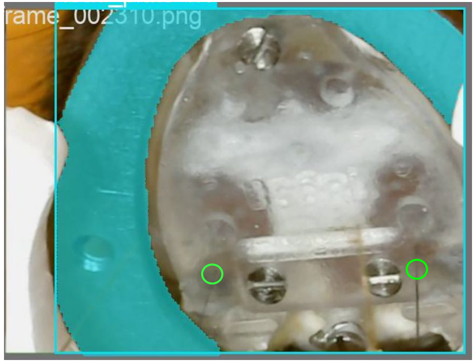
- Goal: locate insertion points for automated probe placement.
- Manual insertion requires precise identification of brain coordinates.
- Automation computes points relative to stable anatomical landmarks.
- Requires a stable reference for accurate calculation.
- Accurate detection is critical to avoid tissue damage.

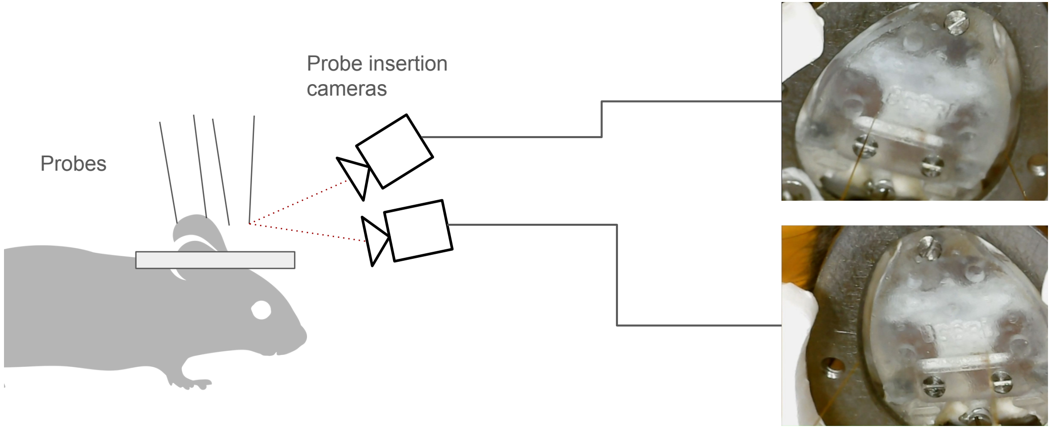


Initial Approach

Designed reference-based localization:

- **Headplate Segmentation Approach:**
  Segment headplate → use as fixed reference → calculate insertion coordinates





- **Alternative Approach:**
  Detect [anatomical features/markers] → establish reference frame → compute insertion points

Why it failed?

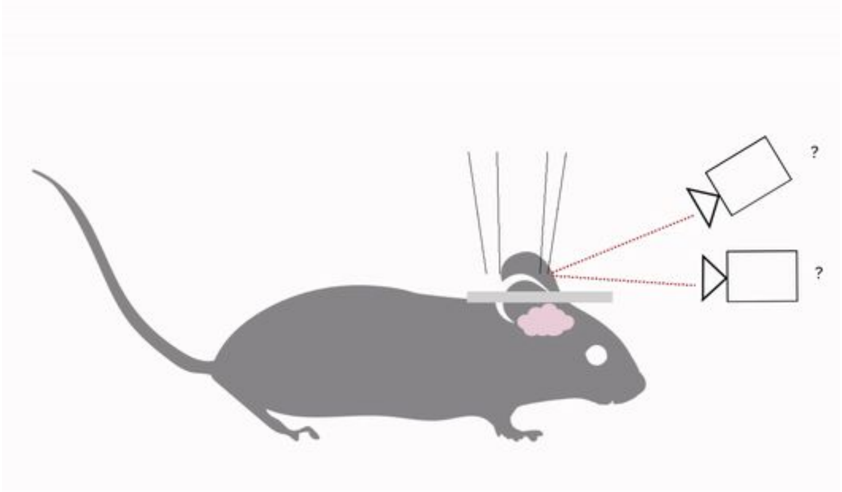
Initial Assumption: Camera would remain fixed throughout all trials

