Project Thesis

Lucerne University of Applied Sciences and Arts

Profile: Mechatronics and Automation

Human Pose Detection for Robotic Scrub Nurse Systems

Author: Daniel Kogan, B. Eng.
Matriculation Number: 23-565-385

Advisor: Prof. Dr. Björn Jensen
Expert: Thomas Estier

Zurich, 17.01.2025

Declaration of Independence

I hereby declare that I completed this work independently and did not use any tools other than those specified. All text excerpts, quotes or content from other authors used were expressly marked as such.

Daniel Kogan

Zurich, 17.01.2025

Abstract

The global shortage of nurses, including scrub nurses, presents a growing challenge for healthcare systems worldwide. Scrub nurses play a critical role in assisting surgeons by passing tools during surgeries, often under demanding and hazardous conditions. To address this challenge, robotic scrub nurses have been proposed to support tool handovers in surgical environments. However, existing solutions primarily focus on speech recognition, gesture recognition, and tool handling, with little emphasis on leveraging human pose estimation to dynamically track personnel in the operating room.

This thesis investigates how human pose estimation can be applied to robotic scrub nurses by evaluating different approaches and developing a Proof of Concept focused on hand detection to facilitate tool handovers. The pipeline integrates MediaPipe Hands, a human pose estimation model optimized for hand poses, with depth data from the Intel RealSense D455 RGB-D camera. These components are implemented in a modular ROS 2 framework, enabling gesture recognition, frame alignment, and motion control. Two control methodologies are developed: a MoveIt 2-based method for sophisticated trajectory planning and a web socket-based approach for rudimentary control relying on the robot’s native controller.

A foundational framework is established to integrate human pose estimation with collaborative robotics, addressing the challenges of dynamic and constrained surgical environments. Initial implementation confirms that hand tracking and robot control are functional. However, further simulation in conditions resembling those of an operating room is needed to fully assess the system’s robustness and effectiveness. The resulting insights from this thesis highlight the system’s potential and open pathways for future research into dynamic obstacle avoidance, advanced tool-specific handover strategies, and real-world system validation.

Keywords: robotic scrub nurse, surgical robotics, human pose estimation, tool handover, gesture recognition

Table of Contents

1 Introduction

Understaffing of nurses is a significant and growing challenge in healthcare worldwide. The World Health Organization (WHO) estimates a global nursing deficit of 12.9 million by the year 2035. This issue is further exacerbated by an aging population. According to the United Nations (UN), the number of people aged 60 and older was 901 million in 2015 and is expected to reach 1.4 billion by 2030 [1, p. 3]. These demographic shifts highlight the urgent need to alleviate the burden on healthcare workers by leveraging technological solutions.

While collaborative robots (cobots) have already been introduced into healthcare, most of these systems are for logistical tasks or patient-centered, with very few designed to directly assist nurses [2, pp. 6-9]. Within nursing, scrub nursing constitutes a specialized and demanding field. Scrub nurses play a critical role in surgical procedures, requiring a high level of skill, precision, and focus. Their primary responsibilities include managing and passing surgical instruments to the surgeon, often under high-pressure conditions. In larger-scale surgeries, scrub nurses must classify and organize numerous surgical instruments, remember their names and usage, and anticipate the surgeon’s needs. The high cognitive load, combined with the physical demands of prolonged surgical procedures, can lead to fatigue and communication breakdowns, particularly during emergency operations or nighttime shifts [3, p. 74].

Developing robotic systems to assist scrub nurses could potentially alleviate some of these challenges. However, existing research in robotic scrub nurse (RSN) systems has largely focused on speech recognition [4], [5], hand gesture recognition [6], instrument detection [3], [4], [5], [7], [8] or instrument anticipation [9]. These approaches provide a foundation but often lack the ability to adapt dynamically to complex surgical environments. Current solutions typically rely on designated hand-off spaces for instrument exchange and pre-defined movement paths and fail to consider the dynamic nature of human movement and interaction in the surgical setting.

1.1 Research Question and Scope

To the best of the author’s knowledge, no existing approach has focused on leveraging human pose estimation (HPE) to optimize collision avoidance and facilitate tool handovers in surgical environments. HPE may offer the potential to enhance safety and efficiency in RSNs, by dynamically tracking human movement. Unlike systems that rely on predefined paths, HPE could enable a more adaptable approach, which is non-trivial given the highly constrained environment during surgical procedures, as described in detail in Chapter 2.1.

The aim of this work is to evaluate state-of-the-art 3D HPE methods and explore their applicability for dynamic robotic assistance in healthcare, with a focus on tracking human motion, incorporating accurate depth information, and enhancing robot interaction in real-world surgical environments.

As a first step toward advancing robotic assistance in surgical settings, this thesis establishes foundational frameworks for 3D HPE and develops a proof of concept (POC) for tool handovers between a cobot and surgical personnel. Future work will expand upon this foundation by utilizing the developed 3D HPE frameworks to capture 3D spatial data of individuals in the operating room (OR), enabling dynamic collision avoidance. Additionally, subsequent works will refine the POC to enhance the tool pickup and handover mechanism, ultimately progressing toward the realization of a fully functional RSN prototype.

1.2 Structure

The structure of this thesis is outlined below, with the following chapters summarized.

Chapter 2 establishes the theoretical foundation of the thesis. It explores the complexities of surgical environments, the principles of HPE, and the state-of-the-art in 2D and 3D HPE methods. Furthermore, it describes the hardware and software frameworks utilized in this work, providing the context for the system's design and implementation.

Chapter 3 focuses on the experimental evaluation of two HPE methods, testing a 2D and a 3D model in a controlled scenario. The experiments aim to explore the capabilities and limitations of each method, providing insights into their potential use in the RSN system and forming the basis for the subsequent method selection. Additionally, this chapter examines monocular depth estimation (MDE) as a potential alternative to RGB-D cameras.

Chapter 4 presents the concept and implementation of the POC. It details the integration of the selected HPE model into a modular ROS2 architecture and demonstrates the system's functionality through a practical demonstration.

Chapter 5 provides a critical discussion of the results, assessing the strengths and limitations of the developed system. It also identifies potential areas for improvement and outlines the necessary steps to enhance the system's capabilities for real-world applications.

Finally, Chapter 6 concludes the thesis and offers a forward-looking perspective on future developments. It summarizes the main findings and discusses the steps required to advance the POC into a fully operational RSN prototype, addressing both technical and practical considerations.

2 Theoretical Foundation

This chapter establishes the theoretical framework for the development of the POC. It begins by examining the dynamic and constrained nature of ORs, highlighting the necessity for adaptable and precise robotic systems. Subsequently, it delves into the technical underpinnings of HPE, providing a comprehensive overview of 2D and 3D approaches, their associated challenges, and the models used to represent human body poses. Finally, the chapter outlines the essential hardware and software components that form the backbone of the POC, including a collaborative robot, depth camera, and the ROS2 framework.

2.1 Surgical Environment

The OR is a highly complex and dynamic environment where various medical professionals, including surgeons, anesthetists, and scrub nurses, must collaborate closely. The spatial constraints and abundance of medical equipment often limit mobility and can hinder effective communication [10, p. 2]. Moreover, the layout of the OR varies depending on the type of surgical procedure, introducing additional challenges for the personnel [1, p. 5]. Figure 1 shows the surgical team in the OR. The yellow arrows display the interaction between the expert operator and anesthetist, scrub nurse, and young assistant within the operating field. The anesthetist and circulating nurse are outside the operating field [11, p. 246]. The image illustrates the significant constraints in the OR, emphasizing the necessity for a RSN to dynamically adapt to the movements of each team member while maintaining efficient operation.

In contrast, as shown in Figure 2, the disparity in organizational structure and environmental setup between typical ORs and those required for RSNs becomes evident. While typical ORs are cluttered and space-constrained, simulated surgeries with RSNs take place in minimalistic and highly organized environments. The reliance of current RSN systems on predefined tool handover positions and their limited spatial awareness might necessitate such structured and decluttered setups, with minimal personnel, to ensure their functionality.

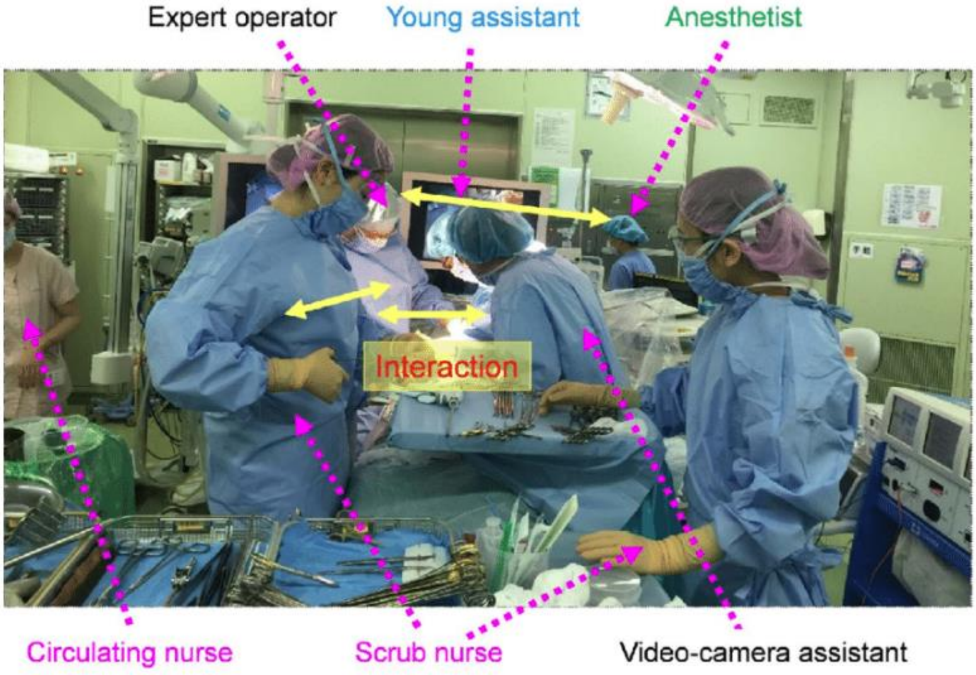


Figure 1: Surgical members in OR [11, p. 246]

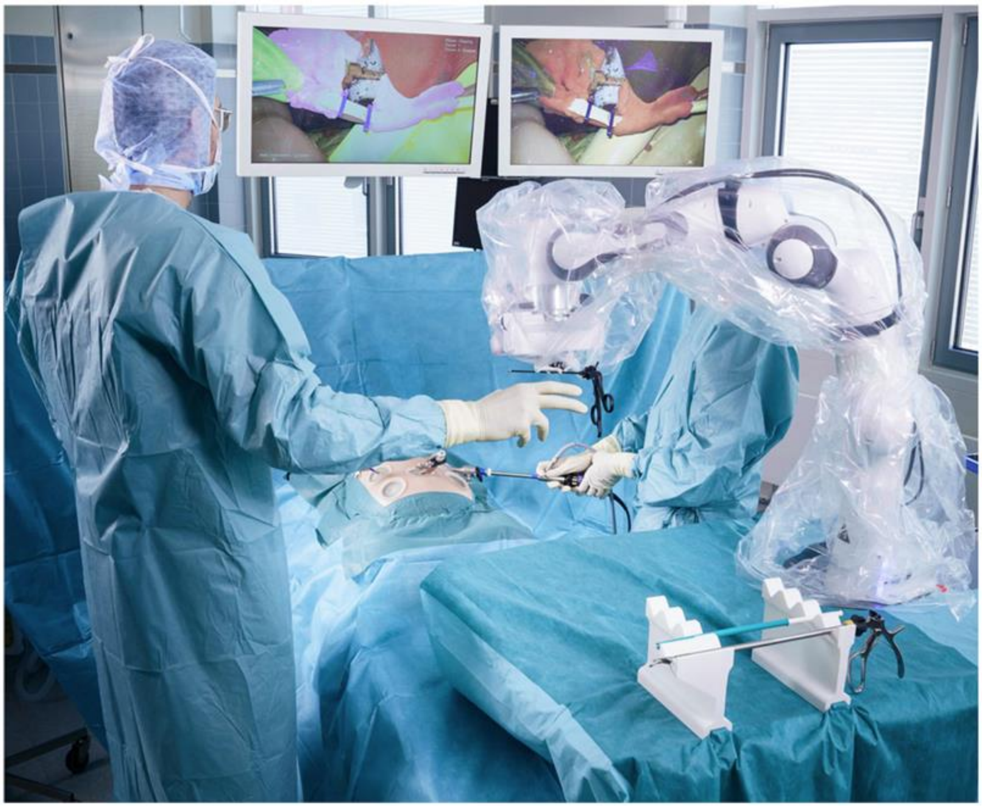


Figure 2: Deployment of a RSN in a simulated surgery [9, p. 3]

2.2 State of the Art in Human Pose Estimation

HPE is a computer vision task that involves detecting and representing human skeletal joints from images or videos to estimate body configuration (pose). Typically applied to monocular images, HPE has become a cornerstone in various fields, including human-computer interaction, motion analysis, and robotics [12, p. 1], [13, p. 2], [14], [15, pp. 1-2], where understanding human movement is critical for seamless collaboration between humans and machines.

Traditional approaches to 2D human pose estimation, which relied on handcrafted features and simplified body representations, laid the groundwork for early research but were limited by low accuracy, high computational complexity, and insufficient ability to handle complex scenarios. As a result, these methods have largely been replaced by deep learning-based approaches [14]. Modern HPE predominantly relies on deep learning due to its superior performance, enabled by advancements in GPUs, the availability of large datasets, and the exceptional capabilities of Deep Neural Networks (DNNs) [12, p. 2], [13, p. 2], [15, p. 1].

Given these advancements, this thesis focuses exclusively on deep learning-based models for HPE, as they represent the state of the art in the field and provide the most viable solutions for addressing the challenges faced in collaborative robotics. Despite their strengths, HPE still encounters significant challenges, which vary depending on whether it involves single-person or multi-person detection, as well as whether it focuses on 2D or 3D pose estimation. Handling complex backgrounds, occlusions, and intricate poses continues to impact its accuracy and robustness. Furthermore, high computational complexity remains a barrier to its scalability in real-world scenarios [14].

This chapter explores the technical aspects of HPE, examining the foundational body models and providing an overview of 2D and 3D HPE approaches, with a specific focus on their application to collaborative human-robot interaction in surgical settings.

2.2.1 Challenges

The following section delves into the challenges associated with HPE, particularly in the context of human-robot collaboration within surgical settings.

Lack of accuracy

“The ultimate goal of any HPE model is to achieve a better accuracy” [14]. Accuracy is paramount in surgical environments, where cobots must precisely detect and localize human body parts to ensure safe and reliable tool handovers and avoid collisions. Inaccurate pose estimations could result in hazardous situations, particularly when working with sharp or delicate surgical instruments. However, it has been observed that many HPE algorithms prioritize other parameters, such as speed or robustness, over accuracy, which can compromise their reliability [14].

Occlusion

Occlusion represents a significant challenge in HPE, arising from various sources such as self-occlusion, occlusion between individuals, or occlusion caused by objects [14], [15, p. 2]. In ORs, the confined space, along with the presence of multiple team members, surgical equipment, and the RSN, can result in significant occlusions. HPE systems must detect poses even when body parts are partially or fully obstructed, as failures in such scenarios could compromise the cobot's ability to respond effectively.

Time complexity

Real-time performance is essential in surgical environments, where immediate responses to human actions are required. However, many HPE algorithms face significant challenges related to time complexity. Deep learning methods can struggle to maintain efficiency if the data is not appropriately preprocessed. In multi-person scenarios, which are common for ORs, the demand for computational resources increases further, making it difficult to balance speed and accuracy [14].

Preprocessing requirements

ORs often present unique challenges, such as varying lighting conditions and the presence of reflective surfaces, which complicate the preprocessing steps required for HPE. These steps include tasks such as background subtraction, data calibration, and image conversion. Ensuring that these preprocessing requirements are met efficiently and robustly is critical for maintaining the reliability of HPE systems in such demanding environments [14].

Depth ambiguity

For 3D HPE from monocular RGB images and videos, depth ambiguity remains one of the most significant challenges. Monocular systems struggle to accurately infer the depth of keypoints, leading to errors in the spatial representation of poses. Some approaches attempt to address these challenges by utilizing additional hardware, such as depth sensors, inertial measurement units (IMUs), or radio frequency (RF) devices [13, pp. 17-18], [15, p. 2]. However, these solutions are often not cost-effective and rely on special-purpose hardware, which may not be practical for widespread deployment in surgical environments.

Lack of 3D training data

A significant challenge in 3D HPE is the lack of sufficient 3D training data. Since manual annotation of 3D poses is both expensive and time-consuming, the availability of datasets paired with accurate 3D annotations is limited. This scarcity poses a challenge for deep learning methods, whose performance heavily depends on the scale and quality of training data annotated with reference 3D poses [15, p. 2].

In the context of surgical settings, this limitation becomes particularly problematic due to the medical workwear commonly worn in ORs. Surgeons and nurses often wear medical gowns, masks, gloves, and, in some cases, specialized eyewear or headgear, which can obscure key body features and joints. These garments not only alter the visible contours of the human body but also introduce additional challenges for pose estimation models, as the distinctive features used for detecting keypoints may be partially or completely concealed. The absence of large, high-quality 3D datasets that account for such real-world surgical scenarios makes it difficult to train models capable of addressing these specific challenges.

2.2.2 Human Body Models

Human body modeling is a fundamental component of HPE. The human body, as a flexible and complex non-rigid object, exhibits many distinct characteristics, such as its kinematic structure, body shape, surface texture, and the positions of body parts or joints. A mature human body model does not need to encapsulate all attributes but should be designed to meet the specific requirements of the task, effectively representing and describing the human body pose [16, p. 4]. HPE models can therefore represent the human body pose using three distinct approaches to meet different requirements. These models rely on the extraction of keypoints and significant features from the input data and are typically based on model-driven techniques [12, p. 2], [14]. The three primary methods for pose estimation are illustrated in Figure 3.

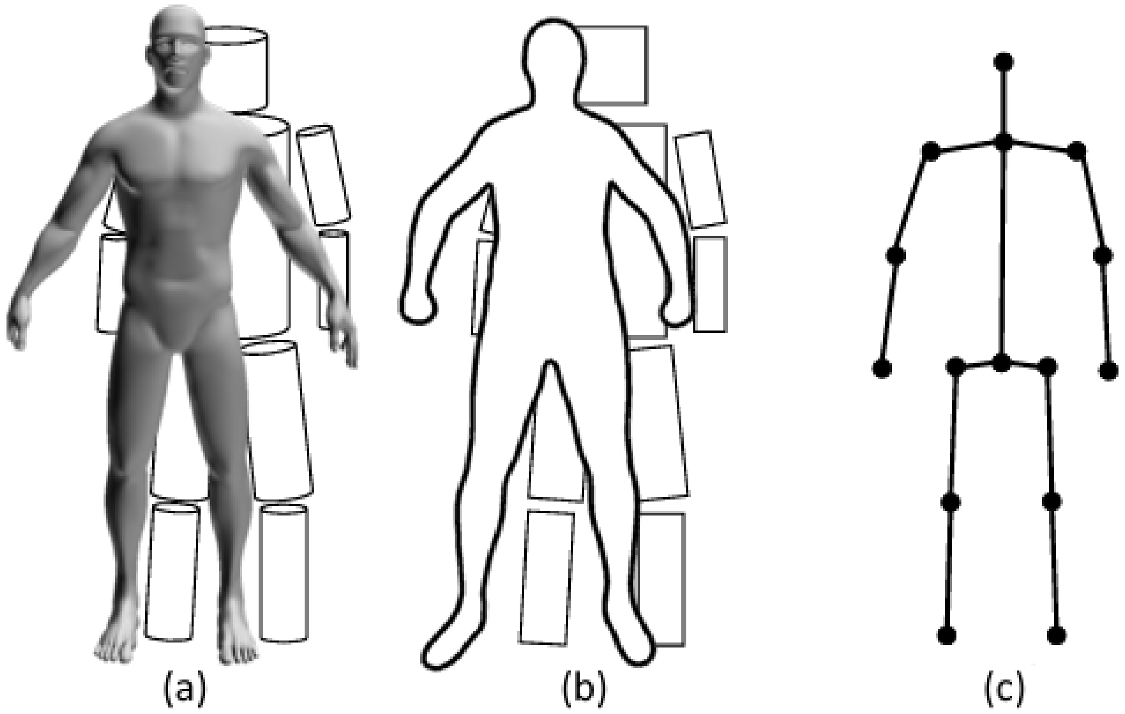


Figure 3: Three types of human body models [16, p. 4]

Volumetric model

Volumetric models or volume-based models, as seen in Figure 3 (a), represent the human body using geometric shapes or triangulated meshes, providing a detailed and deformable representation suitable for the non-rigid nature of the human body [12, p. 2], [15, p. 4], [16, p. 5]. Earlier approaches used simple geometric primitives such as cylinders and cones to approximate body parts. Modern volumetric models, however, predominantly employ mesh representations, often derived from 3D scans, which allow for a more precise depiction of body shapes and poses [15, p. 4], [16, p. 5].

