School of Engineering, University of Western Australia

Depth-Camera-Based Real-Time Human-to-Humanoid Pose Retargeting

Travis Ryan

Word Count: 6542

October 13, 2025

Executive Summary

This project presents the design and implementation of a real-time, depth-camera-based human-to-humanoid pose retargeting system for the Unitree G1 robot. The work addresses a key challenge in humanoid teleoperation: enabling natural and intuitive imitation of human upper-body motion without the use of wearable sensors or costly motion-capture setups. By leveraging low-cost depth sensing and open-source robotics software, the system demonstrates that accurate and stable human motion replication can be achieved using only a single depth camera and standard computing hardware.

The system integrates the Luxonis OAK-D Lite RGB-D camera for visual input, Google's MediaPipe framework for real-time pose estimation, and a ROS 2-based middleware pipeline employing Cyclone DDS for distributed communication. Extracted human keypoints are mapped into the robot's reference frame and solved through inverse kinematics using Pinocchio and CasADi, producing joint commands executed by the Unitree G1's low-level controller. The software architecture was designed for modularity, transparency, and reproducibility, with development supported by simulation testing in the Unitree MuJoCo environment prior to hardware deployment.

Experimental results verified that the system successfully reproduces human upper-body motions such as arm extensions, reaches, and single-arm gestures. The measured end-to-end system latency averaged approximately 300 ms, primarily due to the 180 ms delay from camera frame capture to host delivery. Excluding this hardware-induced latency, the perception and control pipeline operated well within the 150 ms design goal, achieving an effective delay of roughly 115 ms. The system remained stable during continuous operation, demonstrating smooth, realistic motion reproduction suitable for telepresence and gesture demonstration tasks.

Key limitations include monocular depth ambiguity under occlusion, kinematic mismatch between the human operator and the robot's limb geometry, and compounded latency from filtering and communication. Despite these constraints, the project successfully achieved all primary objectives, providing a functional, open-source baseline for low-cost humanoid teleoperation research.

Future work may focus on improving depth accuracy and occlusion robustness through stereo fusion or IMU augmentation, implementing user-specific kinematic calibration, and introducing predictive filtering to further reduce latency. These developments will strengthen the framework's applicability to collaborative robotics, telepresence, and intuitive human-robot interaction.

List of Publications

Submitted for publication:

Zhang, H., Inayat-Hussain, J., Smith, J., Ryan, T., and Bräunl, T. (2025, December 2–4). *A Comprehensive Control Architecture for Humanoid Robots: Integration of Locomotion, Manipulation, and Navigation Subsystems.* 2025 Australasian Conference on Robotics and Automation (ACRA 2025), Perth, Western Australia. Submitted on 17 September 2025.

Acknowledgements

I would like to express my sincere gratitude to **Professor Thomas Bräunl** for his invaluable supervision, guidance, and encouragement throughout the course of this project. His expertise and insight were instrumental in shaping both the technical direction and the overall success of this work.

I would also like to thank **Joel Smith** and **Oliver Zhang**, with whom I worked in parallel on the humanoid robot platform. Their collaboration, technical discussions, and shared ideas greatly contributed to the development and refinement of this project.

Finally, I wish to extend my deepest appreciation to my parents, **Kevin and Debra**, for their unwavering support and encouragement throughout my studies and this project. Their patience and belief in me made this achievement possible.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviation	Definition
AI	Artificial Intelligence
API	Application Programming Interface
DDS	Data Distribution Service (ROS 2 middleware)
DoF	Degrees of Freedom
FPS	Frames per Second
IMU	Inertial Measurement Unit
IK	Inverse Kinematics
ISO	International Organisation for Standardisation
OAK-D	Depth Camera Model (Luxonis)
QoS	Quality of Service
RGB-D	Red, Green, Blue, Depth (camera data)
ROS 2	Robot Operating System 2
SDK	Software Development Kit

Term	Description
CasADi	Framework for nonlinear optimisation problems.
Cobotics	Collaborative robotics.
DepthAI	API for OAK-D cameras providing depth measurement capabilities.
Kalman Filter	Recursive estimator for smoothing noisy measurements.
MediaPipe	Machine learning framework for real-time human pose estimation.
MuJoCo	Multi-Joint dynamics simulator used for robot physics and control validation.
Pelvis Frame	Robot-centric coordinate frame originating at the pelvis centre.
Pinocchio	Rigid-body dynamics library used for kinematic solving.
Teleoperation	Remote control of a robot through real-time human motion input.
Unitree G1	Humanoid robot platform used as the physical testbed in this work.

1 Introduction

Humanoid robots have long represented one of the most ambitious frontiers in robotics research, combining perception, motion control, and human-machine interaction into a unified system capable of replicating human behavior. In particular, the ability for a humanoid robot to imitate human movement in real time is a significant challenge [4], requiring precise sensory data acquisition, accurate kinematic mapping, and low-latency control. Recent advances in motion capture and perception systems have enabled more natural and responsive teleoperation interfaces, allowing robots to mirror human actions through vision-based systems rather than wearable motion sensors or controllers [5, 4].

The challenge addressed by this project lies in bridging the gap between human motion understanding and robotic execution within the field of humanoid teleoperation. This issue has wide-ranging implications for both academia and industry. For the scientific community, developing robust pipelines for real-time motion imitation contributes to the broader understanding of embodied artificial intelligence, human-robot interaction, and perception-driven control. From an industrial perspective, reliable teleoperation frameworks can enhance the capabilities of service and collaborative robots, allowing them to safely operate alongside workers or in environments that are hazardous or inaccessible to humans [6]. This has potential benefits across domains such as healthcare, manufacturing, and remote inspection, where intuitive human-guided control can improve safety and operational efficiency. Furthermore, the outcomes of this project may inform future research directions in cobotics [7] and autonomous systems.

Figure 1.1: Unitree G1 humanoid robot [1].

1.1 Motivation and Context

Traditional methods of humanoid robot control often rely on predefined motion sequences or complex inverse kinematic solvers that require substantial manual tuning. As noted in previous works [4, 5], such approaches lack the intuitiveness and adaptability of direct motion imitation. Real-time motion capture techniques such as those employing inertial sensors or optical tracking, have achieved promising results [4, 5]. However this comes at the cost of requiring complex setups and/or expensive equipment, such as motion capture rigs requiring arrays of cameras, or subjects having to wear multiple markers over their entire body [8]. Such limitations create boundaries to the access of this technology to both individuals and institutions. This in turn can limit its availability to open source development[9].

Recent research in humanoid imitation learning and teleoperation [4, 5, 8] consistently demonstrates that while motion capture and inertial-based systems provide accurate results, their complexity and cost hinder scalability and accessibility. By contrast, advances in vision-based AI frameworks such as *MediaPipe* [10] and *BlazePose* [11] have shown that comparable performance can be achieved using low-cost RGB-D cameras, enabling markerless, real-time human pose estimation. This transition from proprietary, lab-scale systems to open-source, embedded solutions marks a significant shift in the state of the art, with implications for the broadening of humanoid control research and the expansion of collaborative robotics in both academic and industrial settings. When combined with robotic middleware such as ROS2 [12], these perception pipelines create scalable and modular frameworks that support real-time teleoperation and human-robot imitation.

1.2 Project Objectives

The primary aim of this project is to develop a complete and functional software pipeline for upper-body motion imitation on the Unitree G1 humanoid robot [2], enabling the robot to accurately reproduce a human operator's upper-body movements in real time. This capability addresses current limitations in intuitive humanoid teleoperation, where existing approaches rely heavily on costly or invasive motion-capture systems. The proposed system integrates AI-based perception through the *MediaPipe* framework [10], depth sensing using the *OAK-D Lite* camera [13] running Depth-AI [14], and distributed communication via a *ROS2* network. The collected skeletal and depth data are processed and transmitted to the Unitree G1's control software, which calculates inverse kinematics to reproduce the operator's motion within the robot's own kinematic reference frame [15]. Together, these components create an accessible, open-source solution for accurate, low-latency imitation of human motion, contributing to advancements in collaborative robotics and telepresence.

To achieve this aim, the project focuses on the following specific qualitative and quantitative objectives:

- Implement an accurate 3D pose estimation module using *MediaPipe*, capable of detecting upper-body landmarks with an inference rate of 15 fps.
- Integrate the *OAK-D Lite* camera for synchronised RGB-D capture and real-time spatial reconstruction of key upper-body joints.

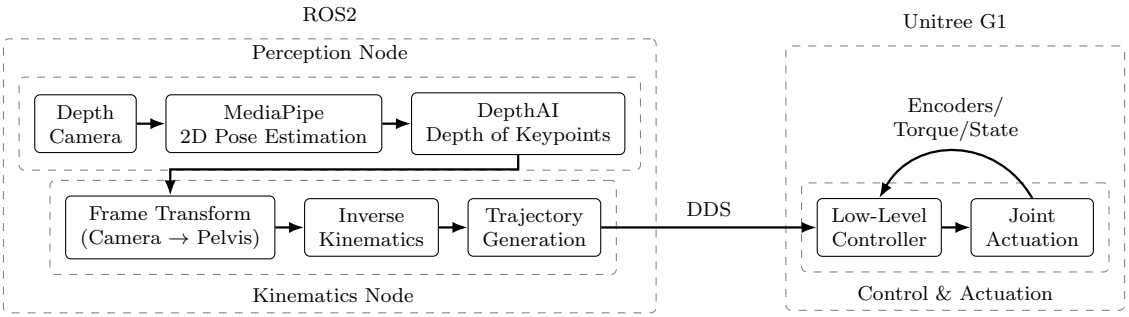


Figure 1.2: Proposed system architecture for perception to control pipeline.

- Establish a reliable *ROS2* communication pipeline with latency under 150 ms to transmit pose data between perception and control nodes in real time.
- Develop a transformation and inverse kinematics layer that maps human joint coordinates into the Unitree G1 reference frame, ensuring correct scaling and orientation across the robot's 7 DoF arms.
- Achieve smooth and stable replication of human upper-body poses on the Unitree G1.

By meeting these objectives, the project advances low-cost, markerless teleoperation by leveraging commodity RGB-D sensing and real-time pose estimation, in line with recent MediaPipe/BlazePose pipelines [10, 11, 16, 17]. The resulting framework provides a foundation for ongoing research into full-body imitation and autonomous human-robot cooperation. Successful implementation will benefit both academic and industrial stakeholders by reducing development costs and enhancing the realism and responsiveness of humanoid telepresence. Furthermore, this work lays the groundwork for future investigations into full-body robot mimicry.

2 Design Methodology

The design methodology adopted for this project follows a structured engineering process comprising of problem analysis, conceptual design, system integration, and iterative testing at outlined in similar works [5, 4, 7]. The objective was to create a real-time, vision-based teleoperation system capable of mapping human upper-body motion to the Unitree G1 humanoid robot with minimal latency and high positional accuracy. Each stage of the design was informed by both practical constraints as well as the technical requirements of perception, communication, and actuation systems in a distributed computing environment.

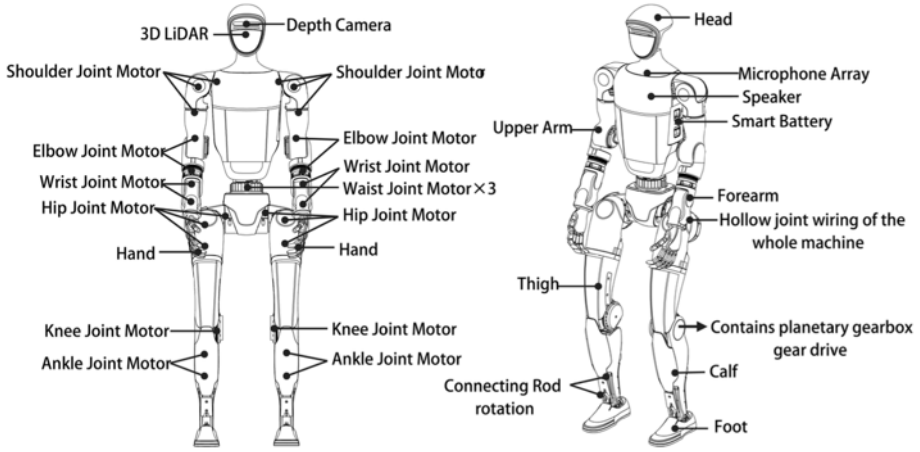


Figure 2.1: Overview of the Unitree G1 [2]

2.1 Design Constraints

Several constraints influenced the design and implementation of the system, these have been categorised as follows:

Hardware and physical constraints: The project utilised a single Luxonis OAK-D Lite depth camera mounted on a fixed stand, with a limited depth sensing range of 0.3-5 m. The camera's field of view (81° diagonal) restricted the maximum workspace and required consistent operator positioning for reliable tracking. The humanoid robot used for this project was the Unitree G1 as this was the only humanoid robot available for research.