- This would allow a one-time calibration using the headplate reference

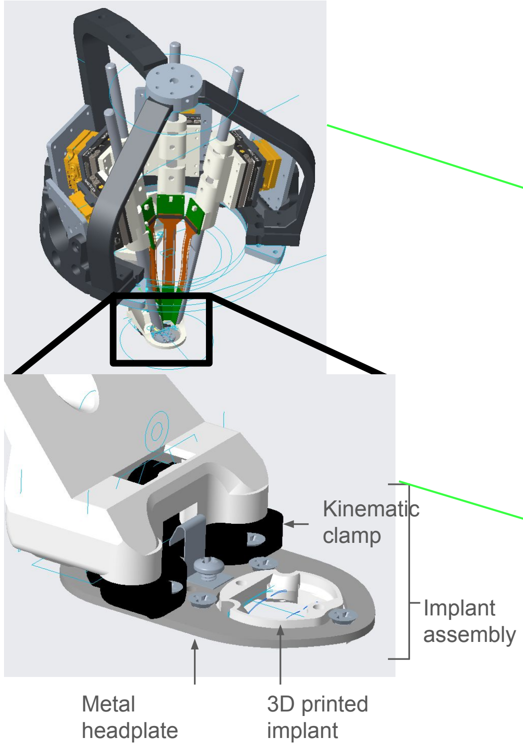
Reality: Camera position is only static **per trial**, not across trials

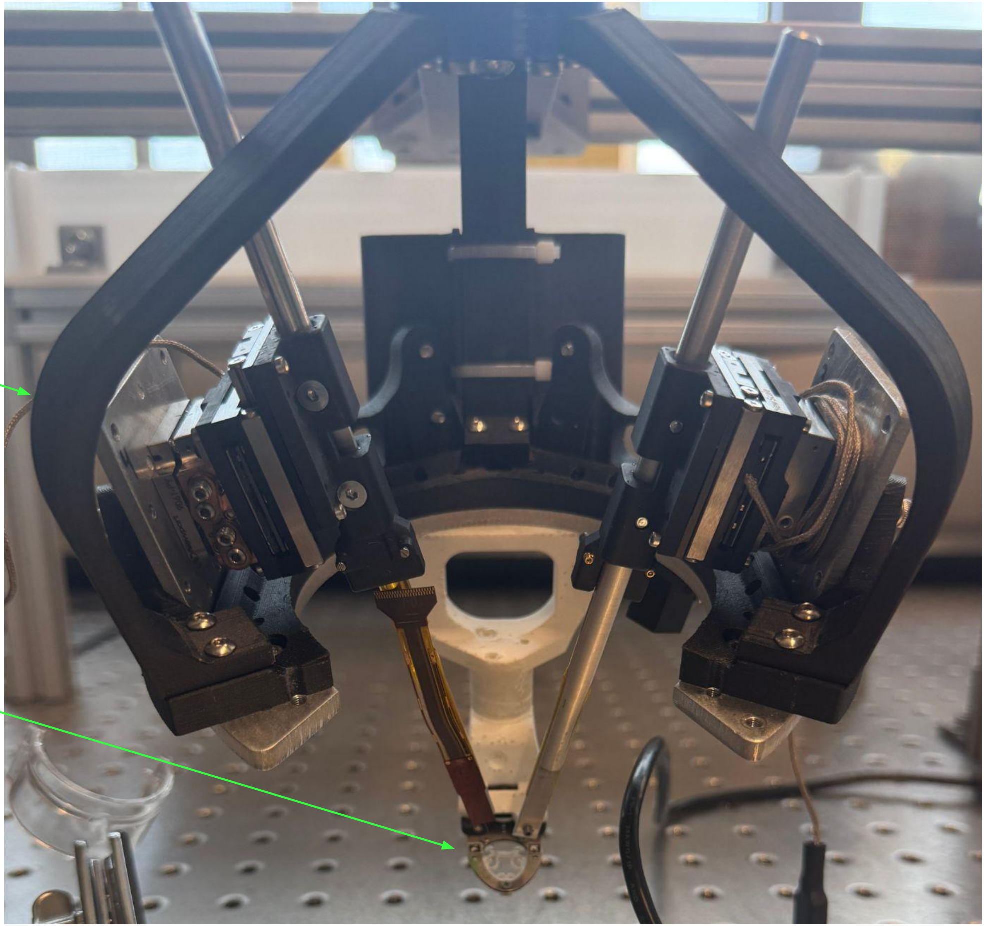
- Camera repositioning between trials changes the reference frame
- Each new trial requires recalibration relative to the headplate

- Camera repositioning alters headplate position, scale, and orientation, invalidating calibrations.
- Small segmentation errors (2–3 px) cause large deviations; sub-millimeter accuracy is required.
- Converting 2D to 3D coordinates on the curved headplate requires precise per-camera calibration.
- Lack of fiducial markers makes repeatable reference point identification challenging.