Volumetric models are known for their ability to capture detailed body contours and surface features, making them valuable for applications requiring high realism. However, their complexity and computational demands can pose challenges for real-time implementations.

Planar model

Planar models, illustrated in Figure 3 (b), also referred to as contour-based models, are primarily used to represent the human body’s appearance and shape by approximating body contours. These models depict body parts using multiple rectangles or boundaries that follow the silhouette of the human figure [14], [15, p. 4], [16, p. 5].

Two widely recognized approaches within planar models are the cardboard model and the Active Shape Model (ASM). The cardboard model represents body parts using rectangular shapes and incorporates foreground color information to enhance its representation. The ASM, on the other hand, provides a more flexible approach by statistically modeling variations in body shape, allowing it to adapt to different contours [15, p. 4].

Kinematic model

Kinematic models, also referred to as skeleton-based or stick-figure models, represent the human body by mapping joint positions and limb orientations based on the skeletal structure [12, p. 2], [14], [15, p. 3], [16, p. 5]. These models are widely utilized in both 2D and 3D HPE due to their simplicity and effectiveness in describing human body configurations [12, p. 2], [16, p. 5]. Figure 3 (c) illustrates the kinematic model.

Traditional kinematic models annotate between 10 and 30 keypoints [16, p. 5]. Advanced approaches, such as AlphaPose [17] and OpenPose [18], extend this to 130–140 keypoints, including hand and facial landmarks, as shown in Figure 4. One recent development, Meta's Sapiens [19] model further expand this to 308 keypoints, encompassing hands, feet, and facial features.

Despite their adaptability, kinematic models have notable limitations. By focusing solely on joint positions and limb orientations, they fail to capture texture, body width, and surface contours, which are crucial for representing the full complexity of the human figure. This restricts their use in scenarios requiring detailed morphology, such as precise human-robot interactions, highlighting the need for complementary methods [14], [15, pp. 3-4], [16, p. 5].

In the following, this work focuses on kinematic models, as they provide a straightforward approach to mapping depth data to specific keypoints compared to volumetric or planar models. Additionally, kinematic models have the lowest computational requirements among all body model types, making them the most practical choice for collaborative robotics in surgical settings, where real time motion planning could be required.

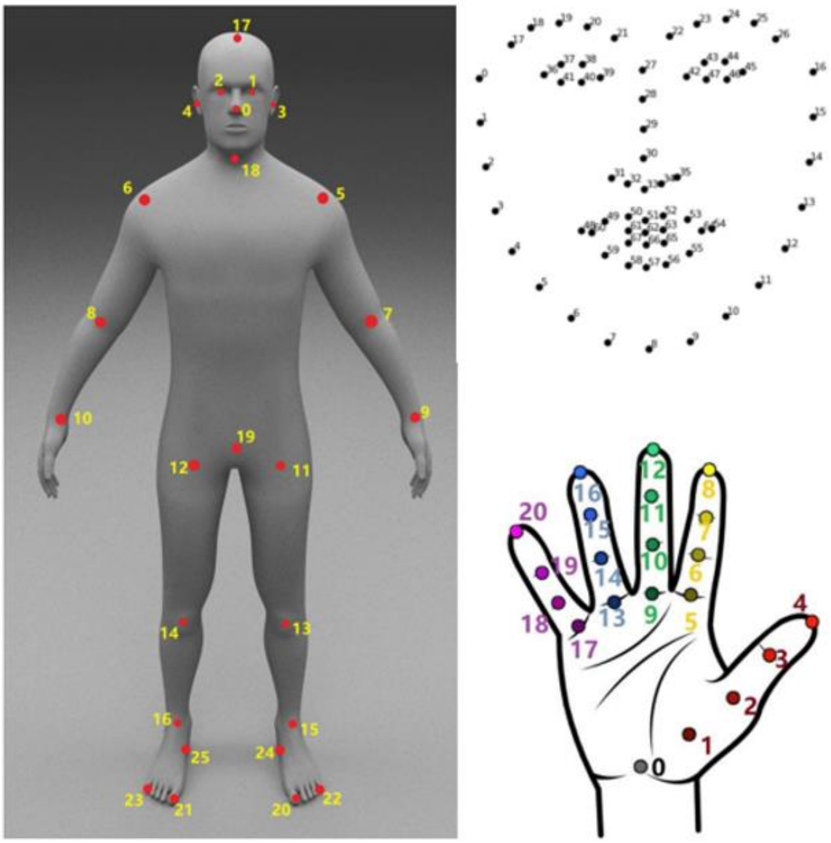


Figure 4: AlphaPose Keypoints [17]

2.2.3 2D Human Pose Estimation

In the literature, 2D HPE models are typically categorized into single-person and multi-person models, which are further subdivided based on their underlying methodologies. Furthermore, a distinction must be made between image-based and video-based approaches, further discussed in Chapter 2.2.3.3. Figure 5 provides an overview of the various 2D HPE approaches.

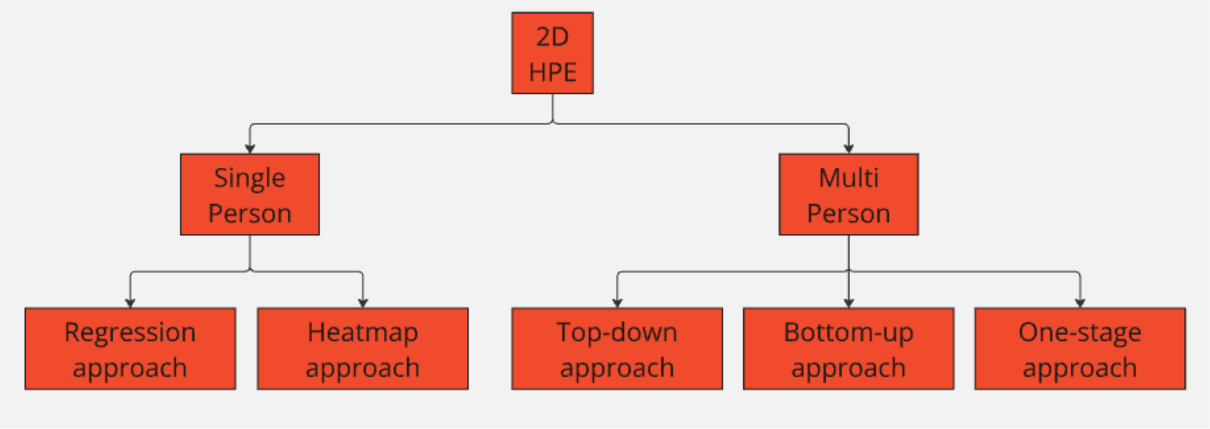


Figure 5: 2D HPE approach overview

2.2.3.1 Single-Person Pose Estimation

Image-based single-person pose estimation (ISPPE) focuses on detecting the pose of a specific individual within an image by locating keypoint positions based on provided position information, such as bounding boxes or initial pose estimations. Deep learning ISPPE approaches can be broadly categorized into two methods, regression-based approaches and heatmap-based approaches [9], [11], [13, p. 4].

Regression-based approach

The regression-based approach in single-person pose estimation, illustrated in Figure 6, derives keypoints directly from input images using an end-to-end framework. These methods aim to predict the precise locations of keypoints, such as shoulders, ankles, or wrists, by mapping feature maps to joint coordinates [12, pp. 3-4], [13, pp. 4-5], [14].

Feature maps, in this context, are intermediate representations produced by the neural network, capturing spatial and semantic information about the image. These maps serve as the foundation for identifying keypoints, with the final output often represented as a vector containing the X and Y coordinates of each keypoint [14].

Regression-based approaches face significant challenges in training stability and precision. Small deviations in keypoint predictions can lead to errors, as the models are sensitive to even slight inaccuracies. Additionally, the direct mapping of feature maps to joint coordinates increases the model's susceptibility to noise, complicating optimization [14]. Techniques such as multitask learning and compositional pose regression have been proposed to address these challenges by improving feature representation and generalization [13, pp. 4-5].

While regression methods have demonstrated considerable success in single-person pose estimation, their reliance on direct keypoint prediction makes them particularly challenging to train for complex poses. As such, robust optimization techniques remain essential for extending their applicability to more diverse scenarios [14].

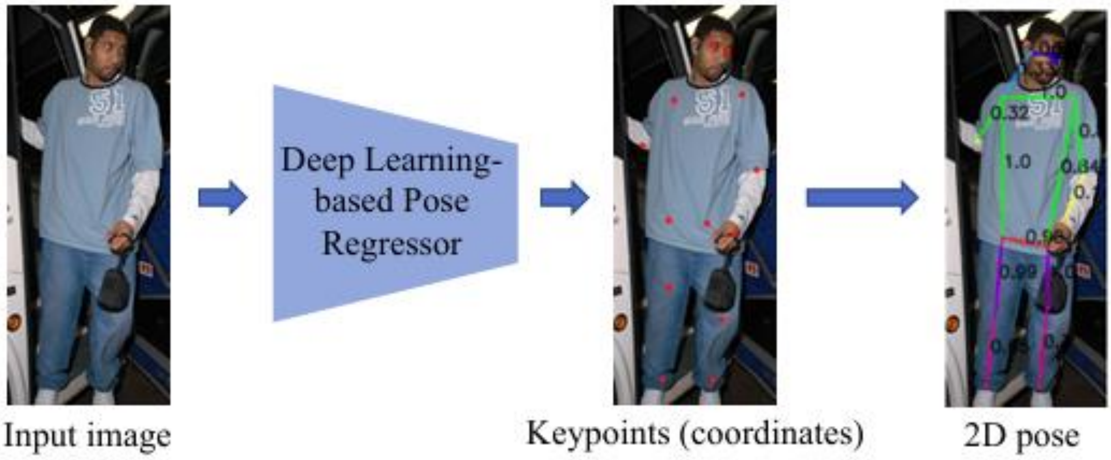


Figure 6: Regression-based approach for 2D single-person pose estimation [13, p. 4]

Heatmap-based approach

The heatmap-based approach, illustrated in Figure 7, is the dominant method for modern HPE tasks due to its ability to provide accurate and robust predictions of keypoint locations [12, p. 4], [13], [14]. Unlike regression-based methods, which directly predict keypoint coordinates as numerical values, heatmap-based methods estimate a probability distribution for each keypoint across the image. These distributions are visualized as heatmaps, where pixel intensities represent the probability of a keypoint being at a specific location. Compared to the joint coordinates predicted by regression-based approaches, heatmaps retain spatial context and enhance the stability and smoothness of the training process [13, pp. 5-6], [14].

One of the key advantages of heatmap-based methods is their ability to handle complex poses and partial occlusions effectively. By integrating body structure information, such as spatial and appearance consistency between joints, these models can improve generalization and accuracy. Additionally, adversarial learning frameworks have been explored to refine predictions further by distinguishing realistic poses from unrealistic ones [13, pp. 5-6].

Despite their advantages, Heatmap-based methods face two key challenges, decoding and encoding. The decoding problem refers to extracting keypoint positions from predicted heatmaps, typically solved by selecting the pixel with the highest intensity or averaging neighboring values. The encoding problem involves generating ground-truth heatmaps from keypoint coordinates by applying Gaussian kernels to represent the probability distribution for each keypoint [14].

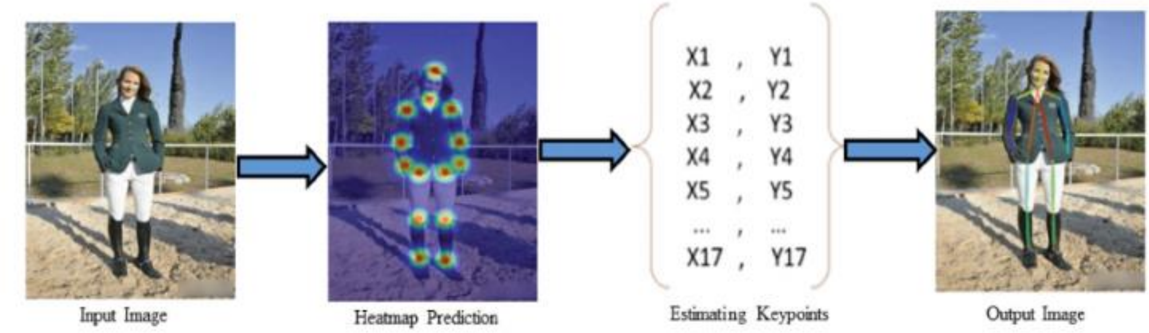


Figure 7: Heatmap-based approach for 2D single-person pose estimation [14]

2.2.3.2 Multi-Person Pose Estimation

Image-based multi-person pose estimation (IMPPE) involves identifying human figures and determining their keypoints within a scene incorporating regression or heatmap-based approaches. Compared to single-person pose estimation, it is significantly more complex due to challenges such as overlapping poses, variations in body shapes, determining the number of people, and correctly grouping keypoints to individuals [12, p. 5], [13, p. 7]. The three primary approaches to IMPPE are top-down, bottom-up, and one-stage methods [12, pp. 5-7].

Top-down approach

The top-down approach, illustrated in Figure 8, is a two-stage framework for multi-person pose estimation. First, a human body or object detector identifies individuals in the image and generates bounding boxes, which are rectangular regions enclosing the detected person. Then, a single-person pose estimator predicts the keypoints for each detected individual within their respective bounding box [12, p. 5], [13, pp. 7-8], [14].

While this method enables accurate keypoint predictions for individual poses, it requires sequential processing of each detected person, leading to longer computational times. Additionally, occlusions and crowded scenes can pose challenges, as errors in the detection stage directly affect the subsequent pose estimation [13, pp. 7-8], [14].

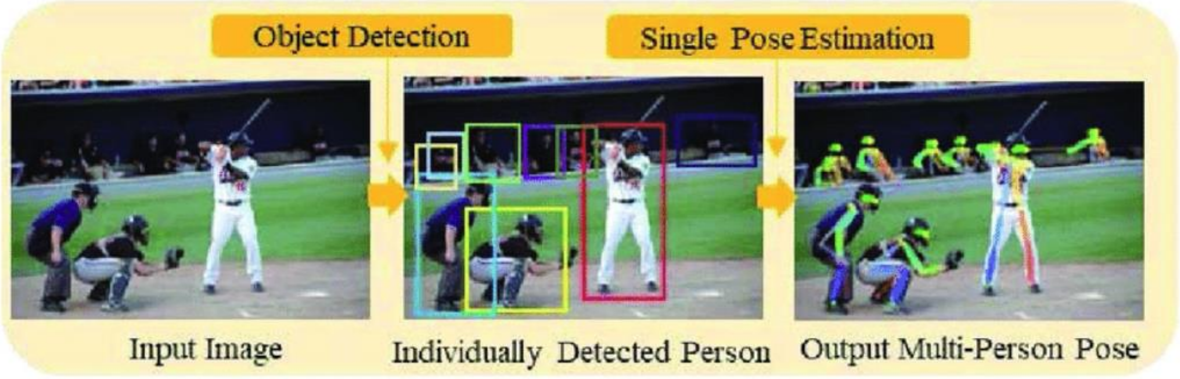


Figure 8: Top-down approach for 2D multi-person pose estimation [20, p. 3]

Bottom-up approach

The bottom-up approach, illustrated in Figure 9, is an alternative framework for multi-person pose estimation. Unlike the top-down method, it begins by detecting keypoints for all individuals in an image simultaneously, without prior identification of individual bounding boxes. These keypoints are then grouped to form skeletons for each person [12, p. 6], [13, p. 9], [14].

One of the primary strengths of the bottom-up approach is its efficiency. Since it processes all keypoints in parallel, it avoids the repeated pose estimation steps required by top-down methods, making it faster, particularly in scenes with many individuals. However, this efficiency comes with limitations. Bottom-up methods may struggle with accuracy, as they lack the ability to zoom into individual instances for detailed feature extraction. Issues such as scale variability, especially when dealing with small individuals in the image, can further reduce accuracy compared to top-down pipelines [12, p. 6], [14].

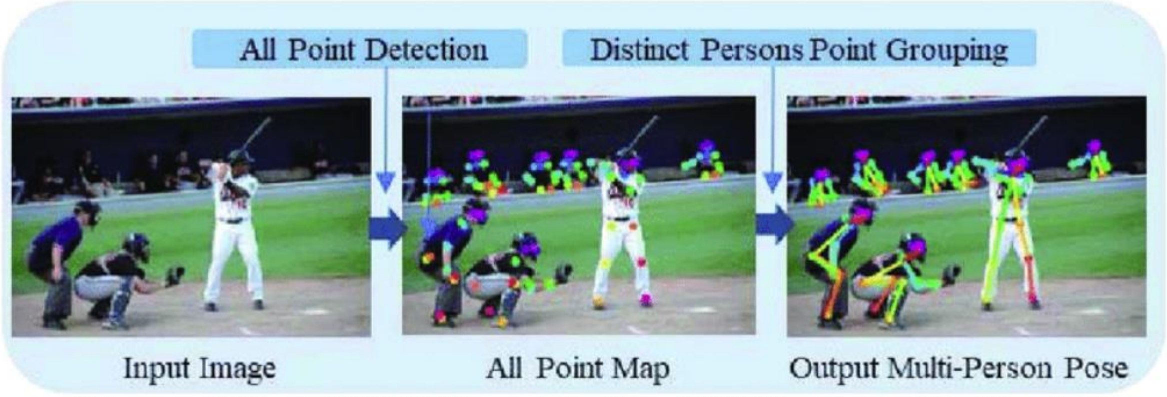


Figure 9: Bottom-up approach for monocular 2D multi-person pose estimation [20, p. 3]

One-stage approach

The one-stage approach is a unified framework for multi-person pose estimation that combines detection and pose estimation into a single pipeline. Unlike top-down and bottom-up methods, it directly predicts keypoints and their associations without requiring separate person detection or keypoint grouping stages [12, p. 7].

One of the key advantages of the one-stage approach is its simplicity and potential for faster processing, as it eliminates intermediate steps. However, this method faces challenges in achieving high precision, as the integration of all tasks into a single network can lead to trade-offs in accuracy and scalability. Additionally, some implementations rely on complex decoders to handle human and keypoint detection simultaneously, which may introduce additional computational demands [12].

2.2.3.3 Image-Based vs. Video-Based Pose Estimation

This section is based on [12, pp. 7-9].

It is important to highlight that approaches may differ when applied to videos. Image-based pose estimation operates on individual frames, estimating poses for one or multiple individuals based solely on the spatial structure of the image. In contrast, some video-based approaches extend these methods by incorporating temporal dynamics to ensure consistency across frames and analyze motion. This added dimension introduces complexity but enables tracking poses over time and provides a richer understanding of movement.

Single-person video pose estimation (VSPPE) focuses on estimating the pose of a single individual throughout a video. It can follow a frame-by-frame approach, where poses are estimated independently for each frame while incorporating temporal consistency, or a sample frame-based approach, which selects key frames to reduce computational complexity.

