Titre: Title:	Multi-Camera Calibration and Real-Time Pose-Guided View Selection for Supervision of Human-Robot Interaction
Auteur: Author:	Alaleh Asaran Darban
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Asaran Darban, A. (2025). Multi-Camera Calibration and Real-Time Pose-Guided View Selection for Supervision of Human-Robot Interaction [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/67793/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/67793/
Directeurs de recherche: Advisors:	Lama Séoud
Programme: Program:	GÉNIE INFORMATIQUE

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Multi-Camera Calibration and Real-Time Pose-Guided View Selection for Supervision of Human-Robot Interaction

ALALEH ASARAN DARBAN
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Juillet 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Multi-Camera Calibration and Real-Time Pose-Guided View Selection for Supervision of Human-Robot Interaction

présentée par **Alaleh Asaran DARBAN**
Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment acceptée par le jury d'examen constitué de :

François GUIBAULT, président
Lama SÉOUD, membre et directrice de recherche
Amir HAJZARGARBASHI, membre

DEDICATION

To my beloved husband, for his unwavering support, encouragement, and belief in me, even during the most challenging moments of this journey.

And to my precious daughter, Delin, whose smile has been my greatest source of strength and inspiration.

This work is for you both....

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to my supervisor, Dr. Lama Seoud. This thesis is the result of a long journey that came with both challenges and moments of growth. Throughout every difficult situation I encountered, she stood by my side with patience, encouragement, and unwavering support, something I will never forget.

Accepting me into her supervision was more than an academic opportunity ; it was a turning point in my life. Her kindness, her belief in me when I needed it most, and her constant presence not only shaped the direction of this research but also offered comfort and strength like that of a caring family member.

She was far more than a supervisor, she was a source of light during the darkest moments, and I will always hold deep appreciation and respect for everything she did for me.

I would also like to thank the National Research Council (NRC) for their generous funding and for providing access to the cobotic platform used in this research. Without their support, this work would not have been possible.

A special thanks to Corentin Hubert and Marie Noel, lab members who greatly assisted in the data collection process. Their dedication and collaboration were invaluable in making this research a success.

RÉSUMÉ

Ce mémoire présente une approche fondée sur la vision pour améliorer l'interaction homme-robot dans des environnements industriels collaboratifs, en s'appuyant sur un système multi-caméras RGB-D. Réalisée dans une cellule cobotique de parachêvement, équipée de six caméras Intel RealSense D455 synchronisées, l'étude aborde trois défis principaux : la calibration des caméras, l'estimation robuste de la pose de l'opérateur humain, et la sélection adaptative des vues de caméras.

Premièrement, un cadre de calibration a été mis en place en utilisant une mire pour effectuer la calibration intrinsèque et extrinsèque stéréo. La précision du calibrage a été évaluée à l'aide de l'erreur quadratique moyenne de reprojection (MSE), après une optimisation non linéaire. Les résultats obtenus ont permis de dégager des pistes d'amélioration avant une utilisation de ces matrices pour la reconstruction 3D.

Deuxièmement, l'estimation de la pose humaine a été étudiée à travers une comparaison entre les frameworks OpenPose et MediaPipe sur un petit jeu de données en haute résolution. Bien que MediaPipe fournisse une structure de points clés plus détaillée, sa complexité computationnelle plus élevée limite son utilisation en traitement temps réel multi-caméras. La version allégée d'OpenPose propose un meilleur compromis entre efficacité et précision des points clés, ce qui en fait le modèle le plus adapté pour cette application.

Troisièmement, une stratégie de sélection de caméras a été mise en œuvre à l'aide d'un classifieur Random Forest. En extrayant des caractéristiques de confiance à partir des points clés détectés, le modèle a permis de sélectionner efficacement l'ensemble de caméras le plus informatif sur la pose de l'opérateur, pour chaque image. L'évaluation en validation croisée leave-one-out a montré d'excellentes performances, avec une précision moyenne de 94,99 %, une précision de 92,3 %, et un rappel parfait de 100 %.

Dans l'ensemble, le système proposé offre une chaîne de perception évolutive et réactive pour les systèmes robotiques collaboratifs. Ce travail propose une méthodologie robuste pour la calibration multi-caméras et la sélection de vues basée sur la pose de l'opérateur, avec des applications potentielles dans des environnements industriels réels.

ABSTRACT

This thesis presents a vision-based approach for enhancing human-robot interaction in collaborative industrial environments using a multi-camera RGB-D system. Conducted within a cobotic part-finishing cell equipped with six synchronized Intel RealSense D455 cameras, the research addresses three main challenges: precise camera calibration, robust human pose estimation, and adaptive camera view selection.

First, a calibration framework was developed using checkerboard-based intrinsic and stereo extrinsic calibration procedures. The accuracy of the calibration was evaluated through mean squared reprojection error (MSE), after non-linear optimization. This setup ensured a consistent coordinate system for 3D reconstruction.

Second, for pose estimation, a comparative study between OpenPose and MediaPipe was carried out on a small high-resolution dataset. While MediaPipe provided a richer keypoint structure, its higher computational overhead made it less viable for real-time multi-camera processing. OpenPose's lightweight configuration offered an optimal trade-off between efficiency and keypoint accuracy, making it the preferred model for this application.

Third, a camera selection strategy was implemented using Random Forest classification. By extracting confidence-based features from detected keypoints, the model effectively selected the most informative subset of cameras for each frame. Evaluation through leave-one-out cross-validation showed excellent performance, with an average accuracy of 94.99

Overall, the proposed system delivers a scalable and real-time perception pipeline for collaborative robotic systems. The work contributes a robust methodology for multi-camera calibration, pose-based camera selection, and vision-guided interaction, with applications extending to real-world industrial deployments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

IETF	Internet Engineering Task Force
OSI	Open Systems Interconnection

CHAPITRE 1 INTRODUCTION

1.1 Context

With the increasing integration of **collaborative robotic systems (cobots)** in industrial settings, ensuring precise environmental perception is essential for enhancing human-robot interaction. Unlike traditional industrial robots, which operate in isolation, cobots are designed to collaborate with human operators within shared workspaces.

This research is conducted in collaboration with the Aerospace Manufacturing Technologies Center (AMTC) at the National Research Council of Canada (NRC), where a prototypical cobotic platform is made available for our experiments. The cobotic cell, illustrated in Figure 1.1, consists of a Universal Robots UR10 mounted on a finishing table, surrounded by six RGB-D cameras placed at different locations to provide a complete view of the workspace.

This cobotic cell is specifically designed for part-finishing operations such as polishing and sanding. In this setup, the operator interacts with the workpiece in a highly intuitive and contactless manner. The entire interaction is achieved solely through vision-based systems and there is no need for physical contact with any interface or control device. This vision-driven workflow enables seamless and efficient human-robot collaboration while maintaining a natural and ergonomic environment for the operator.

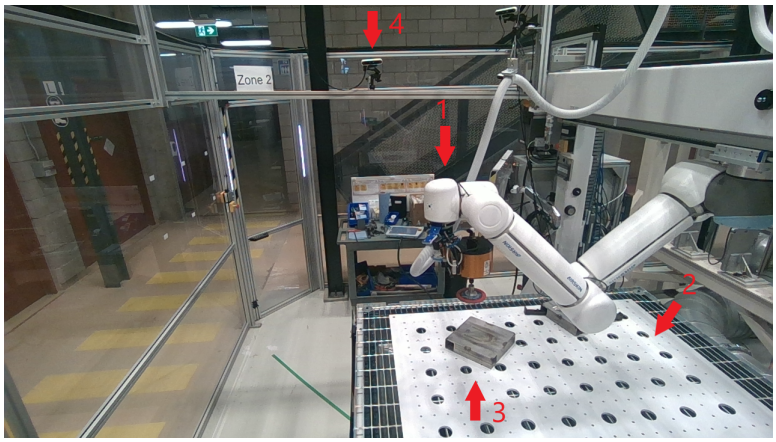


FIGURE 1.1 1 :cobot arm, 2 :table, 3 :work piece 4 :one of the six installed RGB-D cameras.

The work of Corentin Hubert, within our research team, conducted in the same cobotic cell at the National Research Council of Canada, focused on the development of a robust, multi-view gesture recognition framework and resulted in the creation of the publicly available

MuViH dataset. The dataset was designed to enable vision-based, contactless communication between an operator and a collaborative robot by interpreting predefined static hand gestures. To support this, a comprehensive data acquisition campaign was carried out involving 20 different participants. Each participant was instructed to perform a sequence of ten predefined static hand gestures across three spatial zones around a downdraft table, which serves as the central work surface in the cobotic cell. These gestures were selected based on ergonomic constraints and relevance to industrial part-finishing tasks. In this work, camera calibration was conducted independently for each camera using single-view images, without employing any multi-view fusion techniques. Each RGB-D camera captured its own sequence of checkerboard images, and intrinsic parameters were estimated individually using standard single-camera calibration methods. Although multiple cameras were used in the setup, all calibration processing was performed on an image-by-image basis rather than through joint or stereo calibration involving simultaneous multi-view data. Also, data collection was performed using six synchronized Intel RealSense D455 RGB-D cameras placed at two different height levels and distributed around the workspace to maximize visibility while minimizing the risk of occlusion from the operator or the robot. During the acquisition sessions, the configuration of the UR10 cobot was varied between recordings to introduce realistic levels of occlusion and simulate operational variability. All gesture sessions were recorded with precise synchronization among the six RGB-D streams, resulting in a highly redundant multi-view representation of each action. Following the acquisition phase, a large-scale annotation process was undertaken to generate accurate 2D bounding boxes around the hands for each camera stream and frame, facilitating hand gesture recognition. These annotations were used to train a YOLOv8-based hand detector, followed by a gesture classification pipeline using a ResNet-inspired neural network. This extensive dataset and experimental design provide a robust foundation for subsequent research, including the current thesis, which expands the perception pipeline by focusing on calibration accuracy, camera view selection, and human pose estimation to further enhance multi-camera collaboration in the cobotic cell.

One of the key challenges in deploying cobots in **industrial environments** is ensuring that their **visual perception is both accurate and computationally efficient**. In a multi-camera system, three major problems must be addressed : **cameras calibration**, **human pose estimation**, and **camera view selection**.

Camera calibration is a fundamental process that retrieves cameras intrinsic and extrinsic parameters which allow metric understanding of the 3D scene in a unified coordinate system. In cobotic applications, where precise object localization and human tracking are essential, improper calibration can lead to **misalignment, inaccurate depth estimation, and poor pose estimation of the operator**. Given the complexity of a cobotic environment rich in

visual occlusions and cluttered background, calibrating a network of 6 cameras to provide a unified and accurate 3D representation of the workspace is a non-trivial challenge.

Human pose estimation (HPE) is another essential component for tracking an operator's movements. HPE involves detecting keypoints on the human body and reconstructing their 3D positions in space. This functionality is crucial for monitoring operator safety, ensuring proper task execution, and providing real-time feedback to the robotic system. However, achieving robust human pose estimation in an industrial setting is challenging due to occlusions, varying lighting conditions, and the presence of dynamic elements within the workspace.

Camera view selection focuses on dynamically determining **which subset of cameras should be active at any given time**. In fact, keeping all six cameras active simultaneously introduces significant challenges, including **redundant data collection, increased processing time, and unnecessary computational load**. Selecting the most **informative subset of cameras** based on the operator's location, pose visibility, and the current task is critical to optimizing system efficiency.

This research focuses on developing an intelligent **camera calibration, human pose estimation, and view selection strategy** that enhances human-robot collaboration by ensuring that only the most relevant cameras are active while maintaining precise spatial alignment across all viewpoints.

1.2 Problem Statement

Although cobots have been successfully integrated into industrial environments, their effectiveness heavily depends on their **perception capabilities**. Multi-camera vision systems provide a comprehensive view of the workspace, allowing for better tracking and monitoring of human operators. However, using multiple cameras introduces several challenges.

The first challenge is **camera calibration**. In a multi-camera setup, each camera has its own intrinsic and extrinsic parameters, including focal length, lens distortion, position, and orientation. To obtain a unified 3D representation of the environment, all cameras must be carefully calibrated so that their individual frames align within a common coordinate system. Calibration errors can lead to **unreliable skeleton tracking**, which can negatively impact robotic decision-making.

The second challenge is **human pose estimation (HPE)**. In industrial applications, accurately determining the posture and movements of an operator is essential for improving cobot response mechanisms. However, occlusions, lighting variations, and the presence of multiple

objects in the scene complicate the accurate reconstruction of human poses in 3D. Advanced deep learning-based HPE techniques must be integrated into the multi-camera system to ensure robust tracking.

The third challenge is **camera view selection**. While having multiple cameras enhances visibility, **not all cameras provide useful data at all times**. Certain cameras may have an obstructed view due to the robot's position, the operator's movement, or the workspace layout. Moreover, activating all cameras continuously leads to **unnecessary computational overhead, reduced real-time performance, and increased data redundancy**. A key challenge is developing an **adaptive camera selection strategy** that dynamically activates the most relevant subset of cameras based on the operator's location, task requirements, and visibility constraints.

Without addressing these challenges, a cobotic system may suffer from **inefficient data processing, misaligned depth estimation, poor human pose tracking, and increased operational latency**, all of which hinder real-time human-robot collaboration. This thesis aims to develop an effective **calibration, human pose estimation, and camera selection framework** to optimize perception in multi-camera cobotic environments.

1.3 Thesis Structure

This thesis is organized into six chapters :

Chapter 2 - Literature Review : This chapter presents an overview of existing research on **multi-camera calibration techniques, human pose estimation methods, and camera view selection strategies**. It highlights the strengths and limitations of current approaches and identifies gaps that this research aims to address.