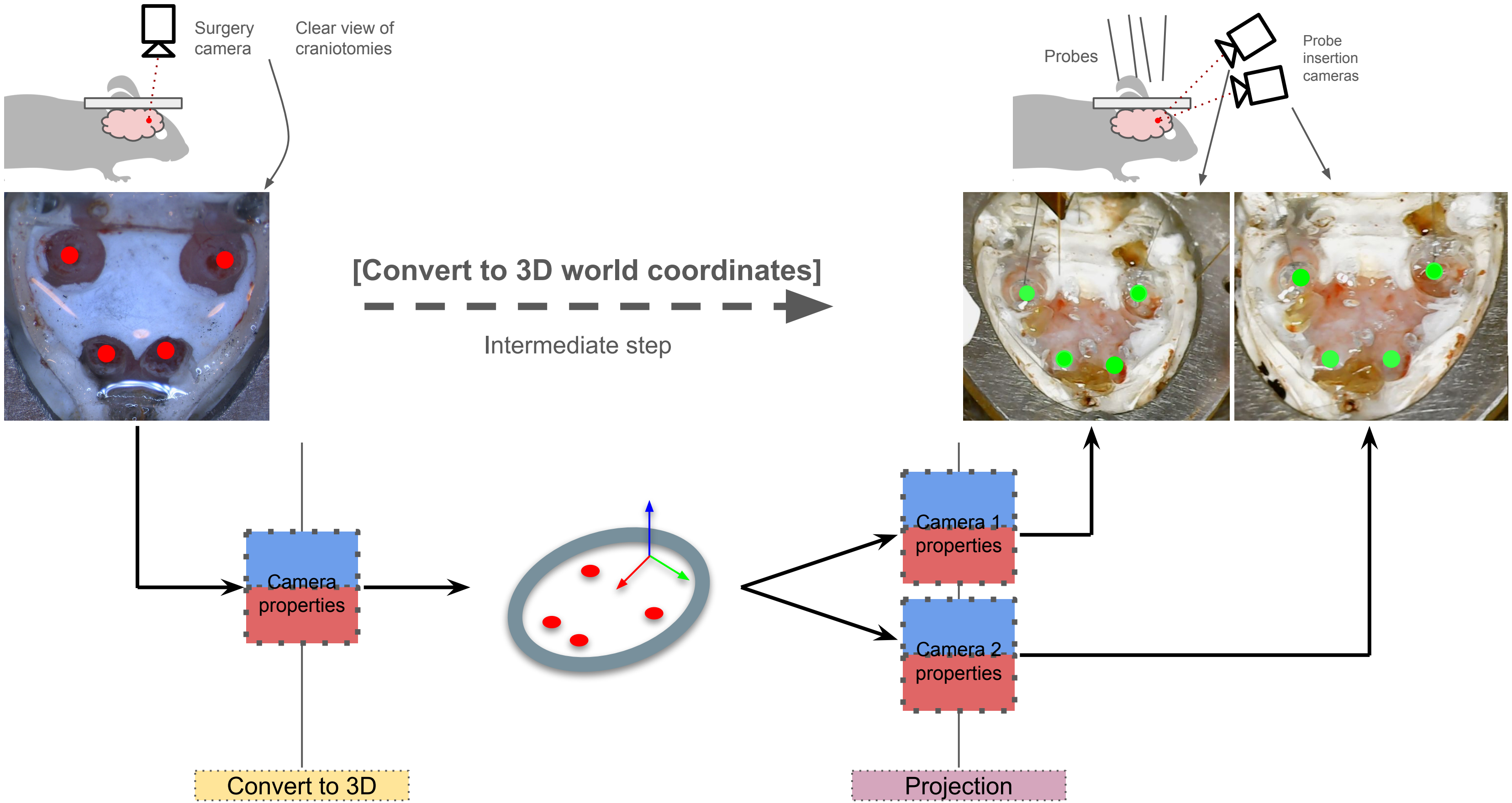


Exo Setup

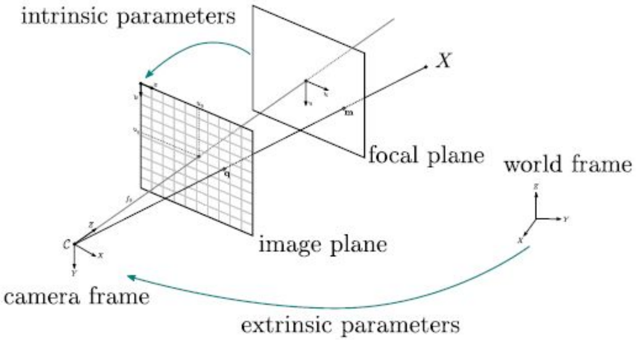

Kinematic clamp
Implant assembly
Metal headplate
3D printed implant



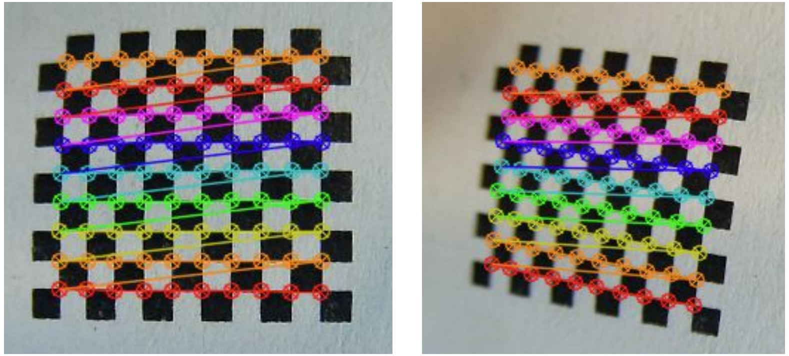
Why 3D conversion is necessary?

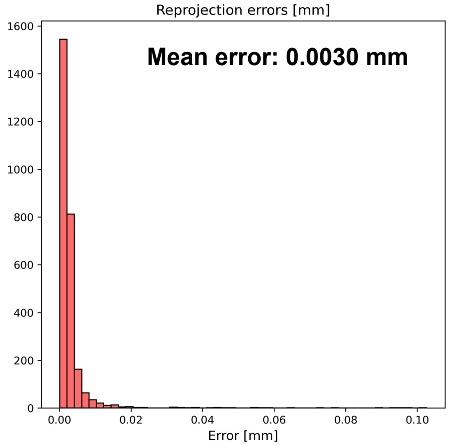


Camera Calibration