Multi-person video pose estimation (VMPPE) applies image-based multi-person methods to video data, often using frame-by-frame processing. To improve tracking, these methods leverage motion dynamics to link poses across frames, addressing issues such as occlusions and redundancies. Both top-down and bottom-up approaches are adapted for video-based scenarios.

While video-based methods enable motion analysis and pose tracking, they introduce challenges such as handling motion blur, ensuring temporal consistency, and addressing higher computational demands.

2.2.4 3D Human Pose Estimation

3D HPE aims to predict the three-dimensional coordinates (x, y, z) of body keypoints from images or videos, extending the spatial understanding achieved in 2D HPE [14]. While significant advancements have been made in 2D HPE, estimating 3D poses introduces unique challenges, such as depth ambiguity through the ill-posed nature of monocular 2D-to-3D projection, where different 3D depths and joint configurations may result in the same 2D projection, and the scarcity of high-quality, annotated 3D datasets [12, p. 9], [13, p. 10], [14]. Most methods rely on monocular RGB images or videos, however some approaches mitigate these challenges by fusing complementary data sources, such as IMUs, depths sensors or other [13, p. 10], [14].

This section gives an introduction into the different approaches for monocular HPE as well as briefly exploring other data sources.

2.2.4.1 Monocular 3D Human Pose Estimation

Methods for monocular 3D HPE can be categorized into top-down, bottom-up, one-stage, and two-stage approaches, with variations across single-person and multi-person tasks, as well as image-based and video-based settings [12, pp. 9-13]. Figure 10 provides a visual overview of the various monocular 3D HPE approaches discussed in this chapter.

The advantages, disadvantages, and underlying frameworks of top-down, bottom-up, and one-stage approaches in 3D HPE closely align with those of their counterparts in 2D HPE for multi-person pose estimation (MPPE), discussed in Chapter 2.2.3.2.

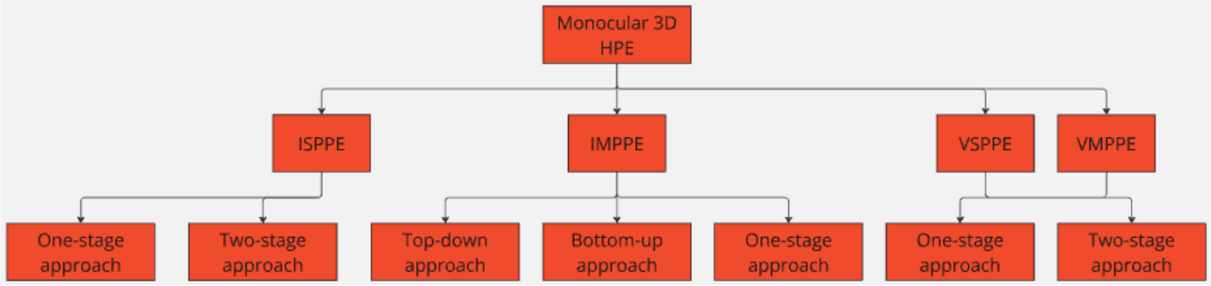


Figure 10: Monocular 3D HPE approach overview

Top-down approach

Involves detecting individuals first, followed by estimating their 3D poses within each detected bounding box. While highly accurate for isolated individuals, this approach is computationally expensive and struggles in crowded scenes or when capturing interactions between individuals. The additional depth information required for 3D HPE further complicates this method compared to its 2D equivalent [12, p. 11], [13, p. 15]. The process of the top-down approach is visually depicted in Figure 11.

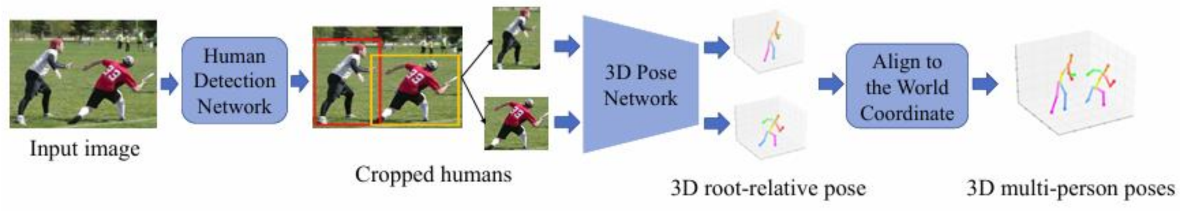


Figure 11: Top-down approach for monocular 3D multi-person pose estimation [13, p. 15]

Bottom-up approach

Begins by detecting all keypoints across the entire image, followed by grouping them into individual skeletons using depth information. This approach is more efficient than top-down methods, as it avoids repeated detection and estimation for each person. However, it is sensitive to variations in scale and often requires additional steps for associating keypoints accurately [12, p. 11], [13, p. 16]. Figure 12 illustrates the framework of the bottom-up approach.

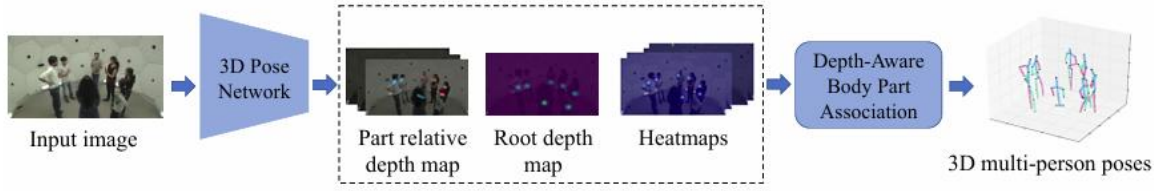


Figure 12: Bottom-up approach for monocular 3D multi-person pose estimation [13, p. 15]

One-stage approach

Directly predicts 3D keypoints from the input image or video without intermediate steps. This end-to-end framework simplifies the pipeline but is computationally intensive and may lack the refinement achievable with multi-stage processes [12, pp. 12-13], [13, p. 11]. As illustrated in Figure 13, the method bypasses intermediate 2D pose estimations, enabling a streamlined yet resource-heavy workflow.

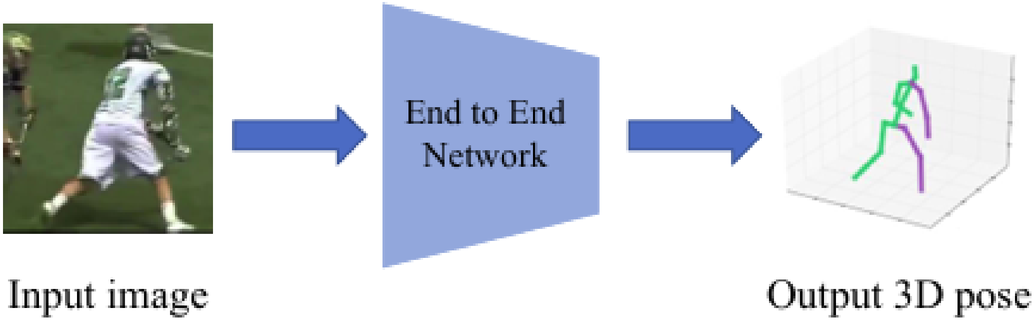


Figure 13: One-stage approach for single-person pose estimation [13, p. 11]

Two-stage approach

Commonly referred to as lifting, this method first estimates a 2D pose, utilizing established 2D HPE techniques, before transforming the 2D pose into a 3D representation through regression or other algorithms. This approach leverages the robustness of reliable 2D pose estimations to achieve higher accuracy in 3D reconstructions. It is particularly effective when applied to tasks with well-annotated 2D data [12, pp. 12-13], [13, pp. 11-12]. The process is visually depicted in Figure 14.

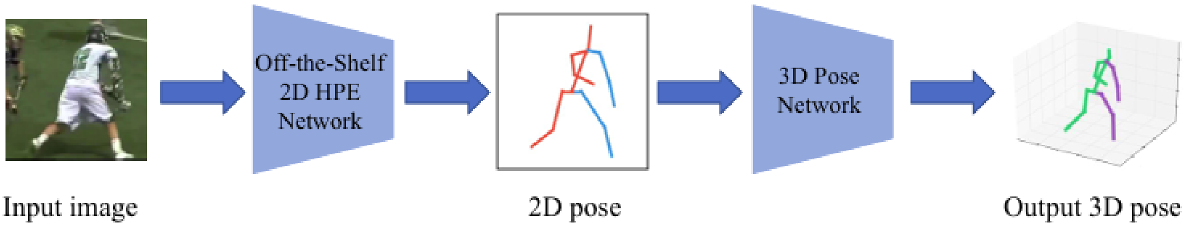


Figure 14: Two-stage approach for single-person pose estimation [13, p. 11]

2.2.4.2 3D Human Pose Estimation from Other Sources

Monocular RGB cameras are the most commonly used devices for 3D HPE [13, p. 17], [14], various alternative sensors can be employed to address specific challenges such as depth ambiguity, occlusions, and limited spatial resolution. These sensors include depth and point cloud sensors, inertial measurement units, RF devices, and other data sources [13, p. 18].

Multi-view

Multi-view 3D HPE, in contrast to the more widely used single-view systems, employ multiple monocular cameras to capture the target from different perspectives, improving the estimation of depth and increasing accuracy, particularly in cases where body parts are occluded in one view but visible in another [13, pp. 16-17], [14].

This approach requires resolving the association of corresponding points across views, often using methods like epipolar geometry or consistency constraints. Although multi-view methods achieve higher precision by leveraging spatial information from multiple perspectives, they come with increased computational costs and complexity in system setup, particularly for multi-person scenarios. Moreover, adapting to changes in camera configurations often requires retraining models, which can be resource intensive [13, p. 17].

Despite their advantages, the necessity of multiple synchronized cameras makes multi-view methods less practical compared to single-view monocular systems in many real-world applications. However, they remain a valuable alternative for tasks requiring high accuracy and resilience to occlusion [13, p. 17], [14].

Depth sensors and point clouds

Depth sensors, such as RGB-D cameras, have become increasingly important in 3D HPE due to their affordability and ability to address depth ambiguity challenges effectively [13, pp. 17-18]. By capturing both color (RGB) and depth (D) information, these sensors provide a comprehensive representation of the scene, significantly enhancing spatial understanding and improving the accuracy of pose estimation [21]. The depth information is typically provided as a depth map, which usually is a grayscale image, representing the distance between the camera and objects in the scene. Each pixel in the depth map encodes depth information instead of RGB information [22], [23], [24].

Point cloud data, commonly generated by LiDAR sensors, offers highly detailed spatial information by measuring precise distances to surfaces within the environment. LiDAR is particularly valued for its invariance to lighting conditions and its ability to deliver accurate depth sensing without interference from light, making it a suitable tool for 3D HPE tasks [25].

Inertial measurement units

Inertial measurement units, devices to track orientation and acceleration, can be effective tools for reconstructing 3D poses without being affected by clothing or occlusions. However, these approaches require the devices to be physically attached to the person being tracked, which imposes practical limitations. Additionally, IMU-based methods are prone to drifting over time, which can impact long-term accuracy [13, p. 18]. As shown in Figure 15, the OpenSenseRT system exemplifies how IMUs, marked with an orientational frame, are utilized to estimate 3D poses.

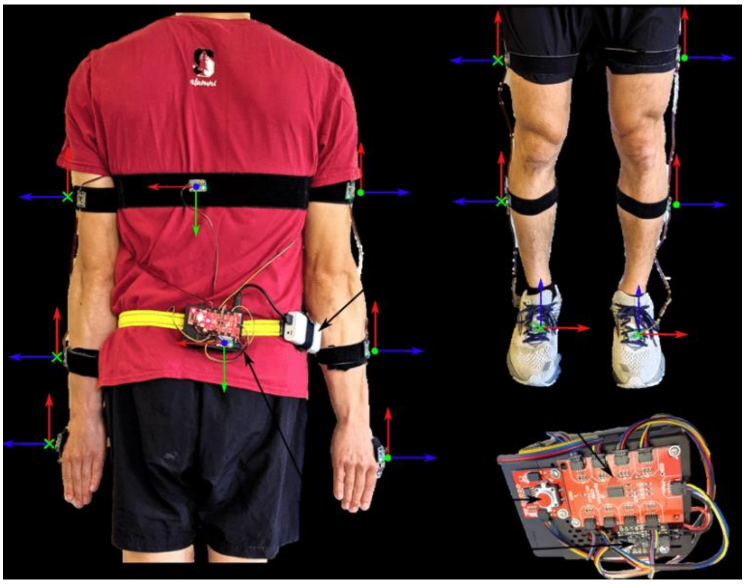


Figure 15: Components of the OpenSenseRT System [26, p. 1]

Radio frequency devices

Radio-frequency-based sensing systems can estimate 3D poses by detecting signals that pass-through walls or reflect off human bodies, offering advantages such as non-visual data collection and privacy preservation. However, RF signals provide lower spatial resolution compared to visual data [13, p. 18].

Other sources

While there is a wide range of other sensor fusion possibilities some examples of other methods include utilizing Non-Line-of-Sight (NLOS) imaging systems, egocentric fisheye cameras, or multiple Autonomous Micro Aerial Vehicles (MAVs) for pose estimation [13, p. 18].

2.2.5 Human Pose Estimation Summary

2D SPPE appears to be considered a mature area, with well-established methods that seem to provide accurate and reliable keypoint detection. These approaches perform well under varied conditions and are effective for most single-person scenarios. While notable progress has been made in 2D multi-person pose estimation, challenges such as handling occlusions and crowded scenes persist. Ongoing research aims to address these issues, and the methods appear to be becoming increasingly reliable for a range of practical applications.

3D pose estimation, particularly monocular approaches, appear to be less developed in comparison. Methods often rely on relative pose estimation to infer spatial relationships between joints but may struggle with depth ambiguity and consistent accuracy. Although advancements have been made, limitations in monocular data seem to continue impacting precision. Metric 3D pose estimation, which requires absolute joint positions in real-world coordinates, seems to face additional challenges due to the need for robust datasets, complex depth estimation, and computational constraints.

2.3 Used Hardware and Software Frameworks

This section provides a detailed overview of the hardware components and software frameworks employed in this work.

2.3.1 UR3e

Collaborative robots are industrial robots designed with advanced safety features to facilitate safe and efficient collaboration with humans. Rather than replacing personnel, cobots are intended to complement human workflows by automating specific tasks. Their safety-focused design includes features such as rounded edges, collision sensors, and force limitations, enabling them to operate safely in shared spaces [27, pp. 222-223]. This ability to safely interact with humans and operate in constrained environments makes cobots an ideal foundation for RSNs.

The UR3e from Universal Robots, shown in Figure 16, was selected for this work due to its suitability for OR conditions. It is a six Degrees of Freedom (DoF) manipulator and the most compact cobot in Universal Robots portfolio. According to the manufacturer, the UR3e is particularly well-suited for automation in confined spaces, thanks to its compact design [28]. With a reach of 500 mm, a footprint diameter of 128 mm, and an IP54 rating, the UR3e is well-suited to the spatial constraints and potential exposure to fluids in the OR. Furthermore, it is capable of handling surgical tools that weigh significantly less than its maximum payload of 3 kg [28], [29].

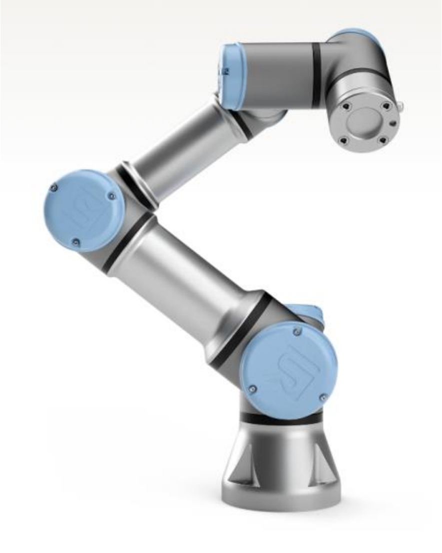


Figure 16: Universal Robot UR3e [28]

2.3.2 Intel RealSense D455

The Intel RealSense D455 depth camera was selected for this work due to its ability to capture both standard RGB images via a dedicated RGB sensor and depth data through a stereo imaging system comprising a left and right imager. Additionally, the integrated infrared (IR) projector enhances depth perception under varying lighting conditions, making the camera highly adaptable to diverse environments [30]. Figure 17 displays the depth camera, including the left and right imagers, the dedicated RGB sensor and the IR projector. For further information on depth sensors, see Chapter 2.2.4.2.

The D455 offers an operational range of 0.6 to 6 meters, which is well-suited for monitoring the workspace of surgical personnel and the RSN. Its depth measurement error remains below 2% at a distance of 4 meters [31], ensuring reliable depth measurements.

Furthermore, Intel provides the RealSense SDK 2.0, a versatile Software Development Kit (SDK) that supports commonly used programming languages in robotics and integrates with various frameworks, including ROS2, as detailed in the subsequent section. Additionally, the SDK is compatible with multiple operating systems, enabling broad adaptability for development environments [32].

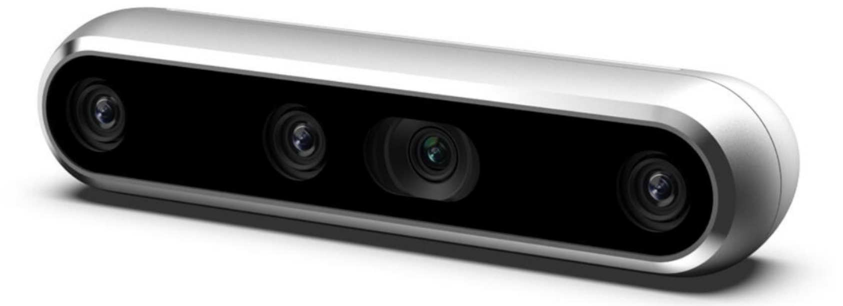


Figure 17: Intel RealSense D455 Depth Camera [31]

2.3.3 Robot Operating System

The Robot Operating System (ROS) [33] is an open-source framework that provides software libraries and tools to support the development of robotic applications. Initially developed by Willow Garage and now maintained by the Open Source Robotics Foundation (OSRF), ROS is a cornerstone of robotics research and academia. The release of ROS2 in 2018 addressed several limitations of its predecessor, making it the predominant choice in academic settings [33], [34, pp. 3-4].

ROS2 facilitates communication between heterogeneous robotic components through its modular, scalable architecture, based on nodes, topics and messages [35]. Enabling the integration of sensors, and other external systems, this functionality is essential for addressing the limitations of commercially available cobots and supporting advanced tasks such as motion control and system coordination [34].

A vital package utilized in this work is MoveIt2 [36], a motion planning framework for ROS2. MoveIt2 provides tools for trajectory planning, collision avoidance, and robotic manipulation. Its modular design incorporates state-of-the-art libraries such as the Open Motion Planning Library (OMPL) for motion planning, implementing algorithms like Rapidly-Exploring Random Tree (RRT) and Probabilistic Roadmap Method (PRM) [1, pp. 22-25], [34, p. 17]. Collision detection is typically managed by the Flexible Collision Library (FCL), ensuring effective planning in constrained and dynamic environments [34, p. 17].

Another key package of this work is the tf2 (transform) library, a core component of ROS2, providing tools to track multiple coordinate frames over time, enabling transformations between frames. By maintaining a buffered tree structure of frames, tf2 allows queries for transformations at specific points in time. Additionally, tf2 facilitates distributed robotic systems by broadcasting and listening to frame transformations across nodes. Transform broadcasters share relative poses of frames, while listeners receive and buffer this information for queries [37], [38].

MoveIt2's graphical interface via the ROS visualizer (RViz) [39] enables visualization, simulation and integration of the planning scene, allowing for validation of planned trajectories prior to execution. For instance, it can generate trajectories within an environment reconstructed from a depth sensor's point cloud [34, p. 17], making it particularly useful in applications requiring precise interaction with external surroundings, such as surgical robotics.

