**Group Number:** 27
**Eric Ouyang**: eouyang@seas.upenn.edu, **Xiaohan Wang:** xhwang@seas.upenn.edu, **Sohan Lele**: lele03@seas.upenn.edu

# Robot Position Estimator

## Summary

This project presents a complete multi view vision pipeline designed to estimate the 3D position of a small and difficult to detect object, specifically the tip of a needle moving through a reflective gel. The problem is challenging because reflections, low contrast, and visual noise make traditional stereo depth recovery unreliable. To address this, we developed a modular system that extracts 2D tip locations from synchronized top and side camera views and combines them into a triangulated 3D trajectory.

The pipeline is built around several key stages. SAM2 based segmentation isolates the needle. Skeletonization reduces the segmentation mask to a stable one pixel centerline. A convolution based endpoint detector identifies tip candidates. A multi view pairing step aligns corresponding detections across cameras. Each stage went through several iterations in order to resolve issues such as glare artifacts in the side view, inconsistent skeleton branches, and frame rate mismatches when decoding data from ROS bags. Later iterations introduced visual overlays that helped verify spatial alignment, along with a synchronized side by side triangulation view to confirm that the two camera streams matched correctly.

We then validated the reconstructed 3D tip trajectory against a deformation model by building a real world coordinate frame, separating translation and bending phases, and comparing the measured curve with the predicted one. The system achieved a relative error of about seven percent. This level of accuracy is strong given the noise introduced by the gel medium and the limited resolution of the cameras.

Overall, this work provides a functional and extensible 3D tracking framework for laboratory and robotics applications. The development process shows the importance of reliable segmentation, careful data handling, and geometric consistency when designing multi view tracking systems.

## Problem Definition and Audience

The goal of our project is to develop a system that can accurately estimate the 3D position of a small, hard-to-detect point, specifically the tip of a needle moving through a clear reflective gel. In the original proposal, our primary objective was to track the needle tip in real time using a multi-view camera setup and a vision-based processing pipeline. Since then, our

work has focused on turning that idea into a functional system by building the stages needed to reliably detect, isolate, and track the tip across both camera views.

For the scope of this course project, the problem statement generalizes beyond a needle-in-gel setup. The pipeline we are building aims to serve as a reusable 3D position estimator for an object where the tracking target is a single small feature visible from a top view and a side view. This meant designing each step of the process: segmentation, skeletonization, tip extraction, and triangulation in a modular way so the approach could be applied to various laboratory or robotic tasks.

The intended audience for this system remains anyone working with 3D tracking problems in controlled environments, such as robotics labs, medical device researchers, or anyone analyzing motion inside a constrained workspace. As we continue developing the pipeline, our results and iteration insights are geared toward people interested in building lightweight tracking systems without relying on expensive motion-capture hardware.

## Solution and Pipeline

Our system follows a multi-stage vision pipeline that converts raw video frames into 2D tip coordinates for both top and side views, which will later be used for 3D triangulation. Throughout the project, we refined each stage based on experimental results, especially the issues discovered during skeletonization and endpoint detection. The finalized pipeline is summarized below.

1. **Needle Detection using SAM2:**
   Each frame from both camera views is processed with Segment Anything Model 2 (SAM2) using point-based prompts. SAM2 consistently isolates the needle from the reflective gel environment, producing clean binary masks. This segmentation step replaced our original stereo baseline, which failed to recover depth inside the gel.
2. **Skeleton Mask Extraction (Skimage):**
   The SAM2 mask is converted into a 1-pixel-wide centerline using the skimage skeletonization function. Skeletonization removes thickness variations and produces a stable geometric representation of the needle for later processing. This step worked reliably for the top view, but required additional tuning for the side view due to glare-induced mask artifacts.
3. **Tip Localization:**
   To identify the needle tip, we convolve the skeleton with a 3×3 neighbor-counting kernel. Pixels with exactly one neighbor are marked as endpoints. In most frames, this detects the true tip and the base of the needle. When skeletonization produces small branching artifacts, additional filtering is applied to discard the false endpoints. The detected tip position is recorded for each frame to form a 2D trajectory.
4. **Multi-View Tracking:**
   The same segmentation, skeletonization, and endpoint detection steps are applied independently to the top and side camera videos. Frame indices are synchronized so that

each pair of detected points corresponds to the same moment in time. This produces two aligned sequences of 2D tip coordinates, the top camera providing (x, y) and the side camera providing (y, z).