Zhang's calibration method





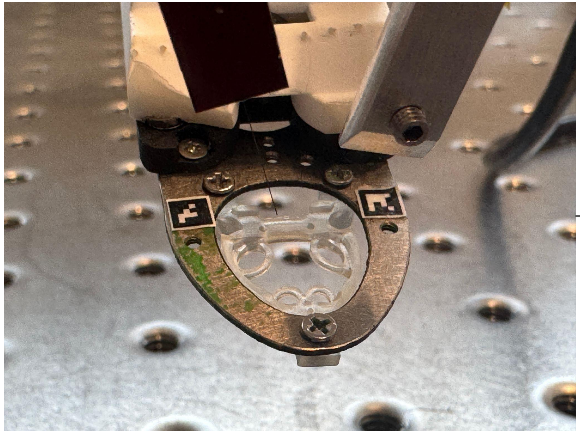
Chessboard Images → Camera Calibration → **Camera Parameters** → 3D coordinates

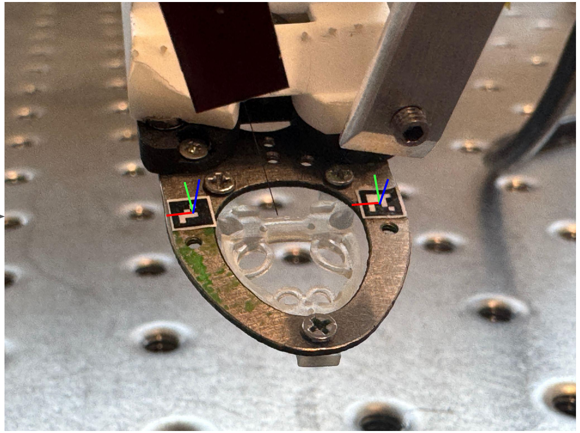
Camera Parameters

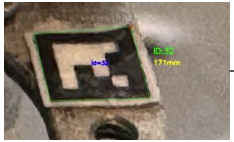
$$K = \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix}$$