Computational and software constraints: The system needed to operate in real time, with an overall perception-to-actuation delay below 150 ms. This imposed constraints on neural network model complexity, data transmission rates, and processing hardware capabilities.

Budgetary constraints: The design intentionally relied on low-cost, open-source hardware and software for control, excluding proprietary motion-capture systems to ensure replicability within typical academic and research budgets.

Mechanical and control constraints: The Unitree G1 robot's upper-body degrees of

freedom, torque limits, and joint velocity bounds defined the allowable motion mapping range and movement speed of the limbs to prevent instability, self-collision and maintain user safety.

2.2 Design Criteria and Evaluation Metrics

To guide design decisions and assess performance, the following criteria were adopted:

Responsiveness: End-to-end system latency maintained below 150 ms to support natural and responsive teleoperation.

Stability: Robot joint trajectories must remain smooth, with minimal motion jitter during steady-state operation.

Scalability and modularity: Each subsystem (3D pose estimation, kinematics calculation, and control) must operate as an independent ROS2 node, allowing reuse and future expansion.

Safety and reliability: The design must prevent unexpected robot movements through motion-limiting thresholds and movement filtering. The robot must also have an emergency stop mechanism in the case of uncontrolled movement or behaviour.

2.3 Ideation and Concept Development

The ideation phase explored multiple alternative approaches to upper-body motion imitation, each with distinct advantages and practical challenges. The evaluation of these methods was guided by criteria such as setup complexity, accuracy, repeatability, and suitability for integration with the Unitree G1 humanoid control system.

1. **Inertial-based control**: Using wearable IMUs, such as the Sony Mocopi system [18], was initially considered for its ability to provide continuous 3D orientation data independent of lighting or occlusion. However, this approach required an extensive calibration sequence prior to each session, which introduced significant setup time and potential for user-dependent error. Furthermore, drift over long durations and the need for wireless synchronisation limited its reliability for repeated experimental runs.

2. **Optical marker-based tracking**: Using multiple external cameras was evaluated as a high-precision alternative capable of producing accurate 3D motion data through triangulation. Despite its precision, this method was deemed impractical within a shared laboratory environment. Maintaining consistent camera placement, lighting conditions, and calibration across multiple uses would be time-consuming and prone to variability, particularly when the workspace was frequently reconfigured or accessed by other users.

3. **Vision-only AI pose estimation**: Using an RGB-D camera was identified as a promising alternative that avoided the need for specialised wearable equipment or extensive calibration. This approach could also leverage advances in AI-based

human pose estimation, specifically the *MediaPipe* and *BlazePose* frameworks [11], to perform real-time skeletal tracking directly from visual input.

Figure 2.2: Luxonis OAK-D Lite RGB-D camera.

After comparative analysis and preliminary testing, the vision-based method was selected as the most practical and scalable solution for this project. The *OAK-D Lite* depth camera was chosen for its ease of access, simplicity, and portability, with the ability to generate synchronised RGB and depth data with onboard processing through the DepthAI framework [14]. Unlike the IMU or multi-camera systems, the OAK-D required minimal calibration and could be deployed quickly in different environments, including both the physical laboratory and from remote locations when work was being done in simulation, away from the physical Unitree G1 robot. This workflow was able to support consistent testing and rapid iteration.
Ultimately, this approach provided an optimal balance between performance, ease of use, and reproducibility, aligning with the project's emphasis on creating an accessible, low-cost, and open-source teleoperation pipeline.

2.4 Design Tools and Software Frameworks

The implementation of this project relied on a multiple open-source software frameworks and simulation tools that were able to support both rapid development and reliable system integration. All development and testing was carried out in an Ubuntu Linux 22.04 [19] environment, chosen as it is required for use with ROS2 Humble. The software was structured around modular design principles, enabling distributed testing, debugging, and ease of transition between simulation and physical deployment.

2.4.1 Software Frameworks

The perception and communication pipelines were implemented entirely in Python [20], ensuring portability and ease of integration with ROS2-based nodes. The software stack included:

- **MediaPipe and DepthAI:** Used for human upper-body pose estimation and depth reconstruction. The *MediaPipe* framework provided a solution for detecting 2D

skeletal keypoints in real time, while the *DepthAI* API enabled direct access to the OAK-D Lite camera for depth mapping.

- **ROS2 Humble:** Adopted as the middleware layer to handle inter-node communication, using the Data Distribution Service (DDS) protocol for low-latency data transfer. This architecture allowed the perception node, transformation node, and control node to operate independently yet synchronously.
- **Unitree SDK2 Python interface:** The project utilised the Unitree Python SDK2 [21] to directly interface with the Unitree G1 robot within ROS2. This Software Development Kit (SDK) allows the nodes via a python interface, to send movement commands, query robot joint states, and subscribe to sensor feedback without needing to reimplement robot-specific communication protocols. The SDK abstracts away much of the complexity of motor control and data serialisation.

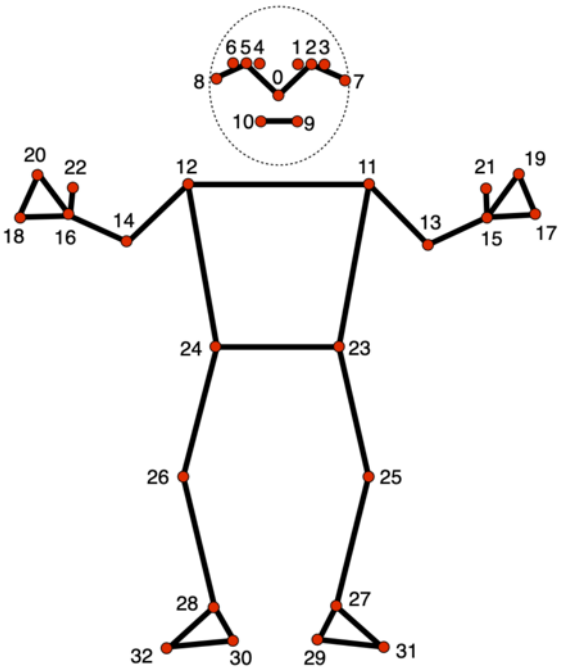


Figure 2.3: BlazePose keypoint layout and index list.

2.4.2 System Integration and Communication

The system architecture was divided into two primary subsystems: the perception node and the actuation node. The OAK-D Lite depth camera was directly connected to the host computer via a high speed USB 3.1 connection. Using this camera data the perception node performed real-time pose estimation and transformed the captured human pose, into the robots coordinate frame. This data was then published over the ROS2 network for the actuation node to consume. The actuation node was responsible for calculating the resulting movement commands and trajectory planning needed to move the robot to the

given coordinates. These motion commands were transmitted using DDS, piggybacking on ROS2 topics via the SDK, to the G1's onboard controller through an Ethernet connection. This configuration ensured high reliability and minimal packet loss, maintaining consistent communication performance throughout testing.

2.4.3 Simulation Environment

Simulation played a critical role in the design and validation stages of the project, allowing rapid iteration and troubleshooting prior to real-world testing. The Unitree MuJoCo environment [3], a customised variant of the MuJoCo physics simulator [22] was used to model the Unitree G1 robot in real time. This simulator provided an accurate representation of the robot's kinematic and dynamic behaviour, allowing motion commands generated by the pipeline to be tested, tuned, and validated before any hardware deployment took place. By leveraging simulation, testing cycles were significantly accelerated, reducing hardware risk and enabling rapid optimisation.

2.4.4 Impact of Simulation and Development Tools

The heavy use of simulation throughout the development process substantially accelerated the project timeline. Algorithms for keypoint mapping, coordinate transformation, and motion smoothing were first validated in simulation before being deployed to the physical robot, reducing the number of physical test cycles required. The ability to easily switch between simulated and real-world operation also enabled reproducible testing under controlled conditions, readily allowing for evaluation of system latency, stability, and accuracy.

Simulation has now become a standard component of modern robotics workflows, providing a safe, scalable, and cost-effective environment for development. This simulation approach encompasses the entire robotics development life cycle, allowing for planning, validation, and testing, in an accelerated timeline, before any hardware deployment takes place [23, 24, 25, 26]. The ever increasing realism of physics-based engines such as MuJoCo and Issac Sim [27] further bridges the simulation-to-real life gap, allowing developers to refine algorithms and benchmark system performance using easily reproducible scenarios [23].

2.5 Standards and Compliance

The design was developed in accordance with the general principles of ISO 13482:2014 [28], which governs the safety requirements for personal care and collaborative robots. Specific clauses relevant to physical contact, motion control, and emergency stop behaviour informed software safety measures. Additionally, ROS2 Quality of Service (QoS) standards were followed to ensure deterministic communication under the DDS protocol.

2.6 Testing and Validation

The testing and validation process was conducted to evaluate the real-time performance, stability, and qualitative motion accuracy of the retargeting pipeline. The assessment focused on metrics that could meaningfully reflect the system's responsiveness and motion

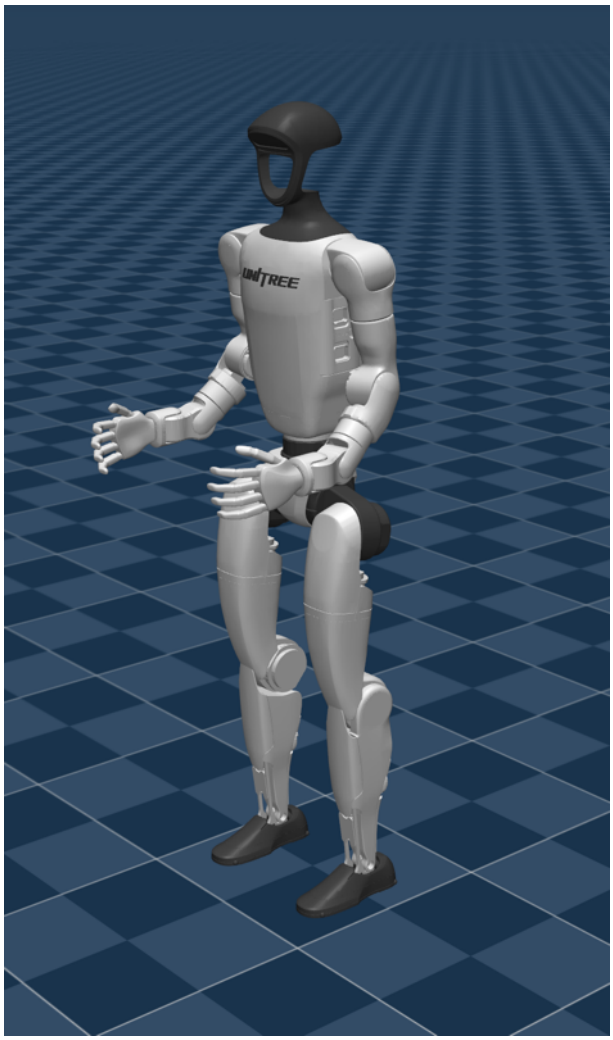


Figure 2.4: Unitree G1 in MuJoCo[3] simulation environment.

faithfulness, rather than direct spatial accuracy, which is impractical due to the non-isomorphic kinematic structure and scale difference between the human operator and the Unitree G1 robot.