In this work, ROS2 with MoveIt2 and tf2 are integral to establishing communication and coordination between the Intel RealSense D455 sensor and the UR3e cobot. ROS2 ensures seamless and reliable data exchange between nodes, while its package MoveIt2 provides advanced trajectory planning capabilities essential for executing collaborative tasks in the OR. These capabilities are underpinned by the frame transformations broadcasted and managed through the tf2 package, which ensures accurate spatial alignment of sensor data with the robot's workspace.

2.3.4 Monocular Depth Estimation

While the primary focus of this work is on HPE models, monocular depth estimation (MDE) is explored as an alternative approach. This section offers a concise introduction to monocular depth estimation models.

MDE refers to the prediction of depth information from a single RGB image, a technique central to applications in computer vision such as 3D reconstruction, robotics, and Simultaneous Localization and Mapping (SLAM). Traditional methods typically depend on stereo images or specialized hardware like LiDAR, however, MDE presents a cost-effective and versatile alternative by leveraging deep learning techniques to infer depth from monocular cues [40, pp. 137-138].

Such models typically use Convolutional Neural Networks (CNN) to extract spatial and semantic features from an input image [41, p. 1]. These features are then processed by regression layers or decoders to predict depth values for each pixel in the image, producing a depth map [24, p. 6], which represents the relative distances of objects in the scene, exemplified in Figure 18. The model learns depth perception by training on datasets containing images paired with ground-truth depth maps [41, p. 3].

Recent advancements, such as the Depth Anything V2 model, have significantly improved the field by offering robust and efficient solutions. Depth Anything V2 is trained on a large dataset consisting of 595,000 synthetic labeled images and more than 62 million real, unlabeled images [42, p. 6]. This enables the model to produce fine-grained and reliable depth predictions across a wide range of scenarios. Furthermore, its universal training approach on heterogeneous data enhances its generalization capabilities, making it highly adaptable to diverse applications [42, pp. 1-2].

Despite these advancements, a critical limitation of MDE is its inability to provide absolute depth measurements in real-world units without additional information. This limitation arises due to the inherent ambiguity in projecting a 3D scene onto a 2D plane, similar to challenges faced in 3D HPE, as discussed in the introduction to Chapter 2.2.4. While efforts have been made to address this issue and bridge the gap between relative and absolute depth maps [43], MDE models continue to provide solely relative depth estimations [44].

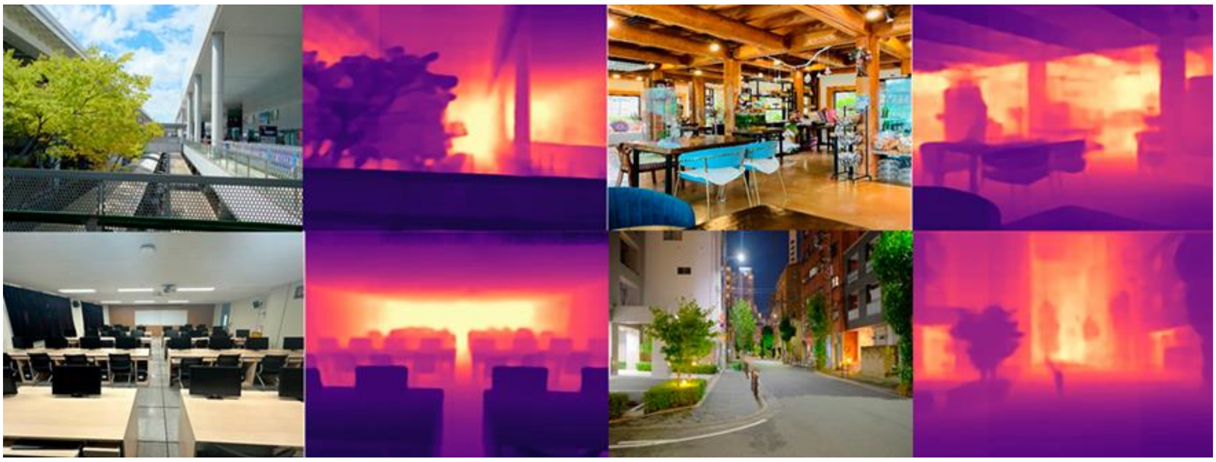


Figure 18: Estimated depth maps from real-life pictures [45, p. 15]

3 Experiments

This chapter provides an overview of the HPE experiments conducted. Selecting suitable HPE models involved extensive research prioritizing credible scientific papers with high citation counts as well as models developed by reputable companies such as Meta and Google. The selection process focused on models that have been actively maintained within the past five years to ensure relevance and the inclusion of established approaches.

Figure 19 summarizes the considered models for testing, highlighting their earliest and latest release dates, suitability for 2D or 3D HPE, single-person or multi-person tasks, input type, number of keypoints, and computational approach. Additionally, the figure includes references to the respective sources. Fields marked with "n/a" indicate that the information was not available at the time of conducting the research.

Model	Earliest release	Latest release	2D/ 3D; SP/MP	Input	No. of keypoints	Approach	Reference
VideoPose3D	2018 Nov	2020 Aug	3D	Monocular	17	Lifting	[46]
OpenPose	2017 Oct, v1.0.0	2020 Nov, v1.7.0	2D MP	Monocular	135	Bottom-up	[18]
AlphaPose	2018 Sep, v0.2.0	2022 Nov, v0.6.0	3D MP	Monocular	135	Top-down	[17]
OpenPifPaf	2019 Mar, v0.2.0	2023 Feb, v0.13.11	2D MP	Monocular	133	Bottom-up	[47]
Detectron2	2020 Feb, v0.1	2021 Nov, v0.6	2D MP; 3D MP	Monocular	17	Top-down	[48]
Monoloco	2019 Jun, v0.1	2021 Sep, v0.7.4.6	3D MP	Monocular, RGB-D	17	Lifting	[49]
ViTPose	2022 Apr	2023 Jan	2D	Monocular	17	Top-down	[50]
YOLOv8	2024 Jul, v8.0.4	2024 Oct, v8.3.4	2D MP	Monocular	17	One Stage	[51]
MMPose	2020 Sep, v0.6	2024 Jul, v1.3.2	2D MP; 3D MP	Monocular	133	Top-down	[52]
Sapiens	2024 Aug	2024 Aug	2D MP	Monocular	308	Top-down	[19]
Blazepose	2021	2023	3D SP	Monocular	33	HeatMap	[53]
PPoseur	2022	2023	2D	Monocular	17	One-stage	[54]
PAFUSE	2024 Jul, v0.1.0	2024 Jul, v0.1.0	3D MP	Multiview	133	Lifting	[55]
MeTRAbs	2020 Aug	2023 Aug	3D	Monocular, Multiview	17	Top-down	[56]
RTMPose	Mar 2023	2023 Dec	3D	Monocular	133	Lifting	[57]
MediaPipe	2019 Jul, 0.5.0	2024 Aug, v.0.10.15	2D	Monocular	33	Top-down	[58]

Figure 19: HPE Models considered for detailed evaluation

Based on the evaluated models, two primary approaches were selected for 3D HPE. The first involves utilizing a pre-trained 3D monocular model capable of producing metric depth estimations, while the second combines a 2D HPE model with depth data from an RGB-D camera to manually lift 3D coordinates. For further information on lifting techniques, refer to Chapter 2.2.4.1, and for details on RGB-D cameras, see Chapter 2.2.4.2. Furthermore, the potential of employing an MDE model, see

Chapter 2.3.4, as an alternative to the depth map provided by the RGB-D camera for lifting 2D HPE into 3D was considered.

3.1 Experiment Environment and Setup

The test environment, depicted in Figure 20, was used to evaluate the depth estimation performance of the selected models. The red annotations indicate the ground truth depth values in millimeters measured from the RGB-D camera to specific objects within the scene. These values serve as a reference for validating the model outputs.

The wrist joint was selected as the primary evaluation keypoint for the tests. This choice was made to enable flexible positioning of the hand in front of various objects with known depths, providing a straightforward way to verify if the estimated depth aligns with the ground truth. However, it should be noted that the thickness of the wrist of 50 mm must be considered when comparing the depth values. The decision to use the wrist instead of other keypoints, such as the head or chest, was influenced by its ability to interact dynamically with different parts of the scene, and its smaller surface area, which poses a greater challenge for depth estimation.

The setup was not optimized for ideal lighting conditions and was designed as a quick and practical testing solution rather than a fully controlled experimental environment. This approach enabled efficient evaluation of the models' performance under non-ideal conditions. To improve readability in the subsequent sections, only the four depth values shown in Figure 20 are presented per test.

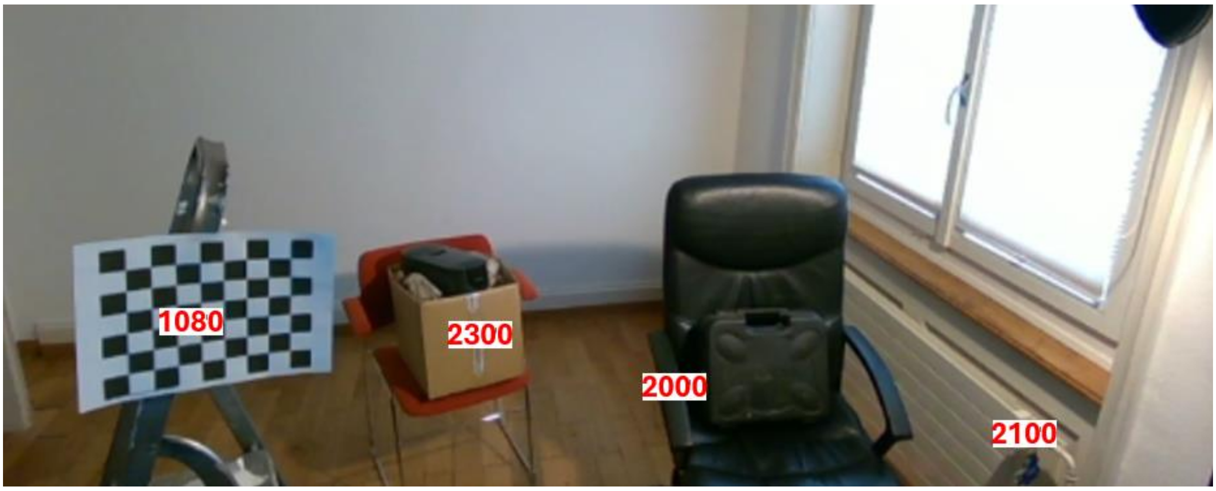


Figure 20: Test environment with selected ground truth depth values annotated in millimeters

3.2 3D Model with Metric Depth Estimation

For the pre-trained 3D monocular model capable of producing metric depth estimations, the Metric-Scale Truncation-Robust Heatmaps for Absolute 3D HPE (MeTRAbs) [56] model was utilized. MeTRAbs was selected due to its distinction as the winner of the 3D Poses in the Wild Challenge [59]. Additionally, its implementation was straightforward, making it a practical choice for this work.

The primary limitation of monocular 3D HPE, as outlined in Chapter 2.2.4, lies in the inherent ambiguity and dimensional loss that occurs when projecting 3D space onto a 2D plane. This limitation results in models excelling at estimating relative poses, that is, the spatial relationships between keypoints but performing poorly in delivering accurate metric depth estimations. However, with a calibrated camera setup, MeTRAbs can natively provide absolute depth estimations.

The model consistently estimates a full-body skeleton, even when parts of the body are occluded or not in frame, as illustrated in Figure 21. However, its performance is most accurate when the full body pose is entirely visible.

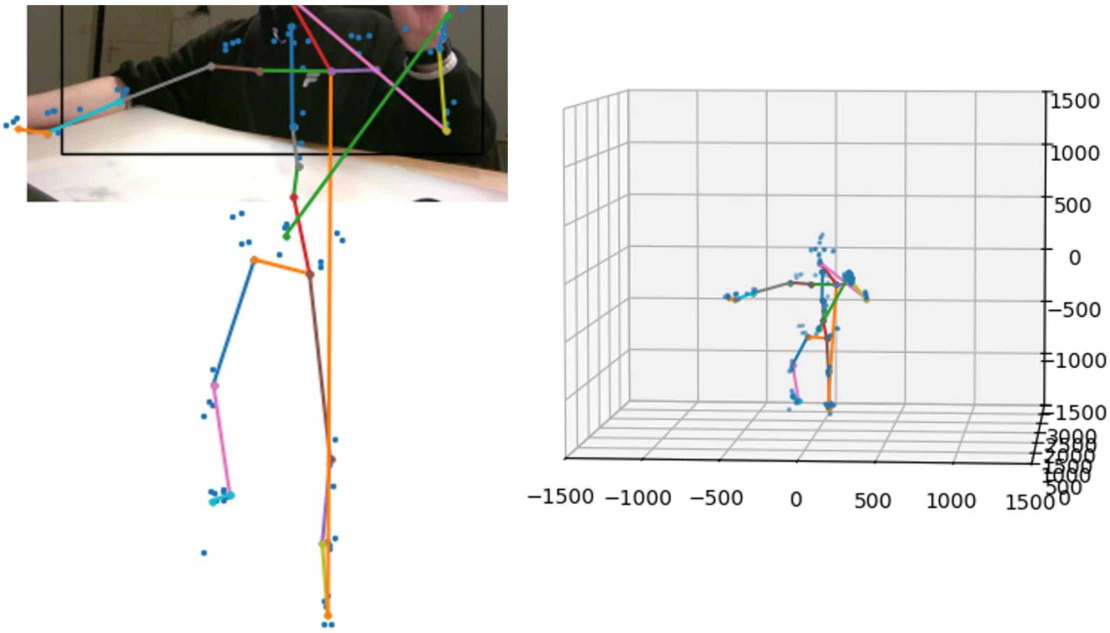


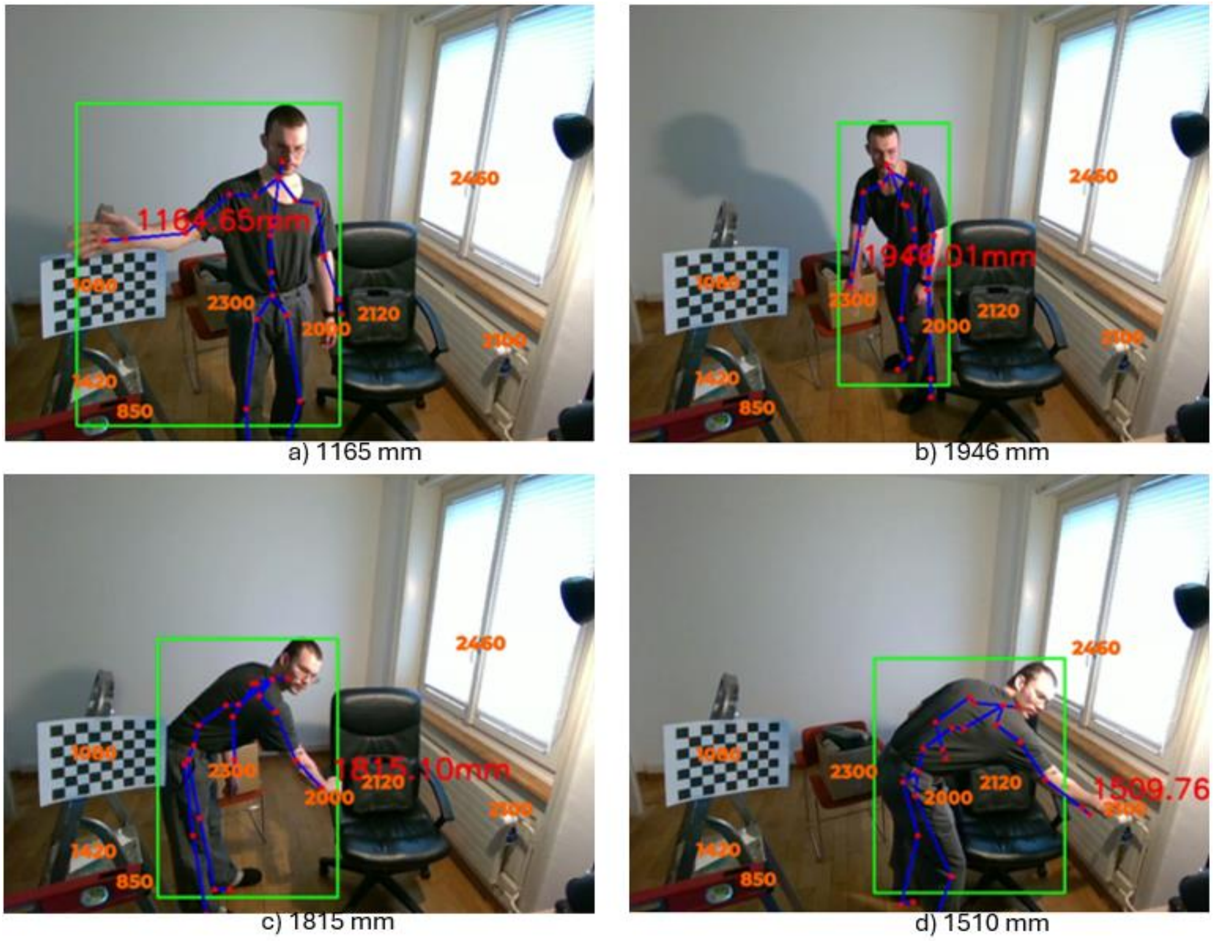
Figure 21: MeTRAbs - Partial body visible in image

For simpler body poses, as shown in Figure 22 a) – c), the depth estimation achieved an accuracy with an error margin of up to 300 mm. In contrast, more complex poses with significant self-occlusion, as illustrated in Figure 22 d), exhibited errors of up to 540 mm. Furthermore, even during static poses, the depth estimation demonstrated fluctuations, with values varying by 200–300 mm over a one-second period.

A notable observation is that the depth estimation was influenced by the overall body configuration. As shown in Figure 23, the left image depicts a closer body configuration, while the right image illustrates a slightly more distant body position, resulting in a depth discrepancy of approximately 140 mm for the static wrist.

Figure 24 highlights the model's challenges with occlusion. Despite the arm being visible, the model was unable to accurately estimate the arm pose and incorrectly placed the wrist and hand keypoints near the body.

The model is unlikely to achieve an error margin smaller than 500 mm in a surgical environment due to the many occlusions caused by the cluttered nature of ORs, and surgical workwear, making it unsuitable for applications requiring accurate metric values or precise spatial understanding from monocular videos. But further testing in a real OR setting is necessary to evaluate its performance under these conditions. However, MeTRAbs could be able to perform within an error margin smaller than 300 mm in optimal settings where the whole-body pose is visible, potentially being beneficial in scenarios where approximate data is adequate, such as initiating a cobot shutdown when entering a large workspace, where such a margin of error is acceptable. Also, it must be stated that MeTRAbs can be utilized within a multi-view setup, which could greatly increase its accuracy. Future research should explore the possibilities of MeTRAbs with alternative setups, such as a multi-view or a 360° camera, provided that the camera's parameters can be accurately determined.



a) 1165 mm

b) 1946 mm

c) 1815 mm

d) 1510 mm

Figure 22: MeTRAbs - Depth estimation of the wrist keypoint for various body poses.