D = [k1, k2, p1, p2, k3, k4, k5, k6]

Detection with ArUco tags







Find marker corners and tag ID

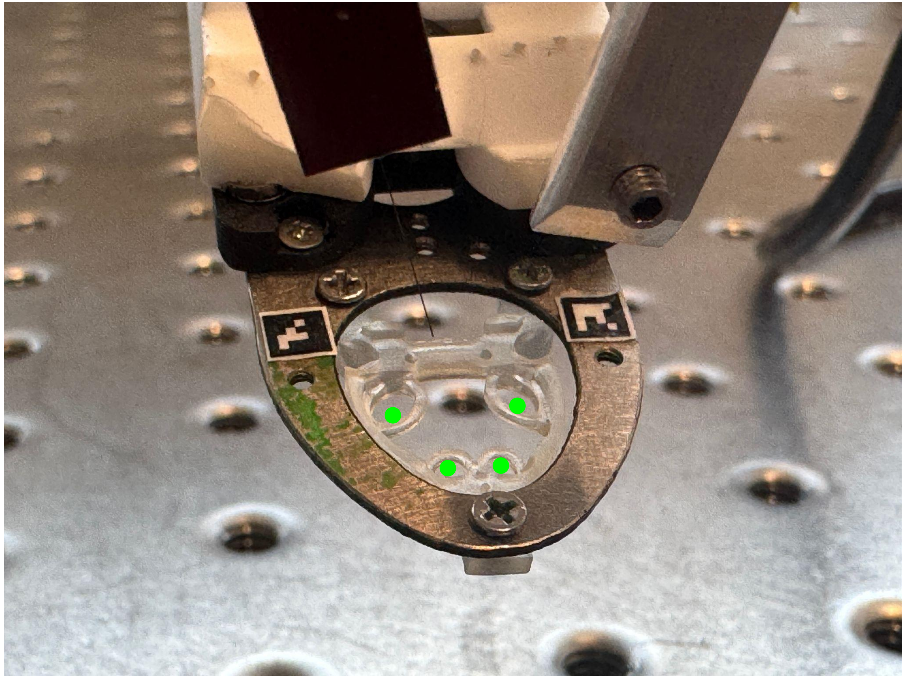
Convert to 3D camera coordinates

Estimate Marker Poses

t = [[tx],
[ty],
[tz]]

rvec = [[rx],
[ry],
[rz]]

Insertion Point Calibration



Process

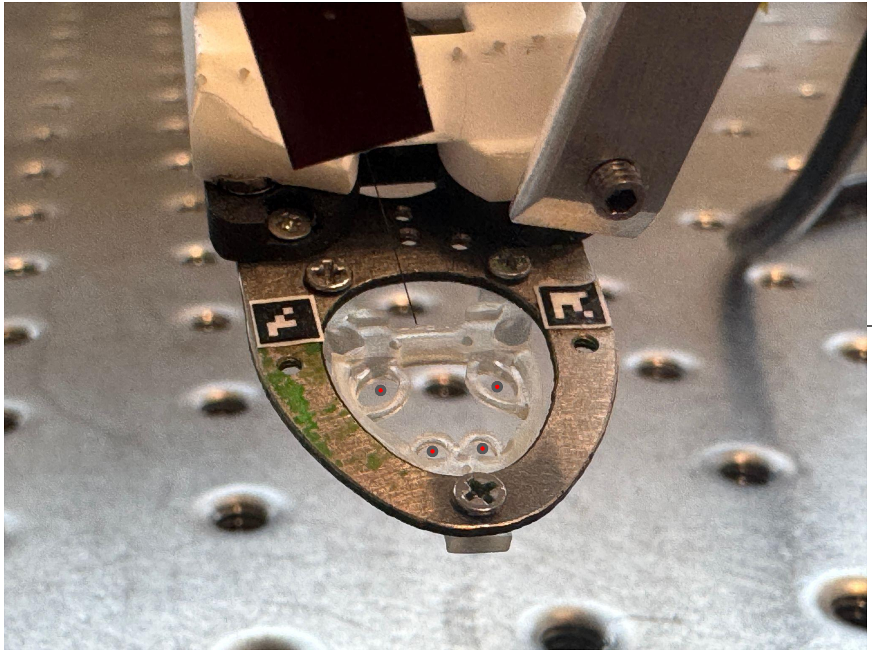
1. Manually identify 4 insertion points on the camera view while all ArUco markers are visible
2. converts 2D pixel coordinates to 3D headplate coordinates using the ArUco marker reference frame

