

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DOCTORAL THESIS

Perception from the Robot's Perspective: Unified Calibration for Human Understanding and Robot-Object Interaction in Human-Robot Collaboration

Ph.D. Candidate:
Davide Allegro

Supervisor:
Prof. Stefano Ghidoni

Co-Supervisor:
Matteo Terreran, Ph.D.

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy in Information Engineering

at the

Department of Information Engineering (DEI)

38^{th} *Cycle*

January 8, 2026

Abstract

Modern robotics is at a turning point: shifting from rigid, pre-programmed automation to dynamic human-robot collaboration, where adaptability and intelligence drive the next generation of industrial production. The goal is no longer to replace human labor, but to augment it, fusing robotic precision and endurance with human creativity and intuition. To realize this vision, robots must not only move but see, understand, and efficiently act within the ever-changing complexity of the real world.

This thesis advances that frontier by proposing a unified framework that bridges perception and action, enabling robots to comprehend and interact within dynamic human environments. The foundation of this framework is a robot-centric perception model, which establishes a coherent and consistent geometric relationship between what the robot perceives and how it moves.

A core contribution of this work is the development of novel camera-to-robot calibration techniques that guarantee consistent spatial alignment across heterogeneous setups. These methods generalize to fixed external camera networks observing robotic arms, multi-camera systems mounted on manipulators, and mobile robots operating in unstructured environments, providing the robot with a unified visual reference, as if multiple eyes were coherently aligned within a single coordinate frame.

To enable effective collaboration, the framework equips robots with robust human understanding capabilities through a novel multi-view 3D human pose estimation method and a real-time skeleton-based action recognition module. Together, they allow the robot to interpret and anticipate human behavior, turning motion into meaningful collaboration.

Finally, the thesis addresses intelligent robot-object interaction through two distinct applications. First, for robot manipulators, we develop a few-shot imitation learning algorithm, guided by a world model, for complex manipulation tasks. Second, for mobile robots, we propose an active mapping strategy for autonomous navigation and reconstruction in unknown environments. This strategy integrates a Fisher Information-driven radiance field (Gaussian Splatting) model to select the most informative viewpoints, while motion prediction optimizes the path to those views. This allows the robot to explore efficiently, focusing attention where learning is most critical.

Together, these contributions unify geometry, semantics, and dynamics into a single perceptual-action loop. The resulting system advances the development of robots that can reason, adapt, and collaborate effectively in unstructured and dynamic environments. All proposed methods are released as open-source software to encourage transparency, reproducibility, and collective progress toward perceptually grounded, human-aware robotic intelligence.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. Stefano Ghidoni and Dr. Matteo Terreran, for their invaluable guidance, support, and encouragement throughout my doctoral journey. Their expertise and insights have been instrumental in shaping this thesis.

I am also grateful to my colleagues and friends at the Department of Information Engineering (DEI) and the IAS-Lab for their collaboration, stimulating discussions, and unwavering support. Special thanks to my lab mates for creating a friendly and inspiring environment.

Finally, I want to extend my heartfelt thanks to my family for their unconditional love, patience, and encouragement. Their belief in me has been a constant source of strength and motivation.

Contents

Abstract iii

Acknowledgements v

1 Introduction 1
- 1.1 Toward Intelligent Robotic Perception and Action . . . 1
  - 1.1.1 Robot-Centric Perception . . . 3
  - 1.1.2 Human Understanding . . . 3
  - 1.1.3 Robot-Object Interaction . . . 4
- 1.2 Thesis Contributions and Goals . . . 4
- 1.3 Thesis Structure . . . 5
- 1.4 List of Works . . . 6

2 Fundamentals 9
- 2.1 2D and 3D Vision Sensors . . . 9
  - 2.1.1 RGB Cameras . . . 9
  - 2.1.2 RGB-D Cameras . . . 10
- 2.2 Camera Modeling and Calibration . . . 11
  - 2.2.1 The Pinhole Camera Model . . . 11
  - 2.2.2 Camera Calibration . . . 11
- 2.3 Hand-Eye Calibration . . . 12
  - 2.3.1 Mathematical Formulations . . . 13
  - 2.3.2 Metric Evaluation . . . 16
- 2.4 Multi-view 3D Reconstruction . . . 18
- 2.5 Deep Learning for Visual Perception . . . 21
  - 2.5.1 Convolutional Neural Networks . . . 22
  - 2.5.2 Transformer-based Architecture . . . 23
- 2.6 Modern 3D Reconstruction with Deep Learning . . . 24
  - 2.6.1 DUSt3R and Its Successors . . . 25
    - MASt3R: Joint 3D Reconstruction and Feature Matching . . . 26
    - MonST3R: Extension to Dynamic Scene Reconstruction . . . 27
    - VGGT: Fully Feed-Forward 3D Reconstruction . . . 27

3 Robot-Centric Perception 29
- 3.1 Related-Works in Camera-to-Robot Calibration . . . 29
- 3.2 Camera-to-Robot Calibration Contributions . . . 30
- 3.3 A Unified Iterative Hand-Eye Calibration Method for Eye-on-Base and Eye-in-Hand Setups . . . 31
  - 3.3.1 Eye-in-Hand . . . 32
  - 3.3.2 Eye-on-Base . . . 33
  - 3.3.3 Experiments and Evaluation . . . 33
    - Simulated Robustness Analysis . . . 33
    - Real-World Validation . . . 34

3.4 Multi-Camera Hand-Eye Calibration . . . 36
3.4.1 Multi-Camera Hand-Eye Calibration for Human-Robot Collaboration in Industrial Workcells . . . 37
3.4.2 A Graph-Based Optimization Framework for Multi-Camera Hand-Eye Calibration . . . 39
3.4.3 The METRIC Dataset for Multi-Camera Calibration Evaluation . . . 41
Synthetic Dataset . . . 41
Real-World Dataset . . . 42
Performance Evaluation on Synthetic METRIC Data . . . 43
Performance Evaluation on Real-World METRIC Data . . . 44
3.4.4 Results on Real-World Industrial Scenarios . . . 45
3.5 MEMROC: Multi-Eye to Mobile Robot Calibration . . . 47
3.5.1 MEMROC Method . . . 47
Ground Plane Detection . . . 48
Ground Plane Validation . . . 49
Camera-to-Robot Calibration . . . 50
Multi-Sensor Joint Optimization . . . 51
3.5.2 MEMROC Dataset . . . 52
3.5.3 Experiments and Results . . . 53
Experiments on Synthetic Dataset . . . 53
Experiments on Real-World Scenarios . . . 55
3.5.4 Discussion . . . 56
3.6 Calib3R: A 3D Foundation Model for Multi-Camera to Robot Calibration and Metric-Scaled Reconstruction . . . 57
3.6.1 Preliminaries . . . 59
3.6.2 Calib3R . . . 60
3.6.3 Experimental Evaluation Protocol . . . 64
Metrics . . . 64
Manipulator-Mounted Camera Validation . . . 65
Mobile Robot Validation . . . 65
Computational Setup . . . 65
3.6.4 Results on Hand-Eye Calibration . . . 65
3.6.5 Results on Motion-Based Calibration . . . 69
3.6.6 Discussion . . . 71

4 Human Understanding **73**
4.1 3D Human Pose Estimation . . . 74
4.1.1 Related Work . . . 75
Multi-view 3D Human Pose Estimation . . . 75
Absolute 3D Human Pose Estimation . . . 75
Occlusion-Aware Human Pose Estimation . . . 76
4.1.2 Multi-view Pose Fusion for Occlusion-Aware 3D Human Pose Estimation . . . 76
Single-view 3D Pose Estimation . . . 76
Multi-view Fusion Module . . . 77
Optimization Module . . . 78
4.1.3 Experiments . . . 79
Experimental Settings and Metrics . . . 79
Results on Human3.6M and Human3.6M-Occluded . . . 81
Results on the Human-Robot Workcell Scenario . . . 82
Ablation Studies . . . 83

4.1.4 Discussion . . . 85
4.2 Online Action Recognition . . . 85
4.2.1 Related Work . . . 87
4.2.2 HAR Framework . . . 88
Sliding Window and Pose Normalization . . . 88
Ensemble of Binary Classifiers . . . 89
4.2.3 Collaborative Draping HAR Dataset . . . 90
4.2.4 Experimental Evaluation . . . 90
Collaborative Draping HAR Dataset . . . 91
Online Action Recognition in Laboratory Environment . . . 91
Online Action Recognition in Industrial Environment . . . 92
Runtime Analysis . . . 95
4.2.5 Discussion . . . 95

5 Robot-Object Interaction **97**
5.1 Dream to Manipulate: Compositional World Models Empowering Robot Imitation Learning with Imagination . . . 98
5.1.1 A Compositional World Model for Robotic Manipulation . . . 98
5.1.2 Related Work . . . 99
World Models . . . 99
Simulation-based Reconstructions . . . 100
Imitation Learning and Data Augmentation . . . 100
5.1.3 Dream-to-Manipulate with Gaussian Splats & Physics Engines . . . 101
Representing 3D Objects with Gaussian Splatting . . . 101
Decomposing the Scene into Object-centric Regions . . . 102
Modelling Robot-Object Dynamics with Object Assets . . . 102
Engineering Considerations for Real-World Deployment . . . 103
5.1.4 Equivariant Transformations for Imitation Learning . . . 104
5.1.5 Experiments . . . 105
One-shot Policy Learning: Single- and Multi-task . . . 105
Analysis and Ablations . . . 107
Experiments with a Real Robot . . . 107
5.1.6 Discussion . . . 109
5.2 Active Mapping for Dynamic Object Exploration . . . 110
5.2.1 Related Works . . . 111
Active Mapping for Static Scenes . . . 111
Mobile Object Passive Reconstruction . . . 112
5.2.2 Paparazzo . . . 112
5.2.3 Problem Formulation . . . 112
5.2.4 Initialization . . . 113
5.2.5 EKF-Based Motion Prediction . . . 113
5.2.6 Object Tracking Mode . . . 115
5.2.7 Object Mapping Mode . . . 117
5.2.8 Experimental Results . . . 118
5.2.9 Evaluation Protocol . . . 121
5.2.10 Results . . . 122
5.2.11 Additional Qualitative Results . . . 125
5.2.12 Additional Analysis . . . 126
5.2.13 Discussion and Limitations . . . 127

6 Conclusion **129**
6.1 Take-Home Messages and Scientific Impact 130
6.2 Limitations and Future Directions 131

List of Figures

1.1 Human-Robot Collaboration scenarios 2

2.1 Examples of RGB cameras: (left) the Sentera 65R ultra-high-resolution sensor, and (right) the Zenmuse P1 full-frame camera with stabilized gimbal. . . 10
2.2 Examples of RGB-D cameras: (left) the ZED 2i passive stereo camera, (center) the Intel RealSense D455 active stereo sensor, and (right) the Microsoft Azure Kinect ToF-based sensor. 10
2.3 Illustration of a camera projection model. Extrinsic parameters transform the 3D point from the world reference frame W to the camera coordinate system C. Intrinsic parameters maps the 3D point from the camera reference frame to the image plane. 12
2.4 Illustration of the two main configurations: on the left, *eye-in-hand* (camera on the end-effector). On the right, *eye-on-base* (fixed camera). 13
2.5 Kinematic loop for hand-eye calibration in eye-in-hand configuration, illustrating the equation $A_iX = XB_i$. The transformation A_i represents the relative motion of the camera ($T^{C_i}_{C_{i+1}}$), while B_i is the relative motion of the end-effector ($T^{E_i}_{E_{i+1}}$) 14
2.6 Kinematic diagram for the eye-on-base hand-eye calibration setup, which solves the equation $A_iX = ZB_i$. This formulation finds two static unknowns: the base-to-camera pose $X = T^C_W$ and the end-effector-to-board pose $Z = T^B_E$. 15
2.7 Geometric model of 3D point triangulation 19
2.8 Incremental Structure-from-Motion pipeline [1] 19
2.9 Sparse 3D reconstruction of Rome with 21K images, resulted from Structure-from-Motion process [1] 20
2.10 Example of a multi-view stereo pipeline. Clockwise: input imagery, posed imagery, reconstructed 3D geometry, textured 3D geometry [2] 21
2.11 Illustration of the convolution operation. The input tensor (blue) is processed by a convolutional filter (orange) to produce an output tensor (green). Each cell in the output corresponds to a weighted sum of a local receptive field in the input, followed by a non-linear activation. 22
2.12 At the left the Transformer architecture. At the right the Multi-Head Attention consisting of several attention layers running in parallel. 23
2.13 Architecture of DUSt3R [3]. Two input images (I_1, I_2) are encoded in a Siamese manner using a shared Vision Transformer encoder. The token representations F^1 and F^2 are processed by transformer decoders that exchange information through cross-attention layers, allowing the model to reason about geometric relationships between views. Finally, two regression heads predict the corresponding 3D pointmaps and their confidence maps, both expressed in the same coordinate frame of I_1. 25