Chapter 3 - Research Objectives : This chapter defines the **specific objectives** of the thesis, providing the rationale for why **camera calibration, human pose estimation, and camera view selection** are essential for improving human-robot collaboration in industrial cobotic environments.

Chapter 4 - Methodology : This chapter details the **experimental setup, dataset acquisition process, calibration methodology, pose estimation techniques, and camera selection models**. It also explains how data is processed, labeled, and used for training and evaluation.

Chapter 5 - Results and Discussion : The results obtained from experimental validation are presented and analyzed in this chapter. The system's performance is assessed based on **calibration accuracy, human pose tracking precision, camera selection efficiency,**

and computational optimization.

Chapter 6 - Conclusion and Future Work : This chapter summarizes the key contributions of the research, discusses its limitations, and suggests directions for future improvements in **multi-camera calibration, human pose estimation, and adaptive camera selection**.

CHAPITRE 2 LITERATURE REVIEW

Human-computer interaction (HCI) or human-machine interaction (HMI) examines the scientific implications and practices of interfaces between people and computers or intelligent agents. This term was first used in 1976. According to [4], there have been five stages in the development of HCI : manual, interactive command language, graphical user interface (GUI), network user interface, and natural HCI. Human-computer interaction deals with methods and tools for designing and evaluating human-computer interfaces and assessing computer usability, as well as broader human-centric issues such as computer interaction with people [5]. As machine technology advances, augmented reality (AR) is a rapidly developing area of HCI because it allows humans to interact with computers and allows them to interact visually with computing devices in a variety of ways, particularly through gestures. Gestures consist of movements made by parts of the human body, such as the face, body, hands, legs, and feet, to convey information. In augmented reality-based applications that involve human-computer interaction, the hand is usually used to recognize gestures more than other body parts, which makes hand gestures very important as an interactive medium [6] .

Perception and interpretation of human behavior, including body language, and hand, and pose estimations, via image-based and non-image-based methods, is a fundamental necessity to enable industrial cobots to be capable of detecting human presence, thereby realizing a secure and intuitive human-robot interaction in the era of the new industrial revolution, namely, industry 4.0.

This section provides a literature review organized as follows : first, an overview of the sensors used for gesture recognition is provided, second, methods for pose estimation and whole body gesture recognition are reviewed, third, some occlusions handling strategies are presented, and finally, methods for camera selection in multiple-camera settings are reviewed.

2.1 Sensors

The development of gesture interfaces, touch screens and augmented and virtual reality have resulted in new usability concerns that need to be studied in a natural environment in an unobtrusive manner. The location of the hand and fingers can be measured with high accuracy with several robust approaches, such as data gloves with electromechanical, infrared, or magnetic sensors.

2.1.1 RGB-D Cameras

The use of image-based solutions enables natural interaction with technology and enables human movement to be tracked and studied without being intrusive.

In recent years, various affordable 3D active imaging systems have been introduced to the market. These systems capture and record information about visible 3D surfaces, including their geometry and appearance. Typically, these technologies are aimed at consumers and embedded in consumer-oriented products. An active 3D imaging system is often referred to as a range camera or RGB-D camera. Structured-light, time-of-flight and active stereo vision are the three fundamental measurement principles that underpin most RGB-D cameras available today [7]. Structured-light involves projecting a pattern of light onto the scene and analyzing the deformation of the pattern to determine depth. Time-of-flight measures the time it takes for a light pulse to travel to the scene and return to the camera to calculate depth. Active stereo vision uses two cameras at different angles to calculate depth and projects a light pattern onto the scene to artificially enhance its texture and thus to facilitate stereo matching. Each of these methods has its own advantages and disadvantages, and the choice of which to use depends on the specific application and environmental conditions. The range of commoditized structured-light systems with temporal encoding is more limited than active stereo sensors. Moreover, in a multiple cameras setting, time-of-flight sensors are sensitive to interference, resulting in erroneous depth measurement. On the opposite, active stereo vision does not rely on light reflected to the sensor, it only projects lights to enhance the texture of the scene and facilitate the stereo matching. Used in a multiple camera setting, each sensor will provide even more texture and thus depth information, which can improve the accuracy and robustness of the 3D reconstruction. This is especially useful in complex scenes with occlusions or reflective surfaces, where multiple camera views can provide different angles and viewpoints to capture the scene more completely.

Depth information is an essential aspect of 3D imaging and computer vision. It provides information about the distance between objects in a scene, allowing for the reconstruction of 3D models and the detection of objects and their movements. With the advancements in depth sensing technology, more accurate and reliable depth information can be obtained, leading to improvements in various fields such as robotics, augmented reality, and autonomous vehicles. In the context of cobotic, a comparison is made between Kinect v1 and Kinect v2 [8]. A Kinect v1 infrared emitter projects infrared dots into the environment, which are deformed by the environment to calculate depth. With its ToF camera, the Kinect v2 projects infrared light into the environment and measures the speed of that infrared light back and forth to calculate the depth of a scene. Kinect v1 sensors have a larger range, but are less accurate

than Kinect v2.

There is also a comparison between a version of Intel's RealSense called the D415 and the Kinect v2. D415 raw data significantly reduces probing form errors, probing size errors, sphere spacing errors, and flatness errors compared to Kinect v2. Intel RealSense D415 was found to be a low-cost device in [9].

Leap Motion and Microsoft Kinect, two commercially available solutions, restrict hand movement to a relatively small area, do not capture all the nuances of rapid hand movements, and are not precise enough to measure finger movement accurately [10].

Depth sensing has shown great potential in the recognition of pose estimation in a cobotic cell and hand tracking for pointing out the part being processed. Some of the studies use a Kinect depth sensor [11] [12]. This is because it offers a cheaper and easier solution as compared with other methods mentioned in [13]. In [14], the authors provide a comprehensive review of the current state-of-the-art in gesture recognition techniques for human-robot collaboration. They discuss various gesture recognition methods, such as vision-based, sensor-based, and hybrid approaches, and their advantages and limitations. They also highlight the challenges and future directions in this field. In the paper , the authors provide a detailed review of the Microsoft Kinect sensor and its applications in computer vision. They discuss the hardware and software components of the Kinect sensor and the various features it provides, such as depth sensing, skeleton tracking, and voice recognition. They also review the applications of the Kinect sensor in various fields, such as healthcare, entertainment, and robotics, and discuss the advantages and limitations of the sensor. Additionally, they highlight the future directions and potential advancements in this field.

Intel's RealSense D455 is a compact stereo RGB-D camera purpose-built for real-time three-dimensional sensing. It couples a 1280 $\times$ 720 pixel RGB module with two infrared imagers that project and capture structured light ; the on-board processor then computes dense depth maps at up to 90 frames s^{-1}. With a 95 mm baseline and an 86°×57° field of view, the camera achieves sub-millimetre depth precision from roughly 0.4 m to 6 m. Automatic dynamic calibration maintains accuracy under changing illumination or vibration, and rolling-shutter compensation limits motion artefacts. The unit is USB-powered, draws little current, and fits easily into mobile robots, handheld scanners, or fixed industrial stations.

A single D455 positioned above the workspace observes the entire arm, the surrounding environment, and the robot simultaneously, enabling multi-person tracking, collision monitoring, gesture recognition, and ergonomic analysis without requiring any wearables. Its centimetre-scale depth accuracy and higher spatial resolution offer more reliable hand-pose estimation for applications such as augmented reality or task-level supervision.

Overall, using the Intel RealSense D455 is a good choice because it is well suited for a multi-camera setting, allowing the capture of highly accurate and detailed 3D models of the scene with high texture information. Compared to other sensors such as the Kinect v1, Kinect v2, and Leap Motion, the Intel RealSense D455 has been found to have significantly fewer errors and a higher level of precision in its depth sensing capabilities, making it a more reliable and accurate option. Additionally, the D455 has a wider field of view and can capture data at higher resolutions than some of the other sensors mentioned [15]. Overall, the D455's combination of high accuracy, precision, and versatility make it a strong choice for a range of computer vision and depth sensing applications.

2.2 Human Pose Estimation (HPE)

Human Pose Estimation (HPE) is a fundamental component in facilitating intuitive human-robot interaction, particularly within collaborative industrial environments. The objective of HPE is to accurately determine the spatial configuration of human joints, enabling machines to interpret and respond to human actions effectively. Over the years, HPE methodologies have evolved significantly, transitioning from traditional model-based approaches to sophisticated deep learning techniques.

2.2.1 Early Model-Based Approaches

The early stages of Human Pose Estimation (HPE) were shaped by model-based techniques that relied heavily on handcrafted features and predefined representations of the human body. These methods were designed to identify individual body parts and reconstruct the overall pose using a set of structural assumptions.

One category of these approaches, known as appearance-based models, focused on using visual cues such as texture, color, and gradients to detect body parts. These models generally lacked flexibility and were sensitive to background clutter and occlusion. To improve robustness, deformable or structural models were introduced, which captured the relationships between body parts through spatial constraints.

A foundational contribution to this line of research was the Pictorial Structure Model [16]. This model represented the human body as a set of parts connected in a graph structure, where each connection encoded the expected spatial relationship between joints. The model balanced local appearance cues with global spatial configurations, making it more resilient to partial occlusions and detection noise. Its formulation allowed an efficient inference using dynamic programming, which was an important step toward practical implementation.

Subsequent developments, such as a research [17], extended this framework specifically for human pose estimation. They incorporated improved part detectors and more expressive spatial models, which allowed for more accurate and flexible pose recovery in real-world images. Their work also addressed people detection in crowded scenes and improved articulation modeling, laying the groundwork for the transition to more data-driven approaches.

Although these early methods provided valuable insights and theoretical foundations, they were often limited by their reliance on hand-crafted features and simplistic appearance models. Their performance degraded significantly in the presence of complex poses, lighting variations, and real-world backgrounds. These limitations became a driving force behind the emergence of deep learning-based approaches, which soon redefined the state of the art in HPE.

2.2.2 Emergence of Deep Learning Techniques

The advent of deep learning marked a significant turning point in HPE. Convolutional Neural Networks (CNNs) demonstrated remarkable capabilities in learning hierarchical feature representations directly from data, leading to substantial improvements in pose estimation accuracy and robustness.

DeepPose was among the pioneering works that applied deep learning to HPE, formulating the task as a regression problem to predict joint coordinates directly from images. This approach showcased the potential of DNNs in capturing complex spatial dependencies inherent in human poses [18].

Building upon this foundation, the Stacked Hourglass Network introduced a multi-scale architecture that processed features at various resolutions, allowing for iterative refinement of pose predictions. This design facilitated the capture of both local and global contextual information, enhancing the model's ability to handle diverse poses and occlusions [19].

2.2.3 Advancements in Lightweight and Real-Time Models

As the demand for real-time HPE applications grew, researchers focused on developing models that balanced accuracy with computational efficiency.

A novel architecture incorporating a channel attention mechanism, PixelShuffle up-sampling, and a Cross-Stage Heatmap Fusion method. Their approach achieved high accuracy while significantly reducing the number of model parameters, making it suitable for real-time applications [20].

Similarly, another research designed a lightweight image scaling network utilizing a non-local convolution operator. Their model demonstrated remarkable improvements in scale invariance and overall accuracy, further contributing to the development of efficient HPE systems [21].

2.2.4 Contemporary Frameworks : OpenPose and MediaPipe

Modern human pose estimation frameworks have advanced significantly by offering robust, real-time, and multi-person tracking capabilities. Two prominent examples such as OpenPose and MediaPipe are widely used across research and industry. Each takes a distinct approach and exhibits trade-offs in computational complexity, runtime speed, and granularity of keypoint representation.

OpenPose, developed by researchers at Carnegie Mellon University, employs a bottom-up approach based on Part Affinity Fields (PAFs). The method first detects all body parts independently and then associates them into individual skeletons. This architecture supports multi-person tracking and delivers highly accurate 2D keypoint estimation. However, its real-time performance is heavily reliant on powerful hardware. On systems equipped with a high-end GPU such as the NVIDIA GTX 1080 Ti, OpenPose can reach approximately 22 frames per second. On CPU-only systems, performance drops significantly, often below one frame per second, rendering it impractical for embedded or mobile applications. The full OpenPose model is roughly 209 megabytes in size, with computational demands reflecting its detailed architecture and comprehensive coverage of body, hand, and facial keypoints [22].

In contrast, MediaPipe, developed by Google, is designed for human pose estimation. It employs a two-stage detector-tracker architecture : the first stage detects the region of interest (ROI) containing the person, and the second stage estimates 33 pose landmarks within this ROI. While this architecture enables real-time performance on devices like smartphones and laptops, it is not devoid of computational demands. The framework indicates that higher accuracy comes at the cost of increased computational load. Moreover, the two-stage process introduces additional latency and memory usage, particularly when handling continuous video streams. Therefore, although MediaPipe is optimized for performance, its computational requirements can be significant, especially in applications demanding high accuracy or processing multiple video streams concurrently [23].

Figure 2.1 illustrates the keypoint configurations used by both frameworks. OpenPose typically relies on an 18-keypoint structure for full-body pose, whereas MediaPipe employs a more detailed 33-keypoint layout. This includes additional joints for hands, feet, and facial landmarks, making it more suitable for fine-grained motion tracking and gesture-based

interaction.