2.6.1 Latency Evaluation

End-to-end system latency was measured as the total delay between human motion occurrence and corresponding robot actuation. This was quantified by timestamping key ROS2 and Camera messages at each stage of the pipeline, from pose estimation to robot execution. The aggregate delay was then computed to evaluate overall system responsiveness. This method assumes that any motion is instantly captured by the camera, as the real life time for the camera to capture a frame can not be easily determined.

2.6.2 Motion Filtering and Stability Analysis

To ensure smooth and natural robot motion, filtering techniques were implemented to mitigate high-frequency noise and estimation uncertainty present in the pose data generated by the vision-based perception. A combination of Kalman filtering [29, 30] and low-pass smoothing [31] was applied throughout the pipeline to help increase stability in the systems

output motion. The Kalman filter provided predictive correction and reduced movement noise from the 3D pose estimation stages, while the low-pass filter suppressed noise in the depth estimates from the RGB-D camera.

Stability was evaluated by analysing the robot's dynamic response rather than steady-state behaviour. As the system operates a continuous control loop, variations in the operator's movement naturally propagate to the robot, preventing the establishment of a meaningful baseline for positional stability. Consequently, the assessment focused on ensuring that all commanded movement trajectories remained smooth, and free from osculations or sudden fluctuations.

2.6.3 Qualitative Pose Assessment

Due to the inherent differences in morphology between the human operator and the humanoid robot, direct numerical comparison of joint angles or Cartesian positions was not feasible. Instead, pose accuracy was assessed qualitatively through visual analysis [32]. This involved recording synchronised video footage of the operator and the robot during both static and dynamic pose sequences. The resulting data was reviewed to assess the degree of visual similarity, motion fluidity, and pose alignment between human and robot. Observers evaluated how closely the robot's end-effectors and limb orientations reflected the operator's intended motion, providing a meaningful measure of the system's imitation fidelity [33].

Figure 2.5: MediaPipe pose landmarking running on RGB-D camera feed.

2.7 Health and Safety Considerations

Health and safety was prioritised throughout the design, testing, and operation of the system. All experiments were conducted within a controlled laboratory environment with defined safety boundaries to ensure safe operation of the Unitree G1 robot. Although the platform is battery powered and therefore presents minimal electrical risk, appropriate handling procedures were followed during charging, transport, and testing to mitigate potential hazards.

A software-based dead man switch mechanism was implemented to prevent unintended robot motion. The robot's control software required continuous activation of a specific button combination on the Unitree G1's controller for any motion commands to be executed. If the operator released the input, the system immediately halted all active motion commands and placed the robot into a safe idle state. This ensured that movements could only occur under deliberate operator supervision, eliminating the risk of unintended activation or false positives from the perception system.

Joint velocity limits and motion thresholds were also applied to restrict the robot's maximum speed and range of motion, reducing the risk of damage to the robot and surroundings. As no human-robot physical contact occurred during testing, the overall risk of injury was considered minimal. Standard laboratory risk assessments were followed in accordance with institutional safety policies, ensuring compliance with best practices for teleoperated robotic experimentation.

3 Results & Discussion

3.1 System Pipeline Overview

The pipeline created for this project was intended to perform mimicry based teleoperation of Unitree G1 via binocular human pose estimation and retargeting over ROS 2. The pipeline was made up of 6 key steps.

The key steps in the pipeline are as follows:

1. **Camera Feed**: Live camera data is captured from the OAK-D lite camera.
2. **2D Pose Estimation**: Each frame of the camera feed is given to the MediaPipe engine to estimate human keypoint markers found in the image.
3. **3D Pose Estimation**: 2D keypoints are fused with depth data from Depth-AI to create a 3D pose estimate. These 3D keypoints are transformed into the robots reference frame originating at the pelvis.
4. **Inverse Kinematics**: The given target locations in the robots frame are then solved, outputting the required joint angles need to reach the pose.
5. **Trajectory Generation**: The torques required to move the robot from current joint angles to the requested joint angles are calculated.
6. **Joint Actuation**: Calculated joint angles and torques are given to the robots controller to actuate the motors.

These key steps lay the ground work for the pipeline. These steps can be grouped together into logical nodes which can then be implemented in code. For this project 2 key nodes were developed and then integrated with the existing software running on the Unitree G1 robot.

3.2 pose_tracker Node

This node creates a DepthAI pipeline (OAK-D) to acquire RGB and stereo depth information, runs a MediaPipe pose estimation model (most accurate/slowest model) on the RGB stream, lifts selected 2D landmarks to 3D using the DepthAI point cloud data (low pass filtering applied), temporally smooths them with per-joint 3D Kalman filters, and publishes pelvis-relative wrist transforms over ROS 2 as a TransformStamped ROS topic.

The node takes the following inputs from the OAK-D lite camera:

1. Colour video stream at 1080p, running at a 30 FPS target.
2. Stereo mono cameras at 400p each, running at a 30 FPS target.
3. Point cloud depth data (calculated from stereo camera offset).The point cloud is aligned to the RGB camera feed.

The core steps of the algorithm in this node are as follows:

1. **2D Pose Estimation.** Each RGB frame is processed using the MediaPipe Pose model configured with the highest complexity level. This setting provides the most accurate landmark detection available in MediaPipe, at the cost of reducing the frame processing rate.

2. **2D to 3D lifting.** For each frame, upper-body keypoints are extracted and mapped to 3D coordinates using the DepthAI point cloud. Each 3D position is computed as the mean of a 3×3 neighborhood around the pixel (x, y) to suppress depth noise. If depth data is invalid or missing, the filter reuses the last valid estimate.

3. **Temporal smoothing.** Each tracked joint is filtered using an independent constant-velocity Kalman filter. The filter state includes both position and velocity in (x, y, z). Each update cycle follows the standard predict-update sequence [34], applied only when a valid measurement is available.

4. **Frame conversions & pelvis-relative vectors.** The 3D keypoints obtained from the DepthAI camera are expressed in the camera optical frame, where $(+Z_{\text{cam}})$ points forward, $(+X_{\text{cam}})$ to the right, and $(+Y_{\text{cam}})$ downward. To represent motion in body-centric coordinates, all keypoints are transformed into a pelvis-centered frame with $(+Z_{\text{pelvis}})$ upward, $(+X_{\text{pelvis}})$ forward, and $(+Y_{\text{pelvis}})$ to the left:

$$\mathbf{p}_{\text{pelvis}} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{p}_{\text{cam}}.$$

   The pelvis origin is defined as the midpoint between the left and right hip joints:

$$\mathbf{p}_{\text{pelvis\_origin}} = \frac{1}{2}\left(\mathbf{p}_{23} + \mathbf{p}_{24}\right) + \begin{bmatrix} 0 \\ 0 \\ Z_{\text{offset}} \end{bmatrix},$$

   where indices 23 and 24 denote the BlazePose hip landmarks and $Z_{\text{offset}} \approx 0.1m$. This offset is because a depth camera observes only the front surface of the hips, the true position of the pelvis center is further offset.

   Since the camera *faces* the user, moving a wrist *toward* the camera reduces Z_{cam}. When converting to the pelvis frame the depth difference between a given point $\mathbf{P}_{\text{cam}}$ and the pelvis origin $\mathbf{P}_{\text{cam\_pelvis\_origin}}$ in the camera frame must be used as the depth distance for the point in the pelvis frame.

$$X_{\text{pelvis}} \;=\; -\,\Delta Z_{\text{cam}} \;+\; Z_{\text{offset}} \;=\; -\,(Z_{\text{cam\_point}} - Z_{\text{cam\_pelvis}}) \;+\; Z_{\text{offset}}.$$

   For each arm, an estimation for orientation of the wrist is reconstructed from the shoulder-elbow-wrist geometry. Let $\mathbf{u} = \mathbf{p}_{\text{elbow}} - \mathbf{p}_{\text{shoulder}}$ denote the upper-arm vector and $\mathbf{f} = \mathbf{p}_{\text{wrist}} - \mathbf{p}_{\text{elbow}}$ the forearm vector, both expressed in the pelvis frame. The primary wrist axis is chosen as the normalised forearm direction:

$$\hat{\mathbf{x}} = \frac{\mathbf{f}}{\|\mathbf{f}\|}.$$

However, $\hat{\mathbf{x}}$ alone is insufficient to define a unique 3D orientation because rotations about the forearm axis remain unconstrained. To make the rotation well-defined, a local orthonormal basis is constructed using the cross product with the upper-arm vector:

$$\hat{\mathbf{y}} = \frac{\hat{\mathbf{x}} \times \mathbf{u}}{\|\hat{\mathbf{x}} \times \mathbf{u}\|}, \qquad thus \qquad \hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}},$$

With this basis defined the corresponding rotation matrix can be constructed:

$$\mathbf{R} = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \end{bmatrix}.$$

The quaternion (q_x, q_y, q_z, q_w) is then extracted from $\mathbf{R}$ and published as the wrist's orientation relative to the pelvis.

This orthonormal basis ensures that the wrist frame remains consistent with the Unitree G1 defined arm coordinate system. the $\hat{\mathbf{x}}$-axis aligns with the forearm, while the $\hat{\mathbf{y}}$-axis always parallel with the elbow joint's axis of rotation. Without this coordinate construction the wrist's roll angle is undefined. Although this model does not fully capture the degrees of freedom of a real human wrist, it provides a close approximation that simplifies the control mapping.

5. **Validity checks.** As the pelvis joints are key in solving the coordinate transforms, frames are skipped if the hips are not detected. Any anomalous transforms that are detected such as NaN's are not published.

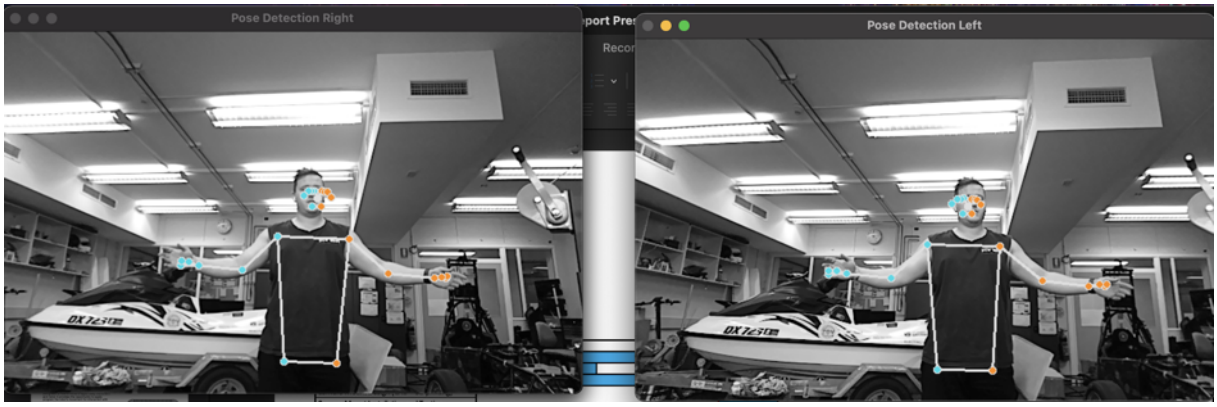


Figure 3.1: MediaPipe pose landmarking running on stereo camera feed.

The outputs from the node are as follows:

1. /pose_tracker/left_wrist (geometry_msgs/TransformStamped) - Published transform from the pelvis frame (parent) to the left_wrist frame (child). The translation component (in meters) encodes the pelvis-relative 3D wrist position, while the rotation component is a quaternion derived from the shoulder-elbow-wrist geometry.

2. /pose_tracker/right_wrist(geometry_msgs/TransformStamped) - Published transform from the pelvis frame (parent) to the right_wrist frame (child). The translation component (in meters) encodes the pelvis-relative 3D wrist position, while the rotation component is a quaternion derived from the shoulder-elbow-wrist geometry.

Implementation notes: Using MediaPipe's highest-complexity model and the described DepthAI stereo pipeline, the node achieves an average update rate of **14.7 FPS** on a modern laptop or PC, with the bottleneck primarily in human pose estimation.