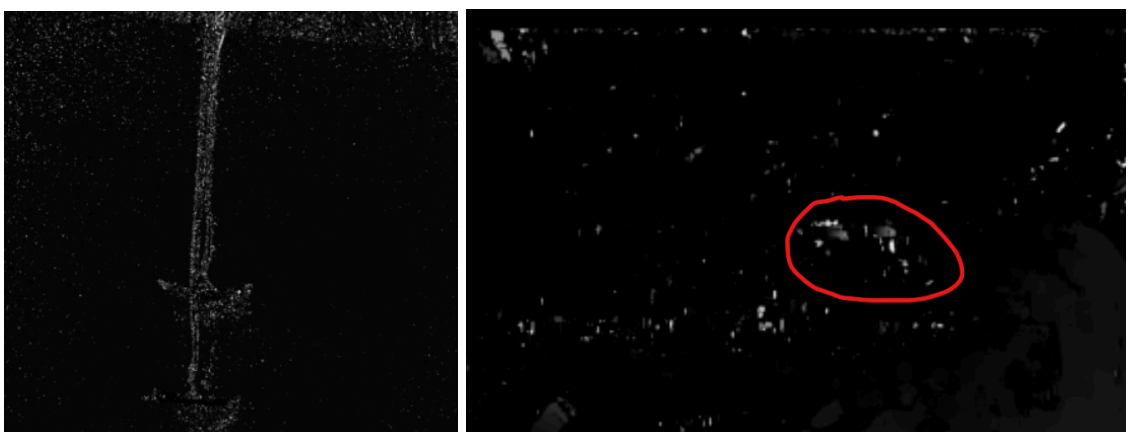
5. **3D Triangulation:**
   The corresponding 2D tip coordinates from the top and side views will be used for triangulation. Using the projection matrices of both cameras, we will compute the 3D location of the needle tip for each frame and reconstruct its full trajectory in real-world coordinates.

## Baseline Results

Before developing the full 3D tracking pipeline, we first evaluated two baseline approaches to understand the difficulty of the needle-in-gel problem.

*Stereo Vision Baseline:*

As described in our project proposal, our initial baseline used a stereo setup to estimate depth by computing disparity between synchronized top and side views. This method performed reasonably well when the needle was fully visible in the air. The structure was clear enough to detect, and the approximate tip location could be inferred. However, once the needle entered the reflective gel, the stereo pipeline failed almost completely. The gel introduced strong reflections, distortions, and low-contrast regions, making it impossible to recover meaningful disparity. This confirmed that stereo alone was insufficient for robust tip tracking inside the gel environment.



*(Figure 1: Stereoview results of needle without gel (Left). Stereo image of needle in gel, circled needle position (Right).)*

*First Development Iteration: SAM2 Segmentation*

To address the limitations of the stereo method, our first iteration tested Segment Anything Model 2 (SAM2) for frame-by-frame segmentation. This significantly improved the visibility of the needle. SAM2 generated clean, consistent masks that captured the full shape of the needle even when the gel caused heavy background noise. Unlike the noisy and unstable stereo disparity output, the SAM2 masks provided clear boundaries and high contrast between the needle and the surrounding gel, making downstream processing (skeletonization and tip

detection) feasible. The example frames below show SAM2's segmentation quality across different needle positions in gel:

*(Figure 2: SAM2 segment of needle in gel in white (Left). SAM2 segment of needle in gel after moving further (Right).)*

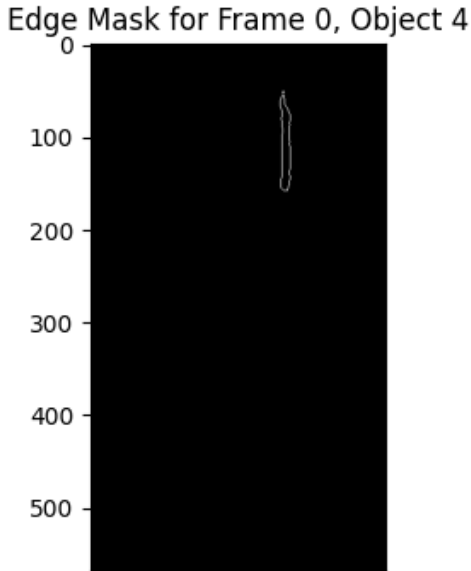
These results established SAM2-based segmentation as a strong baseline and provided the foundation for the later pipeline stages we developed during the midterm phase. During the midterm phase, three major development iterations occurred. Each iteration focused on improving one part of the pipeline. Preprocessing, skeletonization, and endpoint detection were all important steps toward reliable 2D and eventually 3D tip tracking.