2.14 Reconstruction example from 4 random frames of a RealEstate10K indoor sequence, after global alignment. On the left-hand side, the 4 input frames are depicted, and on the right-hand side the resulting point-cloud and the recovered camera intrinsics and poses [3]. 26
2.15 Architecture of MASt3R [4]. Given two input images (I_1, I_2) to match, MASt3R network regresses for each image and each input pixel a 3D point, a confidence value and a local feature. Plugging either 3D points or local features into a fast reciprocal NN matcher yields robust correspondences. Compared to the DUSt3R framework which it builds upon, the contributions are highlighted in blue. 26
2.16 MonST3R processes a dynamic video to produce a time-varying dynamic point cloud, along with per-frame camera poses and intrinsics, in a predominantly feed-forward manner. This representation then enables the efficient computation of downstream tasks, such as video depth estimation and dynamic/static scene segmentation. 27
2.17 VGGT is a large feed-forward transformer with minimal 3D-inductive biases trained on a trove of 3D-annotated data. It accepts up to hundreds of images and predicts cameras, point maps, depth maps, and point tracks for all images at once in less than a second, which often outperforms optimization-based alternatives without further processing. 28

3.1 Illustration of representative human-robot collaboration scenarios highlighting the robot-centric perception problem, which aims to estimate the relative poses between multiple cameras and the robot reference frames denoted with colored arrows. 29
3.2 The two single-camera configurations solved by the unified calibration framework. On the left, the *eye-in-hand* setup, where the camera moves with the end-effector (E_i) and observes a static board B. The unknowns are the end-effector-to-camera pose ($X = T_E^C$) and the board-to-base pose ($Z = T_B^W$). On the right, the *eye-on-base* setup, where a fixed camera C observes a board B moving with the end-effector. The unknowns are the base-to-camera pose ($X = T_W^C$) and the board-to-end-effector pose ($Z = T_B^E$). 32
3.3 Robustness evaluation of all calibration methods under increasing visual detection noise (first column) and initial guess perturbations in translation and rotation (second and third columns). The proposed method demonstrates consistent accuracy across all noise levels, while classical hand-eye calibration methods (Tsai, Daniilidis) exhibit order-of-magnitude degradation in both translation and rotation estimation. 34
3.4 Focus on the result of the evaluation of the proposed hand-eye calibration method vs. Koide's method only. 35
3.5 The two real-world experimental setups used for validation. On the left, the *eye-in-hand* configuration with a ZED camera on the end-effector. On the right, the *eye-on-base* configuration with a fixed Microsoft Kinect V2 observing the robot. 35
3.6 A large industrial robotic workcell equipped with a camera network around an ABB robot arm, enabling human-robot collaboration task in a carbon fiber draping process as foreseen in the DrapeBot European Project. 36

3.7 Multi-camera hand-eye setup, illustrating the geometric transformations optimized through the proposed multi-camera hand-eye calibration method. They include the single board-to-end-effector transformation common to all cameras, the spatial constraints among the cameras, and all the hand-eye transformations. 38
3.8 Graph representation for the multi-camera eye-on-base calibration. **Vertices** (circles) represent the unknown poses to be estimated: the world-to-camera poses $\{T_W^{C_k}\}$, the shared board-to-end-effector pose T_B^E, relative camera poses $\{T_{C_i}^{C_k}\}$ and the board-to-camera poses $T_{B_i}^{C_k}$. **Edges** (rectangles) represent error terms. Fixed parameters (3D board points $\mathbf{P}^B$ and intrinsic parameters K_L) are shown as **hexagons**. 39
3.9 Simulated METRIC robotic workcell from a side view on the left; robotic workcell from a top view on the right, showing the different crucial dimensions. 42
3.10 Large-scale real workcell equipped with a Panda robot arm with a checkerboard attached on its end-effector, surrounded by four Intel RealSense Depth D455 cameras. 43
3.11 Analysis of translation error e_t^{GT} and rotation error e_θ^{GT} for the METRIC simulated workcells as the number of images per camera used in the calibration process varies. Errors are expressed in a logarithmic scale. 45
3.12 On the left, the ABB industrial workcell equipped with four Intel RealSense Depth D455 cameras positioned around the manipulator at an average distance of 5 m. On the right, the KUKA industrial workcell equipped with three Intel RealSense Depth D455 cameras surrounding the manipulator at an average distance of 6 m. 46
3.13 MEMROC Overview. The system is structured into three primary components: the Ground Plane Detection module, responsible for identifying the ground plane; the Ground Plane Validation module, which ensures the ground plane's alignment with the mobile robot; and the Motion-Based Calibration Process, the concluding stage that calibrates the system. 48
3.14 Formulation of the motion-based calibration method. A mobile robot R equipped with a camera C moves from pose i to $i+1$ towards a calibration pattern P in the scene. A_i represents the odometry of the mobile robot with respect to the fixed frame W (i.e., the robot starting position), while B_i represents the camera pose with respect to the calibration pattern. X denotes the camera to robot transformation to be estimated. 49
3.15 Set of 4 images representing 4 different scenarios where the dataset was collected. The first two on the left show two different real-world scenarios, the other two simulated scenarios. 52
3.16 Impact of image quantity on calibration methods, where the lines represent the average errors and the shaded areas indicate the standard deviation, reflecting the variability and consistency of each method. 54
3.17 Evaluation of motion-based calibration method with incremental noise in measurements provided by odometry, with $\lambda \in [0, 10]$. The lines represent the average errors and the shaded areas indicate the standard deviation. 55
3.18 Calibration parameters with average and standard deviation of the Intel RealSense LiDAR L515 sensor estimated by different calibration methods. The expected values were measured by using an AprilTag fixed at a precise distance from the mobile robot in front of the sensors. 56

3.19 Overview of the Calib3R method. A set of RGB images and their corresponding robot poses are used as input. The images are processed by the 3D foundation model MASt3R to generate dense local pointmaps. These pointmaps, together with the robot poses, are then used in a joint optimization that estimates the global scene geometry, performs camera-to-robot calibration, and refines cross-camera alignment. The result is a metric-scaled 3D reconstruction aligned with the robot reference frame. 58
3.20 Formulation of camera-to-robot calibration. Let $\mathcal{R}_i$ denote the robot's reference frame at time step i, either the robot base for a mobile platform (left) or the end-effector for a manipulator arm (right). The robot is equipped with one or more cameras $\mathcal{C}_j$ and moves from pose i to $i+1$ within the environment, capturing images of a static 3D scene during its motion. A_i and $B_{j,i}$ represent the relative motions of the robot and the j^{th} camera, while X denotes the unknown rototranslation matrix describing the pose of the camera with respect to the robot reference frame. 61
3.21 Relation of incremental motion of multiple cameras onboard of the same robot formulated as an hand-eye calibration problem where the unknown matrix $X = T_{\mathcal{C}_m}^{\mathcal{C}_n}$ describes the relative pose among cameras $\mathcal{C}_n$ and $\mathcal{C}_m$. 63
3.22 Franka Pattern dataset: Experimental setup with a 7×10 checkerboard (3 cm squares) placed within the robot's workspace for calibration. 64
3.23 Franka Object dataset: Generic objects are placed in the robot's workspace to support calibration and scene reconstruction without relying on patterns. . 64
3.24 GraspNet-1Billion dataset: Robot setup with generic objects placed in its workspace [5]. 64
3.25 Real-world experimental setup of the MEMROC dataset [6]. 65
3.26 Experimental configuration of the synthetic MEMROC dataset [6]. 65
3.27 Robot from the OpenLORIS dataset [7]. 65
3.28 Experimental settings employed in the synthetic CSE Benchmark dataset [8]. 65
3.29 Camera-to-robot transformation accuracy as a function of the number of input images. Starting from 25 images, 2 random images were removed at each step down to a minimum of 5. 67
3.30 Metric-scaled 3D reconstruction of a scene from the GraspNet dataset [5], generated using only 15 RGB images. Both the estimated camera poses and the corresponding robot poses are expressed in the robot's base frame by applying the kinematics from the initial end-effector pose (at step 0). 68
3.31 Camera-to-robot transformation accuracy as a function of the number of input images. Starting from 25 images, 2 random images were removed at each step down to a minimum of 5. 70
3.32 Metric-scaled 3D reconstruction of the environment using 15 images per camera from a MEMROC dataset sequence [6]. The reconstruction includes the estimated camera poses and the corresponding 15 robot poses, all expressed with respect to the first robot pose reference frame. 71

4.1 Overview of the proposed framework for occlusion-aware 3D human pose estimation. The method first predicts a 3D pose from each single view, then fuses information from multiple views, and finally refines the 3D pose through reprojection error minimization. 77
4.2 Sample sequence from the Human3.6M-Occluded dataset, with occlusions on three of four views. 79
4.3 Sample images from the sequences acquired in real human-robot collaboration scenarios. a) Industrial workcell; b) Laboratory workcell. 80

4.4 Qualitative comparison between our approach and the state of the art on the industrial workcell scenario. 83
4.5 Qualitative results of our approach and state-of-the-art multi-view methods on the laboratory collaborative workcell. 83
4.6 Variation of relative and absolute MPJPE with increasing number of desynchronized cameras. Solid lines represent variation of the relative error, dashed lines represent variation of the absolute errors. 84
4.7 Relative and absolute MPJPE gradually removing the number of viewpoints, with occlusions. Solid lines represent variation of the relative error, dashed lines represent variation of the absolute errors. 84
4.8 Example of a collaborative draping task in an industrial setting. The human operator and robot jointly transport the material from a pick-up table to a mould, where it is manually draped. Human actions are continuously monitored throughout the process to trigger and coordinate robot operations. . . . 86
4.9 Framework overview. Normalized skeletons are stored in a fixed-size buffer of length W to capture current and recent movements. The buffer is processed by a set of N lightweight binary classifiers. Classifier predictions are validated using a per-classifier threshold T_i, selecting the final prediction based on the highest score exceeding a global threshold T_g. 88
4.10 Qualitative results for a lab-simulated collaborative draping task. Main steps of the process are shown at the top: (i) initial Next gesture to start the process; (ii) Pick and (iii) Drape of the ply; (iv) Next gesture to signal end of the process. Per-frame action predictions are reported for the proposed ensemble framework and baselines, compared with manually annotated ground-truth. . 93
4.11 Qualitative results for a real industrial collaborative draping task. Main steps of the process are shown at the top: (i) initial Next gesture to start the process; (ii) collaborative transport and (iii) manual draping of the ply; (iv) Next gesture to signal end of the process. Per-frame action predictions are reported for the proposed ensemble framework and baselines, compared with manually annotated ground-truth. 94