3.3 pose_follower Node

This node subscribes to pelvis-relative wrist transform targets produced by pose_tracker node and drives a dual-arm controller on the Unitree G1. Incoming targets are passed to an inverse-kinematics solver; the resulting joint positions and feedforward torques are commanded to the robot via an arm controller. Only the wrists are tracked for the final arm movement calculations, no other body parts are used. It is important to note that the inverse kinematics solver and arm controller are supplied open source software packages from Unitree [15] and were not created for this project. This open source software was modified to work with system configuration for this project.

The node takes in 2 input parameters at startup to control behaviour. These are mainly used for running the code in simulation mode vs live deployment, however they can also be used to bypass safety features for testing.

- sim (bool, default false) - Flag to signal if the code is being run in simulation mode. This internally sets environment variables and changes command publishing domains.
- safeMode (bool, default true) - Forces the use of a deadman switch from the wireless controller to be active to send move commands. If the deadman switch is released all move commands are blocked from publishing and the robot will hold its current pose. This is used for safety as no hardware E-STOP is present on the Unitree G1. This deadman "switch" consists of the shoulder buttons (R2 + L2) being pressed simultaneously on either side of the controller. Buttons from both sides are used to avoid any false positives from a single button being pressed, for example if the controller is put down and is resting against something.

The node takes the following ROS2 topics as inputs:

1. /pose_tracker/left_wrist (geometry_msgs/TransformStamped) - The estimated left wrist position and orientation in the pelvis frame calculated in the **pose_tracker** node.
2. /pose_tracker/right_wrist (geometry_msgs/TransformStamped) - The estimated right wrist position and orientation in the pelvis frame calculated in the **pose_tracker** node.
3. (unitree_go/WirelessController) - Current controller button states being published by the Unitree G1 Controller. This is used to determine the deadman switch state.

The operation of this node proceeds through the following main stages:

1. **Initialisation.** The node instantiates the dual-arm controller and applies damping to all lower body joints to prevent them from moving. This node will hang and wait for a DDS connection to be established, meaning the robot (or simulation) must be active and ready to run. The inverse-kinematics solver is also instantiated. After

ROS 2 initialisation, the robot is commanded to a neutral "home" posture (see Fig. 2.4, where all upper body joints are set to 0°.

2. **IK solution and Trajectory.** Incoming left/right wrist transforms are cached as the current targets. Before solving, the wrist targets are uniformly scaled to match the robot's arm reach. Operational frames for each wrist are defined at the wrist yaw joints with a small forward offset to represent the palm frame.

   Inverse kinematics is then formulated as a constrained least-squares problem in CasADi/Pinocchio [35, 36]. The solver searches for joint angles that best align each wrist's position and orientation with its target, while respecting joint limits and gently large joint values. To keep motion stable from frame to frame, the solution takes into account the previous joint state and a moving-average filter smooths the result.

   Finally, feedforward torques are computed from inverse dynamics at the solved posture, and both the joint targets and torques are sent to the low-level dual-arm controller.

3. **Safety guarding.** Arm motion is enabled only when the deadman switch is held or when safe mode is disabled from the startup parameters.

4. **Control loop (100 Hz).** A 10 ms timer executes the main control cycle. Each iteration retrieves the current cached joint target positions and current velocities and solves the kinematics. The calculated joint configuration and feedforward torques are then sent to the robot via the low level controller.

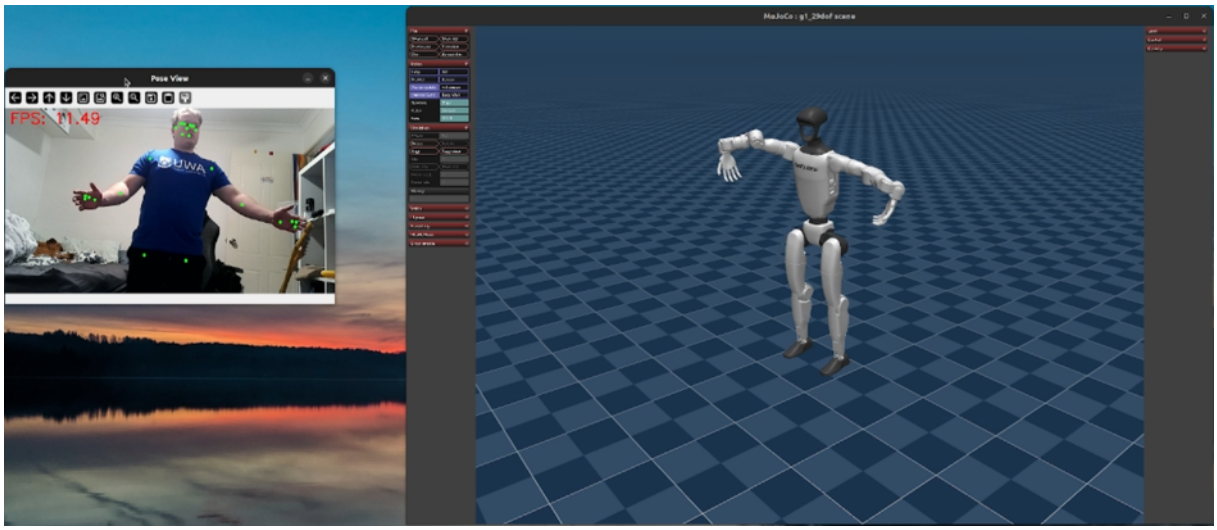


Figure 3.2: Early pose retargeting trial before arm basis and 3D scaling implementation.

The outputs from this node are as follows:

- /lowCmd - A ROS topic carrying low-level commands issued via the controller. This can be directly changed to /arm_sdk to allow for teleoperation whilst other code controls other parts of the robot.

Implementation notes: The homing sequence during initilisation will: wait 5 s for subscriptions, start moving to home position, wait 2 s, then enter the main control loop.

3.4 Pipeline Latency and Observed Response Time

To estimate the end-to-end latency from visual sensing to commanded motion on the Unitree G1 robot, the latency of each major stage of the pipeline was profiled or inferred from known loop frequencies. Table 3.1 summarises the measured and estimated timings for each component.

Table 3.1: Latency characteristics (average) of the pose tracking and control pipeline.

*calculated from frequency.

Pipeline Stage	**Frequency (Hz)**	**Period (ms)**	**Samples**
Camera capture	—	187.5	474
Frame processing	14.7	68.0*	458
Pose publishing	—	5.7	148
IK Solving	—	7.4	253
Move command publishing	110	9.1*	3100
ROS 2 DDS (worst-case, estimated)	—	20.0	—
Unitree G1 motor control loop	500	2.0*	—
Estimated system latency (sum)	—	**299.7**	—

Latency breakdown

The perception-to-actuation delay can be understood sequentially from the camera input to the low-level controller output. The OAK-D Lite camera introduces an average 187.5 ms delay from frame capture to host arrival (measured using the recommended Luxonis time-stamping method [37]).

The visual front-end operates at an average of 14.7 FPS, yielding an image processing latency of approximately 68 ms. This time includes the time to find 3D keypoints and complete data transformations. Once keypoints are extracted and lifted to 3D, the resulting wrist transforms are published to the ROS2 network, reaching the pose_follower node with an average delay of 5.65 ms (measured via end-to-end timestamp comparison).

The pose_follower node executes a control loop running at approximately 110 Hz (measured using ROS2 utilties), producing new joint-space commands every 9.1 ms. These commands are transmitted over ROS2 DDS middleware, for which a very conservative 20 ms network delay is assumed based on reported latency bounds for similar systems [38, 39].

Finally, the Unitree G1's firmware-level motor control loop operates at 500 Hz [40], after which the motion command is executed by the actuators. Summing these nominal contributions yields an average end-to-end latency of approximately **299.7** ms. Under ideal conditions, a human wrist movement should therefore propagate to visible arm motion on the robot within roughly 300 ms of image capture. If the pipeline latency is assessed in isolation without live capture from the depth camera, then the total pipeline latency reduces to **112.2** ms.

Observed delay and discussion

In practice, visual inspection of the robot's response indicates noticeably longer apparent delays, sometimes up too approximately 1 second. The present analysis cannot definitively attribute this discrepancy to a single factor, but several causes are hypothesised.

Firstly, the motion controller intentionally suppresses instantaneous reversals in arm direction to avoid abrupt or unstable transitions, effectively introducing a short hold before reversing movement.

Secondly, temporal filtering is applied at multiple stages: per-joint Kalman filters in the vision pipeline smooth keypoint estimates, and a weighted moving-average filter in the inverse-kinematics solver damps rapid fluctuations in joint movement. While these filters reduce noise and yield smoother trajectories, they inherently introduce phase lag, causing 'information' to propagate with a delay through the system. Further investigation is required to quantify the contribution of each source of delay and to distinguish between smoothing latency, ROS 2 communication overhead, and actuation response.

3.5 ROS 2 Interfaces & Transport

Table 3.2: ROS2 topic overview used by the system.

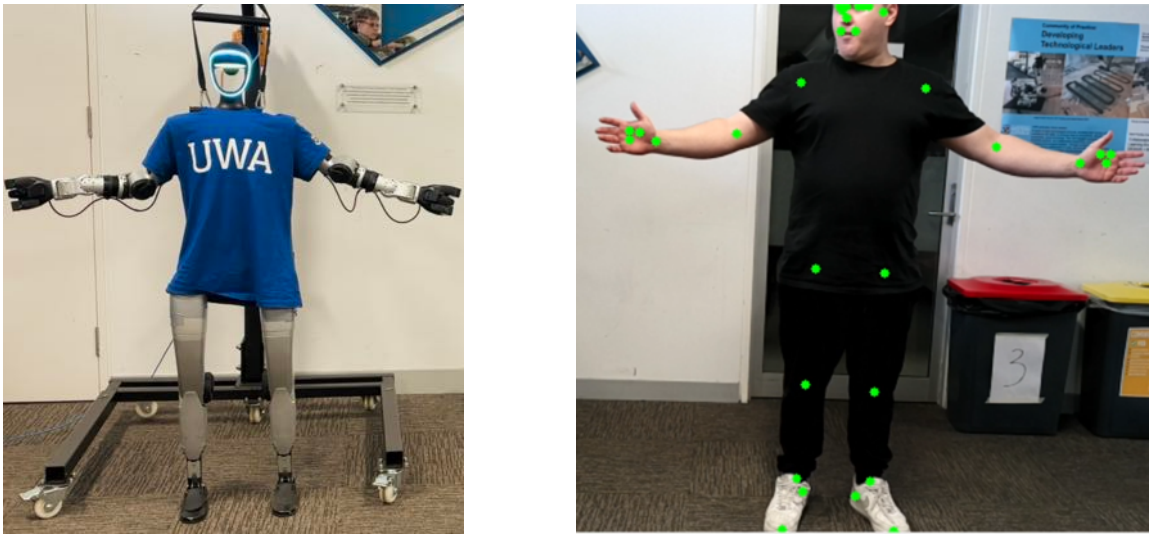
Topic	Type
/pose_tracker/left_wrist	geometry_msgs/TransformStamped
/pose_tracker/right_wrist	geometry_msgs/TransformStamped
/wirelesscontroller	unitree_go/WirelessController
/g1/lowstate	unitree_hg/msg/dds_/LowState_
/g1/lowcmd	unitree_hg/msg/dds_/LowCmd_

Quality of Service (QoS): QoS is used in DDS to ensure that certain behaviours from the data and communicators on the network are maintained. Unless explicitly configured, the Unitree SDK, which communicates over ROS 2 using the Cyclone DDS middleware, inherits the default ROS 2 QoS policies [41]. Specifically, the default settings are:

- **Reliability:** RELIABLE - guarantees that all published samples are delivered to active subscribers, with automatic retransmission if data loss occurs.
- **Durability:** VOLATILE - only delivers data to currently active subscribers, no persistence of messages are available.
- **History:** KEEP_LAST with a default depth of 10 - maintains only the most recent ten samples per topic.
- **Liveliness:** AUTOMATIC - the node is considered alive if any of its publishers sends a message.
- **Deadline and Lifespan:** INFINITE - no timing constraints on expected message arrival or validity duration.