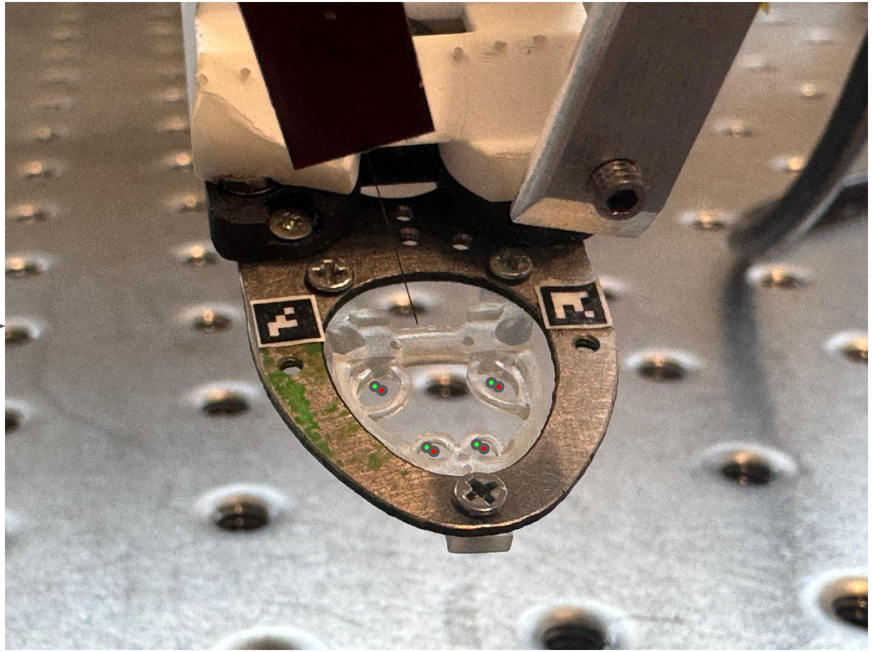
Reference Coordinate System

- **Origin:** Centroid of 4 ArUco tags (44, 32, 29, 49)
- **X-axis:** Direction from tag 44 → 32 (projected to plane)
- **Z-axis:** Average normal vector from all 4 tags
- **Y-axis:** Cross product completing right-handed system