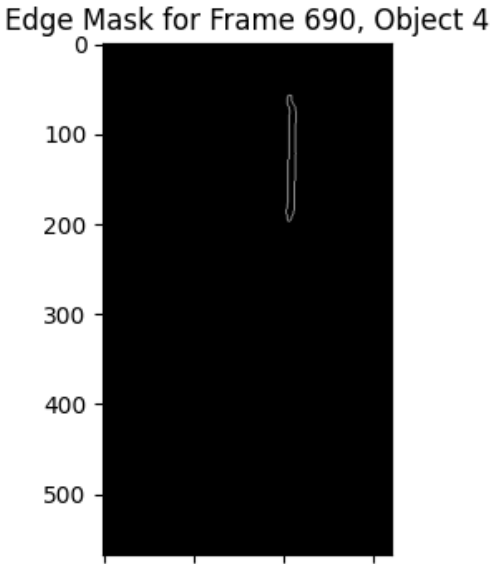
**Iteration 1: Edge Masking / VGGT Exploration**

Our first attempt at post-processing the SAM2 masks involved using OpenCV's erode() and dilate() operations to clean the segmentation, but this actually made the overall mask worse. Instead of sharpening the structure, these operations distorted the shape of the needle and reduced mask consistency across frames. Because of this, the entire segmentation became less reliable for any downstream processing.

We then tried extracting only the contour of the SAM2 mask to create an edge-only representation. However, converting the mask to edges removed all interior information, which made skeletonization impossible since we could no longer extract a "middle" for the needle. The edge maps were also unstable, sometimes splitting into multiple thin curves or losing parts of the boundary altogether. So this approach ultimately reduced the clarity needed for tip detection.

Examples of edge masks:



We also briefly explored using VGGT, since it reconstructs 3D structure from video frames and can theoretically provide the tip position. However, VGGT requires clean object isolation as input. Because our needle must be segmented manually for VGGT to work, it contradicted our goal of a fully vision-based, automated detection pipeline. We decided not to pursue VGGT further.

*Key Takeaways from Iteration 1:*

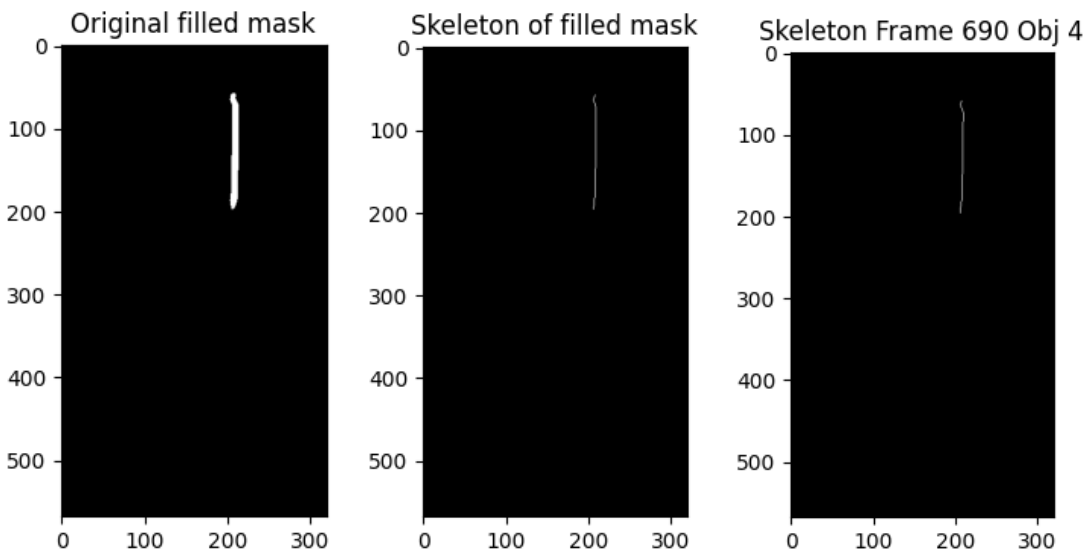
- Edge-only masks were too inconsistent to support tip localization
- Erosion/dilation must be tuned very carefully or avoided
- VGGT is not suitable unless segmentation is solved first

This iteration helped us realize the pipeline needed more stable morphology, not additional edge filtering.