3.6 Qualitative Pose Retargeting Assessment

To qualitatively assess the results of the pose retargeting, three representative human poses were selected and compared against their reproduced pose on the Unitree G1 robot. The reference human poses were extracted through the proposed retargeting pipeline and captured under identical camera conditions.



(a) Robot pose. (b) Human pose.

Figure 3.3: Human vs Robot "arms-outstretched" pose.

(a) Robot pose. (b) Human pose.

Figure 3.4: Human vs Robot "single-arm raise" pose.

3.6.1 Visual comparison.

Across all three cases, the reproduced robot poses show strong correspondence to the source human poses in terms of arm orientation and wrist positioning. For the *arms-outstretched* pose (Fig. 3.3), the robot successfully achieved near-symmetric shoulder and

(a) Robot pose. (b) Human pose.

Figure 3.5: Human vs Robot "forward-reach" pose.

elbow extension, matching the lateral spread and orientation of the human wrists. In the *forward-reach* pose (Fig. 3.5), the robot accurately aligns both hands in front of the torso, preserving approximate shoulder width and wrist height. Finally, in the *single-arm raise* pose (Fig. 3.4), the robot captures the lift of the right arm while maintaining a stable left-arm posture, demonstrating effective decoupling between arms.

3.6.2 Observed discrepancies.

Minor deviations are visible in elbow articulation and wrist roll, which appear slightly under-extended compared to the human reference. In other cases odd positions of wrist joints are present (robots right arm in Fig. 3.4). These offsets likely stem from structural differences between the human arm and the robot arm's 7-DoF kinematics, as well as the simplified quaternion-based wrist orientation model used in the retargeting stage. Error in the MediaPipe pose estimation could also contribute to pose inaccuracy, while noise and depth error from the depth camera certainly also play a role in the mismatches.

In Fig. 3.6, an example of basis degeneration referred to here as basis collapse, in the robot's arm orientation can be observed due to occlusion of the elbow by the wrist. In these instances, when the wrist overlaps the elbow, the depth camera percieves incorrect depth data for the elbow joint. Since the basis for arm orientation is derived from the vectors between the upper arm and the forearm, wrist-elbow occlusion causes the upper-arm vector to become over-extended, pointing directly toward the wrist, while the forearm vector becomes very short and may point in any arbitrary direction due to noise and the elbow and wrist being measured at practically the same spot. Although the x, y, and z coordinates of the pose remain mostly unchanged, the corresponding quaternion orientation becomes inaccurate. This leads to the visible twisting in the robot's right arm shown in the example.

Once the arm becomes twisted, it tends to maintain this unnatural pose even after the occlusion has ended. This occurs because the kinematic solver attempts to find the most direct path to the next target configuration, often reinforcing the incorrect orientation.

Figure 3.6: Example of arm basis collapse due to wrist-elbow occlusion.

3.6.3 Interpretation.

Overall, the qualitative assessment indicates that the system reproduces upper-body motions with high accuracy and lateral/sagittal alignment. Some error exists from depth camera noise, pose estimation misalignment, and in cases of occlusion lack of fallback modeling to counter the basis collapse.

3.7 Discussion

The system achieved its core objective of demonstrating stable, real-time upper-body pose imitation between a human operator and the Unitree G1 humanoid robot. Under nominal conditions, the robot was able to accurately mirror arm gestures and semi-static postures with consistent latency and minimal oscillation. Visual inspection confirmed that wrist and elbow positions maintained plausible alignment with the human operator's movement throughout extended operation, validating the reliability of the perception and control pipeline.

Performance degraded under conditions of rapid motion or partial self-occlusion. Depth estimation errors from the OAK-D Lite and pose ambiguity in MediaPipe occasionally led to erroneous joint detections, especially during occlusion events when hands were perfectly overlapped with elbow keypoints, leading to a collapse of the arm basis. These issues propagated through the pipeline, resulting in temporary loss of wrist targets or unnatural arm orientations. Such events highlight the sensitivity of monocular systems to occlusion and depth error.

Compared with recent hybrid and monocular approaches for motion imitation [32, 33], the proposed system removes the dependency on wearable IMUs or multi-camera calibration, instead achieving full-body tracking through a single depth sensor in real-time. This enables

a fully markerless, low-cost solution suitable for upper-body telepresence and demonstration tasks. While commercial motion-capture systems still offer superior spatial precision and temporal consistency, the achieved imitation fidelity was sufficient for responsive gesture reproduction, validating the practicality of vision-based tracking for humanoid control.

In relation to more recent learning-based approaches [16, 17], this work emphasises deterministic control and transparency over data-driven adaptation. While deep models can achieve smoother temporal consistency, the ROS 2-based pipeline ensures predictable behaviour suitable for safety-critical research environments.

From a middleware perspective, the use of ROS 2 with Cyclone DDS and the Unitree SDK2 demonstrated stable, low latency communication, consistent with performance reported in prior benchmarks [38, 39]. This confirms that the proposed architecture is robust enough for real-time humanoid control in a distributed setup.

The following limitations in the current pipeline implementation have been recognised:

1. **Monocular ambiguity and occlusion.** As the OAK-D Lite camera observes only a single viewpoint, depth estimation degrades under occlusion or rapid movement, occasionally leading to unstable or inaccurate pose reconstruction.

2. **Kinematic mismatch.** The human operator and Unitree G1 differ in joint range and link proportions. The current implementation does not calibrate kinematic scaling based on user proportions, sometimes producing under-extended poses.

3. **Latency sensitivity.** The combined perception and control latency ($\sim$300 ms average) limits responsiveness during fast gestures. Temporal smoothing via Kalman and moving-average filters further increases phase lag.

4 Conclusions & Future Work

Conclusions

This project successfully achieved its primary objective: to design and implement a complete, real-time human-to-humanoid pose retargeting system using a depth camera and the Unitree G1 robot. The developed framework demonstrated that markerless vision-based imitation can reliably replicate upper-body human motions in real-time without the need for wearable sensors or motion-capture infrastructure. The system integrated the OAK-D Lite camera, MediaPipe pose estimation, and a ROS 2, based control pipeline with inverse kinematics and temporal filtering, achieving a stable end-to-end teleoperation system.

Key findings include:

- The perception and control pipeline achieved stable, real-time operation with an average end-to-end latency of approximately 300 ms, sufficient for interactive upper-body mimicry. This latency can be further reduced to approximately 115 ms if the camera capture delay is disregarded from the pipeline latency.
- The vision-based tracking and retargeting produced accurate poses, with the Unitree G1 reproducing a range of human arm configurations in the sagittal and lateral planes.
- Temporal smoothing through Kalman and moving-average filtering effectively reduced noise, though at the cost of increased phase lag in the system.
- ROS 2 and Cyclone DDS provided reliable communication between distributed nodes, validating the chosen middleware for humanoid teleoperation.

All primary project objectives were successfully achieved, including real-time perception, reliable ROS 2 data exchange, and accurate upper-body pose reproduction in both simulation and hardware trials. The system demonstrated stable teleoperation performance and consistent mapping between human motion and robot articulation.

Measured end-to-end latency was higher than initially targeted, averaging approximately 300 ms-primarily due to the $\sim$180 ms input delay introduced by the OAK-D Lite camera during frame capture and host transfer. When this external camera latency is excluded, the software pipeline itself operates well within the 150 ms design goal, achieving an effective processing delay of approximately 115 ms. Remaining performance limitations are largely attributed to depth estimation noise and kinematic mismatch between human and robot arm geometries from lack of calibration.

Overall, the project establishes a robust and extensible foundation for real-time humanoid imitation and teleoperation research, with clear pathways for improvement in sensing latency and user based calibration.

Future Work

While the implemented system demonstrates the feasibility of low-cost, vision-based human-to-humanoid imitation, several ideas for future development have been identified. The following directions are proposed to address the identified limitations and extend the robustness of the current framework:

1. **Enhanced depth and occlusion handling:** The single-view limitation of the OAK-D Lite can be mitigated through multi-view fusion or temporal depth refinement. Incorporating occlusion confidence metrics, or pairing with low-cost IMUs on key joints, would reduce occlusion based pose dropouts and stabilise depth tracking under rapid motion.

2. **User-specific kinematic calibration:** Implementing automatic human-robot scaling through per user kinematic calibration could improve pose accuracy. By calibrating limb lengths and joint limits per user, the retargeting pipeline can compensate for human-robot proportion mismatches, reducing under-extension and maintaining more natural joint mapping across different operators.

3. **Latency mitigation and predictive filtering:** To reduce the perceptible delay introduced by the 300 ms average end-to-end latency, predictive estimation techniques such as learning-based motion forecasting could be explored. These methods would allow future robot poses to be extrapolated in advance of sensor updates, partially offsetting perception and communication delays without sacrificing stability.

In summary, this work demonstrates a practical, reproducible, and open-source baseline for real-time human-to-humanoid teleoperation using affordable hardware and widely available software frameworks. With further refinement in perception accuracy, motion prediction, and safety assurance, such systems have the potential to enable intuitive, accessible, and responsive humanoid interaction in research, industry, and education.

Bibliography

[1] Unitree Robotics, *Unitree G1*, Unitree Robotics, 2025, accessed: 2025-10-08. [Online]. Available: https://www.unitree.com/g1

[2] ——, *Unitree G1 Humanoid Robot User Manual*, Unitree Robotics, 2024, accessed: 2025-09-11. [Online]. Available: https://unitree.com/g1

[3] Unitree Robotics (GitHub), "unitree_mujoco," https://github.com/unitreerobotics/unitree_mujoco, 2025, accessed: 2025-10-08.

[4] I. Almetwally and M. Mallem, "Real-time tele-operation and tele-walking of humanoid robot nao using kinect depth camera," in *2013 10th IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL (ICNSC)*, 2013, pp. 463–466.

[5] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, "Enabling real-time full-body imitation: A natural way of transferring human movement to humanoids," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*. IEEE, 2003, pp. 2368–2374.

[6] F. Semeraro, A. Griffiths, and A. Cangelosi, "Human-robot collaboration and machine learning: a systematic review of recent research," *arXiv preprint arXiv:2110.07448*, 2022.

[7] M. M. Rahman, F. Khatun, I. Jahan, R. Devnath, and M. A.-A. Bhuiyan, "Cobotics: The evolving roles and prospects of next-generation collaborative robots in industry 5.0," *Journal of Robotics*, vol. 2024, pp. 1–22, 2024.

[8] N. Pollard, J. Hodgins, M. Riley, and C. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, 2002, pp. 1390–1397 vol.2.

[9] A. G. Seidle and J. M. Pearce, "Overcoming limitations of proprietary scientific hardware funding," *Journal of the Knowledge Economy*, 2025. [Online]. Available: https://doi.org/10.1007/s13132-025-02783-w

[10] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. [Online]. Available: https://github.com/google/mediapipe

[11] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *arXiv preprint arXiv:2006.10204*, 2020.

[12] Open Robotics, "Ros 2 humble hawksbill," Open Source Robotics Foundation, 2022, accessed: 2025-09-11. [Online]. Available: https://docs.ros.org/en/humble

[13] Luxonis, *OAK-D Lite Datasheet*, Luxonis, Inc., December 2021, technical Datasheet, Revision December 2021. [Online]. Available: https://www.luxonis.com

[14] ——, "Depthai: Spatial ai and depth perception for embedded systems," 2020, accessed: 2025-09-11. [Online]. Available: https://docs.luxonis.com

[15] Unitree Robotics, "xr_teleoperate: Teleoperation of unitree humanoid robots using xr devices," https://github.com/unitreerobotics/xr_teleoperate, 2025, accessed: 2025-09-11.

[16] J.-W. Kim, J.-Y. Choi, E.-J. Ha, and J.-H. Choi, "Human pose estimation using mediapipe pose and optimization method based on a humanoid model," *Applied Sciences*, vol. 13, no. 4, p. 2700, 2023.

[17] S. Negi, M. Garg, H. Maindola, V. Kansal, U. Jain, and S. Bhatla, "Real-time human pose estimation: A mediapipe and python approach for 3d detection and classification," in *Proceedings of the 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*. IEEE, 2023, pp. 128–134.