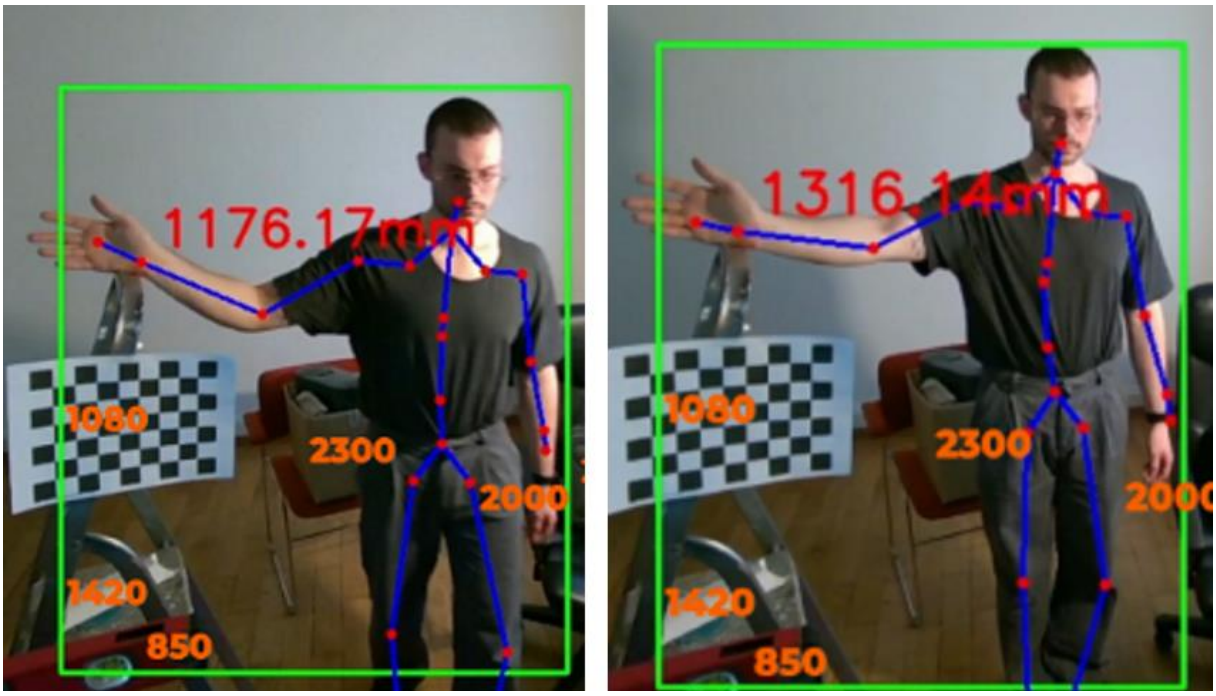


Figure 23: MeTRAbs - Impact of varying body poses on wrist depth estimation.

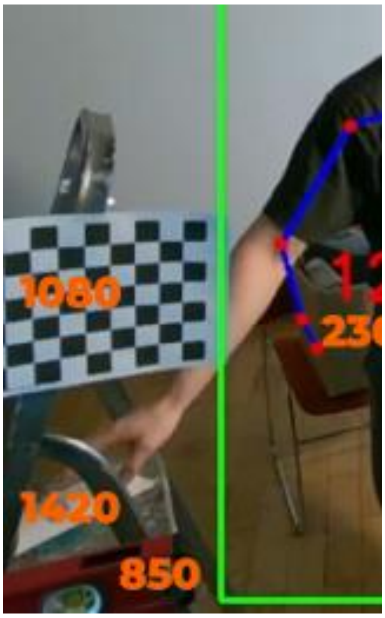


Figure 24: MeTRAbs - Challenges posed by occlusion

3.3 2D Model with RGB-D Data

For the 2D HPE with depth from the RGB-D camera, the eighth version of YOLO (You Only Look Once) [51] was selected due to its widespread usage, comprehensive documentation, and its characteristics as a fast and efficient one-stage model. The choice is further reinforced by the availability of numerous tutorials, resources, and ease of implementation. Notably, several models are built upon different iterations of YOLO, such as MeTRAbs, which utilizes YOLOv3.

As YOLOv8 does not estimate depth, this test primarily evaluates the performance of the 2D HPE model in combination with the depth map.

The model does not consistently estimate a full skeleton when body parts are occluded, as seen in Figure 25,. Additionally, the 2D HPE model demonstrated high accuracy in estimating keypoints, enabling the extraction of plausible depth data, as shown in Figure 26 a) – d). YOLOv8's ability to estimate key body regions under occlusions suggests its suitability for occluded environments like ORs. However, this requires validation through realistic testing. Since this approach relies on the well-established principle of RGB-D cameras, its depth measurements are assumed to be inherently reliable.

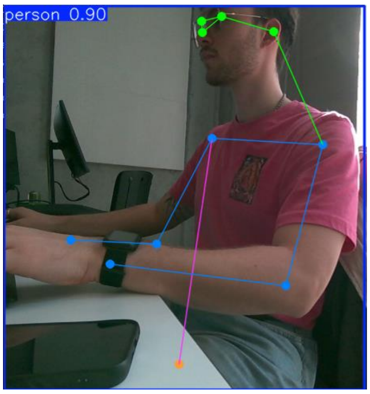
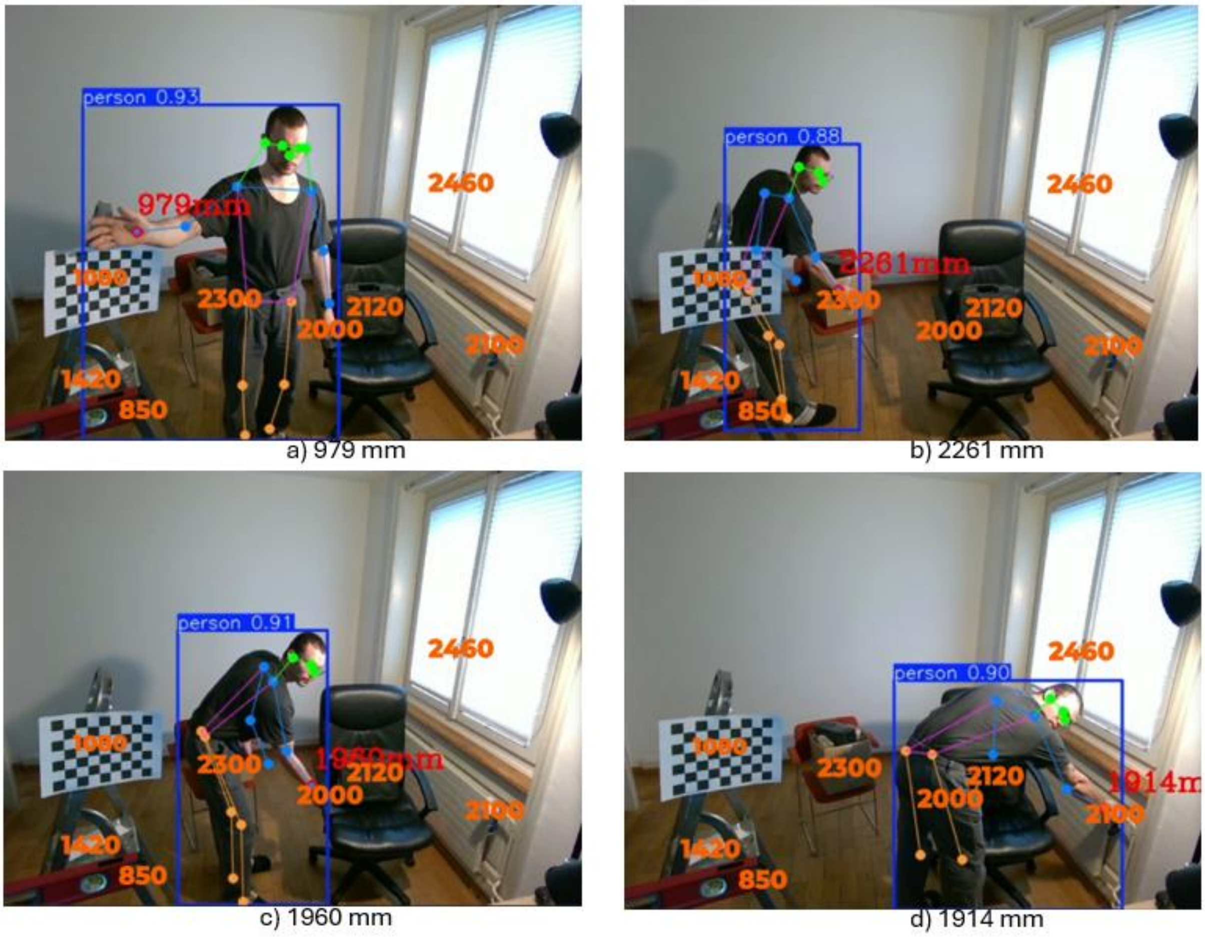


Figure 25: YOLOv8 - Partial body in image



a) 979 mm

b) 2261 mm

c) 1960 mm

d) 1914 mm

Figure 26: YOLOv8 - Depth estimation of the wrist keypoint for various body poses.

3.4 Exploring Monocular Depth Estimation as an RGB-D Alternative

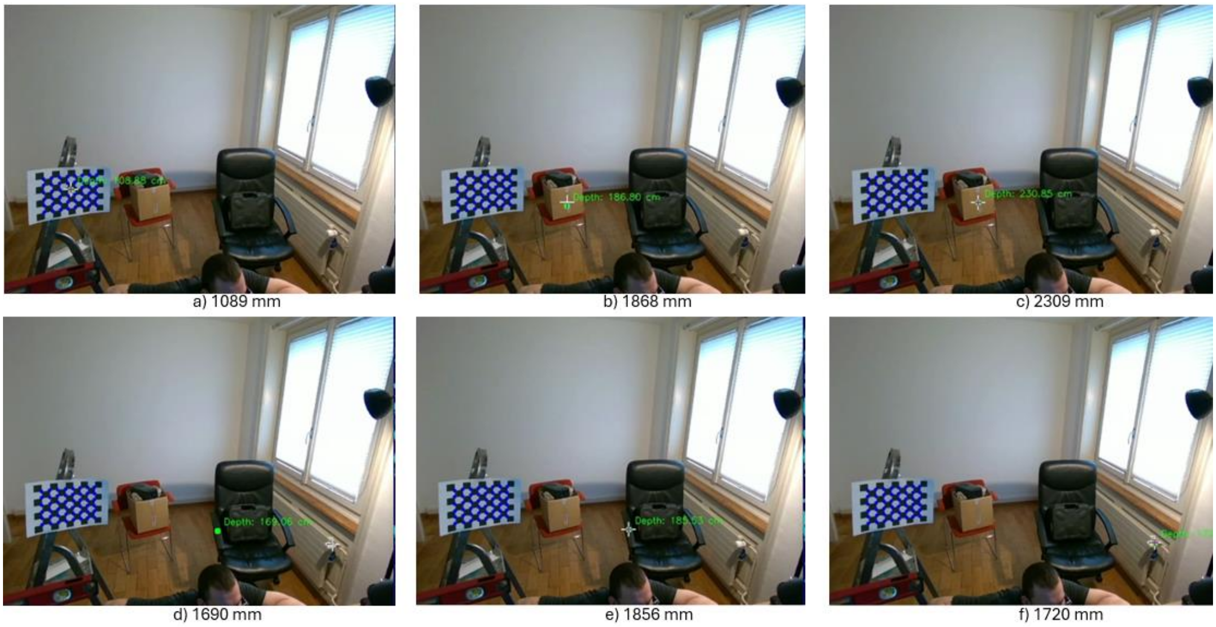
To evaluate whether an MDE model could replace an RGB-D camera for lifting 2D HPE into 3D coordinates, Depth Anything V2 was selected. Depth Anything V1 has been widely used, and V2 was introduced during the course of this work.

As described in Chapter 2.3.4, the primary issue with MDE models, is that their output is limited to relative depth data, interpreting the scene in relation to itself. Consequently, the depth values were not reliable by default and exhibited deviations of over 500 mm in both directions. To address this, the author implemented a scaling approach by providing a known depth value and using it as a scaling factor to calibrate the scene. Several strategies were tested, including defining multiple known points in the room (such as the wall, table, and window frame), providing a single known depth point, and using the surface area of a checkerboard pattern as a reference.

Figure 27 illustrates the results of the last approach. The RGB images show the query point indicated by a green crosshair with the estimated depth annotated in green at the right side of the crosshair. Despite the scaling efforts using the checkerboard reference, significant inaccuracies persisted, with deviations occasionally exceeding 500 mm. Additionally, b) and c), as well as d) and e) show the variations in depth values for the same object, measured within a static scene over a span of 2 seconds.

The most critical issue was that the relative nature of the model caused the overall scene configuration to influence the depth at a given query point. As shown in Figure 28, the position of the person in the scene directly affected the estimated depth at the query point. When the person was standing, they were located between the known depth of the checkerboard and the query point, which led the model to estimate the depth of the query point as being closer to the checkerboard. When the person was seated, the visual connection between the query point and the known depth was lost, resulting in a discrepancy between the estimated depths of the seated and standing person, exceeding 850 mm.

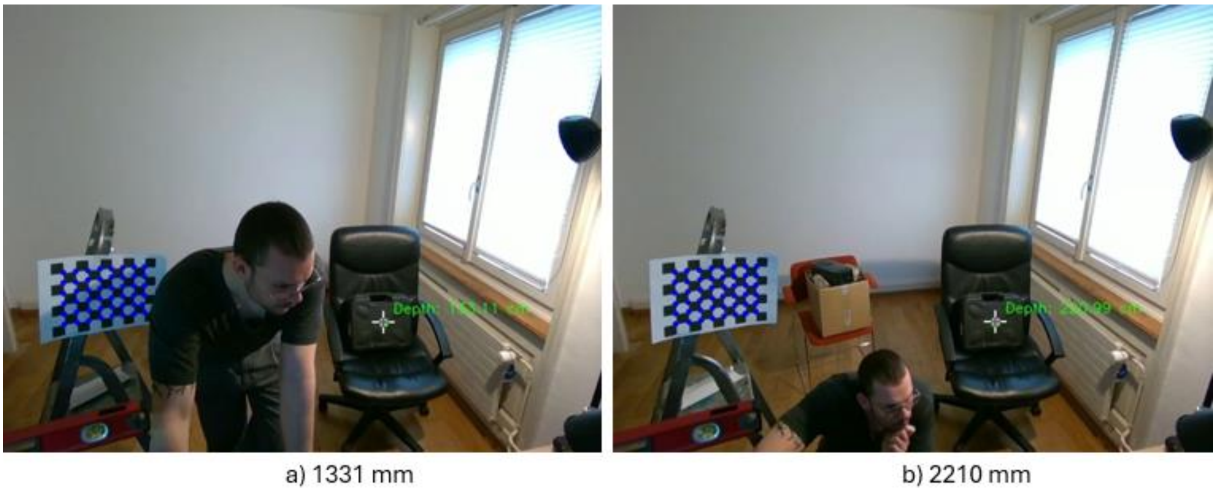
This experiment demonstrated that an MDE model currently cannot replace an RGB-D camera for lifting 2D HPE keypoints into 3D.



a) 1089 mm b) 1868 mm c) 2309 mm

d) 1690 mm e) 1856 mm f) 1720 mm

Figure 27: Depth Anything V2 - depth estimation at query point



a) 1331 mm b) 2210 mm

Figure 28: Depth Anything V2 - Impact of varying scene composition on estimated depth

4 Concept and Implementation

The experiments described in Chapter 3 demonstrates that the approach combining 2D HPE with an RGB-D camera yields the most reliable results for 3D pose detection. A RSN could utilize 3D HPE to facilitate tool handovers, with the first step being the hand tracking and subsequent movement of the cobot towards the hand. The next step in integrating 3D HPE would involve identifying individuals within the environment and treating them as dynamic obstacles to be avoided. This work focuses on the initial step of the process, aiming to develop a POC capable of detecting a hand and facilitating tool handovers using 3D HPE. Future advancements could expand this system by incorporating the second step of the handover process.

To achieve this objective, the dedicated hand pose estimation model MediaPipe Hands [60] was selected. MediaPipe Hands offers straightforward implementation and can run seamlessly alongside YOLOv8, as shown in Figure 29. However, for the scope of this work, only MediaPipe Hands is utilized, as full-body tracking does not provide any additional benefit to the focus of the POC.

Chapter 4.1 introduces the theoretical system architecture, outlining the modular structure of interconnected ROS2 nodes and detailing the mathematical principles underpinning their functionality.

Chapter 4.2 shifts the focus to the operational workflow, demonstrating the system's functionality in practice. This chapter elaborates on the differences between the two implemented control methods, MoveIt2-based and web socket-based and is supported by visual illustrations, providing a comprehensive understanding of the system's behavior.

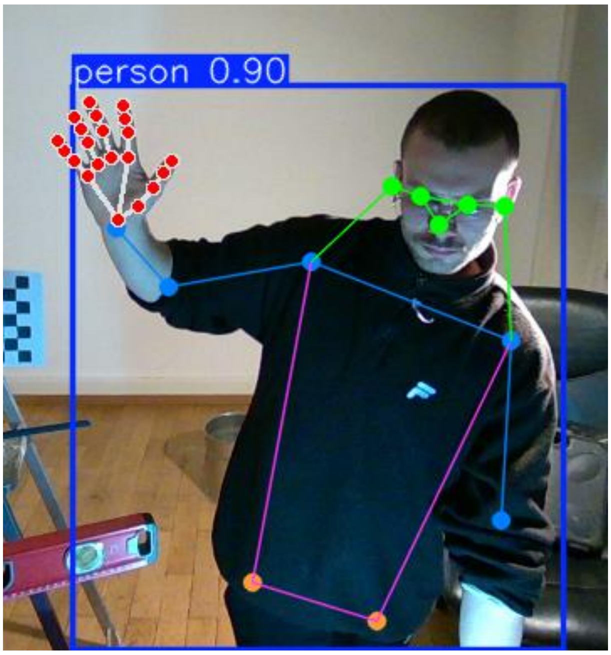


Figure 29: MediaPipe Hands running in parallel with YOLOv8

4.1 System Architecture

The POC is composed of interconnected ROS2 nodes that integrate 3D HPE with robotic control. These nodes collectively process input from an RGB-D camera, detect hand positions, calculate their spatial coordinates, and command the UR3e to execute movements for tool delivery. Figure 30 presents the flowchart of the system architecture, which visualizes the relationships and interactions between the various components of the pipeline. At the core of the system are custom-developed ROS2 nodes, depicted in orange, which serve as the primary functional elements. Data flow is represented by arrows connecting the nodes, with labels adjacent to each arrow indicating the respective ROS2 topics for publishing or subscribing.

The UR Driver, represented in gray, is an essential pre-configured ROS2 package [61], which interfaces with the MoveIt2 planner and is crucial for controlling the robotic arm, as it provides the necessary parameters of the UR3e. RViz Visualization, shown in green, enables the simulation and visualization of the cobot and its environment.

The following subchapters provide a comprehensive explanation of the functionality of each node within the system. Emphasis is placed on their individual contributions to the overarching system architecture and the mathematical principles underpinning their operation.

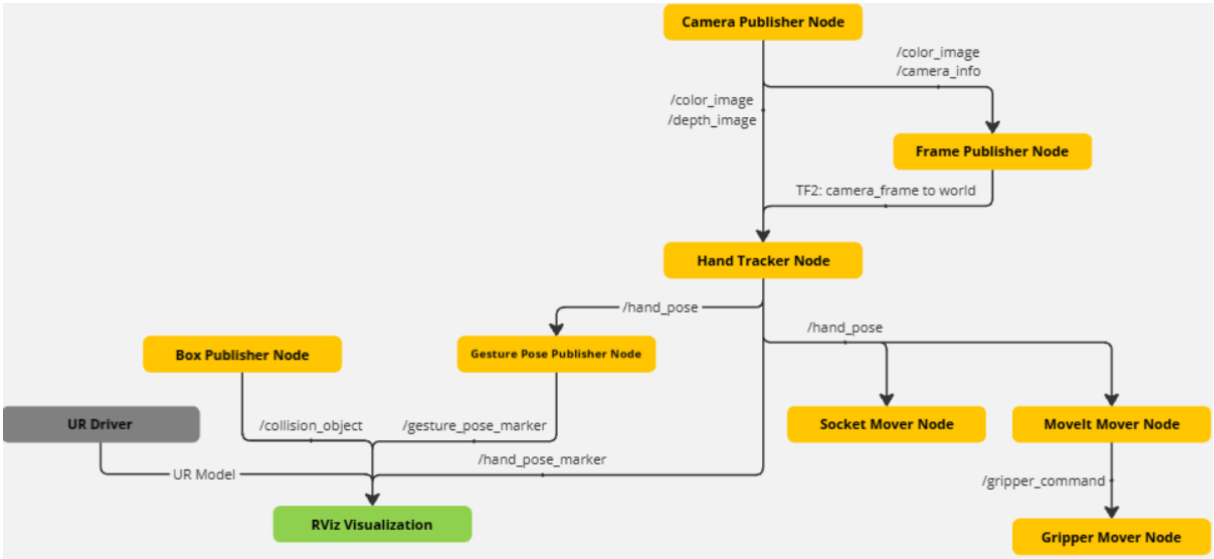


Figure 30: Flowchart of the system architecture

4.1.1 Camera Publisher Node

The pipeline begins with the camera publisher node, which captures RGB and depth frames using an Intel RealSense D455 camera. To enhance the quality of the depth map, the node applies several filtering techniques: decimation filtering is used to downsample the resolution and reduce noise, spatial filtering smooths the depth measurements spatially, temporal filtering stabilizes the depth frames over time, and hole-filling addresses any missing depth data, ensuring a more complete and reliable depth map.