5.1 Overview of imagination with DREMA, which builds a compositional manipulation world model from environment images using object-centric Gaussian Splatting to generate novel demonstrations by transforming real ones. 99
5.2 Steps to create the compositional world model with DREMA: observation of the environment and scene decomposition, representation extraction and future predictions. 101
5.3 The effect of equivariant translation, equivariant rotation, and the object rotation transformations. Top row: start of demonstration. Bottom row: target of demonstration. 105
5.4 Keypoints distribution of original (blue dots) and generated data (red cross) for the *sort shape* task. Data is generated from one examples (left), five examples (middle) and twenty examples (right). 107
5.5 Imagined demonstrations keep improving imitation learning even with increasing number of original data. 108
5.6 Original (top) and imagined demonstration (bottom) after a 90° rotation transformation. 109
5.7 We introduce the novel task of active mapping of moving objects, requiring agents to plan observation trajectories while compensating for target motion. We also propose a method to solve this task and a benchmark for evaluation. 110

5.8 Paparazzo alternates between Object Tracking Mode and Object Mapping Mode based on the confidence of the EKF motion estimate. When the filter is uncertain, the agent prioritizes acquiring stabilizing observations; once confident, it predicts future object motion, generates and propagates candidate viewpoints, and selects the optimal one $(\mathbf{x}^*, i^*)$ that minimizes the final cost function. 112
5.9 The four target objects used in our experiments, featuring different shapes and colors. Each object is evaluated independently across all environments. . 119
5.10 **Generation of candidate viewpoints for Object Mapping Mode.** When the EKF becomes confident, Paparazzo switches from tracking to mapping and evaluates a set of candidate viewpoints $\mathcal{V}$ distributed around the object. The expected information gain (EIG) of each pose, computed using the FisherRF criterion, is visualized here with a color gradient: darker tones correspond to low informativeness, while brighter tones highlight more informative poses for reconstructing the object. 120
5.11 **EKF-based prediction of future object poses.** The EKF predicts the object pose, denoted in orange, over the next N_h steps. The figure shows three examples at steps k+10, k+20, and k+30. These predicted poses are then used to propagate all candidate viewpoints $\mathcal{V}$ into the future, producing a set of $|\mathcal{V}| \times N_h$ future-aligned viewpoints that are subsequently evaluated by the final cost function in Eq. (5.25). 121
5.12 **Visualization of the 3D reconstruction of Object 3 and Object 4 under Stop & Go motion.** We compare the RW, RIS, and TO baselines against our Paparazzo method. Paparazzo produces significantly more complete and geometrically consistent reconstructions. 124
5.13 **Qualitative 3D reconstructions of Object 1 and Object 2 under Stop & Go motion.** Paparazzo consistently produces reconstructions that are significantly more complete, coherent, and stable across both objects. In contrast, the baselines, especially on Object 2, fail to recover the entire frontal surface, leaving substantial portions missing. 125
5.14 **Examples of object trajectories**. The moving target is performing the Bouncing Ball motion in the Denmark, Greigsville, and Ribera scenes (left to right). The agent executes the Paparazzo framework while continuously adapting its motion to track and map the moving object. 126
5.15 **Illustration of Stop & Go motion.** Both panels show the same Stop & Go trajectory executed by the moving object. Across 300 steps, the object pauses once. (a) TO: the agent remains passive during the stop phase, losing valuable time and collecting no new viewpoints, which prevents further progress in the reconstruction. (b) Paparazzo: the agent continues to actively reposition and capture informative views even while the object is stationary, as visible from the additional exploratory camera frustums (in gray). This allows Paparazzo to maintain reconstruction progress during motion interruptions, unlike the TO baseline. 126

List of Tables

3.1 Comparison of hand-eye calibration methods in both configurations. 35
3.2 Main dimensions for the three workcells considered in the synthetic dataset: small, medium and large. 41
3.3 Dimensions for the two workcells considered in the real-world dataset: small and large. 42
3.4 Sensor specifications. 42
3.5 Average errors for all cameras achieved by hand-eye calibration techniques in the METRIC simulated workcells. The upper block reports single-camera methods, while the lower block lists multi-camera approaches. Bold and underlined values indicate the 1^{st} and 2^{nd} top-performing calibration methods for each metric, respectively. 44
3.6 Average errors achieved by hand-eye calibration techniques on the METRIC real workcells. The upper block reports results from single-camera methods, while the lower block corresponds to multi-camera approaches. Bold and underlined values indicate the 1^{st} and 2^{nd} top-performing calibration methods, respectively ("—" denotes non-convergence). 46
3.7 Average error on industrial workcells. 47
3.8 Average error in Camera-to-Robot transformation 53
3.9 Average error in Camera-to-Camera transformation 54
3.10 Average error in Camera-to-Camera transformation 57
3.11 Average error in the estimated camera-to-robot (hand-eye) transformation, computed using images acquired from 25 robot poses in the Franka Pattern dataset (Figure 3.22), Franka Object dataset (Figure 3.23) and GraspNet-1Billion (Figure 3.24). The Pattern column indicates whether the corresponding method requires a calibration pattern. Traditional method can not be used in Franka Object dataset and GraspNet-1Billion benchmark since it is not involved a calibration pattern. 66
3.12 Metric scale accuracy of 3D reconstruction on Franka Pattern dataset. The table reports the mean scale error, the standard deviation and the percentage scale error relative to the true checkerboard square size of 3 cm. Lower values indicate higher metric consistency. 68
3.13 Average error in the estimated camera-to-robot transformation across the real-world and synthetic MEMROC datasets, as well as the OpenLORIS and CSE benchmarks, computed over 25 robot poses. The Pattern column indicates whether the method requires a visible calibration pattern. 69

4.1 The results of evaluation on the Human3.6M dataset. The table presents the MPJPE relative to pelvis for state-of-the-art multi-view human pose estimation methods. 81
4.2 The results of evaluation on the Human3.6M dataset. The table presents the absolute MPJPE for state-of-the-art multi-view human pose estimation methods. 81

4.3 The results of evaluation on the proposed Human3.6M-Occluded benchmark. The table presents the MPJPE relative to pelvis for our method and state-of-the-art multi-view human pose estimation algorithms. 82
4.4 The results of evaluation on the proposed Human3.6M-Occluded benchmark. The table presents the absolute MPJPE for our method and state-of-the-art multi-view human pose estimation algorithms. 82
4.5 The results of evaluation on data sequences collected in real human-robot workcells. The table presents the MPJPE relative to pelvis for our method and state-of-the-art multi-view human pose estimation algorithms. 83
4.6 The table presents the absolute MPJPE on data collected in real human-robot workcells for our method and state-of-the-art multi-view human pose estimation algorithms. 84
4.7 Action recognition results on the Collaborative Draping HAR dataset. Best results are highlighted in **bold**, and second-best results are underlined. 91
4.8 Online action recognition results on continuous sequences from a lab-simulated collaborative draping task. 92
4.9 Online action recognition results on continuous sequences from a real industrial collaborative draping task. 94
4.10 Action recognition performance in terms of runtime and recognition delay on the industrial scenario. 95

5.1 Comparison of trained on original, augmented, and DREMA with imagined and all demonstrations, reporting mean ± std and max success over 5 runs on 50 test environments. 106
5.2 Comparison of PerAct [9] trained on original demonstrations to DREMA trained on only imagination demonstrations and the combination of both in the multi-task setting. The table reports the mean ± std and maximum success rate over 5 test runs. 106
5.3 The mean ± std and maximum success rate of DREMA trained on single-task demonstrations from different types of transformations over 5 test runs on 50 examples. 108
5.4 Localization errors. 109
5.5 In- and out-of-distribution evaluation with real robots. 109
5.6 Results of the *pick block* task with original and imagination demonstrations. . 109
5.7 EKF parameters used in all experiments. 115
5.8 Quantitative results across 6 scenes for our three dynamic motion types (Bouncing Ball (BB), Forward & Backward (FB) and Stop & Go (SG)). Reported values are averaged over all test objects and 5 runs. Each entry shows Coverage (%), Completeness (cm), and Accuracy (cm). 122
5.9 Quantitative comparison of different motion types averaged across all scenes. We evaluate three dynamic behaviors (Bouncing Ball (BB), Forward&Backward (FB), and Stop&Go (SG)) and report results for four test objects. Each cell shows Coverage (%), Completeness (cm), and Accuracy (cm). 124
5.10 Reconstruction coverage (%) for Object 1 and Object 2 under the *Stop & Go* motion pattern. Paparazzo significantly outperforms all baselines. 125

Chapter 1

Introduction

Industrial robots were originally conceived within a paradigm of segregation, reflecting an era in which safety and efficiency were achieved by strict physical separation from human workers. Their design philosophy was therefore centered on executing precise and repeatable operations with superhuman endurance, confined within the rigid, predictable structure of environments, such as manufacturing and automotive assembly, where tasks could be predefined and repeated with minimal variability [10, 11]. For decades, these robots have been the cornerstone of modern production, excelling in speed, accuracy, and repeatability, features that made them indispensable for large-scale tasks such as welding, assembly, and inspection [12, 13, 14]. However, this reliability came at the cost of adaptability: these robots could operate effectively only under fixed, predefined settings and struggled to respond to changes in their surroundings. For example, the presence of an unexpected object on a conveyor belt could bring the entire production line to a halt, as traditional robots lacked the perception and flexibility to handle unforeseen situations.

In more recent years, manufacturing has shifted toward smaller production batches, frequent reconfiguration, and greater product customization [15]. This transformation exposed the limits of rigid, fully automated systems and created a demand for flexibility and adaptability. As factories evolve into dynamic, human-centered workspaces, collaboration between humans and robots emerges as a natural progression, combining human creativity and decision-making with robotic precision and endurance [16]. Rather than replacing human labor, modern robots are designed to augment it, taking on physically demanding, repetitive, or high-risk tasks while humans focus on reasoning, creativity, and fine-motor control.

The demand for physical collaboration led to the emergence of Collaborative Robots, or cobots [17], specifically designed with inherent safety mechanisms, including lightweight structures, rounded edges, and advanced sensing systems that limit forces and immediately halt motion upon unexpected contacts. Their combination of safety, adaptability, and affordability has made cobots ideal to be integrated into human-centered workplaces [18, 19, 20]. Yet, despite their progress, current cobots primarily ensure safe coexistence rather than effective collaboration based on mutual understanding and coordinated interaction. They can share space with humans, but they still lack the ability to understand them or perceive their surroundings. To move beyond coexistence, robots must be able to interpret and adapt to the shared world—marking a fundamental evolution from automation, where robots execute predefined tasks, to collaboration, where they act as intelligent partners that reason and interact safely within shared environments. The research presented in this thesis aims to contribute to this evolution, proposing novel methods to enable robots to collaborate effectively with humans through intelligent interaction and enhanced environmental awareness.

1.1 Toward Intelligent Robotic Perception and Action

Effective Human-Robot Collaboration (HRC) depends on the robot's ability to perceive and make sense of its environment through diverse sensory channels—from vision to tactile and

force sensing, auditory cues, and proximity feedback [21, 22]. Among these perceptual systems, vision stands out as the richest and most versatile source of information, it enables the interpretation of the semantic and geometric structure of its environment, reconstruct a 3D representation of the space, and reason about spatial relationships. However, perception becomes truly meaningful only when it is expressed in the robot's own reference frame, linking what is seen to where and how the robot can physically act.

This principle of robot-centric perception forms a central theme of this thesis and is explored through two complementary scenarios, illustrated in Figure 1.1. The first involves fixed workspaces, where external cameras monitor the shared workspace and human activity, while onboard cameras on the robot arm provide close-up observations of objects from multiple perspectives. The second focuses on mobile platforms equipped with onboard sensors that actively explore their surroundings from varying viewpoints.

In both cases, the goal is to endow the robot with an egocentric understanding of its environment, enabling it to reconstruct a coherent 3D representation of the scene, localize humans, interpret their actions, interact effectively with nearby objects, and plan behaviors that are safe, adaptive, and human-aware.