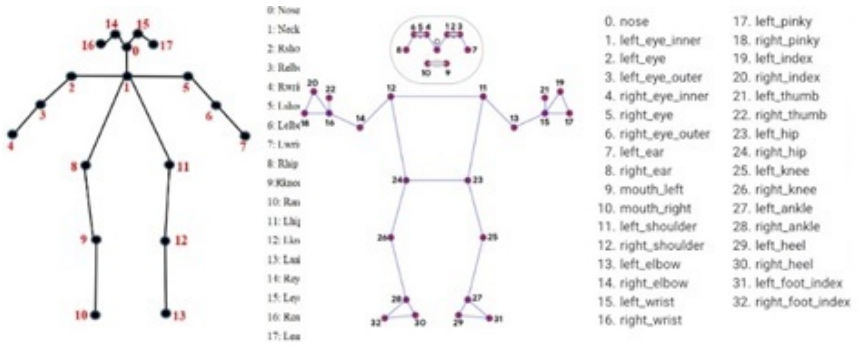


FIGURE 2.1 Keypoint configuration of the two models. Left : **OpenPose** with 18 keypoints [1]. Right : **MediaPipe** with 33 keypoints, including fine-grained facial and hand joints [2].

2.2.5 Application in Human-Robot Interaction

In the context of our project, which focuses on enhancing human-robot collaboration within a cobotic part-finishing cell, the selection of the pose estimation framework plays a critical role. The system must detect human poses in real time, with sufficient accuracy and robustness to support smooth cooperation with the robot.

By combining these insights, we aim to select and tailor the pose estimation method to the real-time requirements and industrial constraints of our cobotic system. The goal is to achieve reliable and seamless human-robot interaction.

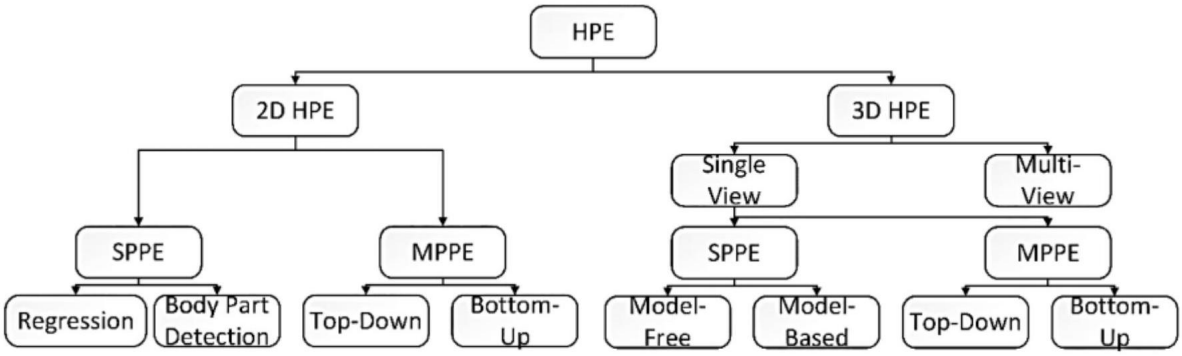


FIGURE 2.2 Taxonomy of 2D and 3D human pose estimation (SPPE refers to single person pose estimation and MPPE to multiple persons pose estimation) [3]

While the choice of pose estimation framework plays a critical role in determining overall system responsiveness and keypoint resolution, real-world deployments introduce further chal-

lenges beyond model architecture and computational constraints. One of the most persistent and difficult problems in human pose estimation, especially in collaborative workspaces like cobotic cell is occlusion. Whether caused by tools, robot arms, or the operator's own body parts, occlusions can significantly degrade keypoint detection accuracy and continuity. The effectiveness of a pose estimation system in industrial environments therefore depends not only on speed and precision, but also on its robustness to partial visibility. The following section explores the techniques that have been developed to mitigate these occlusion-related challenges and maintain reliable skeletal tracking under complex visual conditions.

2.3 Occlusion-Handling

In human pose estimation and hand tracking fields, occlusion handling refers to the ability of a system to accurately detect and track the position and movement of human body parts even when they are partially or completely obscured from view. This is a common challenge in many applications, such as sports analysis, healthcare monitoring, and human-robot interaction, where people may move behind objects or other people. Effective occlusion handling requires the use of sophisticated algorithms that can infer the position and movement of occluded body parts based on contextual information from the visible parts of the body, as well as the ability to recover from tracking failures when body parts are temporarily lost from view.

In [24], the authors evaluate the occlusion handling capability of a 3D human pose estimation framework. The paper proposes a novel metric, called the Occlusion Handling Capability (OHC) score, to quantify the performance of a 3D human pose estimation framework in handling occlusions. To evaluate the OHC score, the authors first define the occlusion status of each joint in the ground truth 3D pose, where a joint is considered occluded if it is not visible in the image. Next, they introduce a noise model to simulate occlusions in the input image, which is used to generate a set of synthetic images with varying degrees of occlusion. In another study [25], The authors propose using machine learning algorithms to predict occlusion, along with advances in tracking, depth sensing, and 3D reconstruction. They also suggest using multiple cameras and sensors to enhance occlusion handling and improve the user's perception of the virtual content. Additionally, the paper highlights the potential of using natural interactions and user-generated content to mitigate occlusion issues. In [26], the authors use feature masking to distinguish between static and dynamic parts of the scene and separate the foreground from the background. The foreground is further split into occluding and occluded regions based on the depth value. The proposed method then uses a deep neural network to predict the motion of the foreground regions and the static background.

The authors evaluate the proposed method on several benchmark datasets, and the results show that it achieves state-of-the-art performance in scene flow estimation while effectively handling occlusions. In [27], the authors propose a robust online visual tracking algorithm that handles occlusions. The algorithm consists of two main components : an object appearance model and an occlusion handling mechanism. The object appearance model is constructed by using a kernelized correlation filter (KCF) and is updated online to account for changes in the object's appearance. The occlusion handling mechanism is based on a combination of a long short-term memory (LSTM) network and a particle filter. The LSTM network is used to predict whether the object is occluded or not, and the particle filter is used to track the object's state when it is occluded. Thus one way to handle occlusion and to have a larger coverage of a scene, is to have multiple cameras

2.4 Multi-camera setups(Multi-camera calibration)

In our research requiring comprehensive scene coverage, such as human-robot interaction within cobotic cells, deploying multiple cameras is essential. However, achieving accurate spatial alignment among these cameras necessitates precise calibration of both intrinsic (internal) and extrinsic (external) parameters. Intrinsic calibration focuses on the internal characteristics of each camera, including focal length, optical center, and lens distortions. Extrinsic calibration, conversely, determines the relative positions and orientations between cameras, ensuring a unified coordinate system for accurate 3D reconstruction [28].

[29] introduced OpenPTrack, a free and open-source software tool for calibrating RGBD camera networks and tracking people. Their method involved simple steps with real-time feedback, which allowed them to estimate camera poses accurately and track people. OpenPTrack utilized a novel method that aligned people detections from all sensors in an x-y-time space to refine camera poses. The authors demonstrated a considerable improvement in people tracking performance compared to Kinect v1, Kinect v2, and Mesa SR4500, making it useful for interactive arts, education, and culture sites to interact with humans and robots.

Traditional calibration methods often rely on reference objects such as checkerboard patterns placed within overlapping fields of view. While effective, these methods are often labor-intensive and not feasible in environments with limited view overlap. As a result, recent studies have introduced more flexible calibration strategies suitable for dynamic or cluttered environments.

One such approach uses human body tracking data to establish initial pose alignment between multiple RGB-D cameras. For example, the Azure Kinect's built-in body tracking system can

provide a coarse global registration, which is subsequently refined through feature matching to achieve accurate extrinsic calibration [30].

An alternative method leverages spherical calibration objects instead of planar ones. Spheres provide visibility from a wide range of angles and distances, making them especially useful when cameras are positioned with limited overlap. By observing a sphere from different views, the system can infer spatial relationships between cameras more effectively [31].

In situations where placing physical targets is impractical, researchers have proposed targetless calibration techniques. These methods analyze environmental features such as lines or edges and apply convergence voting algorithms to infer camera poses. This target-free approach allows for efficient calibration, especially in unstructured environments [32].

These developments contribute to more adaptable and scalable multicamera calibration frameworks, enabling robust 3D reconstruction and pose estimation in real-world cobotic systems.

2.5 Camera selection

The use of camera networks is common in various visual analytics applications, such as video surveillance and crowd behavior analysis. These applications track the location of targets across multiple cameras to determine the position of a target. Automated tracking has become an essential component of visual analytics, especially with the increasing number of cameras at airports, train stations, malls, etc. In [33], DQN reinforcement learning was employed to make camera selection decisions. The action history was encoded using an LSTM-based Auto-Encoder (AE) to learn the policy faster and achieve better performance. They also demonstrated that their method optimizes camera selection and tracking performance on large datasets such as DukeMTMC and CityFlow. A semi-supervised method was later shown to produce comparable results and train in a semi-supervised manner.

In multi-camera systems, such as those used for broadcasting sports, education, concerts, or meetings, a camera is chosen automatically based on the action in the scene. A camera selection method is specific to a particular event. In [34], the authors propose a knowledge-based method to model the functionalities of automatic editing systems based on the analysis of the state of the art, which allows them to propose an automatic camera selection method. Their purpose is to represent the specification and formalization of context-specific knowledge. The proposed model has been successful in the case of municipal council broadcasts, resulting in a context event ontology for the municipal council, consisting of Persons Of Interest (POIs) and Actions Of Interest (AOIs). Moreover, a new method of speaker detection is proposed

to offer live broadcasting with an accuracy greater than 98%.

The summary of what has been done as a literature review is shown in table 2.1. If the paper's primary focus is on recognizing hand gestures or movements, we marked it as "Yes" in the "Gesture-Based" column. If the paper's primary focus is on computer vision techniques such as object detection, image segmentation, or pose estimation, we marked it as "Yes" in the "Vision-Based" column. If a paper does not have a primary focus on either gesture recognition or vision-based techniques, we marked it as "No" in both columns. The table shows that RGB-D and multi-camera were used more recently.

The literature review identified several limitations in the current research on human-robot interaction using RGB-D sensors. One limitation is that studies conducted in a laboratory setting may not fully capture the challenges and complexities of real-world industrial environments. Another limitation is the use of sensors like the Microsoft Kinect, which is no longer available. Additionally, some studies were only evaluated on a small dataset, which may limit the generalizability of the results. Furthermore, some methods may not be suitable for all types of RGB-D data or images/videos. The lack of accuracy, specifically in real-time scenarios, was also noted as a limitation. Other limitations include addressing occlusion, dealing with complex backgrounds. It is important to consider these limitations in our proposed method.

Paper	Gesture-based	Vision-based	single/multiple-cameras	Addressing Occlusion	Type of camera
[1]	Yes	No	Multiple	Yes	RGB-D
[2]	Yes	No	Single	yes	RGB
[3]	Yes	No	Multiple	Yes	RGB
[4]	Yes	No	Single	Yes	RGB
[5]	Yes	No	Single	No	RGB-D
[6]	Yes	No	Single	Yes	RGB
[7]	Yes	No	Single	Yes	RGB
[8]	Yes	No	Single	No	RGB
[9]	Yes	No	Single	No	RGB
[10]	Yes	No	Multiple	Yes	RGB-D
[11]	Yes	NO	Multiple	No	RGB-D
[12]	No	Yes	Multiple	Yes	RGB-D
[13]	Yes	No	Single	Yes	RGB
[14]	Yes	No	Multiple	Yes	RGB-D
[15]	No	Yes	Multiple	No	RGB-D
[16]	No	Yes	Multiple	Yes	RGB-D
[17]	Yes	No	Single	Yes	RGB
[18]	Yes	No	Single	Yes	RGB
[19]	No	Yes	Single	No	RGB
[20]	No	Yes	Multiple	Yes	RGB-D
[21]	No	Yes	Single	Yes	RGB
[22]	No	Yes	Single	Yes	RGB
[23]	Yes	No	Single	Yes	RGB
[24]	No	Yes	Single	No	RGB
[25]	Yes	No	Single	Yes	RGB
[26]	Yes	No	Single	Yes	RGB
[27]	Yes	No	Multiple	Yes	RGB-D
[28]	No	Yes	Single	Yes	RGB
[29]	Yes	No	Single	No	RGB
[30]	Yes	No	Multiple	Yes	RGB-D
[31]	No	Yes	Multiple	Yes	RGB-D
[32]	No	Yes	Single	Yes	RGB
[33]	Yes	No	Single	Yes	RGB
[34]	Yes	No	Single	Yes	RGB
[35]	Yes	No	Single	No	RGB
[36]	Yes	No	Single	Yes	RGB
[37]	Yes	No	Multiple	No	RGB-D
[38]	Yes	No	Multiple	No	RGB-D
[39]	Yes	No	Single	No	RGB
[40]	Yes	No	Single	Yes	RGB
[41]	Yes	No	Single	Yes	RGB-D
[42]	Yes	No	Single	No	RGB
[43]	No	Yes	Multiple	Yes	RGB
[44]	Yes	No	Single	Yes	RGB

TABLEAU 2.1 Comparison of literature reviews

CHAPITRE 3 OBJECTIVES

3.1 Rationale

To ensure reliable pose estimation in complex industrial environments, this work adopts a multi-camera RGB-D setup. This configuration provides a wider coverage of the operator's body, significantly reducing the risk of losing key points due to occlusion. In cobotic cells, where occlusions are frequently caused by tools, robotic arms, or the operator's own limbs, relying on a single viewpoint can lead to interruptions in tracking. A multicamera arrangement ensures that multiple perspectives are available at all times, allowing the system to maintain visibility of essential body joints from at least one angle. Moreover, combining depth information from different viewpoints enhances the accuracy of 3D keypoint localization and enables more robust spatial reconstruction. This is particularly important in real-time applications that depend on consistent tracking for safe and seamless human-robot collaboration.