Output

#2 Detection with Aruco tags

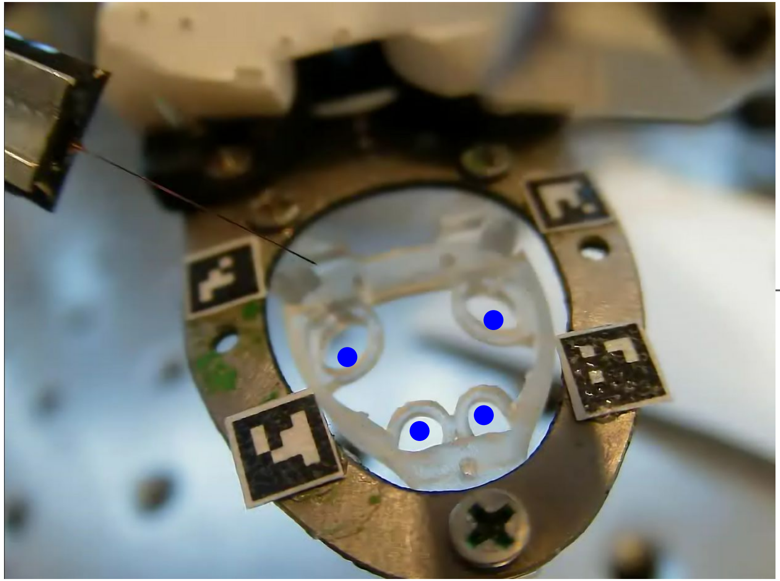




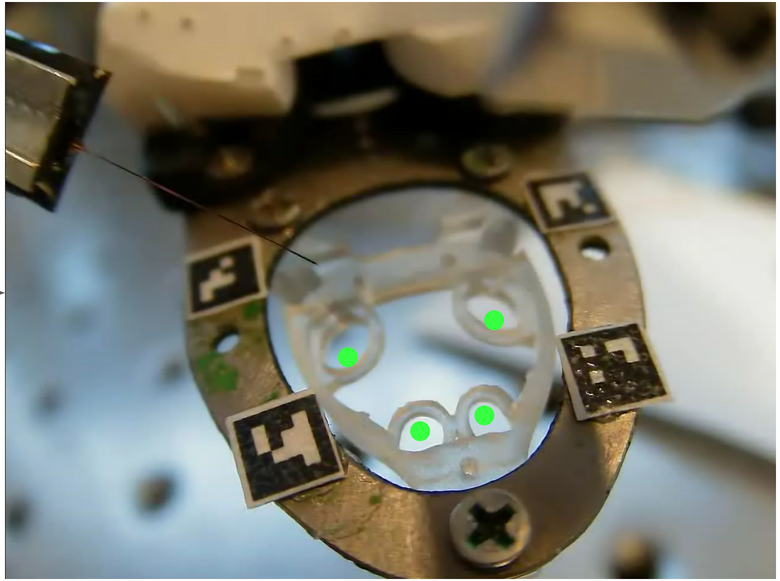
Calibrating Insertion points

Live Detection

#2 Detection with 4 Aruco tags



Calibrating Insertion points



Live Detection

Trajectory Planning

The Goal: Go From Po

Waypoint Structure

position: 3D coordinates (x, y, z)
orientation: Approach direction vector
speed: Velocity scaling factor
waypoint_type: Stage identifier

Trajectory Components

1. Waypoint List: Ordered sequence of positions
2. Distance Metrics: Total path length calculation
3. Time Estimation: Speed-based duration prediction

Point A = The 3 of the probe tip

= The 3D position arget

Waypoint 1
Current Probe position

Waypoint 3
Insertion depth

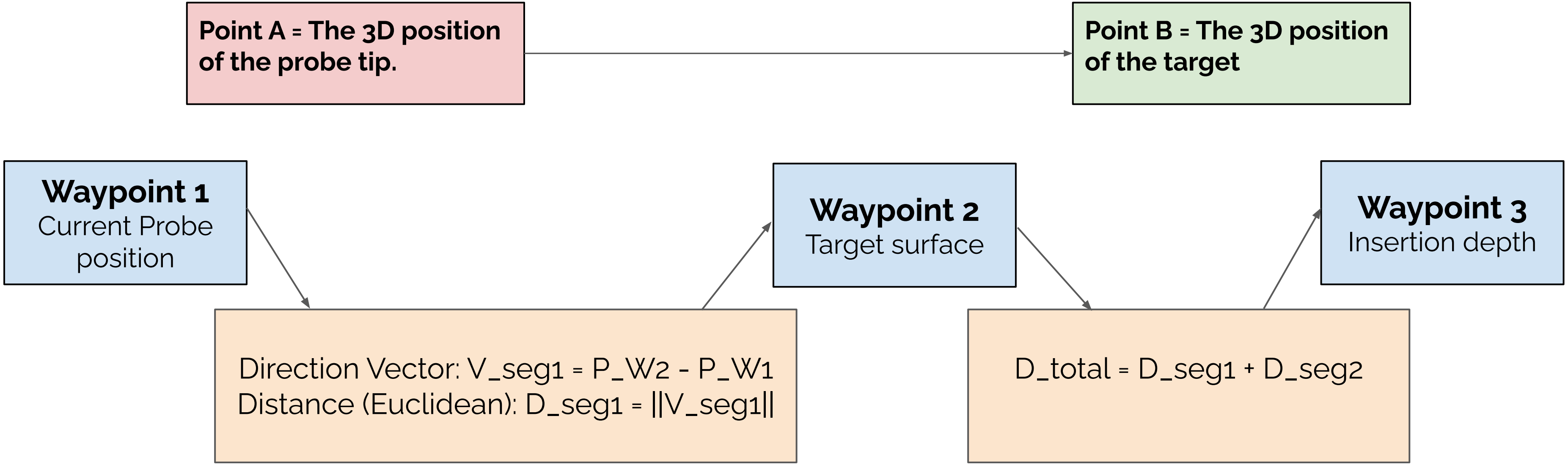
Direction Vector: V_seg1 = P_W2 - P_W1
Distance (Euclidean): D_seg1 = ||V_seg1||