FIGURE 1.1: Illustration of the two main human-robot collaboration settings considered in this thesis. Robots must perceive and act in human-shared environments, both in fixed workcells (e.g., manipulators with external or onboard cameras) and in mobile scenarios (robots equipped with onboard cameras).

Achieving such a level of intelligence requires a seamless integration of perception, reasoning, and action. This thesis builds upon three interdependent research pillars that collectively advance the development of adaptive and human-aware robotic intelligence:

1. **Robot-centric perception**, establishing the geometric relationship between sensor observations and the robot's reference frame, enabling perception from the robot's viewpoint.

2. **Human understanding**, allowing the robot to interpret and anticipate human intentions, fostering natural and intuitive cooperation between humans and robots.

3. **Robot-object interaction**, endowing the robot with the ability to manipulate and explore objects, both static and dynamic, while acting effectively even in uncertain environments.

The synergy among these three research directions provides the foundation to develop robots that not only coexist safely with humans but also become collaborative partners, capable of understanding, adapting, and acting intelligently in complex real-world contexts. Building on these foundations, this thesis introduces a series of methodologies that enable

robots to perceive the world from their own viewpoint, understand human partners, and interact adaptively with dynamic environments.

While this thesis addresses multiple aspects of Human–Robot Collaboration, the core methodological contributions are primarily centered on robot-centric perception, which constitute the fundamental basis upon which the components of human understanding and robot–object interaction are built.

1.1.1 Robot-Centric Perception

A fundamental prerequisite for any perception-based robotic system is the ability to relate visual information to the robot's own reference frame. This problem is commonly known as hand-eye calibration for manipulator arms [23] and as motion-based calibration for mobile robots [24]. Despite their different contexts, they share a common goal: estimating the pose of each camera with respect to a robot reference frame, such as the end-effector frame for manipulators or the base frame for mobile robots. To encompass both configurations, this thesis adopts the general term *camera-to-robot calibration*, emphasizing its broad applicability across robotic platforms.

Existing camera-to-robot calibration methods often suffer from fragmentation: they are typically tailored for either external cameras observing a manipulator or cameras mounted on its end-effector, rarely supporting both within a unified framework. In multi-camera systems, calibration is typically performed independently for each camera [25], deriving relative positions between cameras through chaining transformations or stereo calibration [26], which accumulates error and leads to inconsistent and imprecise results. For mobile robots, planar motion introduces an additional challenge: restricted planar movement often provides insufficient geometric diversity to recover full six degrees of freedom (6-DoF) for the camera [27, 28].

To address these limitations, this thesis introduces three complementary approaches: (i) a unified optimization framework for calibrating single onboard or external cameras, as well as external multi-camera systems surrounding a robot arm; (ii) a motion-based calibration method for multi-camera setups mounted on mobile robots, leveraging ground-plane constraints to recover full 6-DoF parameters; and (iii) a patternless joint calibration and 3D reconstruction framework for multi-camera configurations onboard either a robot arm or a mobile platform, aligning RGB-based reconstructions with the robot frame without using calibration targets.

1.1.2 Human Understanding

Once the robot's perception is spatially aligned within its own reference frame, the next essential step toward effective collaboration is understanding human motion and intention, a prerequisite for safe, intuitive, and proactive cooperation. This thesis addresses this challenge through two fundamental components of human understanding: *3D Human Pose Estimation (HPE)* and *Human Action Recognition (HAR)*.

In 3D HPE, state-of-the-art multi-view approaches typically triangulate 2D keypoints representing the human skeleton [29, 30, 31]. While such methods are effective in controlled laboratory settings, they often fail in real-world HRC scenarios, where occlusions, clutter, and partial visibility lead to incomplete or unreliable 3D pose estimation. To overcome such challenges, this thesis introduces a robust multi-view framework that fuses independently predicted 3D skeletons and refines them through a reprojection-based optimization, enhancing geometric consistency and robustness even under occlusion or limited visibility.

Building on the resulting pose estimations, this work further advances *Human Action Recognition* toward continuous, real-world HRC scenarios. Unlike traditional HAR models trained on short, segmented clips, representing isolated actions [32, 33, 34], real HRC involves ongoing interactions where actions may overlap or transition seamlessly.

To address this, the thesis introduces an online HAR framework that operates directly on the estimated 3D skeletons. The system combines an ensemble of lightweight binary classifiers, each specialized in detecting a specific action, with a causal temporal buffer that accumulates recent motion data to support real-time, frame-by-frame inference. Through this design, the robot can continuously recognize and respond to human actions in dynamic, unstructured collaborative environments, enabling more natural, context-aware interaction.

1.1.3 Robot-Object Interaction

A crucial step toward robotic autonomy lies in the robot's ability to interact effectively with its environment, adapting to new objects, configurations, and dynamics. This thesis investigates robot-object interaction in two complementary contexts: manipulation with robot arms and exploration with mobile robots.

For manipulators, a framework is proposed that enables few-shot learning, allowing the system to generalize from a limited number of demonstrations to unseen object configurations. The approach integrates photorealistic 3D reconstruction via Gaussian Splatting [35], zero-shot visual generalization through foundation models [36], and physics-based validation in simulation [37]. By combining these elements, the system adapts demonstrated trajectories through equivariant transformations between scene geometry and motion, enabling efficient and flexible skill generalization without retraining when encountering new objects or environments.

For mobile robots, this principle of adaptive interaction extends to dynamic exploration, giving the robot the ability to navigate and reconstruct dynamic objects as efficiently as possible. While autonomous exploration in static environments is well studied [38, 39, 40], actively reconstructing moving objects remains a significant open challenge. This requires the robot not only to identify informative viewpoints but also to anticipate the target's motion. To address this, the thesis introduces an active mapping framework that enables simultaneous exploration of unknown environments and reconstruction of unseen dynamic objects. The system selects optimal viewpoints by quantifying the Expected Information Gain (EIG) for the object's Gaussian Splatting model, leveraging Fisher Information, derived from the Hessian of the rendering loss, as a measure of model uncertainty [41]. Crucially, this information-driven selection is integrated with motion prediction: the framework anticipates the object's future trajectory, enabling efficient planning that proactively intercepts the moving target at informative viewpoints for robust reconstruction.

1.2 Thesis Contributions and Goals

Building on these foundations, this thesis aims to advance Human-Robot Collaboration by developing robotic agents capable of integrating multi-sensor observations into a coherent and consistent understanding of the environment, interpreting human behavior, and interacting effectively in unstructured, dynamic settings.

The main contributions are summarized as follows:

Robot-centric perception

- **Unified camera-to-robot calibration:** A novel calibration framework for single- and multi-camera systems, applicable to cameras both surrounding a robot arm and mounted

on its end-effector. The method jointly optimizes all camera poses with respect to the robot base and to each other, achieving globally consistent calibration.

- **Motion-based multi-camera calibration for mobile robots:** A robust approach to estimate full 6-DoF camera extrinsics relative to a mobile robot reference frame, leveraging ground-plane constraints to overcome the degeneracy of planar motion.
- **Patternless joint calibration and metric 3D reconstruction:** A unified optimization framework performing simultaneous camera-to-robot calibration and dense 3D reconstruction using only RGB images, without requiring calibration targets.

Human understanding

- **Multi-view 3D human pose estimation:** A fusion-based approach that aggregates 3D keypoints predicted from multiple views and refines them through reprojection-based optimization, achieving accurate and robust pose estimation even under occlusions and limited visibility.
- **Online human action recognition:** An efficient framework combining lightweight binary classifiers with a causal temporal buffer, enabling real-time recognition of continuous human activities in dynamic environments.

Robot-Object interaction

- **Few-shot imitation learning for adaptive manipulation:** A novel learning paradigm that integrates 3D reconstruction, foundation models, and physics-based simulation, allowing robots to generalize demonstrated skills to unseen objects and configurations.
- **Dynamic exploration and object reconstruction:** A trajectory-planning framework that enables mobile robots to explore and reconstruct dynamic objects using Guassian Splatting and motion prediction for adaptive active mapping.

1.3 Thesis Structure

This section outlines the organization of the thesis.

Chapter 2 provides an overview of the theoretical background relevant to this research. It reviews existing approaches to camera-to-robot calibration, traces the evolution of 3D reconstruction techniques, and summarizes the most important computer vision architectures that underpin the proposed methods.

Chapter 3 introduces the concept of robot-centric perception, addressing the problem of camera-to-robot calibration across different setups: single onboard or external cameras, external multi-camera systems surrounding a robot arm, and multi-camera configurations onboard mobile robots. The chapter culminates with the presentation of a novel joint calibration and reconstruction framework.

Chapter 4 focuses on the human understanding pillar, presenting the proposed method for 3D human pose estimation in human-robot collaboration (HRC) scenarios, designed to overcome occlusions, as well as a human action recognition (HAR) approach that advances interaction in industrial environments.

Chapter 5 explores the robot-object interaction paradigm through two representative setups: object manipulation using a robot arm and object exploration using a mobile robot.

Finally, Chapter 6 concludes the thesis by summarizing the main findings and outlining the key scientific and practical contributions.

1.4 List of Works

Parts of the content presented in this thesis have been previously published or submitted to international conferences and journals. For works where the candidate is not the first author, the specific contribution of the candidate is explicitly indicated.

- [42] **Davide Allegro**, Matteo Terreran, and Stefano Ghidoni, "Calib3R: A 3D Foundation Model for Multi-Camera to Robot Calibration and 3D Metric-Scaled Scene Reconstruction." submitted to IEEE Transactions on Robotics T-RO 2025.

- [43] Niccolò Turcato, Giulio Giacomuzzo, Matteo Terreran, **Davide Allegro**, Ruggero Carli and Alberto Dalla Libera, "Data efficient Robotic Object Throwing with Model-Based Reinforcement Learning" submitted to Transactions on Automation Science and Engineering (TASE) 2025. *The candidate contributed to the design of the multi-camera setup and to its calibration with respect to the robot, which was essential for the experimental validation.*

- [44] **Davide Allegro**, Matteo Terreran, and Stefano Ghidoni, "Multi-Camera Hand-Eye Calibration for Human-Robot Collaboration in Industrial Robotic Workcells." IEEE Robotics and Automation Letters R-AL 2024.

- [45] **Davide Allegro**, Matteo Terreran, and Stefano Ghidoni, "METRIC—Multi-Eye to Robot Indoor Calibration Dataset." Information, MDPI, 2023.

- [46] Leonardo Barcellona, Andrii Zadaianchuk, **Davide Allegro**, Samuele Papa, Stefano Ghidoni, Efstratios Gavves, "Dream to Manipulate: Compositional World Models Empowering Robot Imitation Learning with Imagination", In: The Thirteenth International Conference on Learning Representations ICLR 2025, Singapore. *The candidate conducted the full set of real-world experiments to assess the applicability of the proposed system on real robotic platforms.*

- [47] Matteo Terreran, Laura Bragagnolo, **Davide Allegro** and Stefano Ghidoni "A Lightweight Ensemble Framework for Online Skeleton-Based Human Action Recognition in Industrial Environments", In European Conference on Mobile Robots ECMR 2025, Padua, Italy. *The candidate contributed to the development of the method and of the ensemble architecture, as well as to the real-world experimental evaluation.*

- [6] **Davide Allegro**, Matteo Terreran, and Stefano Ghidoni, "MEMROC: Multi-Eye to Mobile RObot Calibration" Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems IROS 2024, Abu Dhabi.