In modern industrial environments, especially within cobotic cells, enabling intuitive and reliable communication between human operators and robots is a key challenge. While existing literature has proposed various vision-based solutions using RGB-D sensors and machine learning algorithms, many of these approaches still face limitations in terms of robustness, accuracy, and real-time performance under practical working conditions. Specifically, challenges such as occlusion, background complexity, and limited adaptability to new environments often compromise system effectiveness.

The proposed work aims to address these gaps by developing a real-time, vision-based system for human-robot interaction that is both accurate and flexible. Instead of relying on intrusive wearable devices such as data gloves or force sensors, the system utilizes RGB-D cameras that provide both color and depth information, allowing for a more holistic understanding of the operator's position, posture, and gestures. This non-intrusive approach enhances user comfort and is better suited to unstructured industrial settings.

Moreover, the project incorporates deep learning algorithms rather than traditional machine learning methods. Deep learning enables automatic feature extraction and hierarchical representation learning, which improves performance in high-dimensional, noisy data scenarios. These models also offer superior generalization capabilities and scalability, making them appropriate for industrial deployment where adaptability and precision are essential.

The experimental context of this work is a part-finishing cobotic cell developed in collaboration with the National Research Council (NRC). This environment offers a realistic platform

to implement and validate the proposed techniques and investigate their applicability in an actual industrial use case.

It is good to mention that the present work extends two prior research efforts conducted on the same cobotic platform. First, in the study led by Nathan Odic, the focus was on monitoring the minimal distance between the human operator and the robot to ensure safe collaboration. Achieving accurate and reliable distance measurements in such a multi-camera environment requires precise camera calibration. This thesis addresses that requirement by developing a sub-pixel reprojection-error-based calibration pipeline that aligns all camera views into a unified coordinate system, thereby enabling accurate spatial measurements across the workspace.

Second, in the work of Corentin Hubert, the objective was robust hand gesture recognition under realistic industrial occlusion conditions. In that study, the selection of camera views for recognition was performed manually, based on prior knowledge of optimal viewpoints. While effective in controlled experiments, this manual selection is not feasible in real-world industrial settings where operator position and occlusions change dynamically. This thesis proposes an automatic, human-pose-driven camera view selection framework that adaptively identifies the most informative camera views in real time, improving robustness and reducing reliance on manual configuration.

3.2 Specific Objectives

The primary goal of this thesis is to develop a vision-based system capable of localizing the human operator within the collaborative workspace and dynamically determining the most informative subset of RGB-D cameras based on the operator's position and pose. This enables continuous monitoring and gesture interpretation while minimizing computational overhead. By identifying which cameras provide the clearest and most complete view of the operator at any given time, the system ensures reliable keypoint estimation even in the presence of occlusions and workspace constraints. This objective supports real-time human-robot interaction by adapting visual perception to the operator's spatial context. It is realized through three specific and interconnected tasks.

3.2.1 Calibration

The first step is to accurately calibrate the network of six RGB-D cameras installed in the cobotic cell. Both intrinsic and extrinsic calibration are required to create a shared 3D coordinate system that ensures the spatial alignment of data captured from different perspectives.

Intrinsic calibration accounts for camera-specific parameters such as focal length and distortion, while extrinsic calibration aligns all cameras to a common world reference frame. This is essential for integrating depth information and ensuring consistent pose estimation regardless of the operator's position in the workspace.

3.2.2 Pose Estimation

Once the cameras are calibrated, the second objective is to estimate the full-body pose of the human operator. This is achieved using a lightweight deep-learning-based pose estimation tool that processes RGB images. The goal is to achieve a balance between accuracy and real-time performance, allowing the system to track the operator's body continuously and reliably during task execution.

3.2.3 Camera Selection

Running all six RGB-D cameras at high frame rates can create computational bottlenecks. Selecting a subset of cameras, rather than using all six simultaneously, offers several advantages beyond the reduction of computational load. First, it can significantly improve recognition accuracy by prioritizing cameras that currently have the clearest, most unobstructed view of the operator's relevant body parts. In a collaborative cell, visual occlusions caused by the robot, tools, or the operator's own movements can lead to noisy or misleading detections; selecting cameras with optimal viewpoints reduces the impact of such occlusions. Second, it minimizes the influence of perspective distortion and extreme viewing angles, which can degrade pose estimation and downstream decision-making. Third, camera selection enables adaptive reconfiguration of the sensing strategy, focusing resources on the most informative viewpoints for the task at hand and the operator's location, thereby improving system robustness in dynamic industrial conditions. Finally, reducing the number of active cameras in processing can lower data bandwidth requirements, ease network synchronization, and simplify system scaling for larger workcells.

To address this, the final objective is to develop a camera selection strategy that dynamically identifies the most informative views at any given moment. This selection is guided by a confidence-based metric that evaluates the visibility and clarity of detected keypoints. By selecting only two cameras with the highest cumulative visibility scores, the system significantly reduces processing load while maintaining high tracking fidelity. This module is essential for making the system scalable and suitable for real-time deployment in industrial scenarios.

CHAPITRE 4 METHODS

4.1 Camera Calibration

For the camera calibration phase, data were collected with a checkerboard pattern moving intentionally throughout the cobotic workspace. The goal was to ensure that the checkerboard entered the field of view of at least one of the six RGB-D cameras across a variety of angles, distances, and spatial positions, thereby capturing a broad range of poses and transitions. To construct a high-quality calibration dataset, a post-processing step was carried out to synchronize the frames and remove those in which the checkerboard was either occluded or not sufficiently visible. Only frames where the checkerboard could be robustly detected and its corners accurately extracted using standard calibration algorithms were retained. This filtering process resulted in approximately 200 usable frames per camera per participant. From these, about 100 high-quality calibration frames were selected for each camera pair, yielding a robust and diverse dataset suitable for accurate intrinsic and extrinsic calibration of the multi-camera system. Camera selection is a crucial step in the cobotic platform, ensuring that only the most informative and least occluded camera views are used for human pose estimation and gesture recognition. The goal is to reduce computational overhead, improve accuracy, and dynamically adapt to operator movement in real-time.

4.1.1 Synchronized Set of Images

A synchronized set of images consists of frames captured by multiple cameras at the same timestamp or within a short temporal window (60ms). This ensures that corresponding points in different views represent the same moment in time, minimizing discrepancies due to motion.

For calibration purposes, data were collected from four users, each holding a checkerboard and moving it throughout the cobotic cell at various angles and positions to ensure broad coverage across all camera viewpoints. For intrinsic calibration, a total of 250 cleaned (visible checkboard) frames were selected per camera per user. Following synchronization and cleaning across camera pairs, approximately 100 high-quality frames were retained for each pair to perform extrinsic calibration. These frames were specifically chosen for their visual clarity, ensuring that the checkerboard was distinctly visible and its corners reliably detectable across all relevant views.

Camera calibration is a crucial step for reconstructing 3D points accurately in a multi-camera system. The goal is to determine both intrinsic and extrinsic parameters to improve

pose estimation. Each step in this calibration process is necessary to ensure accurate depth estimation and synchronization of multiple cameras.

4.1.2 Checkerboard-Based Calibration

Calibration is performed using a checkerboard pattern, which provides known world coordinates for accurate estimation of camera parameters [35]. A checkerboard is used because it provides well-defined corner points that can be accurately detected and used for geometric transformations.

Checkerboard Specifications : The checkerboard used for calibration consisted of four rows and seven columns of black-and-white squares, forming a total of twenty-eight internal corners. Each square covered a surface area of 625 square millimeters, providing a known physical scale for the calibration process. This metric dimension is essential for computing accurate intrinsic and extrinsic camera parameters. The checkerboard's size and contrast enabled robust corner detection even under varied perspectives, distances, and lighting conditions.

An example of a calibration frame is shown in Figure 4.1. The checkerboard is clearly visible and well-positioned within the field of view, ensuring precise corner extraction and reliable multi-camera calibration.

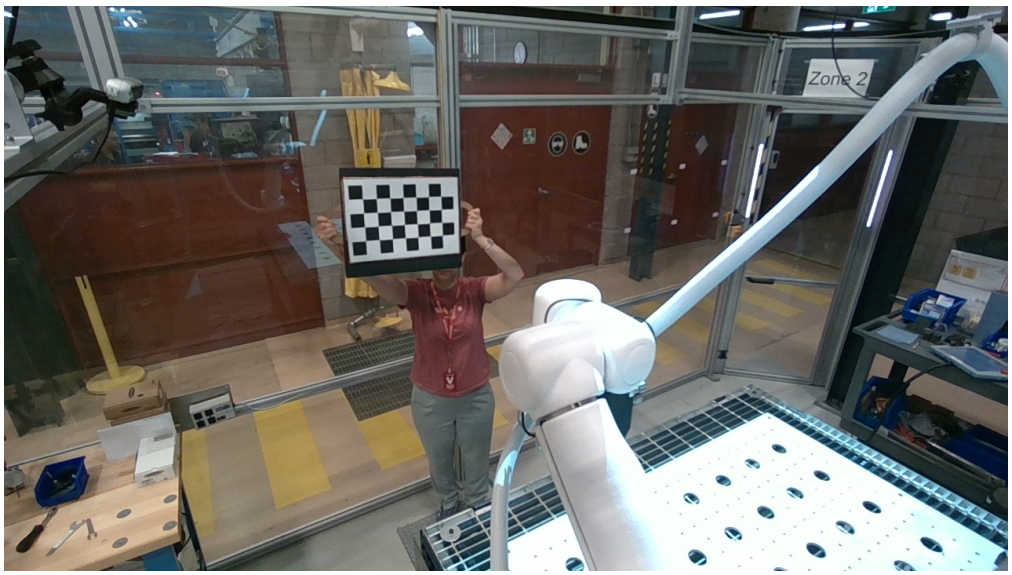


FIGURE 4.1 Example frame used for calibration showing a user holding the checkerboard visible to the camera. Multiple such frames were selected per user across different angles and positions to ensure accurate calibration.

4.1.3 Intrinsic Camera Calibration

Each camera's intrinsic parameters are computed to correct for lens distortions and to understand how the camera maps 3D points to a 2D image.

Each camera has unique internal characteristics, such as focal length and distortion, which must be accounted for to accurately reconstruct 3D geometry.

The intrinsic matrix is :

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

where :

— f_x, f_y are the focal lengths in x and y, expressed in pixels.
— c_x, c_y are the x and y coordinates of the optical center.

Stereo Calibration

Stereo calibration is the process of determining the geometric relationship between two cameras in a multi-camera setup. This involves estimating both the intrinsic and extrinsic parameters that define how the cameras are positioned and oriented relative to each other. The goal is to establish a common coordinate system for reconstructing 3D information from multiple viewpoints.

The resulting R aligns the orientation of the two optical centres while t encodes their baseline in the left-camera frame, thereby defining a common stereo coordinate system [35]. These parameters provide the initial guess for the particle-swarm global refinement ; a concise overview is also given in the OpenCV documentation [1].

The calibration yields two quantities that fully describe the relative pose of the cameras :

— **Rotation matrix** $R \in \mathbb{R}^{3\times3}$, expressing the orientation of the secondary camera with respect to the reference sensor.
— **Translation vector** $t \in \mathbb{R}^3$, giving the baseline displacement between the two optical centres (in the reference frame).

These elements are concatenated to form the homogeneous extrinsic transform

$$T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{4.2}$$

1. `https://docs.opencv.org/master/d9/d0c/group__calib3d.html`

Extrinsic Transformation of the Reference Camera Camera 1 (c_1) is selected as the origin of the global coordinate frame. In computer-vision literature the extrinsic parameters are usually written as the 3×4 matrix $[R \,|\, t]$, while in robotics the same pose is embedded in a 4×4 homogeneous transform. For the reference camera we have no rotation ($R = \mathbf{I}_3$) and no translation ($t = \mathbf{0}_3$), giving

$$[R \,|\, \mathbf{t}]_{c_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{4.3}$$

If a full homogeneous matrix is required, for instance, when chaining with robotic kinematic transforms, we append the row [0 0 0 1] :

$$T_{c_1} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3^\top & 1 \end{bmatrix}, \tag{4.4}$$

Note that R (a pure rotation) must not be premultiplied by an existing 4×4 transform; instead, R and t are concatenated as shown above to form a valid rigid-body transformation.

Computing the Transformations Between Cameras Instead of calibrating each camera directly to Camera 1, intermediate transformations are used to propagate the relationship between cameras step by step. Concatenating transformation matrices offers a practical advantage when the number of stereo image pairs available for direct calibration is limited. Instead of requiring explicit calibration between every camera pair, transformations can be chained through intermediate cameras, effectively linking their coordinate systems. This approach enables geometric alignment across the entire multi-camera network, even when some camera pairs have insufficient or no overlapping field of view for traditional stereo calibration. The transformation from Camera 1 to Camera 3 is computed using the transformation from Camera 1 to Camera 2 and Camera 2 to Camera 3 :

$$T_{c1 \to c3} = T_{c1 \to c2} \cdot T_{c2 \to c3} \tag{4.5}$$

where $T_{A \to B}$ is the transformation matrix between cameras A and B. This method extends iteratively to all cameras :

$$T_{c1 \to cn} = T_{c1 \to c2} \cdot T_{c2 \to c3} \cdots T_{c(n-1) \to cn} \tag{4.6}$$

By applying this method iteratively, all cameras in the setup can be aligned with the reference coordinate frame without requiring direct pairwise calibrations with Camera 1.

Importance of Stereo Calibration Stereo calibration is essential for :

— Establishing a unified coordinate system for all cameras.

— Enabling accurate 3D depth estimation across multiple views.