**Iteration 2: Skeletonization + ROS Bag Decoder**

After the failures of edge masking, we implemented a full skeletonization approach using the skimage library. We converted each filled SAM2 mask into a 1-pixel-wide skeleton. The skeleton preserved the full needle shape while removing thickness, making tip detection more straightforward.

Example skeletonization results:




This worked well on the top camera, but the side camera presented an issue: a strong glare near the top of the gel created a curved, bright region. SAM2 sometimes mislabeled this glare, and the skeleton would bend toward it. As a result, the skeleton occasionally had an unnatural curvature at the start of the needle.

Glare issue example:

To collect additional trial data from the existing hardware setup, we added a ROS bag decoding step. Each experiment in the lab is logged as a large ROS bag file containing all topics, so we wrote a script to automatically extract only the two camera image topics and convert them into MP4 videos. We then used ffmpeg in order to sync both videos' frame rates to 15 and then convert them to individual frames. This gave us paired top and side view sample video frames in

a simple format that we could feed directly into the SAM2 and skeletonization pipeline. The videos were also recorded at different frame rates, so syncing them up so all frames align was crucial. The cameras are lower quality and have a fixed focal length, but this ensures we know what the coordinates are. The frames shown below are example views decoded from one of these ROS bag recordings.

*Key Takeaways from Iteration 2:*

- Skeletonization produced a stable 1-pixel centerline for most frames
- Side-view glare caused early-frame curvature, affecting consistency
- Extracted top and side camera videos from the ROS bag

This iteration gave us the first functional representation of the needle shape across frames.

**Iteration 3: Endpoint Detection and Point Tracking**

Once we had consistent skeletons, we designed a method for identifying the tip endpoints. To do this, we created a 3×3 convolution kernel:

[1, 1, 1]

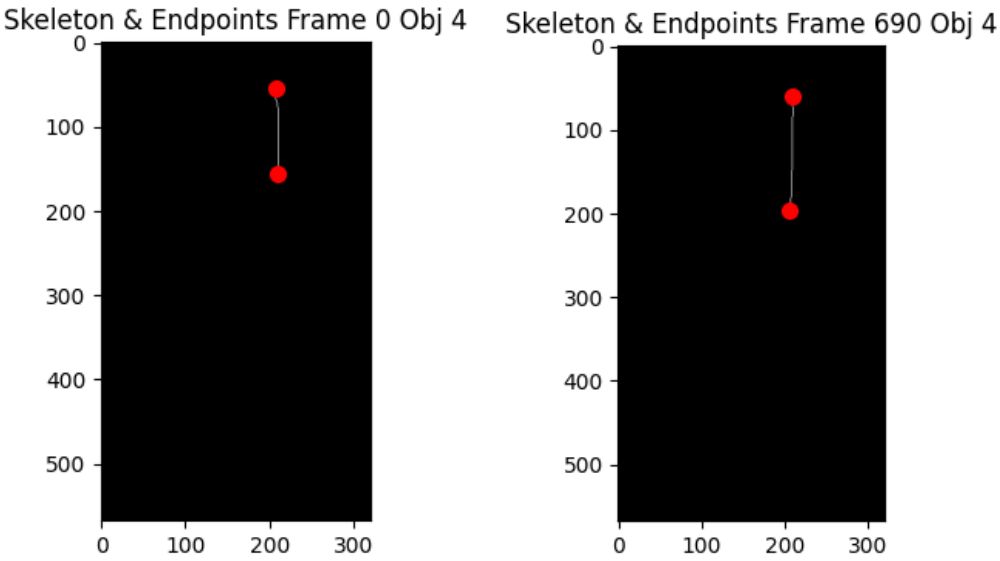
[1,10, 1]

[1, 1, 1]