- [48] Laura Bragagnolo, Matteo Terreran, **Davide Allegro** and Stefano Ghidoni, "Multi-view Pose Fusion for Occlusion-Aware 3D Human Pose Estimation", European Conference on Computer Vision ECCV 2024, Milan, Italy. *The candidate contributed to the implementation of the optimization process based on reprojection error minimization for 3D human pose estimation.*

- [49] Alberto Bacchin, **Davide Allegro**, Stefano Ghidoni and Emanuele Menegatti, "SOOD-ImageNet: a Large-Scale Dataset for Semantic Out-Of-Distribution Image Classification and Semantic Segmentation", European Conference on Computer Vision ECCV 2024, Milan, Italy. *The candidate contributed to the creation and annotation of the dataset, and to the evaluation of Visual Language Models on the proposed benchmark.*

- [50] Daniele Evangelista, Emilio Olivastri, **Davide Allegro**, Emanuele Menegatti and Alberto Pretto, "A Graph-Based Optimization Framework for Hand-Eye Calibration for Multi-Camera Setups." In: IEEE International Conference on Robotics and Automation ICRA 2023, London, UK. *The candidate contributed to the creation of the dataset used to evaluate the method in both real and synthetic scenarios.*

- [51] Daniele Evangelista, **Davide Allegro**, Matteo Terreran, Alberto Pretto and Stefano Ghidoni, "An unified iterative hand-eye calibration method for eye-on-base and eye-in-hand setups." In IEEE 27th International Conference on Emerging Technologies and Factory Automation ETFA 2022, Stuttgart, Germany. *The candidate contributed to the implementation of the proposed method, to the creation of the test dataset, and to part of the experimental evaluation.*

Chapter 2

Fundamentals

This chapter lays out the theoretical and methodological foundations of the thesis, structured around three main themes. It begins by introducing the geometric principles of robotic vision, including sensing modalities, camera modeling, and the hand-eye calibration problem, an essential step for grounding visual perception within the robot's action space. The discussion then moves to the evolution of 3D reconstruction techniques, which enable robots to interpret and interact with their environments. Finally, it explores the emergence of deep learning-based vision models that bring semantic understanding to scene interpretation, tracing their development from Convolutional Neural Networks (CNNs) to Vision Transformers (ViTs) and emphasizing how these architectures have become central to modern, learning-driven approaches to 3D reconstruction.

2.1 2D and 3D Vision Sensors

Visual perception is a fundamental component of autonomous robotic systems, providing the information necessary to interpret and interact with complex environments. Vision sensors can be broadly categorized according to whether they capture 2D appearance and those that measure 3D structure.

2D sensors, such as RGB cameras, provide rich textural and semantic information, while 3D sensors, including RGB-D cameras and LiDAR, deliver accurate metric and spatial data. In practice, robotic perception benefits from integrating both modalities. The fusion of complementary 2D and 3D information enables a more robust and comprehensive understanding of the environment, which is essential for tasks such as navigation, manipulation, and human-robot interaction.

2.1.1 RGB Cameras

RGB cameras capture two-dimensional images that represent the projection of the 3D world onto an image plane. They are typically lightweight, affordable, and capable of acquiring high-resolution imagery rich in color and texture.

This visual data forms the foundation of numerous applications in mobile and aerial robotics, including visual servoing [52], SLAM (Simultaneous Localization and Mapping) [53] and object detection [54]. In these applications, RGB images provide dense visual cues that allow for motion estimation and semantic understanding, even in the absence of explicit depth information. Figure 2.1 illustrates two examples of high-performance RGB sensors used in robotics and aerial applications.



FIGURE 2.1: Examples of RGB cameras: (left) the Sentera 65R ultra-high-resolution sensor, and (right) the Zenmuse P1 full-frame camera with stabilized gimbal.

2.1.2 RGB-D Cameras

RGB-D sensors extend standard RGB cameras by providing a per-pixel depth map aligned with the color image. This alignment enables the acquisition of dense, metric 3D representations of the scene in real-time.

Depth estimation is achieved through several primary technologies:

- **Time-of-Flight (ToF):** Emits modulated infrared light and measures the phase or time delay of the reflected signal to compute depth. ToF sensors deliver dense depth maps in real time but may suffer from multi-path interference.
- **Passive Stereo:** Uses two or more synchronized cameras to infer depth through triangulation of corresponding pixels. Accuracy depends on texture and illumination conditions.
- **Active Stereo:** Combines stereo cameras with an infrared pattern projector, improving correspondence matching in low-texture regions.
- **Structured Light (SL):** Projects a known pattern onto the scene and infers depth from its deformation, achieving high accuracy in controlled environments.

By registering the RGB and depth data in a common reference frame, these sensors generate colored point clouds that encode both appearance (color) and geometry (3D coordinates). This 3D data is essential for tasks like 3D reconstruction [55], object pose estimation [56], and human tracking [57].



FIGURE 2.2: Examples of RGB-D cameras: (left) the ZED 2i passive stereo camera, (center) the Intel RealSense D455 active stereo sensor, and (right) the Microsoft Azure Kinect ToF-based sensor.

2.2 Camera Modeling and Calibration

An accurate camera model is essential for any vision-based robotic system, as it defines the mathematical relationship between 3D world points and their 2D projections in the image. Understanding and estimating this model is a prerequisite for nearly all geometric reasoning tasks, including 3D reconstruction, object and human localization.

2.2.1 The Pinhole Camera Model

The pinhole model is the most common idealization of a camera, representing it as a single viewpoint (the pinhole) through which light rays from the scene pass before intersecting the image plane. This projection maps a 3D point $M = [X, Y, Z, 1]^\top$ in world coordinates to a 2D image point $m = [u, v, 1]^\top$ (expressed in homogeneous coordinates) according to the equation:

$$s\, m = K[R \mid t]\, M \tag{2.1}$$

where s is a non-zero scale factor. This projection is governed by two distinct sets of parameters: the *extrinsic parameters* $[R|t]$ define the pose of the world W, its orientation and position with respect to the camera coordinate frame C. They consist of a rotation matrix $R \in SO(3)$ and a translation vector $t \in \mathbb{R}^3$, forming the rigid-body transformation $T_W^C \in SE(3)$ that maps points from the world frame W to the camera's coordinate frame C (as shown in Figure 2.3). The *intrinsic parameters* $K \in \mathbb{R}^{3\times3}$ model the internal geometry and optical characteristics of the camera:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.2}$$

The pinhole model is an idealization of camera models. The use of real lenses introduce lens distortions (e.g., radial and tangential) that cause the image points to deviate from their ideal projections. These effects are modeled by a set of distortion coefficients, such as the most common $d = [k_1, k_2, p_1, p_2, k_3]$, where k_i represent radial terms and p_i tangential ones. The complete projection is thus a nonlinear mapping $\hat{f}(K, d, R, t, M)$.

2.2.2 Camera Calibration

Camera calibration is the process of estimating this complete set of parameters $\{K, d, [R, t]\}$. This is typically formulated as an optimization problem that seeks to minimize the reprojection error, defined as the squared distance between the observed image points m_i and their projections from the 3D world points M_i according to the estimated model:

$$\varepsilon = \sum_i \left\| m_i - \hat{f}(K, d, R, t, M_i) \right\|_2. \tag{2.3}$$

This optimization typically uses multiple views of a known calibration pattern (e.g., a checkerboard). Classical and widely used methods include Zhang's planar calibration [58] and the refinement techniques from Bouguet [59]. This process yields two crucial results: (1) the camera's static intrinsic parameters (K and d) and (2) the view-dependent extrinsic parameters ($[R, t]$) for each view, representing the pose of the camera relative to the calibration pattern (W).

In this manuscript, intrinsic parameters are assumed to be known through a one-time offline calibration. The subsequent focus shifts to estimating extrinsic relationships between the camera and other coordinate systems, such as the robot's base or end-effector.

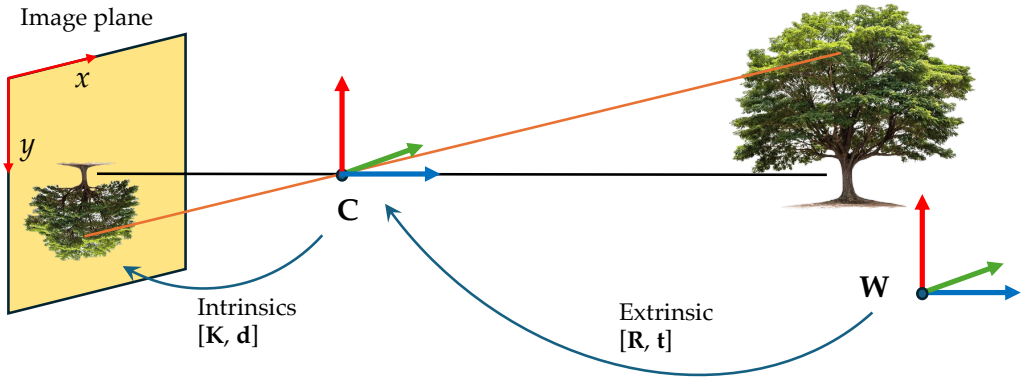


FIGURE 2.3: Illustration of a camera projection model. Extrinsic parameters transform the 3D point from the world reference frame W to the camera coordinate system C. Intrinsic parameters maps the 3D point from the camera reference frame to the image plane.

2.3 Hand-Eye Calibration

In modern robotic systems, visual perception plays a fundamental role in enabling efficiency and autonomy. However, a camera, even when intrinsically calibrated, provides measurements expressed only in its own coordinate system, denoted as C. While a standard extrinsic calibration can relate C to an arbitrary world frame, this frame is typically disconnected from the robot's operational space.

The robot interacts with the environment through its kinematic chain, defined in the base coordinate frame W, which controls the motion of an end-effector (or "hand") E. Consequently, unless the spatial transformation between the camera frame and one of these robot frames is known, the visual information cannot be interpreted in geometrically meaningful terms for control and interaction.

The hand-eye calibration addresses this issue by establishing the spatial bridge between perception and action. The goal is to determine a single, static, rigid-body transformation $X \in SE(3)$ that rigidly links the camera coordinate frame C to a relevant robot reference frame, either the end-effector E or the base W. Using the notation $T_B^A \in SE(3)$ to denote the pose of frame B with respect to frame A, the task consists in finding either $X = T_E^C$ or $X = T_W^C$, depending on the physical configuration. The two primary settings, illustrated in Figure 2.4 are:

- **Eye-in-Hand:** The camera is rigidly mounted on the robot's end-effector E. This provides an active and flexible viewpoint, ideal for close-up inspection and agile manipulation. In this setup, the problem is to find $X = T_E^C$, the static transform from the end-effector to the camera.

- **Eye-on-Base:** The camera is fixed in the environment, often such that it is rigid with the robot's base W, and observes both the workspace and the robot. This configuration offers a global and stable view, preferred for monitoring tasks or supervising human-robot collaboration. Here, the problem is to find $X = T_W^C$, the static transform from the robot's base to the camera.



FIGURE 2.4: Illustration of the two main configurations: on the left, *eye-in-hand* (camera on the end-effector). On the right, *eye-on-base* (fixed camera).

In both configurations, the unknown transformation X is estimated by observing a calibration target (or "board") B from multiple viewpoints. The procedure involves moving the robot to a series of distinct poses. For each pose i, a pair of transformations is acquired: the robot's kinematic chain provides the pose of its end-effector with respect to the base, $T^W_{E_i}$, while the camera, observing the calibration pattern, yields the pose of the pattern relative to itself, $T^C_{B_i}$. The latter is typically obtained through a Perspective-n-Point (PnP) algorithm [60], relying on the camera's known intrinsic parameters K. By collecting such paired measurements across multiple robot configurations, a system of equations can be formulated whose structure depends on the chosen setup.

2.3.1 Mathematical Formulations

The two configurations—eye-in-hand and eye-on-base—lead to distinct mathematical formulations of the hand-eye calibration problem. Both formulations arise from constructing kinematic loops, where two different transformation chains between a pair of coordinate frames are equated.

Formulation 1 — Relative Motion (AX=XB) In the eye-in-hand configuration, the calibration process typically consists of observing a static calibration pattern B from multiple robot poses. Considering two consecutive poses, i and $i+1$, as illustrated in Figure 2.5, the relative motion of the end-effector $T^{E_i}_{E_{i+1}}$ can be expressed in two equivalent ways, corresponding to the *robot path* and the *camera path*.