— Reducing alignment errors and improving multi-camera synchronization.

This hierarchical calibration method ensures that extrinsic parameters are consistently propagated while minimizing calibration complexity.

4.1.4 Stereo Triangulation

Once the intrinsic matrices K_i and extrinsic poses $[R_i \,|\, t_i]$ of the stereo pair are known, each camera's projection matrix is obtained as

$$P_i = K_i \,[R_i \,|\, t_i], \qquad i \in \{1, 2\}. \tag{4.7}$$

Linear triangulation. Given matched image points $x_1 = (u_1, v_1, 1)^\top$ and $x_2 = (u_2, v_2, 1)^\top$, their 3-D position is recovered by intersecting the two back-projected rays. OpenCV's call is

$$\texttt{cv::triangulatePoints}(P_1, P_2, x_1, x_2), \tag{4.8}$$

which internally builds the linear system

$$A = \begin{bmatrix} u_1 P_1^{(3,:)} - P_1^{(1,:)} \\ v_1 P_1^{(3,:)} - P_1^{(2,:)} \\ u_2 P_2^{(3,:)} - P_2^{(1,:)} \\ v_2 P_2^{(3,:)} - P_2^{(2,:)} \end{bmatrix}, \tag{4.9}$$

and solves

$$A\,\mathbf{X} = 0 \tag{4.10}$$

via singular-value decomposition. The resulting homogeneous vector $\mathbf{X}$ is normalised to obtain the world-frame coordinates $(X, Y, Z)^\top$.

4.1.5 Non-linear Global Refinement

Even after Zhang's checkerboard calibration [28], residual errors remain because of lens distortion, asynchronous triggering, and partial field-of-view coverage. We therefore run a particle-swarm optimisation (PSO) that adjusts the six extrinsic poses simultaneously so as to minimize the mean–squared *reprojection* error measured in pixels.

Cost function Let $\mathbf{X}_i \in \mathbb{R}^3$ be the i^{th} checkerboard corner (in millimetres) expressed in the world frame and let $\mathbf{p}_i = (u_i, v_i)^\top$ denote its detected image coordinates (in pixels). For a candidate parameter set the predicted pixel location is $\widehat{\mathbf{p}}_i = (\hat{u}_i, \hat{v}_i)^\top$. Finally, the optimisation target is

$$\text{MSE}_{\text{pix}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \widehat{\mathbf{p}}_i - \mathbf{p}_i \right\|_2^2, \tag{4.11}$$

where N is the total number of corner observations across all calibration images : $N = \sum_{k=1}^{M} n_k$ for M images containing n_k points each.

Interpretation of symbols.

- $\mathbf{p}_i = (u_i, v_i)$ — *measured* 2-D pixel coordinates.
- $\widehat{\mathbf{p}}_i = (\hat{u}_i, \hat{v}_i)$ — *re-projected* pixel coordinates under the current camera model.
- Index i iterates over every detected corner in every stereo image pair ; it does *not* refer to different cameras.

PSO Parameters :

- Particles : 100
- Max iterations : 100
- Inertia weight (w) : 0.7
- Cognitive factor (c_1) : 1.5
- Social factor (c_2) : 1.5

PSO iteratively updates the rotation and translation matrices to improve accuracy.

4.1.6 Metrics for Calibration Accuracy

Calibration accuracy is evaluated using key error metrics that quantify the alignment of detected features with real-world positions. The following metrics are used :

— **Mean Squared Error (MSE)** : Measures the projection accuracy in pixels by computing the average squared difference between the expected and projected points.

The Mean Squared Error (MSE) is used to measure the accuracy of the corners of the checkerboard correspondences between two different cameras. It maps points from one image to another and computes the projection error. A lower MSE indicates better alignment between detected points across images, ensuring more accurate stereo matching.

On the other hand, Root Mean Square Error (RMSE), is the reprojection error computed for a single camera, using only its intrinsic parameters. It quantifies how accurately 3D points are mapped back to the image plane after camera calibration. A lower RMSE means the estimated intrinsic parameters provide a more precise projection of real-world points.

While MSE assesses the accuracy of multi-camera alignment, RMSE evaluates the internal calibration quality of a single camera. Both metrics are crucial for ensuring reliable depth estimation and robust 3D reconstruction.

4.1.7 Synchronization and Image Cleaning

Synchronization ensures that images from different cameras are captured at the same timestamp. This is essential for accurate stereo calibration.

The synchronization process finds pairs of images where the timestamps differ by less than a defined threshold (e.g., 60 ms) :

$$|t_1 - t_2| < 60ms \tag{4.12}$$

This allows the correct association of frames across cameras.

Image cleaning ensures that only images with a detected checkerboard pattern are kept. This removes outlier frames that could negatively impact calibration accuracy.

In the cleaning step :

— The presence of a checkerboard is verified

— Images without detectable patterns are discarded.

This step ensures that only high-quality images are used for calibration, improving overall precision.

4.2 Pose Estimator

4.2.1 Benchmark Data and Scope of our Evaluation

Our primary contribution is a complete pose–guided *camera–view–selection pipeline* for cobotic finishing, validated on our proprietary multi-view industrial dataset (**MuViH**). A preliminary comparison of two off-the-shelf human-pose estimators, OpenPose and MediaPipe Pose, was necessary to decide which backbone to embed in that pipeline. We supplemented the internal quantitative study with a *small, open* image set.

We therefore curated fifteen full-resolution photographs from iStockPhoto [2] depicting adults in varied upright poses (e.g. pointing, leaning, arms raised).

We manually labelled the standard joints corresponding to each application. The Percentage of Correct Keypoints normalised by head length ($\mathrm{PCK_h}$) was computed with a 0.2 h threshold :

$$\mathrm{PCK_h} = \frac{1}{N}\sum_{j=1}^{N}[\,\|\widehat{\mathbf{p}}_j - \mathbf{p}_j\|_2 < 0.2\,\mathrm{head\_len}\,]\,. \tag{4.13}$$

4.2.2 Comprehensive Analysis

To evaluate the performance of OpenPose and MediaPipe, two metrics were considered. The first was PCKh, or Percentage of Correct Keypoints with respect to head size. This metric considers a keypoint correct if it falls within a threshold distance from the ground truth, normalized by the subject's head length. This normalization makes the metric scale invariant, making it especially appropriate for evaluating models under varying human proportions and camera perspectives.

The second metric was runtime per frame, which measures the time required for the model to process a single image. This directly reflects the model's suitability for integration into real time systems, particularly in environments such as cobotic cells, where pose estimation must operate efficiently across multiple camera views without introducing latency.

4.2.3 Operator Selection Strategy :

Since OpenPose detects multiple human skeletons in the scene, the system must determine which skeleton belongs to the operator. To achieve this, the framework **selects the closest person to the camera as the operator** based on the **neck keypoint's y-coordinate**. The detected person with the **lowest y-value** (i.e., highest vertical position in the image) is

2. `https://www.istockphoto.com`

chosen, while all other skeletons are ignored.

4.3 Camera Selection

4.3.1 MuViH Dataset

This study utilizes the Multi-View Hand Gesture Dataset (MuViH) [36], developed by Corentin Hubert et al. The MuViH dataset is specifically designed for human-robot interaction within a cobotic platform and features multi-view hand gesture recognition with occlusion challenges. It has been collected in the cobotic platform at the NRC. In this work, camera visibility was encoded using binary labels, where a value of 1 indicated that a camera had a clear view of the participant performing a gesture, and 0 denoted that the view was obstructed or not informative. These labels were determined based on the known zone in which the gesture was performed and the fixed spatial arrangement of the cameras around the workspace. During data collection, each gesture sequence was associated with one of four predefined zones in the cobotic cell. A zone-to-camera visibility matrix, constructed empirically, defined which cameras had effective viewpoints for each zone. This binary visibility information was used as the ground truth in training the camera view selection model, enabling the system to learn which camera views to prioritize for each gesture occurrence without requiring manual labeling or pose-based supervision.

4.3.2 Subset dataset

For the camera selection part, a representative subset of the MuViH dataset was selected to facilitate focused and efficient analysis of camera view selection in a realistic human-robot interaction scenario. The full MuViH dataset comprises multi-view recordings of twenty participants performing predefined pointing gestures around a collaborative robot, resulting in a large volume of high-resolution RGB-D data. Processing the entire dataset would significantly increase computational cost and introduce unnecessary redundancy.

To ensure diversity while maintaining tractability, a subset was curated from recordings of thirteen participants. This subset was chosen to provide sufficient variability in user height, gesture style, occlusion conditions, and workspace zones. In total, 1300 synchronized frames were extracted, each comprising image pairs captured from selected camera viewpoints, making it well-suited for evaluating camera ranking metrics and view selection strategies.

— **Visibility and confidence scores :** help determine which camera views provide the best pose estimation.

— **Handles Occlusions & Partial Visibility** : Low confidence scores signal occlusions, helping the model select a better camera.
— **Standardized Representation** : Following the COCO format ensures compatibility with machine learning-based models.

4.3.3 Features

To effectively determine the best camera views, a set of features is extracted from each camera's input. These features facilitate the ranking of cameras based on their ability to accurately capture human pose data and contribute to reliable pose estimation.

1. **Pose Representation :** Each camera detects an 18-keypoint human pose, represented as an 18×3 matrix containing the (x, y) coordinates and confidence scores. This standardized format ensures consistent representation of body joints across different camera views, forming the foundation for camera selection.
2. **Feature Vector Transformation :** The extracted pose matrix is converted into a 54-dimensional feature vector, which serves as input to the camera selection model. This transformation enables machine learning models to process pose-related information efficiently in a structured numerical format.
3. **Confidence-Based Camera Ranking :** The model prioritizes cameras based on the confidence scores of detected keypoints. Lower confidence scores indicate potential occlusions or poor visibility, helping to filter out unreliable camera views.
4. **Occlusion Detection and Handling :** By analyzing missing keypoints or low-confidence detections, the system identifies occlusions and assigns lower rankings to obstructed cameras. Cameras capturing occluded body parts provide limited pose information.
5. **Random Forest-Based Feature Learning :** The Random Forest model continuously learns the significance of each feature, refining the camera ranking process across multiple decision trees. Over time, the model adapts, enhancing its capability to select the most suitable camera views based on past data.
6. **Balanced Learning Strategy :** To prevent class imbalance, a Random Under-Sampler (RUS) is applied, ensuring fair representation of all camera views. This technique mitigates dataset bias, preventing the model from favoring specific cameras and improving its generalization capability.

By leveraging these features, the system dynamically selects the most effective camera subset for real-time pose estimation, reducing redundancy, improving computational efficiency, and

minimizing occlusion effects.

4.3.4 Model : Random Forest (RF)

A **Random Forest (RF) classifier** is used to predict the best camera subset in `real-time`. The RF model is selected because :

— High interpretability allows analysis of the importance of the features.
— Fast inference is suitable for `real-time` applications.
— Robustness to nonlinearity, important for handling occlusions and varying viewpoints.

Training Process :

For training the random forest model for camera view selection, the dataset was constructed and annotated using the MuViH zone-to-camera mapping derived from the physical layout of the cobotic cell. Based on the picture 5.1 the cell is divided into four distinct areas (Area 1 to Area 4) based on operator position around the table. Each area has a predefined set of cameras that provide optimal coverage, determined from the multi-camera geometry and occlusion analysis conducted in the MuViH study. This mapping is stored as a `zone_map` dictionary, for example : Zone 1 $\rightarrow$ {C4, C6}, Zone 2 $\rightarrow$ {C2, C4, C5}, Zone 3 $\rightarrow$ {C2, C3, C5}, Zone 4 $\rightarrow$ {C1, C3}.

Ground-truth labels for training were generated automatically by linking the operator's annotated zone (sourced from column 3 of the MuViH annotation CSV file) to the corresponding optimal camera set in the `zone_map` dictionary. For each annotated frame, a binary visibility vector was created, indicating for each of the six cameras whether it belonged to the optimal set for that zone (value 1) or not (value 0). This binary encoding served as the target output for the random forest classifier.

1. **Input Features :** The RF model actually takes six synchronized skeletons, extracted from six synchronized images, as input. These skeletons are represented as (18×3) keypoint matrices, which are then transformed into a 54-dimensional feature vector per image before training. The final input to the RF model consists of concatenated feature vectors from all six images, ensuring that the model evaluates multiple viewpoints simultaneously to select the optimal camera.
2. **Target Output :** The model produces two outputs to optimize camera selection. First, it ranks cameras based on their predicted probabilities using the `predict_proba` function, ensuring that the most relevant views are prioritized. Then, it performs binary classification using the `predict` function, where a probability threshold of 0.5 is applied : if the predicted probability for a camera is greater than 0.5, it is assigned

a label of 1 (indicating that it should be used) ; otherwise, it is assigned a label of 0 (indicating that it should be ignored). This dual-output approach enables the system to not only select the best camera but also quantify the confidence level of each prediction, enhancing the robustness of the camera selection process.

3. **Hyperparameters Optimized :** The Random Forest model is optimized using the following parameters, explicitly set in the implementation :
   — **Number of Trees** (`n_estimators = 100`) : Defines the number of decision trees in the forest, ensuring stable and robust predictions.
   — **Maximum Tree Depth** (`max_depth = None`) : Allows trees to expand fully unless limited by other stopping criteria, enabling the model to capture complex relationships in the data.
   — **Feature Selection Strategy** (`max_features = "sqrt"`) : Selects a subset of features (square root of total features) for each split, balancing computational efficiency and predictive accuracy.