[18] Sony Corporation. (2025) mocopi developer site: Technical specifications. Accessed: 2025-10-08. [Online]. Available: https://www.sony.co.jp/en/Products/mocopi-dev/en/documents/Home/TechSpec.html

[19] Canonical Ltd., "Ubuntu 22.04 lts (jammy jellyfish)," https://releases.ubuntu.com/22.04/, 2022, released April 21, 2022. Accessed: 2025-10-08.

[20] Python Software Foundation, *Python 3 Programming Language*, 2024, version 3.10. Accessed: 2025-10-08. [Online]. Available: https://www.python.org/

[21] Unitree Robotics, *Unitree SDK2: Software Development Kit for G1*, Unitree Robotics, 2024, accessed: 2025-09-11. [Online]. Available: https://github.com/unitreerobotics/unitree_sdk2

[22] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs, C. Sferrazza, Y. Tassa, and P. Abbeel, "Mujoco playground," 2025. [Online]. Available: https://arxiv.org/abs/2502.08844

[23] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve, C. Li, F. Meier, D. Negrut, L. Righetti, A. Rodriguez, J. Tan, and J. Trinkle, "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward," *Proceedings of the National Academy of Sciences*, vol. 118, no. 1, p. e1907856118, 2021, accessed: 2025-10-08. [Online]. Available: https://dartslab.jpl.nasa.gov/References/pdf/e1907856118.full.pdf

[24] A. Afzal, D. S. Katz, C. L. Goues, and C. S. Timperley, "Simulation for robotics test automation: Developer perspectives," *Proceedings of the 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 62–71, 2021.

[25] B. Talbot, D. Hall, H. Zhang, S. R. Bista, R. Smith, F. Dayoub, and N. Sünderhauf, "Benchbot: Evaluating robotics research in photorealistic 3d simulation and on real robots," *arXiv preprint arXiv:2008.00635*, 2020. [Online]. Available: https://arxiv.org/abs/2008.00635

[26] C. K. Liu and D. Negrut, "The role of physics-based simulators in robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. Volume 4, 2021, pp. 35–58, 2021. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-control-072220-093055

[27] NVIDIA Corporation, "Nvidia isaac sim: Robotics simulation and synthetic data generation," https://developer.nvidia.com/isaac/sim, 2025, accessed: 2025-10-08.

[28] *ISO 13482:2014 – Robots and Robotic Devices – Safety Requirements for Personal Care Robots*, International Organization for Standardization Std., 2014, accessed: 2025-10-08. [Online]. Available: https://www.iso.org/standard/53820.html

[29] L. Mochurad, "Implementation and analysis of a parallel kalman filter algorithm for lidar localization based on cuda technology," *Frontiers in Robotics and AI*, vol. Volume 11 - 2024, 2024. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1341689

[30] C. G. Cifuentes, J. Issac, M. Wüthrich, S. Schaal, and J. Bohg, "Probabilistic articulated real-time tracking for robot manipulation," 2016. [Online]. Available: https://arxiv.org/abs/1610.04871

[31] N. R. Shaikh, "Smoothing of a noisy image using different low pass filters," *International Journal of Computer Science Trends and Technology (IJCST)*, vol. 5, no. 2, pp. 261–264, 2017, accessed: 2025-10-08. [Online]. Available: https://www.ijcstjournal.org/volume-5/issue-2/IJCST-V5I2P50.pdf

[32] S. Baek, A. Kim, J.-Y. Choi, E. Ha, and J.-W. Kim, "Human motion retargeting to a full-scale humanoid robot using a monocular camera and human pose estimation," *International Journal of Control, Automation and Systems*, vol. 22, pp. 2860–2870, 2024, published September 2, 2024. [Online]. Available: https://doi.org/10.1007/s12555-023-0686-y

[33] Y. Cho, W. Son, J. Bak, Y. Lee, H. Lim, and Y. Cha, "Full-body pose estimation of humanoid robots using head-worn cameras for digital human-augmented robotic telepresence," *Mathematics*, vol. 12, no. 19, 2024. [Online]. Available: https://www.mdpi.com/2227-7390/12/19/3039

[34] C. Naab and Z. Zheng, "Application of the unscented kalman filter in position estimation a case study on a robot for precise positioning," *Robotics and Autonomous Systems*, vol. 147, p. 103904, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889021001895

[35] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019, received 20 May 2017; accepted 3 March 2018; published 11 July 2018; issue date 14 March 2019. [Online]. Available: https://doi.org/10.1007/s12532-018-0139-4

[36] "Pinocchio: a fast and flexible rigid body dynamics library," https://stack-of-tasks.github.io/pinocchio/, accessed: 2025-10-11.

[37] "Low latency — depthai documentation," https://docs.oakchina.cn/projects/api/tutorials/low-latency.html, 2024, accessed: 2025-10-11.

[38] S. Paul, D. Le Phuoc, and M. Hauswirth, "Performance evaluation of ros2-dds middleware implementations facilitating cooperative driving in autonomous vehicle," in *Proceedings of the Edge AI meets Swarm Intelligence Technical Workshop*. Dubrovnik, Croatia: TU Berlin, September 2024, license: CC BY 4.0. [Online]. Available: https://arxiv.org/abs/2412.07485

[39] T. Kronauer, J. Pohlmann, M. Matthe, T. Smejkal, and G. Fettweis, "Latency analysis of ros2 multi-node systems," 2021. [Online]. Available: https://arxiv.org/abs/2101.02074

[40] U. Robotics, "Direct motor control — unitree sdk2 python," https://deepwiki.com/unitreerobotics/unitree_sdk2_python/3.1-motor-control, 2025, accessed: 2025-10-10.

[41] O. Robotics, "About quality of service (qos) settings — ros 2 documentation," https://docs.ros.org/en/rolling/Concepts/Intermediate/About-Quality-of-Service-Settings.html, 2024, accessed: 2025-10-11.

Appendices

Appendix A: Code Repositories

This appendix lists additional resources and repositories related to the project.

- Project source code: github.com/TravisLRyan/HumanoidMimicry
- Utility scripts made for project: github.com/TravisLRyan/unitreeG1Install

Appendix B: Literature Review

1. Introduction

1.1 Background and Motivation

Recent advancements in human-robot interaction (HRI) have opened up new possibilities in teleoperation and collaborative robotics [1]. In particular, humanoid robots like the Unitree G1 are being used in applications that require intuitive and responsive control for tasks such as object manipulation and remote intervention. Mimicking upper-body human movement provides a natural interface for controlling robots, bridging the cognitive gap between user intent and robot action [2]. This project is motivated by the opportunity to develop a real-time system that enables the Unitree G1 robot to mirror human upper-body motions for use in teleoperated environments.

Figure 1. Unitree G1 Humanoid Robot. [3]

1.2 Problem Statement

The core issue addressed by this project is the lack of an effective, low-latency system for upper-body teleoperation using human 3D pose data. While traditional control systems require manual or scripted commands, a system capable of mimicking human movement in 3D space can offer more intuitive and responsive control [2]. However, challenges

such as the non-isomorphic mapping between human and robot joints, sensor noise, and depth perception inaccuracies hinder the reliability and practicality of such systems in real-world scenarios.

1.3 Project Scope

This project focuses on implementing a depth camera and computer vision-based system to perform real-time upper-body motion capture of a human subject and map those motions onto the Unitree G1 robot. The solution leverages MediaPipe for 2D pose estimation, coupled with an OAK-D stereo depth camera to infer 3D positional data. The data will be processed and transmitted via ROS2 to control the Unitree robot's upper-body joints. The scope is limited to motion above the waist and does not extend to full-body locomotion or autonomous decision-making.

1.4 Research Questions and Hypothesis

The core research question of this thesis is:

How accurately and responsively can a humanoid robot mimic upper-body human movement in real-time using 3D pose estimation from a stereo camera and MediaPipe-based skeleton tracking?

This leads to the research hypothesis:

By combining MediaPipe's 2D pose estimation with depth data from a stereo vision system, it is possible to achieve low-latency, spatially accurate upper-body mimicry in a humanoid robot, even in the presence of non-aligned joint structures between the human and the robot.

1.5 Implications and Significance

Successful implementation of this project could benefit multiple stakeholders. For the scientific community, it advances understanding of robot mimicry using depth-based pose estimation. For industry, especially in fields such as hazardous material handling, elder care, and remote maintenance, the system could enable safer, more intuitive control of robots in environments where direct human presence is infeasible. It also sets the

groundwork for further research into full-body teleoperation and autonomous assistive robotics.

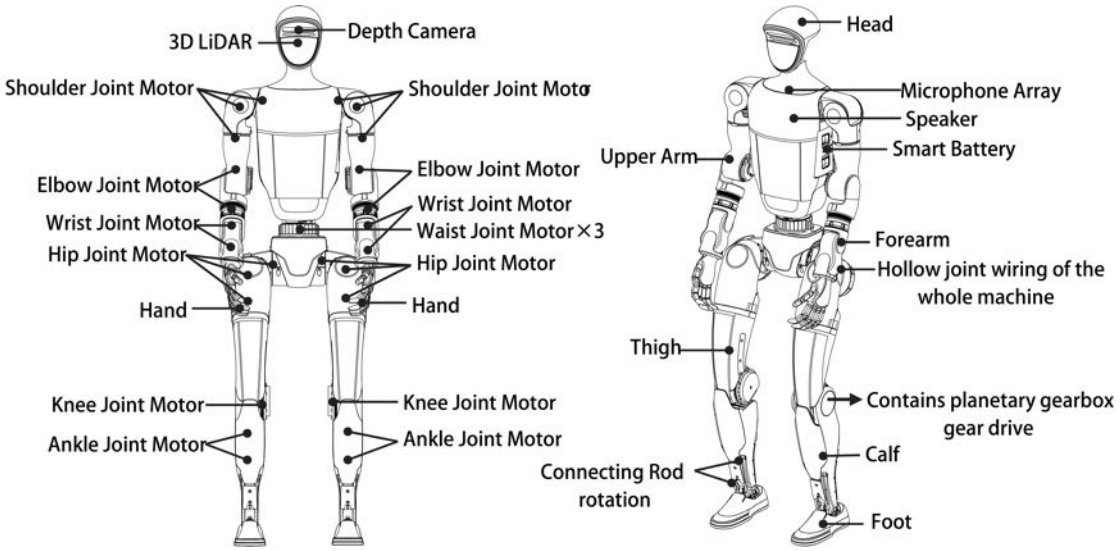


Figure 2. Unitree G1 Robot Overview. [4]

2. Literature Review

2.1 Human-Robot Interaction and Teleoperation

Human-Robot Interaction (HRI) encompasses the study and design of systems in which humans and robots communicate and collaborate, often in shared environments. In collaborative robotics, or "cobotics", robots are not just tools but intelligent teammates capable of adapting to human input at both physical and cognitive levels. These interactions are central in domains such as assistive robotics, telemedicine, industrial automation, and telepresence [5].

Teleoperation, a subset of HRI, involves the control of robotic systems by human operators from a distance. Traditional teleoperation interfaces often suffer from complexity and unintuitive mappings. This is further complicated when a direct one-to-one mapping between human limb motion and robot kinematics is not feasible [6] [7]. In humanoid robots this is particularly pronounced, where differences in degrees of freedom (DOF), joint limits, and dynamics pose challenges to real-time, smooth control.

Recent approaches address these challenges through *motion retargeting*, which involves mapping human pose data, often obtained from depth cameras and pose estimation algorithms like MediaPipe, onto a humanoid model using inverse kinematics and optimization techniques. Relaxed mimicry systems, which allow deviations from strict one-to-one joint mapping, have shown promise in improving the fluidity and safety of telemanipulation tasks by minimizing joint velocity spikes and avoiding kinematic singularities [2].

Moreover, real-time systems that use RGB-D cameras (e.g. OAK-D or Intel RealSense) paired with MediaPipe pose estimation can provide 3D skeletal data robust enough for interaction tasks such as gesture-based control, object handover, and remote manipulation [1]. Machine learning, especially reinforcement learning and long short-term memory (LSTM) models, can be leveraged to handle time-dependent pose data and predict human intent, enabling robots to co-adapt with human partners in shared spaces [8].