On the robot side, the relative motion between two end-effector poses is obtained directly from the robot's kinematic chain. The encoders provide the absolute poses of the end-effector with respect to the robot base W, denoted $T^W_{E_i}$ and $T^W_{E_{i+1}}$. Their composition yields:

$$B_i = T^{E_i}_{E_{i+1}} = (T^W_{E_i})^{-1} \cdot T^W_{E_{i+1}}. \tag{2.4}$$

On the camera side, the same motion can be expressed through the camera's changing viewpoint:

$$T^{E_i}_{E_{i+1}} = T^E_C \cdot T^{C_i}_{C_{i+1}} \cdot T^C_E. \tag{2.5}$$

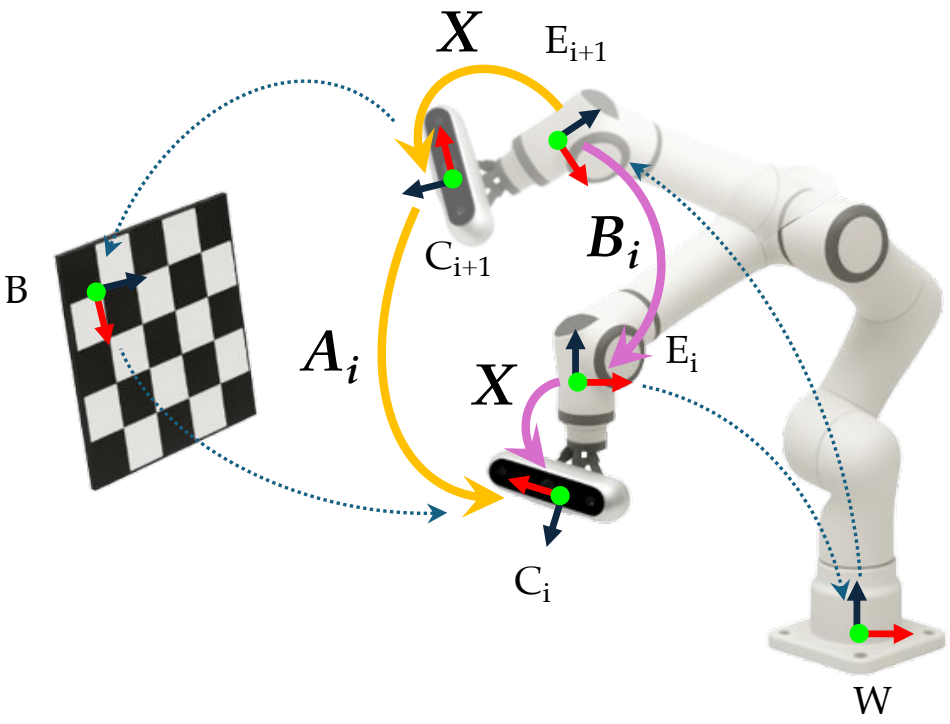


FIGURE 2.5: Kinematic loop for hand-eye calibration in eye-in-hand configuration, illustrating the equation $A_iX = XB_i$. The transformation A_i represents the relative motion of the camera ($T^{C_i}_{C_{i+1}}$), while B_i is the relative motion of the end-effector ($T^{E_i}_{E_{i+1}}$) .

Here, the term $T^{C_i}_{C_{i+1}}$ represents camera's relative motion, referred to as A_i, which can be computed by observing the fixed calibration pattern from the two viewpoints: $A_i = T^{C_i}_{C_{i+1}} = T^{C_i}_B \cdot (T^{C_{i+1}}_B)^{-1}$. The transformations T^C_E and T^E_C correspond to the unknown static transform X and its inverse X^{-1}, respectively.

By equating (2.4) and (2.5), we obtain: $B_i = X^{-1}A_iX$, which can be rearranged into the canonical homogeneous matrix equation:

$$A_iX = XB_i. \tag{2.6}$$

Given a set of N motion pairs $\{(A_i, B_i)\}_{i=1}^N$, the objective of hand-eye calibration is to find the single rigid-body transformation $X \in SE(3)$ that best satisfies this system [61].

Formulation 2 — Absolute Pose (AX=ZB) This second formulation corresponds to the eye-on-base configuration and is generally more versatile, as it allows solving for two unknown transformations simultaneously [62]. Here, the camera is fixed with respect to the robot's world frame W, and a calibration pattern B is rigidly attached to the end-effector E (see Figure 2.6). For any given robot pose i, a kinematic loop can be constructed by expressing the transformation from the robot world frame W to the calibration pattern frame B_i along two alternative paths: On the robot path, the transformation proceeds from the reference frame W to the end-effector E and then to the pattern B, obtaining:

$$T^B_W = T^B_E T^{E_i}_W. \tag{2.7}$$

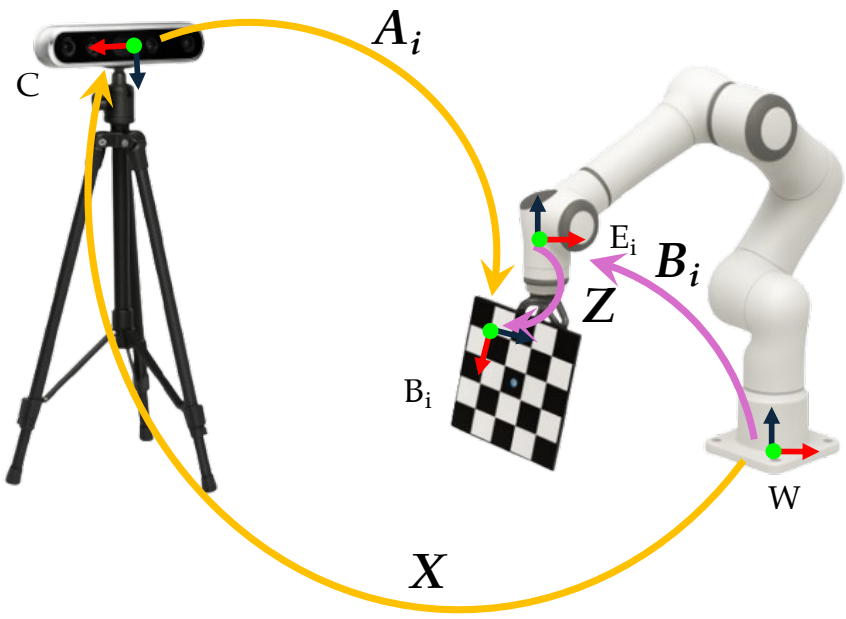


FIGURE 2.6: Kinematic diagram for the eye-on-base hand-eye calibration setup, which solves the equation $A_iX = ZB_i$. This formulation finds two static unknowns: the base-to-camera pose $X = T_W^C$ and the end-effector-to-board pose $Z = T_E^B$.

Conversely, on the camera path, the same transformation can be expressed through the fixed camera:

$$T_W^{B_i} = T_C^{B_i} T_W^C. \tag{2.8}$$

Equating the (2.7) and (2.8) yields the fundamental relationship:

$$T_C^{B_i} \cdot T_W^C = T_E^B \cdot T_W^{E_i}. \tag{2.9}$$

In this equation, the unknown static transformations are $X = T_W^C$, representing the pose of the robot base with respect to the camera, and $Z = T_E^B$, representing the pose of the robot end-effector with respect to the calibration pattern. At each step i, two measurable quantities are available. The first is $A_i = T_C^{B_i}$, which describes the absolute pose of the camera with respect to the pattern, obtained from visual observations. The second is $B_i = T_W^{E_i}$, representing the pose of the robot base with respect to robot end-effector, provided by the robot's forward kinematics.

Substituting these definitions into the kinematic loop equation, we obtain the canonical form:

$$A_iX = ZB_i. \tag{2.10}$$

This formulation is particularly suited for eye-on-base configuration, where neither the camera's position in the world frame nor the mounting of the calibration pattern on the end-effector is known beforehand. While the relative motion formulation $(AX = XB)$ estimates a single unknown transformation X, assuming the calibration target to be fixed in the environment, the absolute pose formulation $(AX = ZB)$ requires solving for two static transformations simultaneously: the world-to-camera pose X and the end-effector-to-board pose Z.

Although this formulation is typically associated with the *eye-on-base* configuration, it can be equivalently applied to the *eye-in-hand* setup by simply reassigning the roles of the

known and unknown transformations. In the *eye-in-hand* case, the camera is mounted on the end-effector, while the calibration pattern is fixed in the workspace. Consequently, the static transformations to be estimated become:

$$X = T_E^C\,, \qquad Z = T_W^B\,, \tag{2.11}$$

where X represents the end-effector pose with respect to the camera reference frame, and Z represents the robot base with respect to the calibration pattern coordinate system. The kinematic loop can be derived in an analogous way by expressing the transformation from the end-effector frame E to the board reference frame B along two possible paths:

$$T_{C_i}^B \cdot T_E^C = T_W^B \cdot T_{E_i}^W \tag{2.12}$$

leading again to the canonical absolute-pose formulation:

$$A_i X = Z B_i\,, \tag{2.13}$$

where now $A_i = T_{C_i}^B$ and $B_i = T_{E_i}^W$.

Geometric calibration provides the spatial grounding necessary for perception, ensuring that visual measurements can be expressed in a consistent reference frame. However, to enable a robot to act purposefully within its environment, it must not only perceive isolated features but also reconstruct and interpret the surrounding 3D structure. The following section therefore focuses on 3D scene reconstruction, a fundamental step toward building a spatial understanding that supports autonomous and efficient robot motion.

2.3.2 Metric Evaluation

Evaluating the accuracy of a hand-eye calibration is not trivial, as the ground truth (GT) transformation is often unknown in real-world scenarios. Therefore, different metrics are used, which can be broadly divided into ground-truth-based and residual-based evaluations.

Ground Truth Error When the GT hand-eye transformations are known, this is the most direct metric. Given an estimated hand-eye transformation $\hat{X} \in \mathbb{R}^{4\times4}$ (with rotation $\hat{R}$ and translation $\hat{t}$) and the corresponding ground truth X_{GT} (with rotation R_{GT} and translation t_{GT}), the error is split into rotational and translational components.

The translation error e_t^{GT} is the Euclidean distance between the estimated and true translation vectors:

$$e_t^{GT} = \|\hat{t} - t_{GT}\|_2\,. \tag{2.14}$$

The rotation error e_θ^{GT} is the angular difference between the estimated and true rotation matrices. It is computed as the angle of the relative rotation matrix $R_{err} = R_{GT}^T \hat{R}$:

$$e_\theta^{GT} = \text{angle}(R_{err})\,, \tag{2.15}$$

where the angle() function extracts the rotation angle from the axis-angle representation of R_{err}. If the Z transformation is also being solved (as in $AX = ZB$), the same metrics are applied to $\hat{Z}$ and Z_{GT}.

Kinematic Chain Residual Error When ground truth is unavailable, a common metric is the residual error of the kinematic chain. This metric does not measure accuracy against a "true" value, but rather the consistency of the solution with the measurements. It quantifies how well the estimated transformations satisfy the governing equation across all N measurements. The formulation depends on which equation is being solved.