   These hyperparameters are selected to enhance model performance, prevent overfitting, and ensure computational efficiency in `real-time` camera selection.
4. **Real-time Processing** : At each frame, all cameras extract features, and the RF model ranks cameras based on their feature importance. The top-ranked cameras remain active while others are ignored, ensuring computational efficiency and reducing redundant views.

4.3.5 Split of the Data

The dataset for camera selection consists of frames collected from multiple cameras capturing the same scene. Data is split into training and testing sets as follows :

— **Dataset Composition :** Multi-view images from six cameras, labeled based on the best subset of cameras.
— **Train-Test Split (80-20%)** :
  — **80% Training :** Used to train the RF model.
  — **20% Testing :** Used to evaluate performance on unseen data.

4.3.6 Leave-One-Out Cross-Validation (LOOCV)

To ensure generalization, we use **Leave-One-Out Cross-Validation (LOOCV)** :

1. Consider N=13 users. Remove one user from the dataset and use it as the test set.

2. Train the model on the remaining $N - 1$ users.
3. Test the model on the removed user.
4. Repeat for all users and compute the average performance.

LOOCV is chosen because it is ideal for small datasets with limited participants and ensures robust performance evaluation across different users.

4.3.7 Metrics

To evaluate the performance of the camera selection model, multiple metrics are used :

— **Accuracy :** Measures the proportion of correctly selected cameras.

$$\text{Accuracy} = \frac{\text{Correct Camera Selections}}{\text{Total Camera Selections}} \tag{4.14}$$

— **Precision and Recall :**

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.15}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.16}$$

— **Normalized Discounted Cumulative Gain (nDCG) :** Evaluates how well the model ranks the best cameras.

$$\text{nDCG} = \frac{DCG}{IDCG} \tag{4.17}$$

where DCG scores the ranked camera subset and IDCG represents the ideal ranking.

CHAPITRE 5 RESULTS AND DISCUSSIONS

5.1 Calibration Accuracy

The calibration process was evaluated in terms of both intrinsic and extrinsic accuracy. Intrinsic parameters were obtained through single-camera calibration, while extrinsic parameters were computed using stereo calibration between Camera 1 (reference camera) and each of the other five cameras. Calibration accuracy was assessed using the mean squared reprojection error (MSE) before and after optimization.

Intrinsic Parameters

The intrinsic matrices for each of the six `RGB-D` cameras, as estimated through calibration, are as follows. These matrices represent the internal geometry of the cameras, including focal lengths and principal points :

$$K_1 = \begin{bmatrix} 525.80 & 0.00 & 649.19 \\ 0.00 & 533.55 & 364.42 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \quad K_2 = \begin{bmatrix} 901.24 & 0.00 & 651.67 \\ 0.00 & 987.37 & 352.56 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$K_3 = \begin{bmatrix} 530.93 & 0.00 & 636.63 \\ 0.00 & 535.21 & 366.10 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \quad K_4 = \begin{bmatrix} 541.57 & 0.00 & 656.17 \\ 0.00 & 572.17 & 369.52 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$K_5 = \begin{bmatrix} 608.19 & 0.00 & 618.02 \\ 0.00 & 615.68 & 368.42 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \quad K_6 = \begin{bmatrix} 653.53 & 0.00 & 641.93 \\ 0.00 & 604.52 & 399.10 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

To validate the calibration process, we compared these results with the factory-provided intrinsic parameters from the RealSense D455 devices. These manufacturer-reported matrices are derived from internal calibration during production :

$$K_1^{\text{manufacturer}} = \begin{bmatrix} 644.618 & 0.00 & 656.458 \\ 0.00 & 643.832 & 358.013 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \quad K_2^{\text{manufacturer}} = \begin{bmatrix} 638.667 & 0.00 & 645.022 \\ 0.00 & 637.174 & 363.420 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$K_3^{\text{manufacturer}} = \begin{bmatrix} 645.000 & 0.00 & 645.841 \\ 0.00 & 643.521 & 362.563 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \quad K_4^{\text{manufacturer}} = \begin{bmatrix} 644.817 & 0.00 & 637.674 \\ 0.00 & 643.132 & 358.101 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$K_5^{\text{manufacturer}} = \begin{bmatrix} 642.836 & 0.00 & 650.272 \\ 0.00 & 641.204 & 357.739 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \quad K_6^{\text{manufacturer}} = \begin{bmatrix} 644.283 & 0.00 & 649.796 \\ 0.00 & 642.607 & 362.962 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

When comparing the calibration results to the manufacturer's values, it is evident that the estimated focal lengths and principal points are reasonably close for most cameras, validating the reliability of the checkerboard-based calibration procedure. However, some discrepancies, particularly in Camera 2 and Camera 3, suggest slight deviations due to limited frame coverage or insufficient angle diversity during calibration. These variations remain within acceptable tolerances for RGB-D applications and are further minimized during stereo calibration and reprojection error optimization.

Extrinsic Calibration and Reprojection Error

Stereo calibration was performed between Camera 1 and each of the remaining cameras. The rotation matrices $R_{1,j}$ and translation vectors $t_{1,j}$ express the spatial transformation from Camera 1 to Camera j, for $j = 2$ to 6. The calibration accuracy was evaluated by computing the mean squared reprojection error (MSE), which measures the squared distance between the original and reprojected checkerboard corners in pixel space. Table 5.1 summarizes the initial and optimized MSE values for each stereo camera pair. The results show a significant decrease in error after optimization, indicating strong geometric consistency across views.

Tableau 5.1 Initial and final mean squared reprojection error (MSE) for each stereo calibration pair involving Camera 1.

Camera Pair	**Initial MSE(pixel)**	**Final MSE(pixel)**
1–2	123.58	0.32
1–3	2600.59	4.46
1–4	15.81	0.48
1–5	174.70	0.46
1–6	4.17	0.04

The variation in stereo calibration accuracy observed across different camera pairs can be attributed in part to the physical layout of the camera setup within the cobotic cell.

Figure 5.2 presents a visual example of the reprojection result obtained for one of the image pairs used in the stereo calibration process, specifically between Camera 1 and Camera 2. In this image, the detected true corners of the checkerboard are marked in blue, while the

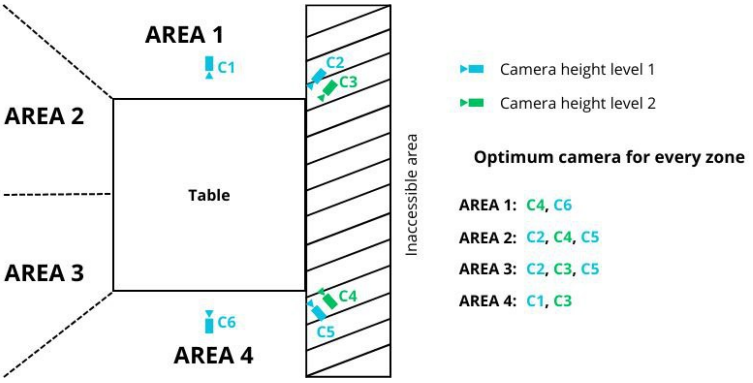


FIGURE 5.1 Camera layout used in the MuViH dataset, adapted from Hubert et al. The cameras are distributed asymmetrically around the collaborative robot at two height levels to optimize workspace visibility.

reprojected corners based on the estimated camera parameters are shown in red. This overlay demonstrates a high degree of alignment between the true and reprojected points. The mean squared reprojection error (MSE) for this camera pair was calculated to be only **0.046 pixels**, highlighting the effectiveness and precision of the calibration framework. Both qualitatively, in terms of visual alignment, and quantitatively, in terms of low reprojection error, the result confirms that the calibration method produces geometrically consistent projections suitable for accurate stereo vision applications in the cobotic cell.

The following matrices show the estimated rotations and translations between Camera 1 and the other five cameras :

$$R_{1,2} = \begin{bmatrix} -0.75 & -0.30 & 0.59 \\ 0.33 & 0.60 & 0.73 \\ -0.57 & 0.74 & -0.35 \end{bmatrix} \quad t_{1,2} = \begin{bmatrix} -54.19 \\ -92.79 \\ 376.40 \end{bmatrix}$$

$$R_{1,3} = \begin{bmatrix} -0.19 & -0.28 & 0.94 \\ 0.73 & 0.60 & 0.33 \\ -0.65 & 0.75 & 0.09 \end{bmatrix} \quad t_{1,3} = \begin{bmatrix} -164.75 \\ -4.84 \\ 167.37 \end{bmatrix}$$

$$R_{1,4} = \begin{bmatrix} -0.80 & -0.56 & 0.23 \\ 0.03 & 0.36 & 0.93 \\ -0.60 & 0.74 & -0.29 \end{bmatrix} \quad t_{1,4} = \begin{bmatrix} -173.32 \\ -57.87 \\ 280.44 \end{bmatrix}$$

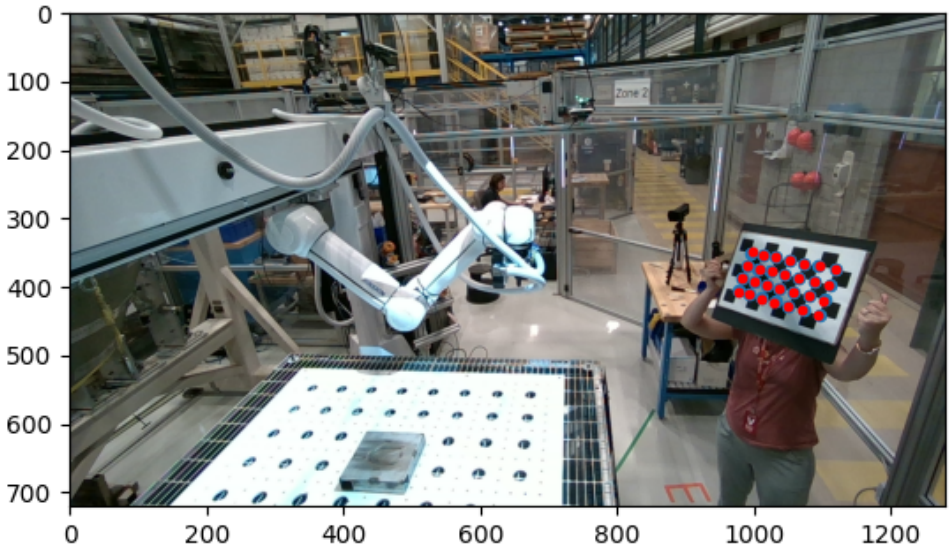


FIGURE 5.2 Blue dots represent the true corners in Camera 2's frame, while red dots represent the reprojected corners based on the estimated calibration parameters (from Camera 1's frame). MSE is 0.046 pixels.

$$R_{1,5} = \begin{bmatrix} -0.34 & -0.82 & 0.46 \\ 0.68 & 0.11 & 0.73 \\ -0.65 & 0.56 & 0.51 \end{bmatrix} \quad t_{1,5} = \begin{bmatrix} -27.63 \\ -63.46 \\ 432.31 \end{bmatrix}$$

$$R_{1,6} = \begin{bmatrix} -0.79 & -0.59 & 0.37 \\ 0.40 & 0.07 & 0.91 \\ -0.46 & 0.87 & 0.13 \end{bmatrix} \quad t_{1,6} = \begin{bmatrix} -48.06 \\ -107.10 \\ 312.19 \end{bmatrix}$$

5.1.1 Stereo-Triangulated Distance Validation

To assess the accuracy of the stereo triangulation process, each camera pair was evaluated using annotated points corresponding to two circular holes on the cobotic cell's tabletop. The real-world center-to-center spacing between these circles was obtained from the engineering drawing shown in Figure 5.4, which specifies a distance of 50.8 mm between adjacent holes.

This dimension was used as the ground truth for error computation.

After executing the calibration code, the intrinsic and extrinsic parameters of each stereo pair were applied to a stereo triangulation procedure. This produced the three-dimensional coordinates of the annotated points in each image pair. The Euclidean distance between the reconstructed 3D points was then computed and compared to the ground truth to yield the absolute error in millimeters.

The stereo-triangulation evaluation was carried out on five different stereo pairs, each composed of Camera 1 and one of the other cameras in the setup (C_2–C_6). Figure 5.3 shows these camera pairs : the top row displays the first camera's view for each pair, while the bottom row displays the corresponding second camera's view. In each image, a checkerboard calibration target is present to ensure adequate coverage for accurate calibration. The tabletop with its circular holes is also visible, serving as the reference object for the triangulation validation.

The results, summarized in Table 5.2, indicate that the absolute distance error for all pairs lies between approximately 5 mm and 9 mm. Considering the inherent depth noise of RGB-D sensors, residual calibration imperfections, and manual point selection uncertainty, this range represents a robust level of accuracy. In practice, achieving a sub-centimeter error in a multi-camera RGB-D setup is a strong indication that the calibration process was successful [**?**].