Figure 3. OAK-D Lite RGB-D Stereo Camera. [9]

2.2. Human Pose Estimation Techniques

Human pose estimation has become a common problem in computer vision, enabling machines to interpret, predict, and interact with human movement. Traditionally, pose estimation relied on handcrafted features like silhouettes or edge histograms; however, the field has shifted towards deep learning models capable of extracting 2D or 3D joint positions with high precision [10]. One dominant pipeline, especially for 3D estimation, is the two-stage approach: estimating 2D joint coordinates from RGB images and subsequently "lifting" them to 3D space. This separation allows leveraging large-scale 2D annotated datasets while addressing the depth ambiguity inherent in monocular vision [10].

MediaPipe Pose, is a lightweight and real-time capable model developed by Google [11], which utilises a BlazePose [12] convolutional neural network (CNN) to estimate human poses. It serves as an effective 2D keypoint extractor and has been widely adopted for deployment on mobile devices. It identifies 33 body landmarks per frame and operates efficiently without dedicated GPUs, which is particularly valuable in mobile robotic applications [7]. However, transforming these 2D landmarks into reliable 3D poses remains challenging due to ambiguities and occlusions.

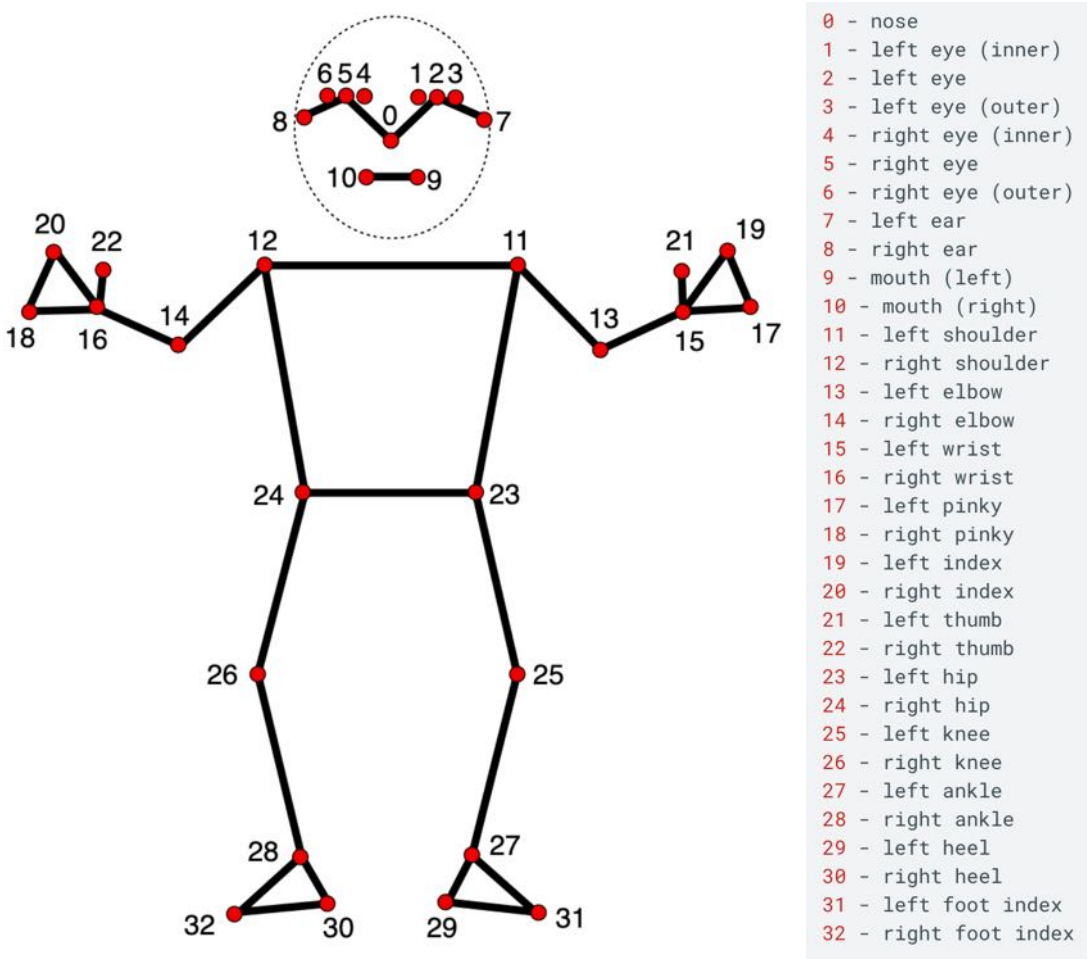


Figure 4. BlazePose Keypoint Landmarks Index. [13]

Several approaches attempt to overcome this gap. Martinez et al. [10] proposed a surprisingly effective deep feedforward network that maps 2D joint locations to 3D poses, significantly outperforming more complex end-to-end systems. Their findings emphasize that visual understanding, rather than the lifting itself, is often the primary bottleneck in pose estimation [10]. In contrast, Kim et al. [7] proposed an optimization-based method to estimate 3D joint angles by matching MediaPipe-derived 2D poses to a humanoid model using a univariate Dynamic Encoding Algorithm for Searches (uDEAS). Their approach introduces additional constraints such as centre-of-mass deviation and joint angle limits to improve accuracy and realism, particularly under conditions where deep networks fail due to unseen postures [7].

Furthermore, combining 2D skeleton data with real-time depth information from sensors such as the OAK-D enables more accurate 3D point cloud generation. This fusion significantly reduces error margins in spatial pose estimation, even at varying camera

distances, thereby enhancing the robot's ability to assess and respond to human movement [1] [9] [14].

Collectively, these methods underline the strength of hybrid architectures that combine lightweight 2D detectors with optimization or depth-assisted 3D estimation frameworks. They are particularly well-suited for robotic mimicry applications where real-time performance, robustness, and adaptability to novel postures are essential.

2.3 3D Pose Estimation Using RGB-D Sensors

The estimation of 3D human pose using RGB-D sensors represents a practical and effective approach for teleoperation and human-robot interaction. These sensors, such as the OAK-D, Intel RealSense, and Microsoft Kinect, provide synchronized colour (RGB) and depth (D) data that can be fused to reconstruct human skeletons in three dimensions. This fusion overcomes the inherent ambiguity present in monocular 2D images and enables more accurate spatial reasoning in robotic mimicry systems.

A common pipeline involves first extracting 2D body keypoints using pose estimation frameworks like MediaPipe, followed by aligning these points with depth data to infer their 3D coordinates. The RealSense-based method by Chao et al. [1] illustrates this clearly: they used MediaPipe for 2D landmark detection, then matched each joint with the corresponding pixel's depth value obtained from the RealSense D455 camera to produce 3D coordinates. The 3D skeleton was visualized and validated through a point cloud representation constructed using the RealSense SDK and OpenGL.

2.4 Evaluation Methodology

To evaluate the accuracy of their 3D estimation approach, Chao et al. [1] conducted controlled experiments at fixed subject distances of 1.0, 1.5, and 2.0 meters from the camera. At each distance, the error between the measured joint positions and ground truth coordinates was computed. They reported a depth error margin of less than 1.5%, demonstrating the robustness of their fusion method even at varying depths.

This form of validation is crucial in applications like teleoperation, where spatial fidelity of joint positions directly affects control precision and safety. The rationale behind using fixed-distance measurements is to ensure repeatability and isolate the effect of depth resolution, which degrades with distance in stereo vision systems.

Other researchers, such as Kim et al. [7], adopted an alternative strategy by validating their optimization-based pose estimation using mean joint coordinate error and joint angle error relative to known humanoid models. They reported an average joint position error of 0.097 m and a mean angular error of 10.02°.

2.6 Identified Gaps in the Literature

Lack of Real-Time, Markerless 3D Pose Estimation Systems for Humanoid Control: While studies such as Martinez et al. [10] and Kim et al. [7] propose accurate pose estimation pipelines using 2D-to-3D lifting and humanoid modelling respectively, they typically rely on offline datasets or lab-controlled environments (e.g., Human3.6M). Others works such as Lee et al. [15] mimic only static poses. Real-time, markerless systems using RGB-D cameras (e.g., OAK-D) for controlling a physical robot are rarely demonstrated in practical deployments.

Sparse Integration Between MediaPipe and Real-Time Robotics Platforms: MediaPipe is commonly used for real-time body tracking, but most studies [11] [16] focus on pose classification or fitness monitoring rather than control output for physical robots . The integration of MediaPipe with real-world robotics control, especially humanoids, is not comprehensively explored.

Limited Handling of Life Scale Humanoid Robots: Whilst some studies have been done on larger humanoid robots [17], most works [15] [6] [2] [8] have involved smaller scale robots such as the NAO or done completely in simulation. As commercial grade humanoid robots are newly emerging in the field, a lack of studies controlling large humanoids like the Unitree G1, using pose estimation have been undertaken.

3. Methodology

To accomplish the project objectives of enabling intuitive upper-body teleoperation of the Unitree G1 humanoid robot, a structured multi-phase approach will be adopted. This includes iterative development, integration, and testing of a vision-based motion capture and robot control system. Once this framework has been developed and if time permits, other pose tracking technologies may be implemented and tested to compare against the baseline of the proposed computer vision driven system.

3.1 Techniques

The core methodology involves 3D human pose estimation using RGB-D data, skeletal keypoint extraction via deep learning, and real-time teleoperation control via kinematic mapping. MediaPipe will be used for 2D pose landmark detection, and these will be lifted to 3D using parallax data from the OAK-D stereo depth camera. ROS 2 will provide the communication framework between perception, processing, and the Unitree robot control interface.

To address the non-isomorphic nature of human and robot joints, custom inverse kinematics (IK) heuristics will be explored, incorporating smoothing filters and coordinate transformation to ensure stable mimicry.

3.2 Equipment

1. Unitree G1 humanoid robot.
2. OAK-D stereo depth camera for real-time RGB and depth capture.
3. Development workstation running Ubuntu via Parallels on macOS for cross-platform deployment.
4. Software stack: ROS 2 Humble, Python (OpenCV, DepthAI), MediaPipe.

3.3 Procedures

The process will begin with data acquisition from human pose demonstrations in controlled environments. Depth and RGB data will be collected simultaneously to calibrate the 3D pose lifting algorithm. Once accurate keypoints are mapped, a custom

ROS node will translate joint positions into control commands using the Unitree G1's API. Each stage will be validated using simulation testing and then deployed to the robot platform, ensuring both responsiveness and safety in mimicry control.

Iterative testing will identify performance bottlenecks, noise sources, and mapping errors. These will be addressed through filtering techniques and retargeting adjustments. A final demonstration will evaluate system effectiveness in manipulation tasks involving common objects on a table.

4. Timeline and Risk Management

4.1 Schedule

The project is structured across two academic semesters in 2025, with each phase progressing through a clear sequence of milestones. The Gantt chart in Appendix A, outlines task durations, overlaps, and dependencies to guide implementation and evaluation.

4.2 Semester 1 (2025) - Pose Capture and Mapping Phase:

This semester focuses on the development and integration of the human pose estimation pipeline. Key tasks include:

1. MediaPipe Implementation and Testing (Weeks 1-5)
2. Camera Mount Design, Installation, and Testing (Weeks 4-6)
3. Depth-Based Body Tracking and 3D Pose Estimation (Weeks 5-10)
4. Initial Mapping of Human Joints to Robot (Weeks 7-12)

As of Week 10, the project is slightly ahead of schedule, with work starting to implement motion tracking onto the robot.