For $A_iX = ZB_i$ (Absolute Pose): this metric evaluates how well $\hat{X}$ and $\hat{Z}$ fit the equation. For each measurement i, we compute the "left-hand side" pose $T_{LHS,i} = A_i\hat{X}$ and the "right-hand side" pose $T_{RHS,i} = \hat{Z}B_i$. Let $A_i = (R_{A_i}, t_{A_i})$, $B_i = (R_{B_i}, t_{B_i})$, $\hat{X} = (\hat{R}_X, \hat{t}_X)$, and $\hat{Z} = (\hat{R}_Z, \hat{t}_Z)$. The residual translation error e_t^{res} is the average Euclidean distance between the translational parts:

$$e_t^{res} = \frac{1}{N}\sum_{i=0}^{N-1} \left\| (R_{A_i}\hat{t}_X + t_{A_i}) - (\hat{R}_Z t_{B_i} + \hat{t}_Z) \right\|_2 . \tag{2.16}$$

The residual rotation error e_θ^{res} is the average angular difference between the rotational parts:

$$e_\theta^{res} = \frac{1}{N}\sum_{i=0}^{N-1} \text{angle}((R_{A_i}\hat{R}_X)^T(\hat{R}_Z R_{B_i})) . \tag{2.17}$$

For $A_iX = XB_i$ (Relative Motion): this metric evaluates how well the single unknown $\hat{X}$ fits the equation. The "left-hand side" is $T_{LHS,i} = A_i\hat{X}$ and the "right-hand side" is $T_{RHS,i} = \hat{X}B_i$. The residual translation error e_t^{res} is:

$$e_t^{res} = \frac{1}{N}\sum_{i=0}^{N-1} \left\| (R_{A_i}\hat{t}_X + t_{A_i}) - (\hat{R}_X t_{B_i} + \hat{t}_X) \right\|_2 . \tag{2.18}$$

The residual rotation error $e\theta^{res}$ is:

$$e_\theta^{res} = \frac{1}{N}\sum_{i=0}^{N-1} \text{angle}((R_{A_i}\hat{R}_X)^T(\hat{R}_X R_{B_i})),. \tag{2.19}$$

A low residual error suggests a stable and consistent solution.

Reprojection Error The reprojection error quantifies the accuracy of the estimated calibration by measuring the average pixel discrepancy between the detected 2D image points and their corresponding 3D calibration pattern points projected onto the image plane. This metric, expressed in pixels, depends on the physical configuration of the system (e.g., *eye-in-hand* or *eye-on-base*) and reflects how accurately the estimated transformations relate the camera observations to the robot's coordinate system.

Let $\hat{f}(K, d)$ denote the camera projection function, parameterized by the intrinsic matrix K and distortion coefficients d. Let P_j^B be the j^{th} 3D corner of the calibration board expressed in the board frame B, and $p_{i,j}^D$ the corresponding 2D corner detected in image i. The reprojected point $p_{i,j}^P$ is obtained by transforming P_j^B from the board frame into the camera frame C_i corresponding to the i^{th} robot pose.

- For **Eye-on-Base** configuration, the unknown transformations are $\hat{X} = T_W^C$ and $\hat{Z} = T_E^B$, while the known robot pose is $B_i = T_W^{E_i}$. The 3D point in the camera frame is computed as:

$$P^C = \left(T_W^C \cdot T_{E_i}^W \cdot T_B^E\right) \cdot P_j^B = \left(\hat{X} \cdot B_i^{-1} \cdot \hat{Z}^{-1}\right) \cdot P_j^B . \tag{2.20}$$

  The corresponding projection onto the image plane is:

$$p_{i,j}^P = \hat{f}\left(K, d, (\hat{X} \cdot B_i^{-1} \cdot \hat{Z}^{-1}), P_j^B\right) \tag{2.21}$$

- For **Eye-in-Hand** configuration, the unknown transformations are $\hat{X} = T_E^C$ and $\hat{Z} = T_W^B$, while the known robot pose is $B_i = T_{E_i}^W$. The 3D point in the camera frame C_i at pose i is given by:

$$P^{C_i} = \left(T_E^{C_i} \cdot T_W^{E_i} \cdot T_B^W\right) \cdot P_j^B = \left(\hat{X} \cdot B_i^{-1} \cdot \hat{Z}^{-1}\right) \cdot P_j^B \,. \tag{2.22}$$

  The corresponding reprojected point on the image plane is:

$$p_{i,j}^P = \hat{f}\left(K, d, (\hat{X} \cdot B_i^{-1} \cdot \hat{Z}^{-1}) \cdot P_j^B\right) \tag{2.23}$$

Finally, the overall Reprojection Error e_{reproj} is computed as the Root Mean Square Error (RMSE) over all N robot poses and M detected corners:

$$e_{reproj} = \sqrt{\frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left\| p_{i,j}^P - p_{i,j}^D \right\|_2^2} \tag{2.24}$$

A low reprojection error (e.g., sub-pixel) indicates that the estimated transformations $\hat{X}$ and $\hat{Z}$ are highly consistent with the original visual measurements.

2.4 Multi-view 3D Reconstruction

While hand-eye calibration aligns the robot's perception within its own action space, a fundamental challenge remains: constructing a consistent 3D model of the surrounding environment. Such a model allows the robot to reason about spatial relationships, navigate safely, and interact effectively with objects in its workspace.

This capability becomes essential when dedicated depth sensors (e.g., RGB-D cameras) are unavailable or unreliable, such as in outdoor scenes with strong illumination or in the presence of reflective or transparent surfaces. In these situations, 3D reconstruction relies solely on geometric principles to infer scene structure directly from multiple RGB images, without requiring active depth sensing.

The goal of multi-view 3D reconstruction is to jointly estimate the spatial coordinates of scene points and the poses of the cameras that observed them.

The central problem of 3D reconstruction lies in a fundamental interdependency:

- Estimating the 3D position of a point requires knowledge of the camera poses from which it is observed.
- Estimating the pose of a camera requires correspondences between image features and their 3D locations.

The geometric mechanism that enables the recovery of 3D structure from known camera poses is **triangulation**. As illustrated in Figure 2.7, if a 3D point $M = [X, Y, Z, 1]^\top$ is observed from two or more viewpoints with known camera poses (C_i and C_j), its image projections (m_i and m_j) define 3D viewing rays originating from the respective optical centers.

In practice, due to image noise and calibration errors, the rays do not intersect perfectly. Triangulation is therefore formulated as an optimization problem that seeks the 3D point M minimizing the reprojection error, i.e., the distance between the observed image projections m_i and the reprojected estimates $\hat{f}(K_i, R_i, t_i, M)$, assuming no distortions:

$$\min_M \sum_i \left\| m_i - \hat{f}(K_i, R_i, t_i, M) \right\|_2 , \tag{2.25}$$

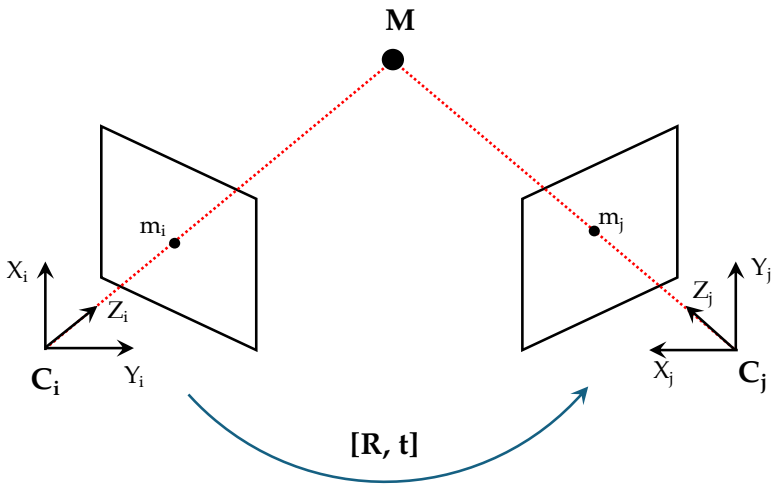


FIGURE 2.7: Geometric model of 3D point triangulation

Triangulation thus allows to recover 3D structure, but it presupposes known camera motion, specifically, the poses $[R_i|t_i]$. This raises a fundamental question: how can structure and motion be recovered when neither is known in advance?

Structure-from-Motion (SfM) This interdependency is addressed by the Structure-from-Motion (SfM) technique [1]. SfM recovers both the (sparse) 3D structure of the scene and the camera poses (the motion) from an unordered collection of images exhibiting sufficient overlap.

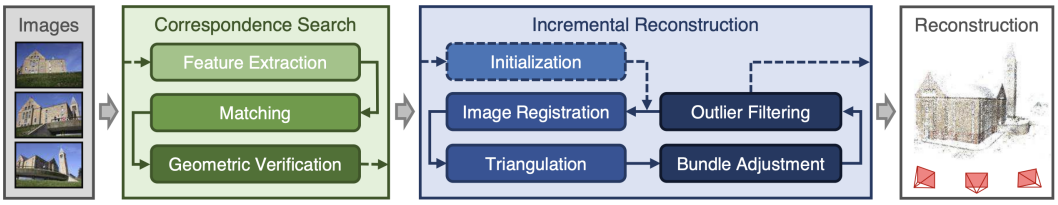


FIGURE 2.8: Incremental Structure-from-Motion pipeline [1]

SfM is typically a multi-stage pipeline, as illustrated in Figure 2.8, which can be summarized as follows:

- **Feature extraction and matching:** For each image, distinctive keypoints are detected and described by feature vectors using methods such as SIFT [63] or ORB [64]. These descriptors, designed to be robust to changes in scale, rotation, and illumination, are then matched across images to establish 2D-2D correspondences.
- **Geometric verification:** These 2D-2D correspondences inevitably contain outliers (incorrect matches). To filter them, the Essential Matrix E [65] for intrinsically calibrated cameras or the Fundamental Matrix F [66] for uncalibrated cameras is estimated within a robust framework such as RANSAC (Random Sample Consensus) [67]. These matrices encode the epipolar geometry between two views, for a pair of corresponding points in normalized image coordinates, m_i and m_j, the constraint is:

$$m_j^\top E_{ij} m_i = 0 \quad \text{or} \quad m_j^\top F_{ij} m_i = 0 \tag{2.26}$$

defining the geometric constraint that imposes that a point in one image must lie on a specific line (the epipolar line) in the other image. Only correspondences that satisfy this constraint (inliers) are retained.

- **Initialization:** This crucial stage initiates the 3D reconstruction. An initial pair of images is selected (often one with many inliers and sufficient baseline). The Essential Matrix E is decomposed to recover the relative rotation R and translation (up to scale) t between the two cameras. A coordinate system is established (e.g., $C_1 = [I|0]$ and $C_2 = [R|t]$), and triangulation is employed to compute the initial 3D positions of common correspondences, creating the sparse "seed map".

- **Incremental reconstruction:** The reconstruction is then extended by adding one image at a time. For a new image, 2D-3D correspondences are established between its 2D keypoints and the existing 3D map points. The Perspective-n-Point (PnP) [68] problem is then solved (often within RANSAC) to estimate the 6-DOF pose ($[R|t]$) of the new camera k. Once the new camera's pose is known, it is used (along with other registered cameras) to triangulate new 3D points from keypoints that had not been previously reconstructed using (2.25), thereby expanding the 3D map.

- **Bundle Adjustment (BA):** As more images are added, small errors accumulate, leading to "drift" in the reconstruction. Bundle Adjustment (BA) [69] is a global, non-linear optimization that jointly refines all estimated parameters (all camera poses $\{R_i, t_i\}$ and all 3D point coordinates $\{M_j\}$) by minimizing the total reprojection error over all N cameras and M points:

$$\min_{\{R_i,t_i\},\{M_j\}} \sum_{i=1}^{N} \sum_{j=1}^{M} v_{ij} \left\| m_{ij} - \hat{f}(K_i, R_i, t_i, M_j) \right\|_2 \tag{2.27}$$

  where v_{ij} is a binary variable (1 if camera i observes point j, 0 otherwise), and m_{ij} is the observed projection. This step is critical for achieving a geometrically consistent and accurate reconstruction

The output of SfM is a set of accurate camera poses $T_{C_i}^{W}$ and a sparse 3D reconstruction of the environment.



FIGURE 2.9: Sparse 3D reconstruction of Rome with 21K images, resulted from Structure-from-Motion process [1]

Multi-View Stereo (MVS) While SfM yields a sparse geometric representation, many robotic and vision applications require dense, metrically complete reconstructions. Multi-View Stereo (MVS) [70] addresses this.