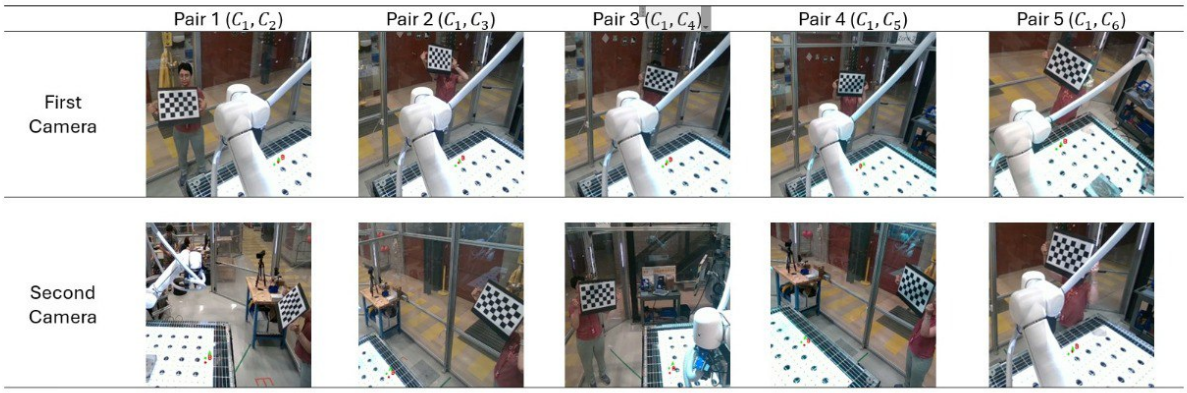


FIGURE 5.3 Stereo image pairs used in the triangulation validation. Each column represents one stereo pair composed of Camera 1 and another camera in the system (Pairs 1–5 : C_1–C_2, C_1–C_3, C_1–C_4, C_1–C_5, and C_1–C_6). The top row shows the first camera's view, and the bottom row shows the corresponding second camera's view for each pair.

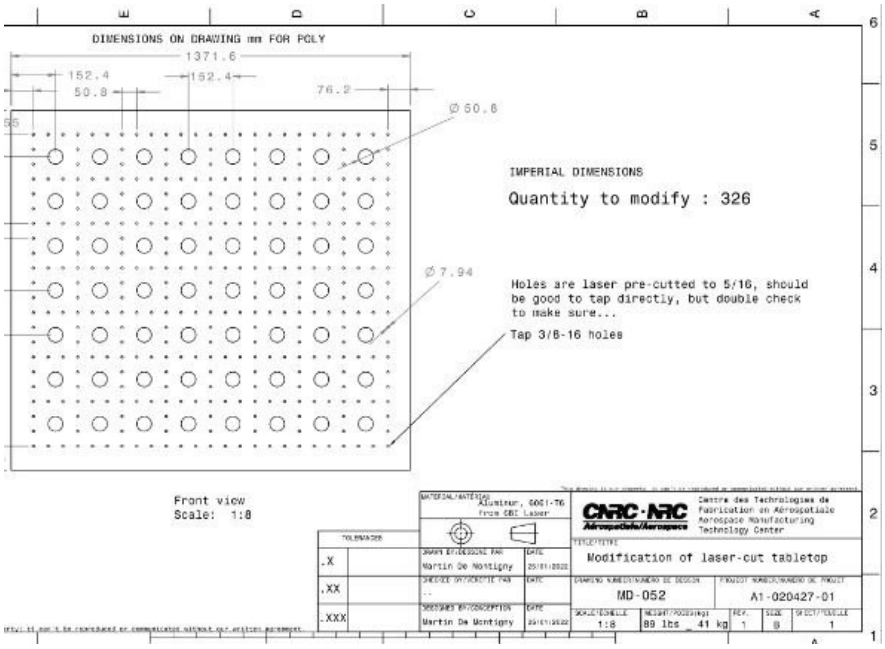


FIGURE 5.4 Engineering drawing of the cobotic cell tabletop, indicating the precise 50.8 mm hole spacing used as the ground truth for stereo-triangulation validation (adopted from NRC team).

5.2 Pose Estimator Selection

In the context of real-time camera view selection within a collaborative robotic (cobotic) cell, the choice of pose estimation framework plays a critical role in system responsiveness and reliability. To identify the most suitable solution, we conducted a comparative evaluation between two widely used frameworks : OpenPose and MediaPipe. Both were tested using the same small high-resolution dataset manually assembled from royalty-free images on the iStockPhoto platform. This dataset contains fifteen full-body images of individuals captured in varied standing poses, such as reaching, pointing, leaning, or with one or both arms raised. The subjects include both male and female adults, dressed in different clothing styles and captured under well-lit conditions. All images were saved in JPEG format at resolutions of approximately 1500 $\times$ 2000 pixels, allowing detailed keypoint analysis. This dataset was designed to provide a controlled, diverse testbed for visual inspection and keypoint evaluation in pose estimation frameworks.

OpenPose, evaluated in its lightweight configuration trained on the COCO dataset, detects eighteen human keypoints corresponding to major joints and body extremities. This version is specifically optimized for applications where speed is a priority, offering a favorable balance between keypoint accuracy and computational efficiency. In our tests, the lightweight

Tableau 5.2 Stereo-triangulated distance between annotated table circles for each camera pair, with ground truth 50.8 mm. Errors of 5–9 mm indicate high calibration accuracy for this multi-camera RGB-D system.

Camera pair	Measured distance (mm)	Abs. error (mm)	MSE (mm^2)	Best pixel MSE (px^2)
$C_1 \times C_2$	56.377	5.577	31.1029	0.434
$C_1 \times C_3$	44.865	5.935	35.2242	20.320
$C_1 \times C_4$	57.715	6.915	47.8172	5.117
$C_1 \times C_5$	45.201	5.599	31.3488	286.693
$C_1 \times C_6$	56.123	5.323	28.3343	0.234

OpenPose model achieved a PCKh (Percentage of Correct Keypoints with respect to head size) of approximately 83.2 percent on the small dataset and maintained an average inference runtime of about 0.08 seconds per frame on a standard GPU. This PCKh score was computed based on manual annotation of keypoints in the dataset(ground-truth), which was carried out to enable objective evaluation of model accuracy. Therefore, the evaluation in this work is not only qualitative, but also includes quantitative comparison using annotated ground truth. OpenPose's ability to preserve pose consistency under partial occlusion and its robustness to lighting variations make it well suited for the industrial conditions found in our cobotic environment.

MediaPipe, on the other hand, uses a top down architecture and detects a denser skeletal structure composed of thirty three keypoints. This allows for finer localization of hand, foot, and facial landmarks, which is particularly useful for detailed gesture analysis. However, this increased granularity comes with a computational cost. MediaPipe's average runtime was measured at 0.19 seconds per frame on the same hardware, which is significantly slower than OpenPose and introduces latency that can affect the timeliness of camera view decisions. While its PCKh was slightly higher at 85.4 percent, the performance gap in runtime presents a challenge for camera systems where inference speed must remain consistently low to avoid processing bottlenecks.

Figure 5.5 presents a qualitative comparison between the two frameworks. As shown, MediaPipe provides a more detailed skeletal output, while OpenPose offers a structurally coherent pose with noticeably faster inference.

Given the demands of our system, including synchronized camera input, robustness to occlusion, and strict real time processing requirements, OpenPose was selected as the more appropriate framework. While MediaPipe's detailed skeletal output may benefit certain fine grained tracking applications, its increased inference time and GPU load reduce its practicality for fast, cycle accurate camera view decisions in our application. The lightweight OpenPose

FIGURE 5.5 Qualitative comparison between **OpenPose** (top row) and **MediaPipe** (bottom row) on the same test images. MediaPipe yields a higher number of keypoints but requires more computation, whereas OpenPose provides structurally consistent poses with lower latency.

configuration, by contrast, enables low latency multi stream processing while maintaining competitive accuracy, making it ideal for integration with our camera selection module in the cobotic cell.

5.2.1 Qualitative Results of selected Pose Estimator

Figure 5.6 illustrates the skeleton predictions produced by the lightweight OpenPose model for a single gesture observed simultaneously by six different RGB-D cameras placed around the cobotic cell. The person is performing a pointing gesture, and the model successfully detects all major joints including shoulders, elbows, wrists, hips, knees, and ankles. Despite changes in viewpoint, occlusions, and lighting conditions, the keypoints are consistently localized across views, confirming the robustness of the lightweight OpenPose configuration. In particular, the system retains anatomical coherence in estimating joint connections even when the operator's limbs are partially obscured by the robot structure or appear foreshortened due to camera angle. This strong spatial coherence across multiple views further supports the viability of this model for downstream tasks such as triangulation and camera view selection in cobotic environments.

5.3 Camera Selection Results

The camera selection model was trained and evaluated on pose-derived features from thirteen users, with the aim of identifying the most informative set of camera views per frame in a cobotic environment. A Random Forest classifier was employed to make these predictions based on extracted features, including per-joint confidence scores, positional coordinates,

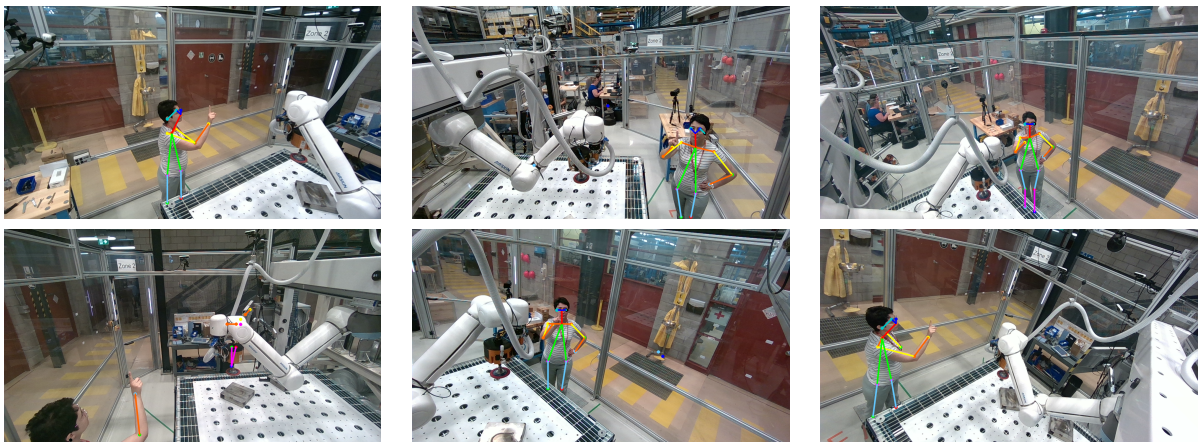


FIGURE 5.6 Pose skeleton outputs from six different RGB-D camera views using lightweight OpenPose. Despite varying angles and partial occlusions, all major joints are accurately detected, preserving the structure and gesture of the human operator.

and the number of visible keypoints.

Among these, the feature importance analysis revealed that the **confidence values of detected keypoints** were the most influential predictors in the model. This aligns with the intuition that frames where joints are confidently detected are more informative for spatial reasoning and gesture interpretation. High-confidence keypoints allow the model to reliably determine operator posture, which in turn makes it easier to identify which camera views are best suited for coverage.

The **number of visible keypoints** was also found to be a highly significant feature. Since occlusions and viewpoint changes frequently affect visibility in multi-camera systems, the presence of more visible joints in a given frame increases the likelihood that the camera has a clear, unobstructed view of the operator. Therefore, frames with a greater number of detected keypoints serve as stronger candidates for selection.

Together, these two features—keypoint confidence and keypoint visibility—played a dominant role in the camera selection decision process. Their significance underlines the importance of pose estimation quality as a prerequisite for intelligent view selection in cobotic systems. This observation also provides further motivation for our choice of a pose estimator that prioritizes both detection accuracy and real-time efficiency.

5.3.1 Qualitative Results

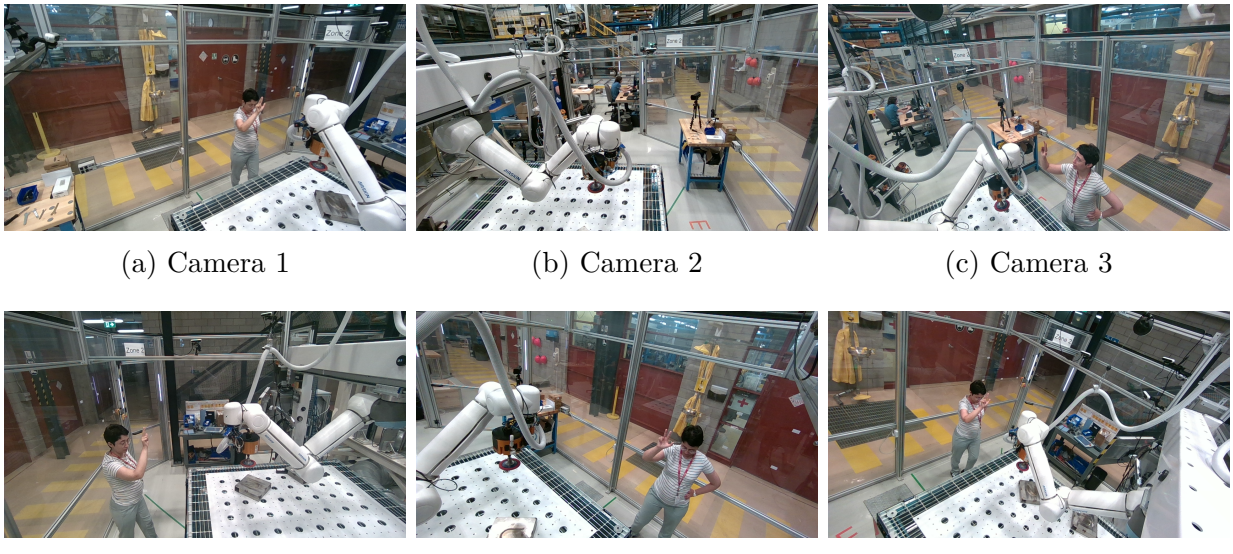
To better illustrate how our model selects the most informative camera views, we present a qualitative example5.5 involving a set of six synchronized frames captured from different

angles in the cobotic workspace. For each of these views, a skeleton was extracted from the operator's pose, and a set of features was derived. These features were then passed through a trained **Random Forest classifier**, which estimated the probability of each camera being the most relevant view.