When convolved with the skeleton, this kernel counts the number of neighboring skeleton pixels. Points with only one neighbor correspond to endpoints. We then overlaid the detected endpoints on top of the skeletonized mask to verify accuracy, and also exported the endpoint

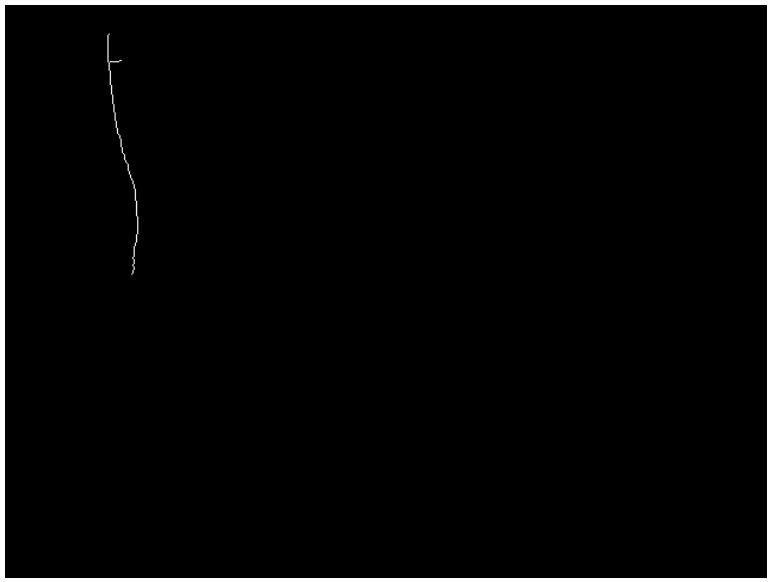
coordinates to a .json file so we could perform simple analysis and automatically remove any unwanted middle points.

Example endpoint detection results:




Most frames produced clean endpoints, and the tip was correctly identified. However, in some cases, skeletonization produced a third small branch, which created an unwanted extra endpoint.

Example of third-branch failure:



This happened mainly in frames where illumination changed abruptly or when the base pixel range shifted unexpectedly across frames.

*Key Takeaways from Iteration 3:*

- The convolution-based endpoint detector worked very reliably when the skeleton was clean
- Extra branches in the skeleton created false endpoints and need additional filtering
- The tip endpoint is consistently the uppermost point in the skeleton, so future iterations may sort endpoints spatially to reject false branches

Together, these iterations formed the backbone of our midterm pipeline and revealed the remaining challenges that shaped the next stages of development.

## Iteration Results

### Iteration 4: Overlay on Original Video

In this iteration, we integrated the processed skeleton and detected tip locations back onto the original RGB video frames. This step served as an important qualitative validation tool, allowing us to visually confirm that the pipeline's outputs remained spatially aligned with the real needle across all the frames. Compared to viewing only the binary masks or skeletons, the overlays made discrepancies such as glare-induced bending or small segmentation errors much easier to identify.

To support this visualization, we extended our pipeline to automatically load each segmented SAM2 mask, compute its corresponding skeleton, extract endpoints, and render those results directly onto the raw camera frames. The script processes every frame in sequence and outputs two synchronized overlay videos (one for the top camera and one for the side camera). The resulting annotated views gave us a frame-by-frame understanding of how reliably the tip was being tracked and whether any downstream filtering would be necessary.

