Gesture-Guided AI Task Planning for Virtual Surgical Robots in AMBF

Yuvraj Singh, Taiwen Gao, Shaurya Mallampati, Arghya V, Dhatri Medidhi, Safwan Mohammed, Rudraansh

Abstract

This project developed a pipeline for gesture-based task planning in surgical robotics using the Asynchronous Multi-Body Framework (AMBF). The system's goal was to capture hand movements from a standard webcam and convert them into structured surgical commands that could be planned and executed in a da Vinci–style simulation. The architecture combined gesture recognition with MediaPipe and OpenCV, an intent parser that encoded commands into JSON, a ChatGPT-based task planner constrained by schemas, a limited library of deterministic surgical skills built in ROS, and a safety supervisor intended to check for unsafe actions.

We successfully built the initial pipeline: gestures were recognized, converted into JSON, processed by ChatGPT, and executed as basic robotic actions in AMBF. However, the later phases of the project—particularly the implementation of a full safety supervisor and a broader skill library—were not completed. The outcome demonstrates a functional proof-of-concept but falls short of a complete end-to-end surgical workflow.

Introduction

Robotic platforms like the da Vinci system have transformed minimally invasive surgery by increasing precision and dexterity. Current systems, however, remain tied to teleoperation, where surgeons must directly manipulate robotic controls. This allows accuracy but restricts the role of autonomy in surgery.

Gesture-based control offers a more natural interface. Instead of issuing direct motor commands, a surgeon could signal high-level instructions—such as "clamp vessel"—through simple hand motions. An AI system could then translate these symbolic gestures into structured robotic actions, shifting the surgeon's role toward oversight while the robot executes low-level details.

This project aimed to test that idea using AMBF, an open-source simulator for robotic systems. The pipeline combined computer vision for gesture recognition, structured intent parsing, a large language model for task planning, deterministic skill modules, and a safety supervisor. While not all elements reached completion, the system did demonstrate that natural gestures can be linked to robotic actions through AI-based planning.

Methods

The pipeline began with gesture recognition. A consumer-grade webcam recorded video at 30 fps, and frames were processed by MediaPipe Hands, which identified twenty-one landmarks per hand. These landmarks—covering fingertips, joints, and palm centers—were then analyzed with OpenCV. The classifier measured distances and angles between points and applied thresholds to categorize gestures. For example, a pinch gesture was detected when the thumb and index finger were close, while an open palm was classified when fingertips spread beyond a certain radius. Each recognized gesture was mapped to a symbolic label such as “clamp” or “retract.”

An intent parser converted these labels into JSON structures. A pinch detected near a simulated vessel, for instance, might produce `{action: "clamp", target_hint: "nearest vessel"}`. The parser also filtered out uncertain gestures and maintained a short buffer to reduce accidental triggers.

The planning stage used the ChatGPT API with a JSON schema to ensure consistency. The schema only allowed certain primitives like clamp, retract, or incise, and required fields such as target and trajectory. This kept outputs predictable and machine-readable. For example, `{action: "clamp", target_hint: "nearest vessel"}` could become `{action: "clamp", target: "vessel_03", trajectory: "safe_path_A"}`. Function calls were also supported, letting the planner query for targets or safe paths when needed.

The skill library contained deterministic ROS nodes implementing atomic surgical actions. Each skill accepted structured inputs and produced AMBF motor commands. The clamp skill, for example, aligned the tool with a vessel and applied force to close; retraction moved tissue meshes backward; incision advanced a scalpel along a straight-line path. These skills were parameterized so they could adapt to different targets.

The safety supervisor was designed as a monitoring process. Its rules included force thresholds, clearance checks, and collision avoidance with no-go zones. While partly implemented, the supervisor was not fully integrated.

AMBF served as the environment, simulating the da Vinci robot with realistic dynamics. Commands traveled from the webcam through each stage of the pipeline and ultimately executed in the simulator.

The work was divided into phases. Phase 1 set up AMBF, ROS nodes, and gesture recognition. Phase 2 integrated ChatGPT with the planner and demonstrated gesture-driven execution in AMBF. Phase 3 attempted to add the safety supervisor, and Phase 4 was meant to show a complete task under supervision. Phases 3 and 4 were only partially completed.

Results

The prototype worked in its initial scope. Five gestures were recognized with over ninety-five percent accuracy under controlled conditions. JSON commands were consistently valid, and ChatGPT returned plans with latency under 200 ms. Skills executed in AMBF with smooth, repeatable results—for instance, a pinch gesture successfully triggered a clamp action on a vessel.

The limitations were clear. The safety supervisor remained incomplete, so safety could not be guaranteed. The skill library included only a small set of atomic actions, preventing more advanced procedures. The final demonstration of a full surgical task was not realized.

Discussion

The project shows that connecting gesture recognition, structured parsing, and AI-driven planning to robotic execution is feasible. ChatGPT proved effective as a symbolic planner when outputs were constrained by schemas.

There were practical challenges. Gesture recognition degraded in poor lighting or with occlusion. ChatGPT raised issues of determinism and cost if scaled further. Most importantly, the absence of a functional safety supervisor blocked progress toward complex tasks, since safe operation is critical in any surgical setting.

Even so, the results establish that a low-cost, accessible pipeline for gesture-driven robotic interaction is possible and provide a clear roadmap for further work.

Conclusion

The system demonstrated the conversion of hand gestures into structured plans executed by a simulated surgical robot in AMBF. The base pipeline functioned, but advanced goals like robust safety monitoring and an expanded skill set were not achieved.

Despite incomplete progress, the project highlights the potential of natural human input as an interface for surgical robotics. It offers a foundation for future research that could combine computer vision, AI planning, and robotic control in ways that improve both usability and autonomy.