In this specific example, the model assigned the following probabilities to each camera : 0.78 (Camera 1), 0.03 (Camera 2), 0.13 (Camera 3), 0.90 (Camera 4), 0.59 (Camera 5), and 0.90 (Camera 6). The selection framework uses a threshold of 0.5 to determine which camera views should be selected or excluded based on their predicted probabilities. As a result, Camera 4, Camera 5, Camera 6, and Camera 1 were selected. This decision corresponds to a binary activation vector of **[1, 0, 0, 1, 1, 1]**, where '1' indicates a selected view and '0' denotes an excluded one.

The selected views provide rich visual coverage of the scene. Camera 1, Camera 4, camera 5 and Camera 6 capture the operator from different angles, both offering clear visibility of the operator's skeleton for interpreting interactions in a cobotic cell.

This example confirms the model's ability to assign probabilistic relevance scores and intelligently activate up to four camera views with the highest probabilities exceeding a threshold of 0.5, ensuring efficient and context-aware visual monitoring in real time.



(a) Camera 1 (b) Camera 2 (c) Camera 3

(d) Camera 4 (e) Camera 5 (f) Camera 6

FIGURE 5.7 Six synchronized camera views from a single frame. Based on predicted probability, the model selected Cameras 1 (0.78), 4 (0.90), 5 (0.59), and 6 (0.90) as the most informative. Cameras 2 (0.03) and 3 (0.13) were excluded due to lower predicted probability.

5.3.2 Quantitative Results

To evaluate the effectiveness of the proposed camera view selection strategy, we conducted a quantitative analysis using a Leave-One-Out Cross-Validation (LOOCV) approach. The dataset consisted of 13 users, each performing the same collaborative task under the same camera setup. In each fold of LOOCV, the model was trained on data from 12 users and evaluated on the one remaining user. This process was repeated until every user had been used once as the test subject. The final performance metrics were computed as the average across all 13 folds.

5.4 Camera Selection Results

The camera selection model was evaluated across thirteen users to verify its consistency and generalizability in dynamic cobotic environments. Each input frame was represented by confidence scores extracted from 18 human keypoints, and a Random Forest classifier was used to predict which camera views were most informative. The model was configured to activate up to three cameras whose predicted probabilities exceeded a 0.5 threshold.

The camera selection model was evaluated across thirteen users to verify its consistency and generalizability in dynamic cobotic environments. Each input frame was represented by confidence scores extracted from 18 human keypoints, and a Random Forest classifier was used to predict which camera views were most informative. The model was configured to activate up to three cameras whose predicted probabilities exceeded a 0.5 threshold.

The results, summarized in Table 5.3, demonstrate that the system performs consistently and accurately across different users. The average **precision** achieved was **92.3%**, meaning that when the system predicted a camera view as optimal, it was correct in the vast majority of cases. The **recall** was even more impressive, reaching **100%**, indicating that the model successfully captured all the relevant views without missing any important frame—a critical requirement in cobotic environments where safety and visibility are essential.

In addition to precision and recall, we also evaluated the ranking quality using the **Normalized Discounted Cumulative Gain (NDCG)**, which assesses the system's ability to not just identify the best view but also rank other views based on relevance. The NDCG score averaged **0.9938**, confirming that the system's predicted rankings were nearly perfect.

Lastly, the overall **accuracy** of the classifier was measured at **86.4%**, reinforcing the reliability of using pose-based features combined with machine learning for dynamic camera selection. These metrics collectively highlight the strength of our approach : high performance across multiple dimensions, generalizability across users, and suitability for real-time

deployment in collaborative human-robot workspaces.

Tableau 5.3 Camera selection performance metrics per user

User	Precision	Recall	Accuracy	NDCG
User 1	0.928	1.000	0.940	0.847
User 2	0.922	1.000	0.941	0.878
User 3	0.929	1.000	0.947	0.858
User 4	0.938	1.000	0.945	0.864
User 5	0.921	1.000	0.951	0.858
User 6	0.921	1.000	0.945	0.907
User 7	0.939	1.000	0.943	0.870
User 8	0.931	1.000	0.957	0.849
User 9	0.918	1.000	0.949	0.886
User 10	0.928	1.000	0.950	0.846
User 11	0.918	1.000	0.943	0.874
User 12	0.918	1.000	0.947	0.831
User 13	0.925	1.000	0.950	0.843
Average	**0.923**	**1.000**	**0.950**	**0.864**

To further support the system's applicability in real-time cobotic environments, we measured the total runtime of the camera selection pipeline, from pose estimation to final ranking decision. Using the lightweight configuration of OpenPose, the average inference time per frame for pose extraction was approximately **0.08 seconds** on a standard GPU. The subsequent feature extraction and Random Forest, based camera ranking step introduced negligible additional overhead, with an average runtime of less than **0.01 seconds** per frame. As a result, the entire camera selection process completes in approximately **0.09 seconds per frame**, confirming that the system is well suited for real-time operation at frame rates

CHAPITRE 6 CONCLUSION

6.1 Summary of Works

This thesis explored the development of a vision-based system for enhancing human-robot interaction in a collaborative industrial setting. Within a cobotic part-finishing cell equipped with six Intel RealSense D455 RGB-D cameras, the work addressed three key research challenges : camera calibration, human pose estimation, and adaptive camera view selection.

The first objective was to establish a reliable calibration pipeline to align all camera views into a unified coordinate system. A checkerboard-based approach was implemented to estimate the intrinsic and extrinsic parameters of each camera. Calibration accuracy was assessed through the reprojection error metric. The most mean squared reprojection errors (MSE) is appropriate. These results indicate successful geometric alignment and validate the calibration process across a multi-camera RGB-D setup.

The second objective involved applying human pose estimation using a method that offers both accuracy and computational efficiency. A comparative study led to the selection of a lightweight OpenPose model, which proved more suitable than MediaPipe for real-time, multi-camera deployment. OpenPose provided robust joint detection even under occlusion, with acceptable latency and resource usage.

The third objective focused on intelligent camera view selection to ensure that the system continuously prioritized the most informative viewpoints with regards to the operator while minimizing processing redundancy. A Random Forest classifier was trained using pose-related confidence features and evaluated across annotated samples. The resulting model achieved 94.99% accuracy, 92.3% precision, and 100% recall in identifying the best subset of camera views per frame.

Altogether, the proposed system met its goals by delivering a scalable and responsive solution for operator monitoring in industrial environments. It offers accurate pose detection, real-time adaptability, and efficient use of multi-camera data.

6.2 Limitations

While the system performed well in controlled settings, several limitations should be noted. One key limitation concerns the generalization of the calibration results. Although reprojection errors on checkerboard images were significantly reduced, applying the calibration

parameters to general scenes occasionally led to projection inconsistencies. This suggests that while the calibration was sufficient for geometric alignment, it may not fully capture all scene-level distortions.

In addition, the use of 2D keypoint estimators with triangulation introduces sensitivity to occlusion, synchronization offsets, and viewpoint disparity. Although the camera selection module helped reduce such issues by focusing on informative views, the system's accuracy still depended on the visibility of keypoints in at least two camera perspectives.

6.3 Future Research

To address these limitations and extend the capabilities of the system, several research directions are recommended.

First, while the current stereo calibration approach yielded satisfactory reprojection errors, its accuracy in non-checkerboard scenes can be improved. Future work should explore alternative calibration methods such as bundle adjustment, which jointly optimizes camera parameters and 3D scene geometry across all views. This could improve consistency in more complex and dynamic environments.

Second, further enhancements in the pose estimation pipeline could involve depth-guided refinement, hybrid RGB-D models, or learning-based fusion techniques to boost precision during occlusion or partial visibility.

Third, the view selection strategy could benefit from temporal coherence and higher-level reasoning. Sequence-based learning models or reinforcement learning agents could learn policies that anticipate occlusions and proactively select camera combinations that maintain joint visibility over time.

Finally, deploying the system in an operational industrial environment would allow for assessment of its long-term robustness and usability. Feedback from real operators and integration with robotic control systems would also support the transition from experimental prototype to production-ready application.

REFERENCES

[1] Q. Lei, H.-B. Zhang, J.-X. Du, T.-C. Hsiao, and C.-C. Chen, "Learning effective skeletal representations on rgb video for fine-grained human action quality assessment," *Electronics*, vol. 9, no. 4, p. 568, 2020.

[2] Google AI, "Mediapipe pose landmarker," https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker, 2023, accessed : April 14, 2025.

[3] L. K. Topham, W. Khan, D. Al-Jumeily, and A. Hussain, "Human body pose estimation for gait identification : A comprehensive survey of datasets and models," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–42, 2022.

[4] F. Ren and Y. Bao, "A review on human-computer interaction and intelligent robots," *International Journal of Information Technology & Decision Making*, vol. 19, no. 1, pp. 5–47, 2020.

[5] H. Park and S. McKilligan, "A systematic literature review for human-computer interaction and design thinking process integration," in *International Conference of Design, User Experience, and Usability*. Springer, 2018.

[6] C. Kerdvibulvech, "A review of augmented reality-based human-computer interaction applications of gesture-based interaction," in *International Conference on Human-Computer Interaction*. Springer, 2019.

[7] M.-A. Drouin and L. Seoud, "Consumer-grade rgb-d cameras," *3D Imaging, Analysis and Applications*, pp. 215–264, 2020.

[8] M. Tölgyessy, M. Dekan, Ľ. Chovanec, and P. Hubinský, "Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2," *Sensors*, vol. 21, no. 2, p. 413, 2021.

[9] T. Guzsvinecz, V. Szucs, and C. Sik-Lanyi, "Suitability of the kinect sensor and leap motion controller-a literature review," *Sensors*, vol. 19, no. 5, p. 1072, 2019.

[10] V. Lyubanenko and et al., "Multi-camera finger tracking and 3d trajectory reconstruction for hci studies," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2017.

[11] M. M. Loper, M. J. Lee, and G. D. Hager, "Mobile human-robot teaming with environmental tolerance," in *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2009.

[12] D. Surdilovic, K. Yamane, K. Okada, and Y. Nakamura, "Compliance control with dual-arm humanoid robots : Design, planning and programming," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2010.

[13] J. Han, S. Zhang, and K. Xue, "Enhanced computer vision with microsoft kinect sensor : A review," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[14] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration : A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, 2018.

[15] Intel Corporation, "Intel realsense depth camera d455," https://www.intelrealsense.com/depth-camera-d455/, 2021, accessed : April 14, 2025.

[16] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[17] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited : People detection and articulated pose estimation," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1014–1021.

[18] A. Toshev and C. Szegedy, "Deeppose : Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 1653–1660.

[19] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision–ECCV 2016 : 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, vol. 14. Springer International Publishing, 2016, pp. 483–499.

[20] X. Qin and et al., "Lightweight human pose estimation : Cvc-net," *Multimedia Tools and Applications*, vol. 81, no. 13, pp. 17 615–17 637, 2022.

[21] R. Sun and et al., "Research on human pose estimation," in *2022 IEEE 5th International Conference on Electronics Technology (ICET)*. IEEE, 2022.

[22] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 7291–7299.

[23] 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.

[24] M. Ghafoor and A. Mahmood, "Quantification of occlusion handling capability of 3d human pose estimation framework," *IEEE Transactions on Multimedia*, 2022.

[25] M. C. d. F. Macedo and A. L. Apolinario, "Occlusion handling in augmented reality : Past, present and future," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1606–1618, 2021.

[26] X. Xiang, R. Abdein, N. Lv, and J. Yang, "Self-supervised learning of scene flow with occlusion handling through feature masking," *Pattern Recognition*, p. 109487, 2023.

[27] T. Rathnayake, A. Khodadadian Gostar, R. Hoseinnezhad, R. Tennakoon, and A. Bab-Hadiashar, "On-line visual tracking with occlusion handling," *Sensors*, vol. 20, no. 3, p. 929, 2020.

[28] J. Zhang, J. Zhu, H. Deng, Z. Chai, M. Ma, and X. Zhong, "Multi-camera calibration method based on a multi-plane stereo target," *Appl. Opt.*, vol. 58, no. 34, pp. 9353–9359, Dec 2019. [Online]. Available : https://opg.optica.org/ao/abstract.cfm?URI=ao-58-34-9353

[29] M. Munaro, F. Basso, and E. Menegatti, "Openptrack : Open source multi-camera calibration and people tracking for rgb-d camera networks," *Robotics and Autonomous Systems*, vol. 75, pp. 525–538, 2016.

[30] Y. Xia, Y. Cai, W. Liu, and Z. Sun, "Rgb-d camera calibration with body tracking and feature matching," *Sensors*, vol. 21, no. 3, p. 1013, 2021.

[31] J. Kim, H. Kim, C. Kim, and J. Kim, "Sphere-based calibration method for multiple rgb-d cameras," *Optics Express*, vol. 28, no. 13, pp. 19 058–19 072, 2020.

[32] J. Wu, K. Zhang, and L. Zhao, "Targetless calibration of multiple rgb-d cameras using line feature convergence voting," *arXiv preprint arXiv :2404.13949*, 2024. [Online]. Available : https://arxiv.org/abs/2404.13949

[33] A. Sharma, S. Anand, and S. K. Kaul, "Intelligent camera selection decisions for target tracking in a camera network," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 3388–3397.

[34] F. Lefevre, V. Bombardier, P. Charpentier, and N. Krommenacker, "Context-based camera selection from multiple video streams," *Multimedia Tools and Applications*, vol. 81, no. 2, pp. 2803–2826, 2022.

[35] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[36] C. Hubert, N. Odic, M. Noël, S. Gharib, P. Debanné, L. Séoud, and S. Zargarbashi, "MuViH : Multi-View Hand gesture dataset for hand and gesture recognition," 2025. [Online]. Available : https://doi.org/10.5683/SP3/JZJTGG