The node publishes the RGB image (/color_image), depth image (/depth_image), and intrinsic camera parameters, including focal lengths and principal points, on the (/camera_info) topic. These outputs form the basis for subsequent HPE, scene calibration, and coordinate transformation.

4.1.2 Frame Publisher Node

The frame publisher node facilitates spatial alignment between the camera frame (camera_frame) and the robot's base frame (world). By utilizing an ArUco board placed in the environment, this node computes and broadcasts static transformations between camera_frame (c), aruco_board_frame (b), and world (w) via the ROS2 tf2 library. These transformations establish a consistent spatial relationship necessary for accurate hand position detection and robotic control:

Equation 1: Camera frame transformation

$${}^{w}_{c}T = {}^{w}_{b}T * {}^{b}_{c}T$$

Each transformation matrix T is defined through the rotational matrix R and translation vector t:

Equation 2: Transformation matrix

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

The transformation from the camera frame to the ArUco board frame ${}^{b}_{c}T$ is established during node initialization by detecting the predefined position of the ArUco board in the environment. Subsequently, the transformation from the ArUco board frame to the world frame ${}^{w}_{b}T$, which is coincident with the robot base, is predefined and hardcoded. However, this transformation can be adjusted to account for the specific translation and orientation required for the placement of the ArUco board relative to the robot base. These transformations are critical for accurately converting detected hand positions from camera coordinates into the robot's workspace coordinates. Figure 31 illustrates the spatial relationship between the ArUco board frame (b), the camera frame (c), and the robot base frame (w), all annotated in yellow. It also depicts the positioning of the ArUco board in front of the collaborative robot, with the RGB image providing an overview of the physical setup in the top-left corner.

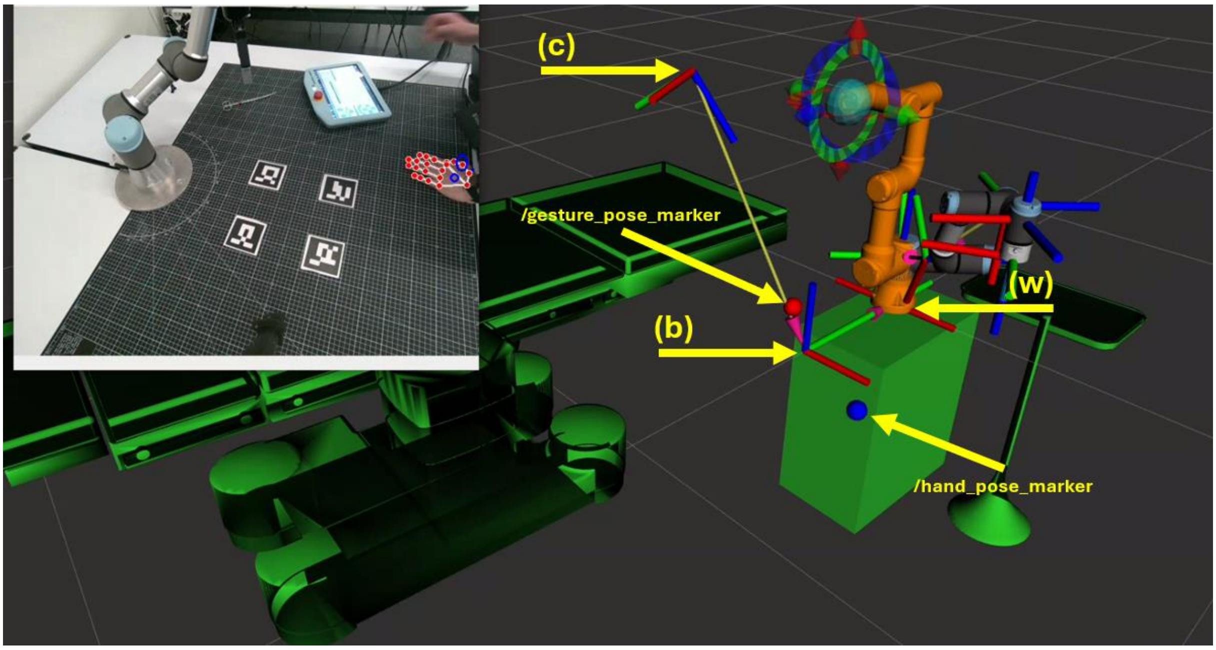


Figure 31: Visualization in RViz

4.1.3 Hand Tracker Node

The hand tracker node detects hand positions in 3D space using the MediaPipe Hands framework. It subscribes to the /color_image, /depth_image, and /camera_info topics. From the RGB images, the node identifies landmarks of a single hand and retrieves the depth values at the pixel coordinates u and v of the detected keypoints from the depth image. These pixel coordinates, combined with the corresponding metric depth Z_c, are then transformed into fully 3D metric camera frame coordinates (X_c, Y_c, Z_c) by utilizing the camera intrinsics, as described by Equation 3.

Equation 3: Pixel-to-camera-frame coordinate conversion

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \frac{(u - c_x) * Z_c}{f_x} \\ \frac{(v - c_y) * Z_c}{f_y} \\ Z_c \end{bmatrix}$$

Where Z_c is the depth at pixel coordinates (u, v). f_x and f_y denote the focal lengths in the x- and y-directions, respectively, while , c_x and c_y represent the coordinates of the principal point.

The computed 3D camera coordinates are then transformed into the robot base frame using the transformation matrix ${}^{w}_{c}T$ provided in Equation 1, which is broadcasted by the frame publisher node. This transformation ensures that the detected hand positions are accurately localized within the robot's workspace.

Additionally, the hand tracker node recognizes a specific gesture, a double close-open motion shown in Figure 32, by temporally tracking the binary state of the hand. The state is classified as either 'open,' where the detected fingertip positions are above their corresponding base joints, or 'closed,' when this condition is not met for the fingers, excluding the thumb.

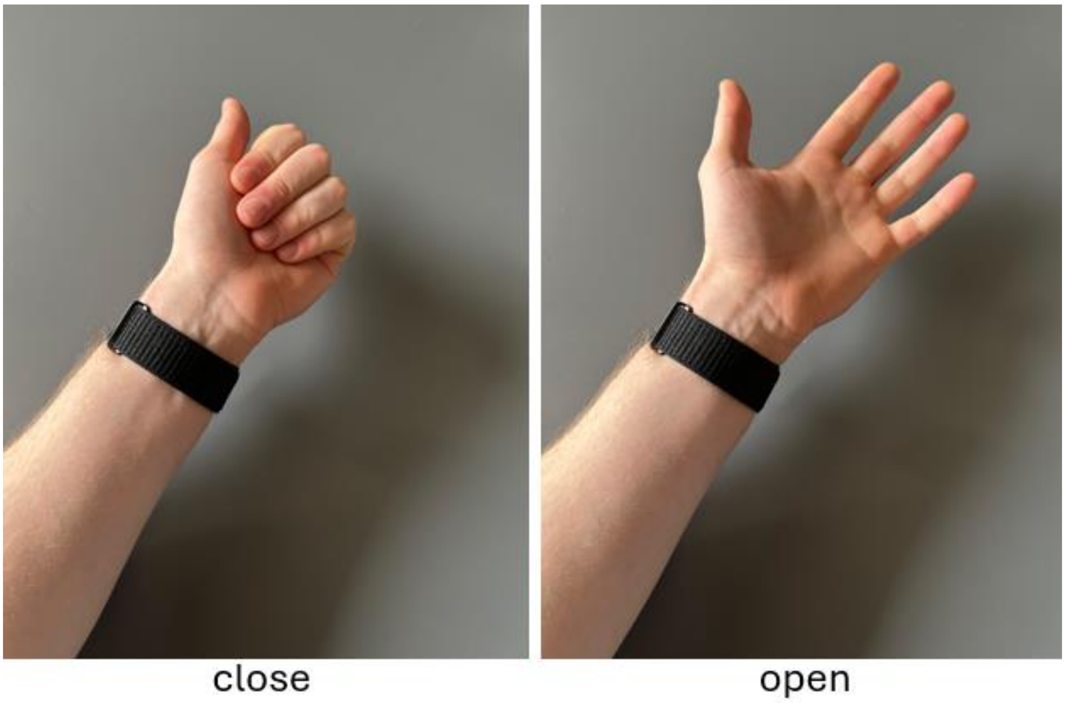


Figure 32: Hand states for close-open gesture recognition

The node is designed to identify a valid gesture as comprising four state transitions (close-open-close-open) within a one-second interval. This threshold was deliberately chosen to avoid false positives caused by unintended transitions, such as when a hand briefly enters or exits the camera frame. By requiring four state transitions in a single second, the system ensures that only intentional gestures are recognized and published. This mechanism provides robustness, enforces reliability, and minimizes the likelihood of erroneous gesture detection.

The detected hand position is published to the /hand_pose_marker topic, enabling its dynamic visualization in RViz as a blue spherical marker. This representation facilitates system debugging and validation by providing real-time feedback on the accuracy of the hand position detection. Figure 31 displays the detected hand, represented as a blue sphere, aligned with the physical hand visible in the RGB image.

4.1.4 Gesture Pose Publisher

The gesture pose publisher node subscribes to the /hand_pose topic and visualizes the detected hand gesture in RViz as a red spherical marker by publishing /gesture_pose_marker. In contrast, the /hand_pose_marker topic dynamically publishes markers corresponding to the detected hand by the camera in robot base frame. Figure 31 illustrates the node's functionality, displaying the last detected gesture in red.

4.1.5 Box Publisher Node

The box publisher node defines and publishes a rectangular static collision object within the MoveIt2 planning scene in RViz to represent physical obstacles in the robot's workspace. This object models the surface on which the robot is mounted, ensuring the robot avoids collisions with it during motion path planning in MoveIt2.

4.1.6 Robot Control Nodes

The robotic arm can be controlled using two methods, direct TCP/IP socket communication or MoveIt2-based motion planning.

Socket mover node

The socket mover node provides low-level control of the UR3e robot via TCP/IP, sending native URScript commands to the cobot. It supports both linear movements (movel) and joint-based movements (movej). Additionally, the gripper is controlled through a separate TCP/IP connection, enabling commands to open or close the gripper. The socket mover serves as a fallback method, offering basic functionality for executing movements when the advanced trajectory planning capabilities of MoveIt2 are unavailable. Further details can be found in Chapter 5.2.

MoveIt mover node

The moveIt mover node utilizes MoveIt2 for motion planning and execution. It subscribes to the /hand_pose topic and calculates trajectories to move the end-effector of the robotic arm to the detected hand position. MoveIt2 is able to ensure safe operation by accounting for obstacles defined in the planning scene, such as the static box published by the box publisher node and can be used in subsequent works to enable dynamic collision avoidance planning. The gripper commands are decoupled in this pipeline, with the gripper mover node managing gripper actions based on boolean messages published to the /gripper_command topic.

4.1.7 Launch Files

The pipeline is executed using ROS2 launch files, ensuring modularity and flexibility in node initialization and configuration. The MoveIt2-based launch file initializes the required nodes for MoveIt2 control, including the camera publisher, frame publisher, hand tracker, box publisher, moveit mover, and gripper mover nodes. Alternatively, the web socket launch file configures the pipeline for socket-based control by substituting the moveit mover node with the socket mover.

4.2 System Demonstration and Workflow

The operational workflow of the MoveIt2-based system is depicted in Figure 33. This section elaborates on each step of the process, accompanied by illustrative images to provide a comprehensive understanding of the system's functionality in practice.

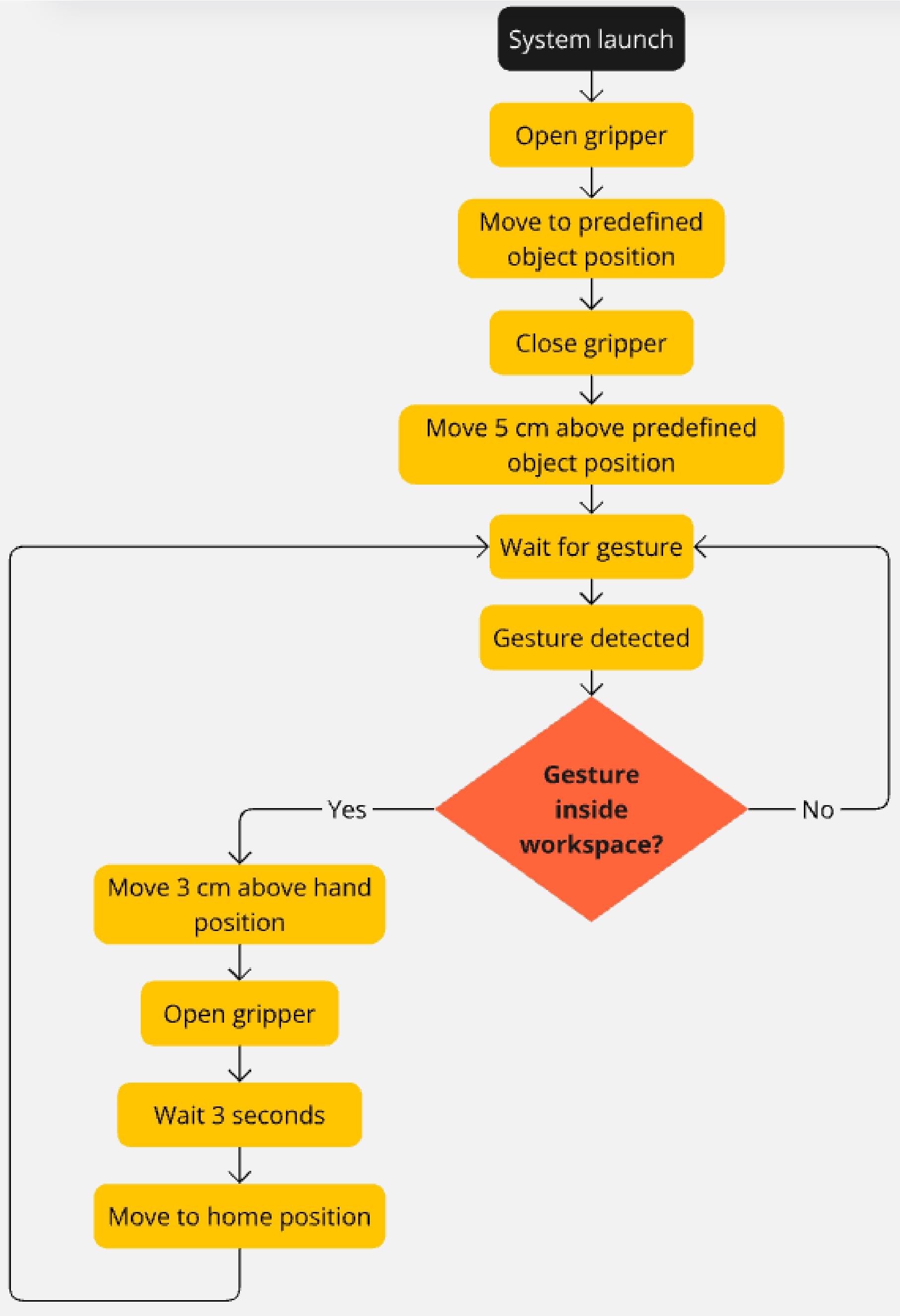

System launch
Open gripper
Move to predefined object position
Close gripper
Move 5 cm above predefined object position
Wait for gesture
Gesture detected
Gesture inside workspace?
Yes
No
Move 3 cm above hand position
Open gripper
Wait 3 seconds
Move to home position

Figure 33: Moveit2-based operational workflow

The operational workflow of the system is initiated upon executing one of the launch files. As the first step, the system sends a signal to open the gripper, regardless of its current state. Subsequently, the cobot moves to a predefined object position. In the demonstration setup, the object is represented by surgical scissors, shown in the middle of Figure 34.

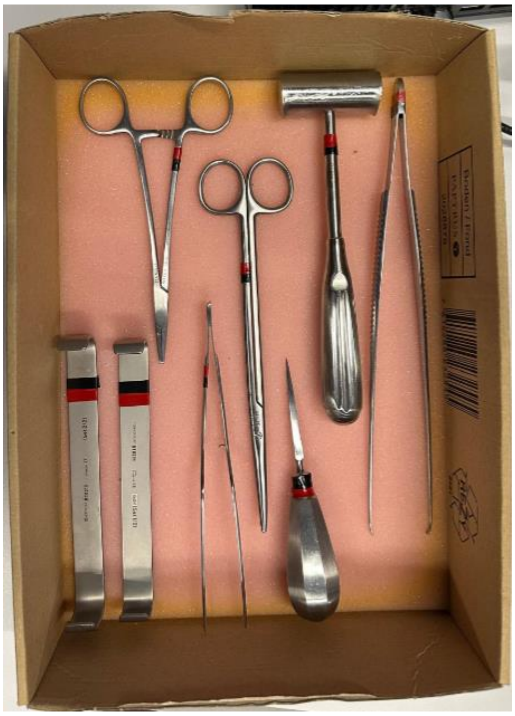


Figure 34: Surgical tools

Once the cobot reaches the object position, the gripper closes to grasp the object, and the system enters an idle state, waiting for a gesture to be recognized. This idle state, with the cobot holding the surgical scissors, is illustrated in Figure 35.

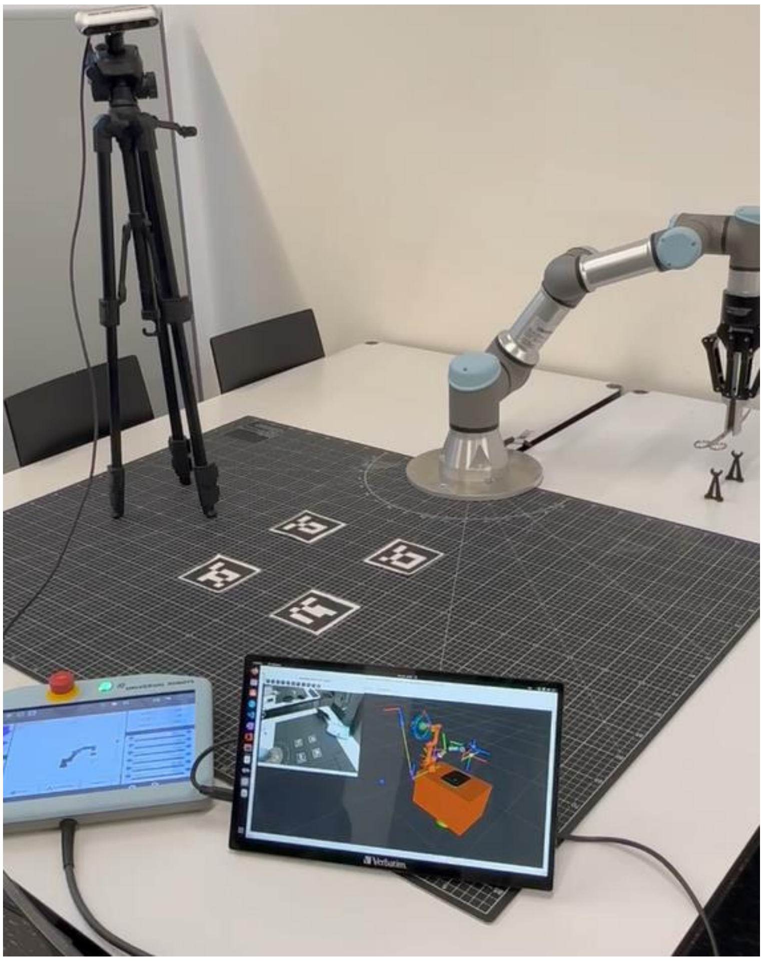


Figure 35: Idle state after object pick-up

Upon successful recognition of the gesture, the cobot moves 30 mm above the detected position relative to the tip of the gripper. Figure 36 shows the path taken when using the MoveIt2 configuration.