Overlay of Skeleton and Detected Tip on Original Video Frames

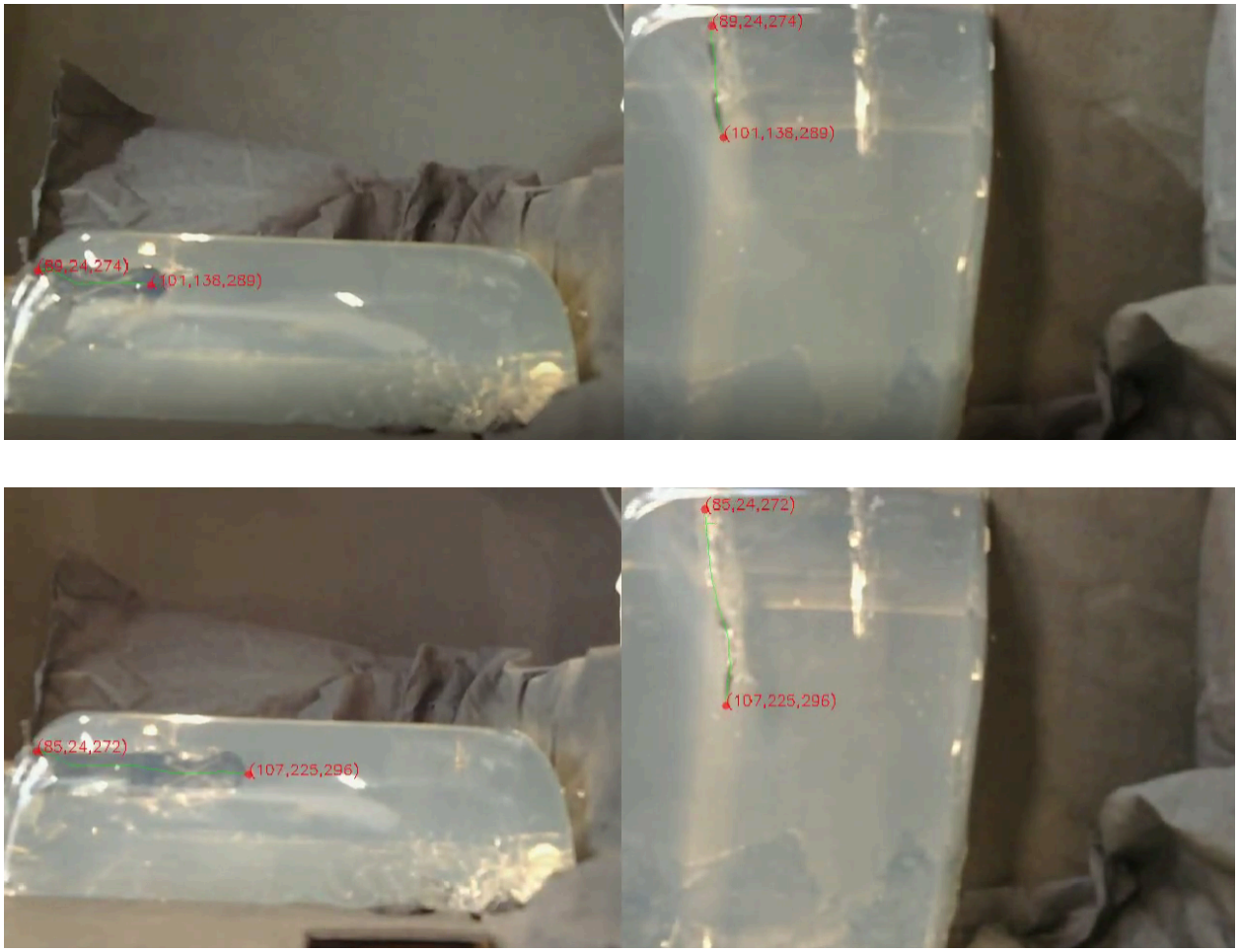
The overlay pipeline also included several practical improvements:

- Automated directory creation and file handling, allowing the entire sequence to be processed in a single run
- Consistent rendering of skeletons and detected endpoints, blended onto the video frames for maximum visibility
- Frame-index tracking, ensuring future stages (triangulation) could use synchronized endpoints from both camera views without mismatch
- Diagnostic printouts, showing the first and last frames with detected points to quickly validate the continuity and integrity of the tracking data

This iteration marked the first point where the full visual behavior of the system could be inspected, and it confirmed that the earlier segmentation and skeletonization stages produced sufficiently reliable outputs for multi-view triangulation in the next phase.

**Iteration 5: Triangulation Side by Side Video**

To begin reconstructing the 3D trajectory of the needle tip, we combined the synchronized 2D endpoint detections from the top and side camera views into a unified coordinate representation. The two annotated videos generated in the previous iteration were placed side-by-side, enabling us to visually confirm frame alignment and verify that corresponding tip locations were being matched correctly across the two views.