D_total = D_seg1 + D_seg2

$D = \mathrm{sqrt}((X\_W1 - X\_W2)^2 + (Y\_W1 - Y\_W2)^2 + (Z\_W1 - Z\_W2)^2)$

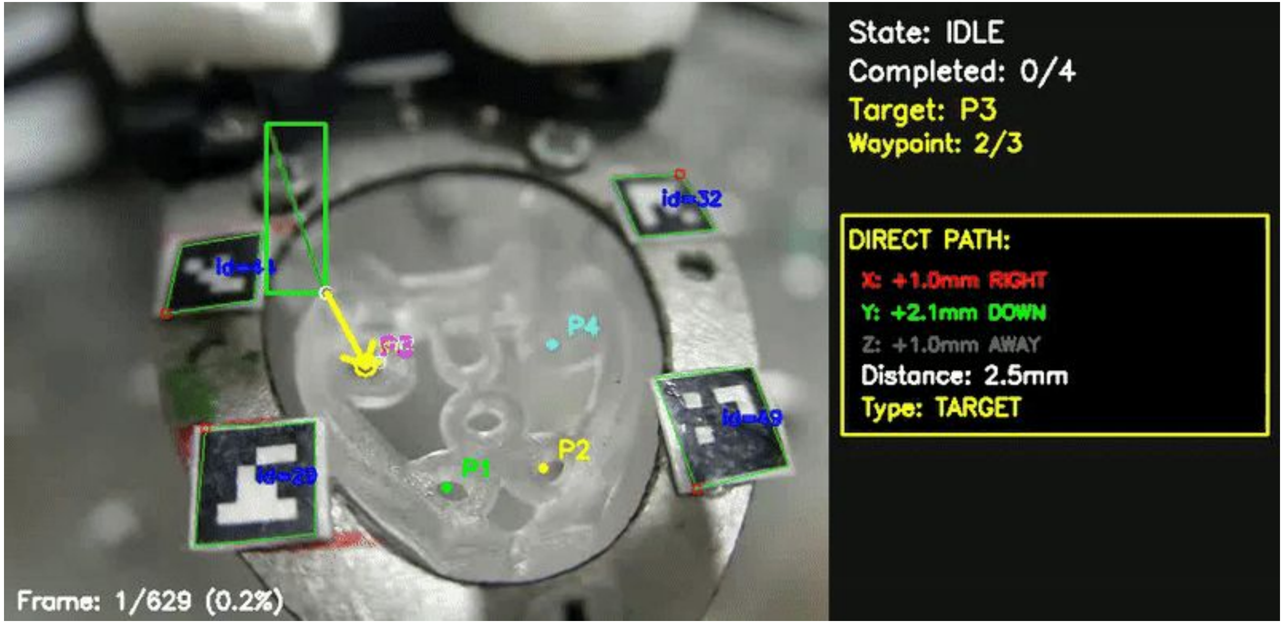
Trajectory Planning

The Goal: Go From Point A to Point B



$$D = \mathrm{sqrt}\left((X\_W1 - X\_W2)^2 + (Y\_W1 - Y\_W2)^2 + (Z\_W1 - Z\_W2)^2 \right)$$

Results



Trajectory Visualization:

- Completed waypoints (gray)
- Current waypoint (cyan)
- Approach phase (yellow)
- Insertion phase (red)
- Retraction phase (purple)

Next Steps

- Validate Accuracy: Perform trial insertions to quantify system accuracy and repeatability.
- Automate GUI: Integrate vision software with the New Scale motor controller.
- Add Multi-Camera Tracking: Implement a stereoscopic camera system to improve 3D tracking.
- Scale to Four Probes: Expand the framework to manage four automated insertions simultaneously.

Impact

Higher-Quality, Reproducible Data

- Computer vision enables consistent and precise insertions.
- Ensures probes are in the correct location, leading to reliable scientific data.

Increased Safety & Cost-Effectiveness

- Real-time feedback (like bend detection) prevents probe breakage.
- Saves money and minimizes tissue damage, improving animal welfare.

Accelerated Scientific Discovery

- Reduces human error and speeds up the entire experimental process.
- Opens the door to new, complex experiments (e.g., multi-probe insertions).

Special Thanks

Travis and **Suahasa** for helping and guiding me throughout the project

Michael for explaining and entertaining all of my questions to understand the previous work and the setup

Thank you

Questions?