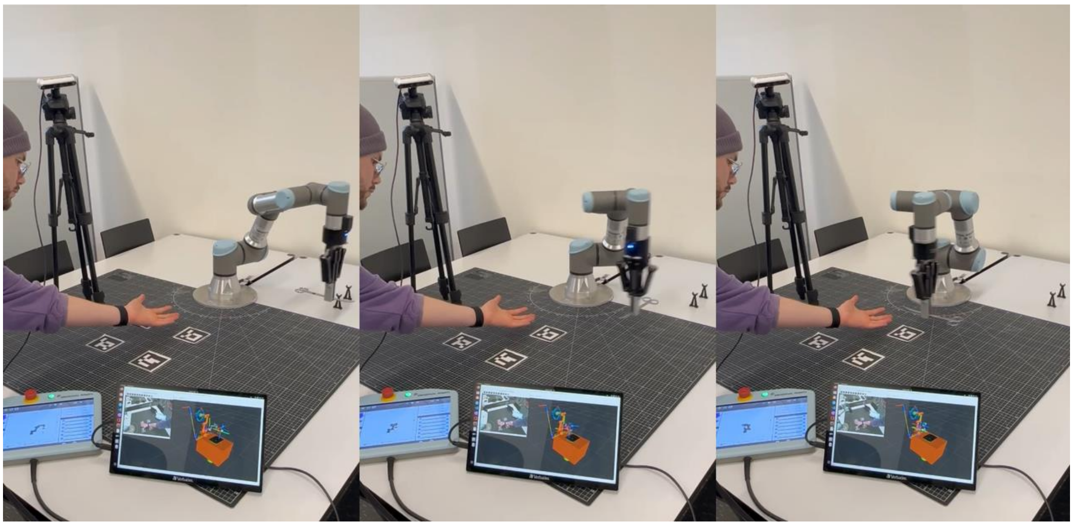


Figure 36: MoveIt2 motion

When reaching the detected position, the gripper opens to release the surgical scissors into the operator's hand. Figure 37 depicts the moment when the scissors fall into the operator's hand.

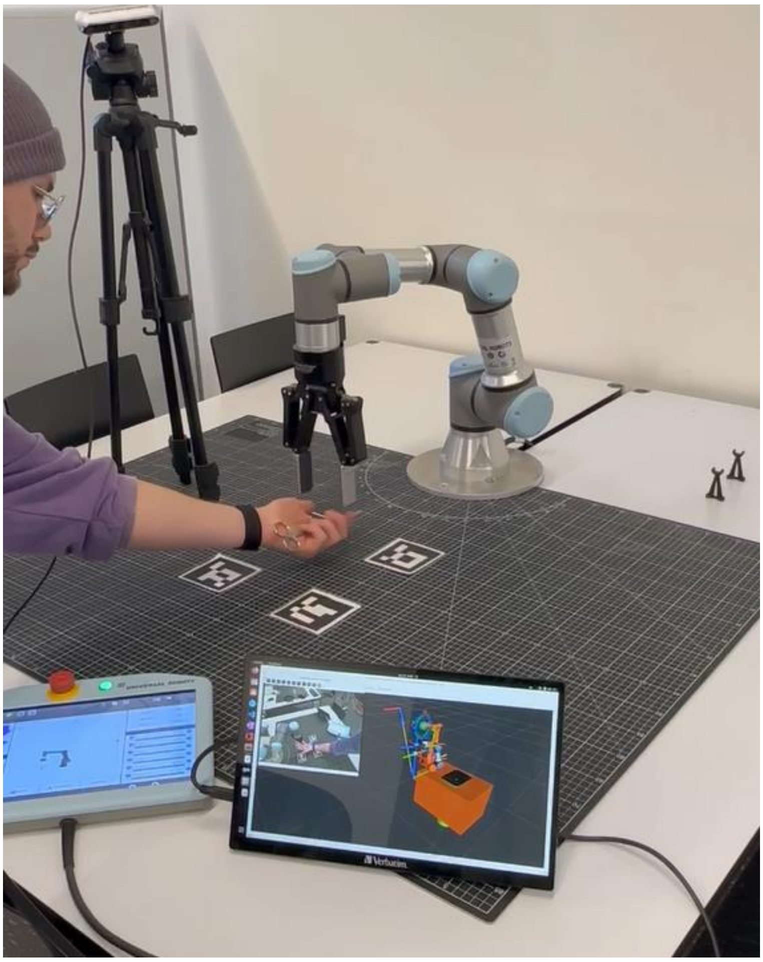


Figure 37: Hand-over of object

After releasing the tool, the system waits for 3 seconds before transitioning to a predefined home position, as demonstrated in Figure 38**.** From this point onward, the cobot remains responsive to gestures, continuously monitoring for subsequent activations. Upon detecting a new gesture within the workspace, the cobot moves to the specified position. At this stage, the system still sends an opening signal to the gripper upon reaching the target position. However, unlike the initial cycle, no closing signals are transmitted. As a result, the gripper remains inactive. Consistent with the initial cycle, the cobot pauses at the target position for 3 seconds before returning to the predefined home position.

If the detected hand position lies outside the cobot's workspace, the system generates an error message in the terminal, and the cobot waits for the next gesture to be performed.

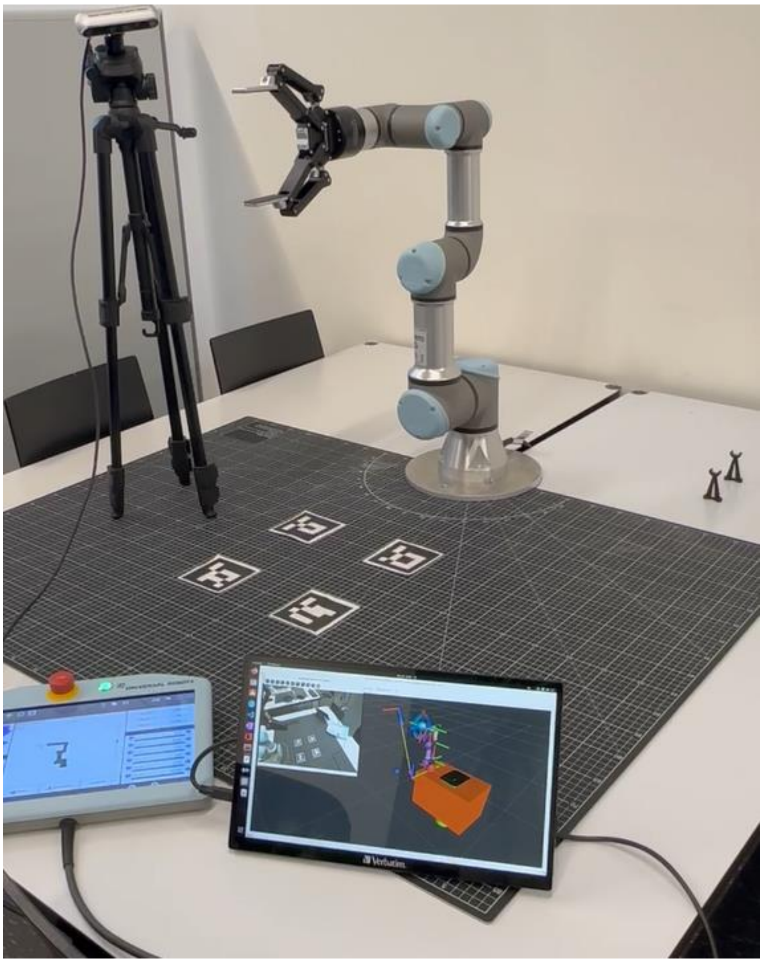


Figure 38: Predefined home position

The MoveIt2 and TCP/IP control methods operate on a similar principle but exhibit some differences. MoveIt2 provides position feedback, allowing the system to confirm whether the target position has been reached before issuing the command to open the gripper.

In contrast, the web socket-based control does not include a feedback implementation over TCP/IP. As a result, predefined wait times are hardcoded into the workflow. In this case, the robot moves to the hand position, and timers are initiated simultaneously. Specifically, a 4-second timer dictates when the gripper opens, while an 8-second timer triggers the robot's return to the home position. This lack of feedback introduces a reliance on fixed delays, making the web socket implementation less adaptive compared to the MoveIt2-based approach.

5 Discussion

The following chapter critically evaluates the developed POC, examining its limitations and areas for future development. By reflecting on challenges encountered during testing and implementation, this discussion provides insights into the system's current capabilities and outlines the necessary steps to advance the POC into a fully operational RSN prototype.

This chapter is structured into four sections. Chapter 5.1 evaluates the system's capability to track and interpret hand gestures, addressing challenges associated with the RGB-D camera. Chapter 5.2 examines the limitations of the robot control methods, highlighting areas for improvement. Chapter 5.3 examines the physical constraints of the cobot's operational range and its ability to manage and interact with tools. Finally, Chapter 5.4 explores safety considerations and strategies for improving the system's awareness and interaction with static and dynamic elements in the OR.

5.1 Gesture Recognition and Depth Accuracy

The gesture recognition system demonstrates reliable performance under favorable conditions, such as a bird's-eye view with consistent overhead lighting. However, its overall precision and response speed remain untested. Without comparative evaluations or benchmark testing, the system's performance in real-world applications and its effectiveness compared to existing systems designed to approach hands and handover objects remain unclear. Furthermore, real-world environments, particularly ORs with adjustable pivotable lamps and reflective objects, introduce challenges that may compromise the system's robustness. Direct lighting into the RGB-D camera lens or horizontal camera orientations under standard ceiling lighting were observed to reduce the reliability of hand depth measurements. Testing the Intel RealSense D455 camera under OR-like conditions is essential to identify its limitations and refine its setup. Adjustments to the applied filters could enhance robustness but are unlikely to fully address this limitation. Finding the optimal placement of the camera within the OR will be a crucial challenge, which involves balancing proximity to the target scene, avoidance of hand-occlusion, and mitigation of direct light interference. It is presumed that the 2D HPE model is not the system's performance bottleneck, which opens the possibility to explore alternative depth sensors with proprietary pose estimation models, such as the ZED2 camera with the ZED SDK. Existing literature highlights the ZED2 camera's potential for higher precision, particularly at longer ranges, making it a promising candidate for future evaluations [62], [63, p. 24].

Initially, calibration was performed using a single ArUco marker, but this approach resulted in inconsistent transformations, leading to positional offsets of up to 50 mm. These inaccuracies caused misalignment between the tool and the hand, compromising the tool handover process. Subsequently, a dynamic calibration method employing an ArUco board with four markers was tested, enabling camera repositioning while maintaining alignment. However, this approach introduced another issue: occlusions of the ArUco board markers by arms, hands, or the cobot led to shifts in the coordinate frame, resulting in subsequent displacements of the camera position, as depicted in Figure 39. While mitigating this issue is achievable, addressing it was beyond the scope of this work. The final implementation adopted a static transformation frame broadcaster, requiring an unobstructed view of the ArUco board during system initialization and prohibiting subsequent camera movement. Future iterations of the system could prioritize resolving the frame shifts caused by occlusion, thus enabling greater flexibility and more dynamic setups.

A point of critique concerns the hand tracking node, which exhibits limitations in both multi-hand tracking and the implementation of gesture recognition. Currently, the system is restricted to tracking only one hand. While MediaPipe Hands inherently supports multi-hand tracking, no reliable method was developed to identify and designate which hand should be tracked. As a result, the system's functionality was limited to single-hand tracking, simplifying the demonstration but constraining its applicability in real-world scenarios. Additionally, the gesture recognition implementation is rudimentary, relying solely on comparing the y-coordinates of the fingertips to their respective base joints to classify the hand as 'open' or 'closed.' While this approach works for a quick demonstration, it is not adaptable to varying camera and hand orientation configurations. To enhance robustness, future implementations should consider more advanced methods, such as calculating the Euclidean distance between the fingertips and the wrist and comparing it to the corresponding distances of the base joints.

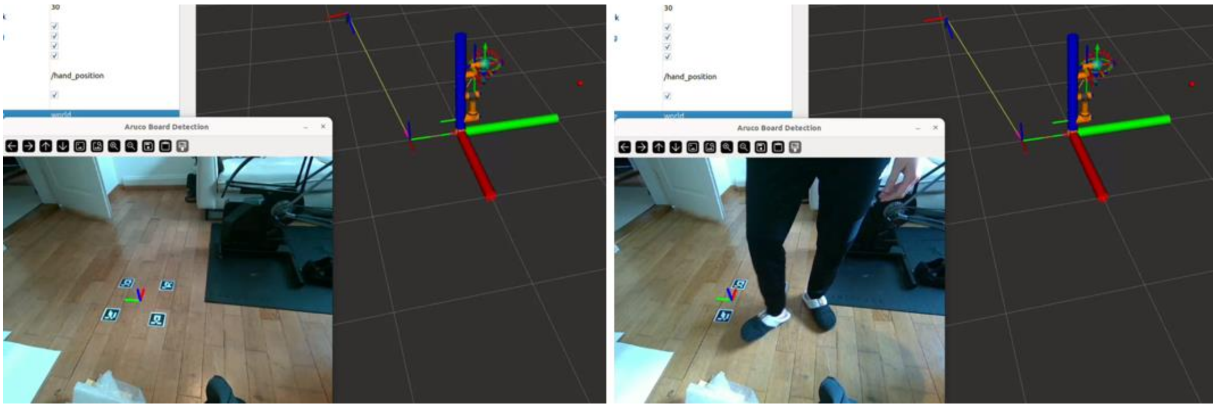


Figure 39: Shifting of occluded ArUco board frame

5.2 Robot Control and Path Planning

MoveIt2 provides advanced trajectory planning, the potential to integrate obstacle avoidance in future developments, and position feedback to confirm task completion before gripper actuation. However, it poses several challenges, primarily due to limitations associated with the UR ROS2 drivers. For instance, the trajectory depicted in Figure 36 highlights inefficiencies and potential safety risks, with the cobot approaching the hand on the same plane and rotating the tool, briefly directing the scissor tip toward the operator’s hand during its approach. Additionally, the cobot frequently rotated multiple times around its joints during the executed paths, while inconsistently high joint accelerations often triggered safety stops, and infeasible paths occasionally led to system crashes. Such issues underscore the need for extensive fine-tuning. Initial improvements should focus on modifying the motion planning to ensure the cobot consistently approaches the operator’s hand from above, minimizing the risk of injuries. However, as highlighted in [1, pp. 59-60], the UR ROS2 drivers, in combination with MoveIt2’s planning scene, pose significant challenges for fine-tuning due to inconsistencies in the configuration files needed for MoveIt2 and provided by the drivers. This source suggests considering an alternative cobot with more reliable drivers unless compatibility between the UR3e and MoveIt2 improves. While these challenges may be manageable for simple path planning, they might become a critical limitation for dynamic obstacle avoidance. Another issue is the absence of gripper and tool geometries in the planning scene, which can result in the cobot colliding with itself. Incorporating the gripper and tool geometries into the RViz planning scene would help constrain and optimize motion paths, significantly enhancing operational efficiency.

In contrast, the web socket-based implementation was designed as a fallback which offers simpler, linear trajectories and avoids many of the complexities associated with MoveIt2, such as unnecessary rotations. However, this method lacks position feedback and relies on hardcoded delays, including a 4-second timer for gripper opening and an 8-second timer for returning to the home position. While functional, this reliance on fixed timers limits the system’s adaptability. Although incorporating position feedback into the web socket control is feasible and would enhance reliability by reducing dependence on predefined delays, this enhancement was outside the scope of this thesis. Due to the needed advanced trajectory planning capabilities and potential for integrating dynamic obstacle avoidance the focus lied on MoveIt2. As such, it should remain the primary focus for future developments, with efforts directed toward addressing its current limitations.

5.3 Workspace and Tool Handling

The limited workspace of the UR3e cobot presents a significant operational constraint. Due to the fixed horizontal orientation of the end-effector, the gripper's length cannot be fully utilized to extend the workspace. For instance, while the cobot's maximum horizontal reach at the base plane measured approximately 460 mm, its effective range for elevated targets diminished considerably due to the orientation of the end-effector, compelling operators to position their hands uncomfortably close to the cobot. As illustrated in Figure 40, the cobot operates near its maximum stretch at around 430 mm on the base plane, highlighting these limitations. Employing a larger cobot, such as the UR5e with an extended workspace of 850 mm, could address this constraint by offering increased reach [64].

This POC is limited to picking up a single predefined tool and does not include strategies for subsequently picking up tools or dynamically adapting the handover process based on the specific tool geometry. By focusing solely on the x, y, z coordinates of the hand while neglecting its orientation, the motion planning is unable to adapt the end-effector orientation in response to the operator's hand pose. Consequently, the tool cannot be positioned in an ergonomic or task-specific manner during handovers. Additionally, the system does not incorporate any handover feedback mechanisms, such as leveraging the UR3e's integrated force sensors or the vision system, to confirm successful tool grasping. Instead, the gripper opens automatically upon reaching the goal position, which increases the risk of the tool being dropped if the operator moves their hand unexpectedly. A potential improvement could involve verifying whether the /hand_pose_marker approximately coincides with the /gesture_pose_marker before executing the gripper opening command, reducing the likelihood of falsely releasing tools.

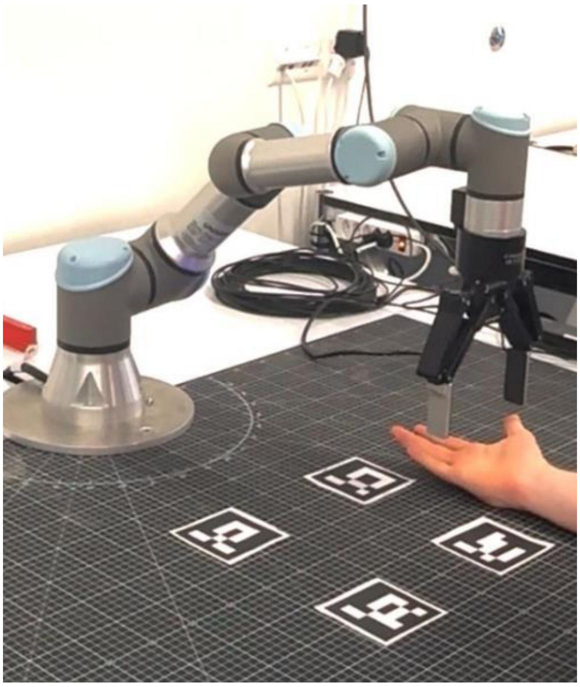


Figure 40: UR3e operating near its workspace limit

5.4 Collision Avoidance and Human Pose Estimation

The POC accounts only for the table as a static collision object, represented through the box publisher node, described in Chapter 4.1.5. While effective for demonstrating basic functionality, this limited approach does not consider other critical environmental features, such as walls, cabinets or surgical equipment. Expanding the box publisher node to incorporate the environment’s static objects would generate a more comprehensive and detailed planning scene, significantly improving the system’s collision avoidance and adaptability in surgical settings. A more versatile approach would be to generate a point cloud through LiDAR of the operational environment and treat the surfaces as static obstacles.