Frames from the Final Triangulated Overlay Video

Because each camera observes a different projection of the needle, the coordinate ordering differs between the two views. In our setup:

- **Left (side) camera:** outputs coordinates in the form **(z, y)**
- **Right (top) camera:** outputs coordinates in the form **(y, x)**

To construct a consistent 3D point (x, y, z), we paired points from the two cameras based on their shared y-coordinate, which is visible in both views. For each frame, the algorithm identifies the closest matching pair of tip detections from the two camera streams by comparing their y-values. After pairing, the coordinates are reassembled into a single 3D point using:

- **x** from the *top* view
- **y** from the shared coordinate
- **z** from the *side* view

Before triangulation, we also filtered out any spurious detections caused by the "third-branch" skeletonization failure identified in Iteration 3. Removing these interior artifacts ensured that only true endpoints were used for reconstruction.

Across all processed frames, the triangulation procedure produced a continuous and stable 3D tracking of the needle tip. The side-by-side visualization shown above reflects the final matched output used for subsequent 3D analysis.

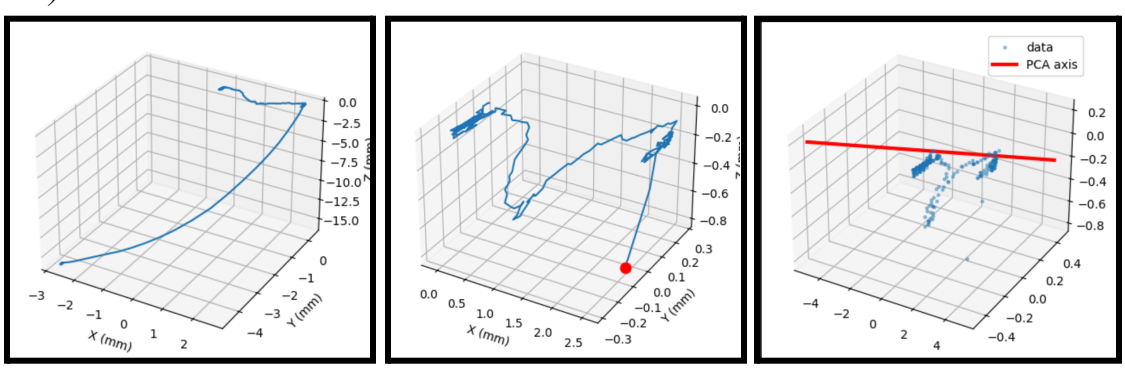
**Iteration 6: Real world validation**

To evaluate the reliability of our vision-based tracking system, we validated the reconstructed 3D needle trajectory against the predictions generated by our mathematical deformation model. This comparison allowed us to assess whether the recovered motion and geometry were consistent with the expected physical behavior of the needle.

a. Coordinates Reconstructions:

Using the 3D tip positions obtained from Iteration 5, we reconstructed the needle's motion in a world coordinate frame. To establish this frame, we first commanded the needle to perform a short translation motion. This provided a known direction of travel that could be used as a reference axis.

*(Images: left to right - raw reconstructed trajectory, zoomed trajectory, PCA-estimated axis)*




To align the coordinate system:

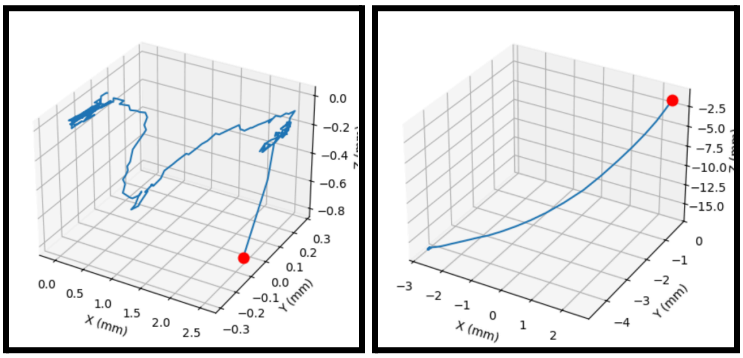
- The **x-axis** was defined by applying PCA to the translation segment of the motion (right image)
- The **y-axis** was taken to align with gravity, based on the experiment's physical orientation
- The **z-axis** was reconstructed using the cross product:

$$z = np.cross(x, y)$$

This produced an orthonormal basis suitable for comparing the measured tip trajectory with theoretical predictions.

b. Data Wrangling: Separate data into different motion parts.