4.3 Semester 2 (2025) - Teleoperation and Evaluation Phase:

This semester shifts focus to robot-side development and system refinement:

1. Continued Joint Mapping and Manipulation Algorithm Development (Weeks 1-7)
2. Testing and Debugging (Weeks 4-10)
3. Optional Lower-Body Extension (Weeks 6-10, contingent on feasibility)
4. Final System Integration and Evaluation (Weeks 9-11)
5. Final Report and Presentation Preparation (Weeks 10-12)

4.4 Critical Path:

The critical path includes tasks that are essential for downstream progress, where any delay could directly impact the project timeline. These tasks are:

1. MediaPipe Implementation → MediaPipe Testing
2. Camera Mount Design →Installation and Testing
3. Body Tracking using Depth Camera →3D Pose Estimation
4. Mapping Human Joints to Robot (spanning both semesters)
5. Manipulation Algorithm Development
6. Testing and Debugging
7. Final Integration and Evaluation
8. Final Report and Presentation

Delays in any of these could compromise the integration timeline in Semester 2. Due to the serial nature of most tasks, delay in any of these milestones will cause delays in the project.

4.2 Risk Management

The following section addresses possible risks associated with the project and how they will be mitigated or dealt with.

Safety Risks

Risk:	Collision between the robot and nearby personnel or objects during operation and testing.
Likelihood:	Moderate
Consequences:	Minor injury or equipment damage.
Management Strategy:	The robot will operate in a controlled environment with limited human presence. ROS-based safety zones and velocity limiters will be configured, and the robot will be equipped with a dead man stop mechanism.
Likelihood After Management Strategy:	Low

Financial Risks

Risk:	Loss or damage to the OAK-D depth camera or Unitree G1 hardware.
Likelihood:	Moderate
Consequences:	High financial burden due to the cost of replacement and delays in project execution.
Management Strategy:	All hardware will be handled in accordance with manufacturer
Likelihood After Management Strategy:	Low

5. Progress to Date

Significant progress has been achieved in the first half of the project, with core technical milestones either completed or advancing ahead of schedule. The foundational system for 3D human pose estimation and ROS2-based data integration is operational, and early steps toward real-time humanoid control have begun.

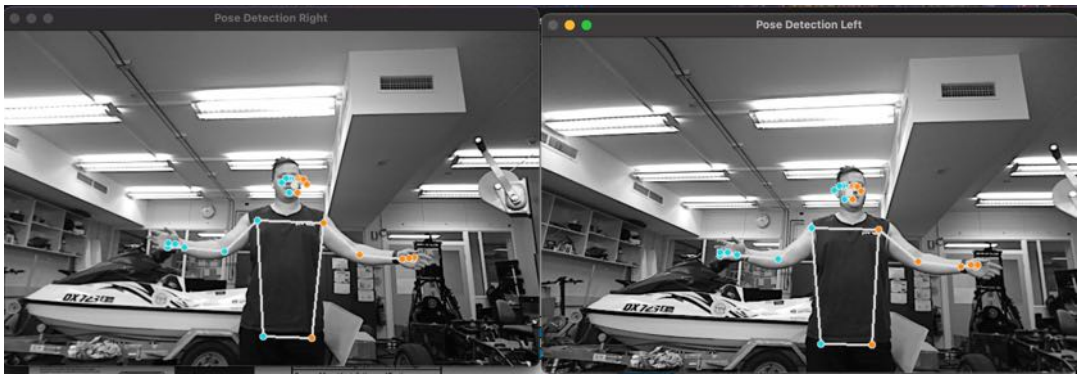


Figure 5. Media Pose Running on OAK-D Stereo Camera.

5.1 Completed Tasks

As referenced in the Gantt chart in Appendix A, the following tasks have been completed or are in advanced progress:

MediaPipe Implementation and Testing: MediaPipe has been successfully implemented to run in real-time at 30 FPS (approx.), accurately detecting upper-body keypoints. The system shows good robustness in various postures under consistent lighting conditions.

DepthAI Integration: The OAK-D pipeline has been configured to capture depth data aligned with the 2D keypoints provided by MediaPipe. This has enabled 3D coordinate generation for each detected joint.

ROS2 Network Integration: The full perception pipeline has been "rosified" and deployed on an Ubuntu system, with the 3D joint data published over the ROS2 network. This data is successfully visualised in RViz, confirming spatial accuracy and correct formatting.

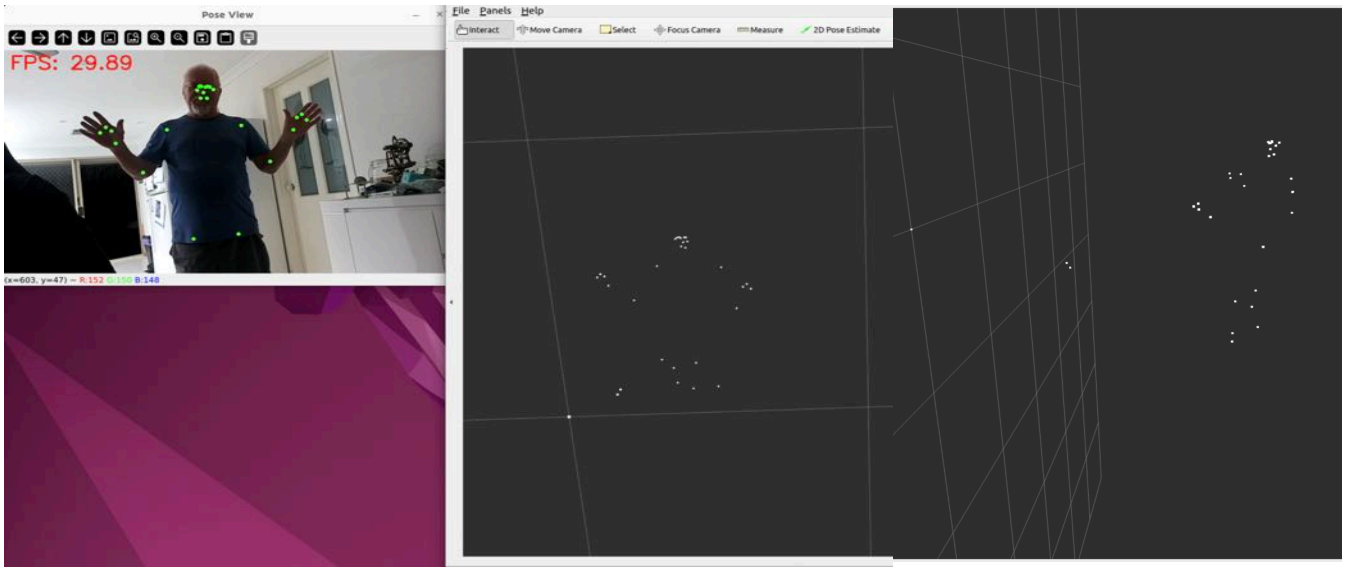


Figure 6. Left: Live camera feed with media pipe skeleton points, Middle: 3D points published over ROS2 network (front view), Right: 3D points published over ROS2 network (Side View)

5.2 In Progress / Delayed Tasks

Initial Robot Control Work:
Implementation of robot motion using the Unitree G1 SDK has begun slightly ahead of schedule. Early tests suggest that upper-body joint movements can be successfully replicated using the real-time skeleton data.

Camera Mount Fabrication:
The physical neck-mounted camera support, originally scheduled for completion by Week 6, has not yet been fabricated. In the interim, the camera is operating from a fixed location. This delay has not yet impacted software development but may limit real-world mobility testing until resolved.

5.3 Preliminary Findings and Lessons Learned

Pipeline Latency and Stability:
The ROS2-integrated pose estimation system performs reliably in controlled environments, but minor instability was observed in low-light or high-occlusion scenarios. Temporal smoothing or filtering may be necessary for robot stability during rapid pose transitions.

Depth Accuracy:
Depth readings from the OAK-D have proven sufficient for spatial estimation at close range (approx. 1-2 meters), aligning with initial planning assumptions. Fixes need to be made to remove floating keypoints that lose tracking from visualisations.

SDK Limitations:
Early interaction with the Unitree SDK has revealed documentation gaps and limited abstraction layers. Custom wrapper nodes may be required to streamline manipulation command integration via ROS2.

5.4 Next Steps

1. Complete the fabrication and testing of the camera neck mount to enable robot-mounted tracking during movement.
2. Continue developing and validating the mapping between human joints and robot actuators, including inverse kinematics constraints.
3. Begin integration of manipulation tasks, including gesture recognition (e.g., open/closed hand states) using additional image classification models or MediaPipe hand landmarks.
4. Begin evaluating the feasibility of using Oculus Quest's built-in hand tracking to replace or supplement current pose capture methods. If successful, this will supersede the lower-body mimicry task from Semester 2 and allow for a more immersive first-person teleoperation system.

5.5 Changes to Timeline

The "Lower Body Extension" task will likely be replaced with an Oculus Quest-based teleoperation interface, which aligns better with the project's evolving capabilities and user-control goals.

Despite the minor delay in the camera mount construction, most core milestones are on or ahead of schedule, particularly in perception and ROS2 integration.

6. Bibliography

[1] C.-L. Yang, S.-Q. Wu, S.-J. Chen, T.-C. Kao, C.-H. Huang, E. Liou, P.-T. Lin and K.-L. Hua, The Fusion and Verification of 2D Human Skeleton and 3D Point Cloud based on RealSense, Taiwan: 2023 International Conference on Advanced Robotics and Intelligent Systems (ARIS 2023), 2023.

[2] D. Rakita, B. Mutlu and M. Gleicher, A Motion Retargeting Method for Effective Mimicry-based Teleoperation of Robot Arms, Madison: Department of Computer Sciences, University of Wisconsin–Madison, 2017.

[3] Unitree Robotics, "Unitree G1," 2025. [Online]. Available: https://www.unitree.com/g1. [Accessed 8 May 2025].

[4] Unitree Robotics, "G1 User manual v1.1," 2024. [Online]. Available: https://reliablerobotics.ai/wp-content/uploads/2025/03/G1-User-Manual_compressed.pdf. [Accessed 8 May 2024].

[5] M. Rahman, F. Khatun, I. Jahan, R. Devnath and A.-A. Bhuiyan, "Cobotics: The Evolving Roles and Prospects of Next-Generation Collaborative Robots in Industry 5.0," *Journal of Robotics,* vol. 2024, p. 22, 2024.

[6] Y. Ou, X. Li, J. Hu, Z. Wang, Y. Fu and X. Wu, A Real-Time Human Imitation System Using Kinect, Springer Science+Business Media Dordrecht, 2015.

[7] J.-W. Kim, J.-Y. Choi, E.-J. Ha and J.-H. Choi, Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model, Busan: Applied Sciences, 2023.

[8] F. Semeraro, A. Griffiths and A. Cangelosi, Human-robot collaboration and machine learning: a systematic review of recent research, 2022.

[9] Luxonis, "OAK-D Lite (default USB boot," 2021.

[10] J. Martinez, R. Hossain, J. Romero and J. J. Little, A simple yet effective baseline for 3d human pose estimation, Vancouver: International Conference on Computer Vision, 2017.

[11] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. George and M. Grundmann, MediaPipe: A Framework for Building Perception Pipelines, 2019.

[12] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang and M. Grundmann, BlazePose: On-device Real-time Body Pose tracking, Seattle: Workshop on Computer Vision for Augmented and Virtual Reality, 2020.

[13] Google, "Pose landmark detection guide," Google, 13 Jan 2025. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker. [Accessed 8 May 2025].

[14] S. Macenski, T. Foote, B. Gerkey, C. Lalancette and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics, 2023.

[15] V. V. Nguyen and J.-H. Lee, "Full-body imitation of human motions with kinect and heterogeneous kinematic structure of humanoid robot," in *International Symposium on System Integration*, Kyushu, 2012.

[16] S. Negi, M. Garg, H. Maindola, V. Kansal, U. Jain and S. Bhatla, Real-Time Human Pose Estimation: A MediaPipe and Python Approach for 3D Detection and Classification, Dehradun: International Conference on Technological Advancements in Computational Sciences, 2023.

[17] M. Riley, A. Ude, K. Wade and C. G. Atkeson, “Enabling Real-time Full-Body Imitation: A Natural Way of Transferring Human Movement to Humanoids,” in *Proceedings of the 2003 IEEE International Conference on Robotics & Autamatioo*, Taipei, 2003.

Appendices

Appendix A - Project Gantt Chart