Dynamic obstacle avoidance, on the other hand, necessitates robust spatial understanding, including the ability to detect, track and interpret human poses within the environment. One potential enhancement involves deploying the tested 3D HPE approaches with a 360-degree camera. Such a setup could provide a holistic view of the OR, capturing human poses around the surgical workspace and enabling the possibility to dynamically add the personnel into the planning scene. However, the skeletal models utilized by the tested HPE approaches represent humans as a series of keypoints and neglect the volumetric nature of the human body. This limitation could be mitigated by approximating body segments with cylindrical volumes, interpolating depth values between keypoints to model the physical space occupied by a person. Another approach could involve replacing HPE models with segmentation models, which assign each pixel representing a human in the image to that individual within the scene [65]. By mapping these pixels, segmentation models can provide detailed planar information about the visible surfaces of human bodies. When augmented with depth data from the camera, this information could be used to approximate the spatial volume of individuals more effectively. Additionally, the segmentation-based method would allow for the identification of the spatial extent of humans within the scene, offering a more comprehensive understanding of their positioning and geometry compared to traditional HPE models. This method would offer a more detailed and realistic representation of humans in the environment, surpassing the keypoint-based connections currently employed. Alternatively, transitioning to HPE models that directly employ volumetric body representations could further improve the system's ability to accurately model and avoid collisions.

The HPE models employed in this work could not be tested in an environment closely resembling a real OR. It is possible that the robustness of the system may decline when confronted with features such as medical scrubs, monotonous colors, and frequent occlusions typical of surgical settings. Therefore, it would be essential to evaluate the system in such an environment, or at the very least, under conditions mimicking an OR, such as testing with an operator wearing surgical attire.

A logical next step would involve training the employed models using data specifically gathered from OR environments to improve their adaptability and reliability. However, as noted in Chapter 2.2.1, manually labeling 3D data is a highly time-intensive task, presenting a significant challenge for practical implementation in the short term. Leveraging alternative approaches, such as synthetic data generation or semi-supervised training techniques, could offer viable solutions to this limitation. However, training and deploying 2D models is inherently less complex than managing direct 3D HPE models, highlighting the practicality of maintaining the current 2D HPE approach.

6 Conclusion and Outlook

This chapter provides a summary of the key findings and contributions of this work, highlighting the achievements of the developed POC. The conclusion reflects on the technical and conceptual advancements demonstrated by the POC, emphasizing its role as a foundation for future developments. The outlook builds upon these achievements, presenting opportunities and strategies to further refine and expand the system, paving the way for its transformation into a fully operational RSN prototype capable of addressing the challenges faced in surgical environments.

6.1 Conclusion

A global deficit of nurses, including scrub nurses, has placed increasing pressure on healthcare systems worldwide. Scrub nurses play a vital role in surgical procedures, managing complex tasks under high-pressure conditions where precision and efficiency are paramount. This work sought to address these challenges by developing a foundational POC for a RSN system capable of performing tool handovers.

The first part of this work involves a comprehensive evaluation of state-of-the-art 2D and 3D HPE approaches to identify the most suitable methods for collaborative robotics in surgical settings. This evaluation highlighted the strengths and limitations of various methods, showing that a 2D HPE model such as YOLOv8, combined with depth estimation using the Intel RealSense D455 RGB-D camera, could achieve accurate spatial localization. Additionally, MeTRAbs, a 3D HPE model, demonstrated potential for tasks where precision is less critical. These insights were crucial for guiding the design and implementation of the POC.

Building on this foundation, MediaPipe Hands, a 2D HPE model specifically designed for hand pose estimation, was used to create a pipeline for hand tracking. The model, together with the depth data from the Intel RealSense D455 camera, was integrated into a modular ROS2 architecture designed for scalability and adaptability. Key modules included gesture recognition, frame alignment, and motion control, each contributing to a cohesive and flexible system capable of dynamic human-robot interaction. Two distinct control methodologies were implemented: a MoveIt2-based approach offering advanced trajectory planning, and a web socket-based approach prioritizing simplicity and ease of use.

The system's ability to recognize a double close-open hand gesture enabled intuitive tool handovers, showcasing the potential of gesture-based interaction in human-robot collaboration. Key technical challenges, such as calibration and frame alignment, were addressed to ensure sufficient spatial accuracy for controlled robotic movements and reliable tool delivery.

Beyond these technical achievements, this thesis contributes to the field of collaborative robotics by demonstrating that HPE can effectively enable task-specific robotic assistance in surgical settings. The integration of HPE with robotic control highlights the feasibility of developing systems that alleviate the cognitive and physical burdens on medical professionals, offering practical solutions to the global shortage of healthcare workers.

In conclusion, this thesis successfully developed a POC that combines 2D HPE with RGB-D-based perception, enabling intuitive interaction and reliable robotic control to address the critical needs of surgical tool handovers. The results establish a robust foundation for future advancements in RSN systems, demonstrating their potential to enhance safety, efficiency, and collaboration in surgical workflows. By laying this groundwork, the thesis contributes to advancing innovations in ORs, with the goal of supporting scrub nurses and addressing critical challenges in surgical environments.

6.2 Outlook

The POC establishes a foundation for robotic tool handovers, but significant advancements are needed to develop a fully functional RSN prototype. Key areas for improvement include tool selection, grasping, path planning, and handover strategies, as well as expanded functionality for managing used tools.

Future systems could leverage LLMs for auditory commands or predictive machine learning algorithms to anticipate required tools, reducing response times and streamlining workflows. Segmentation models might enable accurate identification and differentiation of surgical instruments, ensuring precise and individualized tool handling.

To address the unique geometry and purpose of each instrument, advanced gripping strategies or custom tool holders could be essential, ensuring secure handling while simplifying grasp complexity. Additionally, path planning would need to account for static obstacles and moving personnel, avoiding collisions and ensuring proper tool orientation. The deployment of 3D HPE models on a 360° camera system could further enhance spatial awareness and motion planning.

Furthermore, handover strategies should prioritize ergonomic delivery, such as aligning scissors with the operator's fingers for immediate use. Feedback mechanisms, including visual or force sensing, could confirm secure grasps before releasing tools, increasing reliability.

With these advancements, the RSN prototype has the potential to revolutionize surgical workflows, addressing pressing challenges in healthcare and paving the way for safer, more efficient, and highly collaborative ORs. This vision underscores the importance of continued research and development, ensuring that robotic systems can fully realize their role in supporting medical professionals during surgeries.

List of Figures

List of Abbreviations

2D	Two-dimensional
3D	Three-dimensional
ASM	Active Shape Model
CNN	Convolutional Neural Network
DNN	Deep Neural Network
FCL	Flexible Collision Library
GPU	Graphics Processing Unit
HPE	Human pose estimation
IMPPE	Image-based multi-person pose estimation
IMU	Inertial Measurement Unit
IR	Integrated infrared
ISPPE	Image-based single-person pose estimation
LiDAR	Light Imaging, Detection and Ranging
MAV	Micro Aerial Vehicles
MDE	Monocular depth estimation
MPPE	Multi-person pose estimation
NLOS	Non-Line-of-Sight
OMPL	Open Motion Planning Library
OR	Operating Room
OSRF	Open Source Robotics Foundation
PRM	Probabilistic Roadmap Method
RF	Radio frequency
RGB	Red Green Blue
RGB-D	Red Green Blue-Depth
ROS	Robot Operating System
RRT	Rapidly-Exploring Random Tree
RSN	Robotic scrub nurse
RViz	ROS visualizer
SDK	Software Development Kit
SLAM	Simultaneous Localization and Mapping
UN	United Nations
UR3e	Universal Robot Model 3e
VMPPE	Multi-person video pose estimation
VSPPE	Single-person video pose estimation
WHO	World Health Organization

References

[1] S. Goebel, "A Dynamic Motion Planning Approach for Robotic Scrub Nurses," *Project Thesis,* 2024.

[2] G. T. Babalola, J.-M. Gaston, J. Trombetta and S. T. Jesso, "A systematic review of collaborative robots for nurses: where are we now, and where is the evidence?," *Frontiers in Robotics and AI,* vol. 11, no. 1, 2024.

[3] A. Nakano and K. Nagamune, "A Development of Robotic Scrub Nurse System - Detection for Surgical Instruments Using Faster Region-Based Convolutional Neural Network," *Journal of Advanced Computational Intelligence and Intelligent Informatics,* vol. 26, no. 1, pp. 74-82, 2021.

[4] M. Kuluru, S. Sirasala, V. Jammalamadaka, M. Spiller, T. Sühn, A. Illanes, A. Boese and M. Friebe, "Collaborative Robot as Scrub Nurse," *Current Directions in Biomedical Engineering,* vol. 7, no. 1, 2021.

[5] C. Perez-Vidal, E. Carpintero , N. Garcia-Aracil, J. Sabater-Navarro, J. Azorín, A. Candela and E. Fernandez, "Steps in the development of a robotic scrub nurse," *Robotics and Autonomous Systems,* vol. 60, no. 6, pp. 901-911, 2012.

[6] M. G. Jacob, Y.-T. Li and J. P. Wachs, "A gesture driven robotic scrub nurse," *IEEE International Conference on Systems, Man, and Cybernetics,* 2011.

[7] S. Li, J. Wang, R. Dai, W. Ma, W. Y. Ng, Y. Hu and Z. Li, "RoboNurse-VLA: Robotic Scrub Nurse System based on Vision-Language-Action Model," *arXiv preprint arXiv:2409.19590,* 2024.

[8] A. Ezzat, A. Kogkas, J. Holt, R. Thakkar, A. Darzi and G. Mylonas, "An eye-tracking based robotic scrub nurse: proof of concept," *Surgical Endoscop,* vol. 35, no. 9, pp. 5381-5391, 2021.

[9] L. Wagner, S. Jourdan, L. Mayer, C. Müller, L. Bernhard, S. Kolb, F. Harb, A. Jell, M. Berlet, H. Feussner, P. Buxmann, A. Knoll and D. Wilhelm, "Robotic scrub nurse to anticipate surgical instruments based on real-time laparoscopic video analysis," *Communications Medicine,* vol. 4, 2024.

[10] C. Göras, U. Nilsson, M. Ekstedt, M. Unbeck and A. Ehrenberg, "Managing complexity in the operating room: a group interview study," *BMC Health Services Research,* vol. 20, no. 1, 2020.

[11] T. Ayabe, M. Tomita, R. Maeda and M. Okumura, "Implementation of Resilience Engineering for Operating Room. Unveiling the Hidden Interactions among Multi-Professionals in a Surgical Team," *Surgical Science,* vol. 11, no. 09, pp. 242-256, 2020.

[12] Z. Tunegová, "Evaluation of Human Pose," *Master's Thesis,* 2023.

[13] C. Zheng, W. Wenhan, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz and M. Shah, "Deep Learning-Based Human Pose Estimation: A Survey," *ACM Computing Surveys,* vol. 56, no. 1, 2023.

[14] S. Dubey and M. Dixit, "A comprehensive survey on human pose estimation approaches," *Multimedia Systems,* vol. 29, no. 2, 2022.

[15] D. Zhang, Y. Wu, M. Guo and Y. Chen, "Deep Learning Methods for 3D Human Pose Estimation under Different Supervision Paradigms: A Survey," *Electronics,* vol. 10, no. 18, 2021.

[16] Y. Chen, Y. Tian and M. He, "Monocular Human Pose Estimation: A Survey of Deep Learning-based Methods," *Computer Vision and Image Understanding,* vol. 192, 2020.

[17] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li and C. Lu, "AlphaPose: Whole-Body Regional Multi-Person," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 6, pp. 7157-7173, 2023.

[18] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose:RealtimeMulti-Person2DPose," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 43, pp. 172-186, 2021.

[19] R. Khirodkar, T. Bagautdinov, J. Martinez and S. Zhaoen, "Sapiens: Foundation for Human Vision Models," in *arXiv preprint arXiv:2408.12569*, 2024.

[20] C. Park, H. S. Lee, W. J. Kim and H. B. Bae, "An Efficient Approach Using Knowledge Distillation Methods to Stabilize Performance in a Lightweight Top-Down Posture Estimation Network," *Sensors,* vol. 21, no. 22, 2021.

[21] R. Bashirov, A. Ianina, K. Iskakov, Y. Kononenko, V. Strizhkova, V. Lempitsky and A. Vakhitov, "Real-time RGBD-based Extended Body Pose Estimation," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2021.

[22] "Depth Map from Stereo Images," Open Source Computer Vision, [Online]. Available: https://docs.opencv.org/3.4/dd/d53/tutorial_py_depthmap.html. [Accessed 04 01 2025].

[23] "Python OpenCV – Depth map from Stereo Images," GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/python-opencv-depth-map-from-stereo-images/. [Accessed 04 01 2025].

[24] A. Jan and S. Seo, "Monocular Depth Estimation Using Res-UNet with an Attention Model," *Applied sciences,* vol. 13, no. 10, 2023.

[25] Y. Ren, X. Han, C. Zhao, J. Wang, L. Xu, J. Yu and Y. Ma, "LiveHPS: LiDAR-based Scene-level Human Pose and Shape Estimation in Free Environment," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2024.

[26] P. Slade, A. Habib, J. L. Hicks and S. L. Delp, "An open-source and wearable system for measuring 3D human motion in real-time," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING,* vol. 69, no. 2, pp. 678-688, 2021.

[27] M. Javaid, A. Haleem, R. P. Singh, S. Rab and R. Suman, "Significant applications of Cobots in the field of manufacturing," *Cognitive Robotics,* vol. 2, pp. 222-233, 2022.

[28] "UR3e Ultra-lightweight, compact cobot," Universal Robots, [Online]. Available: https://www.universal-robots.com/products/ur3e/. [Accessed 26 12 2024].

[29] "UR3e Technical Specification," Universal Robots, [Online]. Available: https://www.universal-robots.com/media/1807464/ur3e-rgb-fact-sheet-landscape-a4.pdf. [Accessed 26 12 2024].

[30] Intel, "Intel RealSense Product Family D400 Series Datasheet," 2024.

[31] "Intel RealSense Depth Camera D455," Intel, [Online]. Available: https://www.intelrealsense.com/depth-camera-d455/. [Accessed 26 12 2024].

[32] "Intel RealSense SDK 2.0," Intel, [Online]. Available: https://www.intelrealsense.com/sdk-2/. [Accessed 26 12 2024].

[33] S. Macenski, B. Gerkey, C. Lalancette and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics,* vol. 7, no. 66, 2022.

[34] A. Bonci, F. Gaudeni, M. C. Giannini and S. Longhi, "Robot Operating System 2 (ROS2)-Based Frameworks for Increasing Robot Autonomy: A Survey," *Applied Sciences,* vol. 13, no. 23, 2023.

[35] "ROS2 Documentation Humble: Understanding nodes," Open Robotics, [Online]. Available: https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Nodes/Understanding-ROS2-Nodes.html. [Accessed 26 12 2024].

[36] "MoveIt 2 Documentation," PickNik Robotics, [Online]. Available: https://moveit.picknik.ai/main/index.html. [Accessed 26 12 2024].

[37] "tf2: Package Summary," Open Robotics, [Online]. Available: https://wiki.ros.org/tf2?. [Accessed 02 01 2025].

[38] "Tf2," Open Robotics, [Online]. Available: https://docs.ros.org/en/jazzy/Concepts/Intermediate/About-Tf2.html. [Accessed 02 01 2025].

[39] "ROS Package Summary: rviz," [Online]. Available: http://wiki.ros.org/rviz. [Accessed 28 12 2024].

[40] N. Padkan, P. Trybala, R. Battisti, F. Remondino and C. Bergeret, "Evaluating Monocular Depth Estimation Methods," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. XLVIII, no. 1/W3-2023, pp. 137-144, 2023.

[41] F. Khan, S. Salahuddin and H. Javidnia, "Deep Learning-Based Monocular Depth Estimation Methods—A State-of-the-Art Review," *Sensors,* vol. 20, no. 8, 2020.

[42] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng and H. Zhao, "Depth Anything V2," *arXiv preprint arXiv:2406.09414,* 2024.

[43] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka and M. Müller, "ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth," *arXiv preprint arXiv:2302.12288,* 2023.

[44] "Metric and Relative Monocular Depth Estimation: An Overview. Fine-Tuning Depth Anything V2," Hugging Face, [Online]. Available: https://huggingface.co/blog/Isayoften/monocular-depth-estimation-guide?. [Accessed 27 12 2024].

[45] S.-J. Oh and S.-H. Lee, "A Novel Method for Monocular Depth Estimation Using an Hourglass Neck Module," *Sensors,* vol. 24, no. 4, 2024.

[46] D. Pavllo, C. Feichtenhofer, D. Grangier and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019.

[47] S. Kreiss, L. Bertoni and A. Alahi, "OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 8, pp. 13498-13511, 2021.

[48] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo and R. Girshick, "Detectron2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2. [Accessed 29 12 2024].

[49] L. Bertoni, S. Kreiss, T. Mordan and A. Alahi, "MonStereo: When Monocular and Stereo Meet at the Tail of 3D Human Localization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021.

[50] Y. X. J. Zhang, Q. Zhang and D. Tao, "ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation," *arXiv preprint arXiv:2204.12484,* 2021.

[51] "Ultralytics YOLOv8," Ultralyitcs Inc., [Online]. Available: https://docs.ultralytics.com/de/models/yolov8/. [Accessed 29 12 2024].

[52] "OpenMMLab Pose Estimation Toolbox and Benchmark," MMPose Contributors, 2020. [Online]. Available: https://github.com/open-mmlab/mmpose. [Accessed 29 12 2024].

[53] 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.

[54] W. Mao, Y. Ge, C. Shen, Z. Tian, X. Wang, Z. Wang and A. v. d. Hengel, "Poseur: Direct Human Pose Regression with Transformers," *arXiv preprint arXiv:2201.07412,* 2022.

[55] N. Samet, C. Romme, D. P. and E. V. , "PAFUSE: Part-based Diffusion for 3D Whole-Body Pose Estimation," *arXiv preprint arXiv:2407.10220,* 2024.

[56] I. Sárándi, T. Linder, K. O. Arras and B. Leibe, "MeTRAbs: Metric-Scale Truncation-Robust Heatmaps for Absolute 3D Human Pose Estimation," *IEEE Transactions on Biometrics, Behavior, and Identity Science,* vol. 3, no. 1, pp. 16-30, 2021.

[57] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li and K. Chen, "RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose," *arXiv preprint arXiv:2303.07399,* 2023.

[58] 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," *arXiv preprint arXiv:1906.08172,* 2019.

[59] "First 3D Poses in the Wild Challenge," Max-Planck-Gesellschaft zur Förderung der Wissenschaften e.V., [Online]. Available: https://virtualhumans.mpi-inf.mpg.de/3DPW_Challenge/. [Accessed 30 12 2024].

[60] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," *arXiv preprint arXiv:2006.10214,* 2020.

[61] "Universal_Robots_ROS2_Driver," GitHub, Inc., [Online]. Available: https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver. [Accessed 01 01 2025].

[62] A. K. Ramasubramanian, M. Kazasidis, B. Fay and N. Papakostas, "On the Evaluation of Diverse Vision Systems towards Detecting Human Pose in Collaborative Robot Applications," *Sensors,* vol. 24, no. 2, 2024.

[63] A. T. Vladimir Tadic, Z. Vizvari, M. Klincsik, Z. Sari, P. Sarceviv, J. Sárosi and I. Bíró, "Perspectives of RealSense and ZED Depth Sensors for Robotic Vision Applications," *Machines,* vol. 10, no. 3, pp. 1-27, 2022.

[64] "UR5e Lightweight, versatile cobot," Universal Robots, [Online]. Available: https://www.universal-robots.com/de/produkte/ur5-roboter/. [Accessed 03 01 2025].

[65] "ultralytics Instance Segmentation," Ultralytics Inc., [Online]. Available: https://docs.ultralytics.com/tasks/segment/. [Accessed 04 01 2025].

[66] "Stereolabs Docs: API Reference, Tutorials, and Integration," Stereolabs Inc., [Online]. Available: https://www.stereolabs.com/docs. [Accessed 30 12 2024].

[67] "ZED 2: Versatile stereo camera for spatial perception," Stereolabs Inc., [Online]. Available: https://www.stereolabs.com/en-de/products/zed-2. [Accessed 30 12 2024].