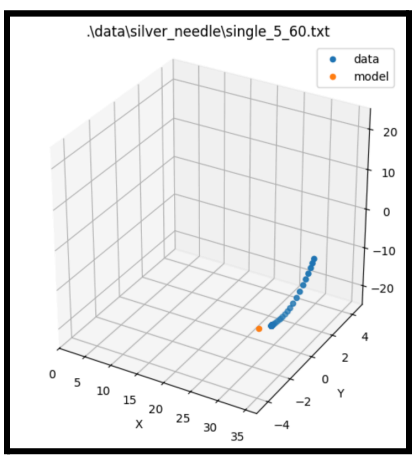
Before comparing the final tip position to the model, we isolated the bending segment of the motion. The raw 3D trajectory contains both translation and bending phases, so filtering and segmentation were necessary to extract the portion relevant to deformation analysis.




Applying smoothing and stability filters produced a much cleaner representation of the bending path, enabling a more meaningful comparison with the model.

c. Data Visualization: Plotting out the prediction result vs. collected data result.

We then plotted the reconstructed bending curve against the model's predicted deformation.




The final result shows:

- **Distance error:** 2.547 mm
- **Total needle length:** 37 mm
- **Relative error:** 7.075%

Given the limitations of the imaging setup, noise in the gel medium, and the relatively low resolution of the cameras, this accuracy is strong. The reconstructed geometry closely tracks the predicted bending behavior, demonstrating that the vision pipeline provides sufficiently reliable measurements for physical validation.

## Timeline, Milestones, and Member Duties

**Timeline**

- **By midterm (COMPLETED):**
  - Implement SAM2-based segmentation for both top and side views
  - Add skeletonization to obtain 1-pixel centerlines of the needle
  - Develop an endpoint detector and demonstrate 2D tip tracking across frames
- **By December 5 (COMPLETED):**
  - Achieve full 3D point tracking, including triangulated outputs with accurate real-world measurements
  - Align the top and side camera streams and overlay their tracked 2D tip positions
  - Perform 3D triangulation of the tip using calibrated camera parameters
  - Convert triangulated points into real-world coordinates and evaluate tracking accuracy

**Milestones**

- Obtain accurate and consistent masks using SAM2 segmentation (COMPLETE)
- Implement 2D tip tracking from the top camera view (COMPLETE)
- Extend the system to the side view and verify stable skeletons and endpoints (COMPLETE) (with some remaining glare issues)
- Overlay synchronized top/side trajectories and compute 3D tip positions via triangulation (COMPLETE)
- Next: Integrate world coordinate calibration and real measurement scaling for quantitative analysis (COMPLETE)

**Member Duties**

All members will collaborate on data wrangling, model implementation, testing, and reporting to maintain consistent progress throughout the project. To ensure balanced contributions, tasks are divided by focus area:

| Member | Primary Roles |
|---|---|
| Eric Ouyang | Data collection, model setup, and SAM2 integration |

| | |
|---|---|
| Xiaohan Wang | Data processing, model refinement, and triangulation |
| Sohan Lele | Experiment coordination, documentation, and presentation preparation |

**Future Work**

Several directions remain for extending and strengthening the system developed in this project:

*Real-time Processing:*
Implementing a fully real-time version of the pipeline would allow the needle tip to be tracked live during experiments, enabling immediate feedback and potential closed-loop control.

*Path Prediction:*
Incorporating predictive models such as motion forecasting or deformation priors could help anticipate the needle's future trajectory and improve stability when visual data becomes noisy or ambiguous.

*Improved Robustness:*
While the current system performs well on cleaner datasets, some trials exhibit noticeable discrepancies between predicted and measured tip positions. Future work should focus on reducing sensitivity to glare, segmentation variability, and skeletonization artifacts to ensure consistent performance across all experimental conditions.

**Lessons Learned**

Throughout the development of this system, we gained several important technical insights:

*Segmentation Tools:*
We learned how to effectively use modern segmentation models such as SAM2, including tuning prompts and handling challenging visual conditions like glare and low contrast.

*Data Wrangling and Processing:*
The project reinforced the importance of careful data cleaning, filtering, and structuring, especially when synchronizing multi-view datasets and removing noisy or inconsistent detections.

*Skeletonization Techniques:*
Implementing skeletonization taught us how algorithmic choices affect geometric stability, and how small artifacts can significantly impact downstream endpoint detection.

*Model and Resolution Considerations:*
When applying vision-transformer-based systems, maintaining consistent image resolution across frames proved critical for ensuring stable outputs and minimizing segmentation drift.