MVS leverages the camera poses, now known and refined by SfM, to compute dense depth maps. For each pixel in a reference image, MVS searches for the most probable correspondence in neighboring images. Because the epipolar geometry is known, this search is not performed over the entire image but is constrained to the corresponding epipolar line, using photometric similarity measures (e.g., Normalized Cross-Correlation, NCC). Once a consistent match is found, depth is estimated via triangulation. Finally, the per-view depth maps are fused into a single, dense 3D point cloud or polygonal mesh.

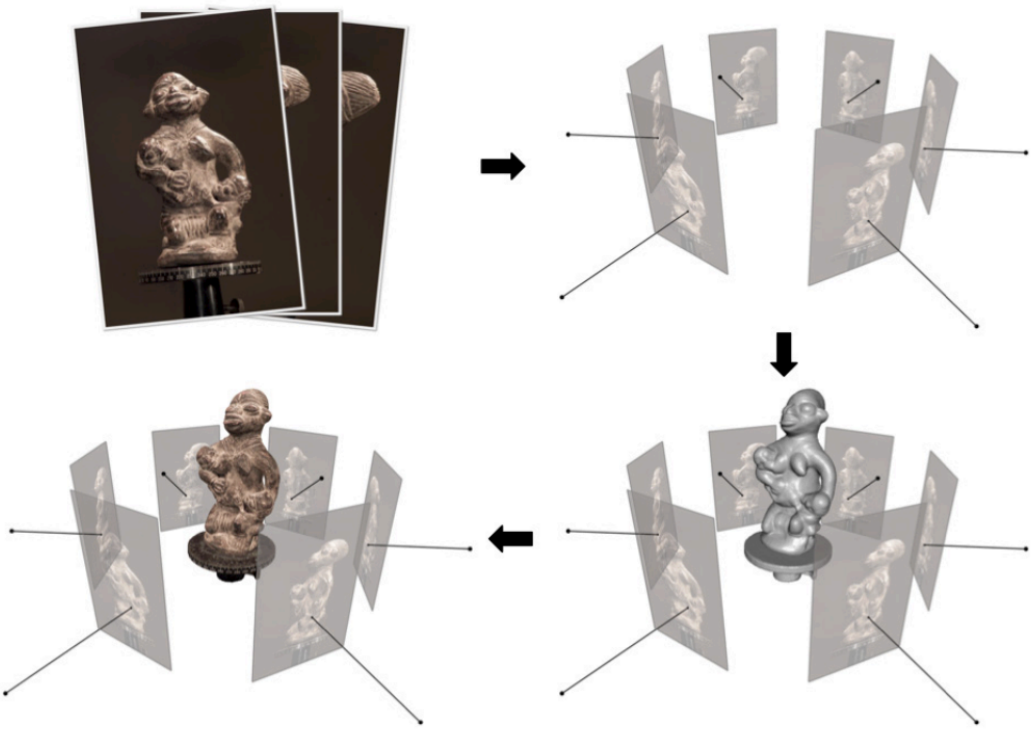


FIGURE 2.10: Example of a multi-view stereo pipeline. Clockwise: input imagery, posed imagery, reconstructed 3D geometry, textured 3D geometry [2]

This classical pipeline (Image Collection → Feature Matching → SfM → MVS), while conceptually elegant, remains practically fragile. It relies heavily on the ability to detect and match accurate features and requires significant image overlap (often exceeding 60%) for stable matching. Real-world conditions, however, often violate these assumptions due to low-texture surfaces, occlusions, illumination changes, or wide-baseline motion, causing feature-based methods to fail. These limitations have marked a progressive shift toward end-to-end learning-based solutions.

2.5 Deep Learning for Visual Perception

The fragility of classical geometric methods and early computer vision systems stems from their reliance on hand-engineered features. The advent of deep learning has radically transformed visual perception, enabling models to learn rich semantic and structural information directly from raw sensory data. This paradigm shift, from engineered rules to data-driven representation learning, has provided the tools to overcome the core limitations of the classical pipeline. This section presents a concise overview of the two foundational architectures of

modern visual perception: Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs).

2.5.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [71] were the first fundamental building block of the deep learning revolution in vision. Their success stems from the convolutional layer (Figure 2.11), which exploits spatial locality and parameter sharing to efficiently capture hierarchical visual patterns like edges, textures, and object parts. Formally, a convolutional

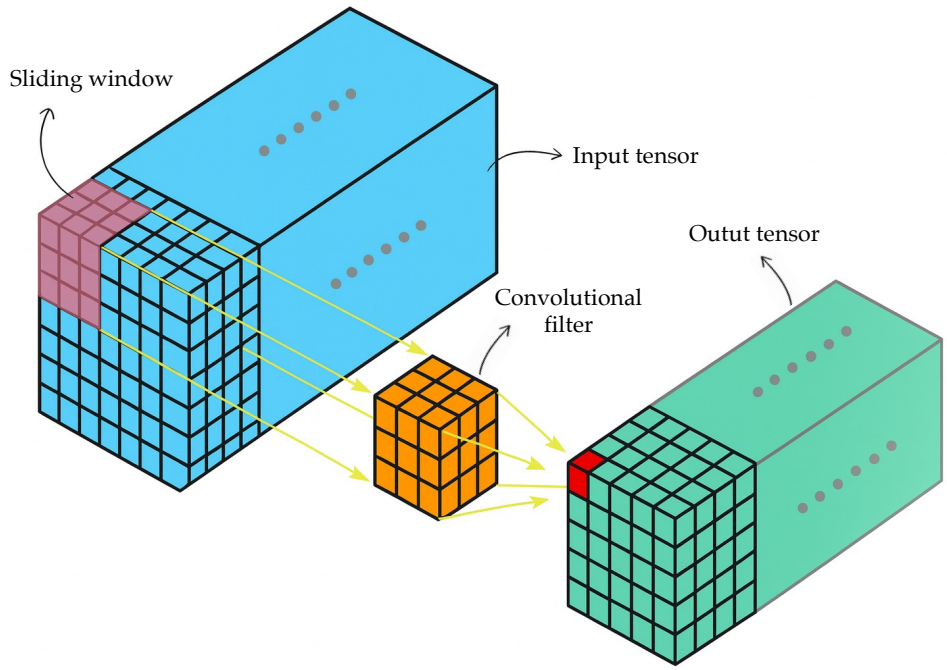


FIGURE 2.11: Illustration of the convolution operation. The input tensor (blue) is processed by a convolutional filter (orange) to produce an output tensor (green). Each cell in the output corresponds to a weighted sum of a local receptive field in the input, followed by a non-linear activation.

layer applies a bank of learnable filters $\mathbf{W}^{(l)} \in \mathbb{R}^{k_h \times k_w \times C_{in} \times C_{out}}$ to an input tensor $\mathbf{X}^{(l)} \in \mathbb{R}^{H \times W \times C_{in}}$, producing an output feature map:

$$\mathbf{X}^{(l+1)}_{h,w,c} = \sigma\left(\sum_{i=1}^{k_h}\sum_{j=1}^{k_w}\sum_{u=1}^{C_{in}} \mathbf{W}^{(l)}_{i,j,u,c} \cdot \mathbf{X}^{(l)}_{h+i,w+j,u} + b^{(l)}_c\right), \tag{2.28}$$

where $\sigma(\cdot)$ is a nonlinear activation function (e.g., ReLU) and $b^{(l)}_c$ is the bias term. Stacking multiple convolutional and pooling layers allows hierarchical feature extraction: shallow layers respond to low-level cues (edges, gradients), whereas deeper layers encode high-level abstractions (object parts, semantics) [72]. This makes CNNs highly efficient and effective in several tasks, such as object recognition, segmentation, and pose estimation.

Classic architectures such as *AlexNet* [73], *VGG* [74], and *ResNet* [75] have demonstrated how depth and residual connections significantly improve generalization and representation capacity [49]. To enable pixel-wise predictions, architectures like Fully Convolutional Networks (FCNs) [76] replaced fully connected layers with convolutional counterparts, producing dense outputs that preserve spatial coherence. These ideas inspired more advanced encoder-decoder networks such as U-Net [77] and DeepLabV3+ [78], widely adopted in

robotics for tasks including semantic segmentation, image classification, and 2D/3D human pose estimation [79, 80].

Despite their effectiveness, CNNs suffer from an intrinsic limitation: their receptive field grows only locally through successive layers. Capturing long-range dependencies across distant regions of an image remains difficult, motivating the development of architectures capable of global reasoning.

2.5.2 Transformer-based Architecture

Transformers, originally introduced for sequence modeling in natural language processing [81], have redefined representation learning by replacing convolutional operations with self-attention mechanisms. The fundamental insight is that relationships between all elements in the input can be learned directly, allowing the model to capture both local and global dependencies without relying on fixed receptive fields. The key idea of attention is to dynamically weight the contribution of different input elements depending on their relevance to a given query. Figure 2.12 illustrates the transformer architecture and the attention mechanism.

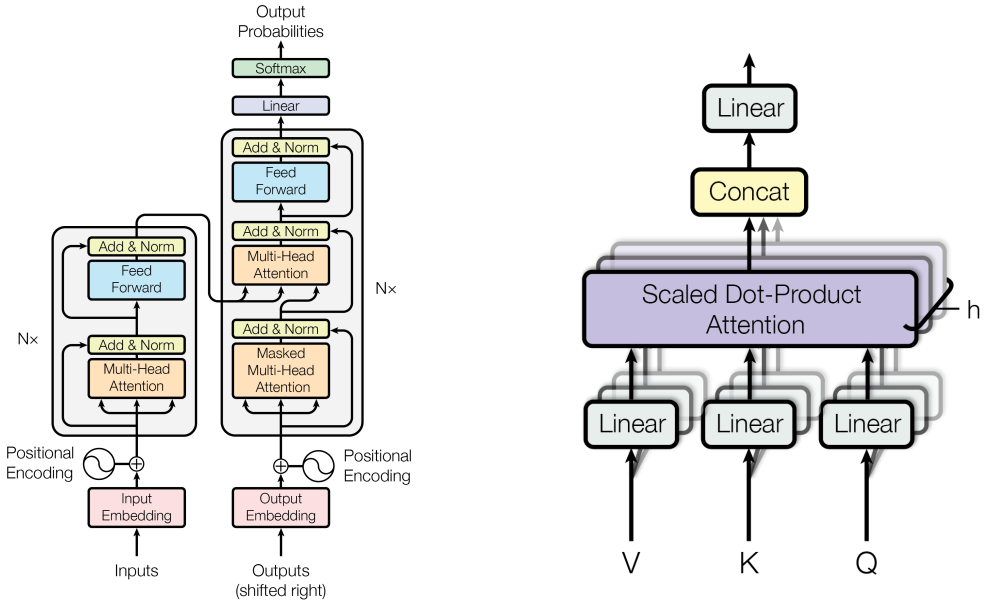


FIGURE 2.12: At the left the Transformer architecture. At the right the Multi-Head Attention consisting of several attention layers running in parallel.

Given an input sequence $\mathbf{X}$ of N tokens, each represented by a D-dimensional embedding, the model computes three projections:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V, \tag{2.29}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D \times D_k}$ are trainable matrices that generate queries, keys, and values, respectively. The attention operation is then defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_k}}\right)\mathbf{V}. \tag{2.30}$$

where $\sqrt{D_k}$ is a scaling factor to stabilize gradients. This operation allows each token to attend to all others, producing context-aware representations. To enhance representational

capacity, h attention heads are computed in parallel:

$$\text{MultiHead}(\mathbf{X}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O, \tag{2.31}$$

where each $\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$ and $\mathbf{W}_O \in \mathbb{R}^{(hD_v)\times D}$ is a learned projection matrix. Each transformer layer includes multi-head attention, a position-wise feedforward network, residual connections, and normalization layers, producing highly expressive hierarchical features.

The attention mechanism has a complexity of $O(N^2D)$, which can be computationally expensive for large N. However, its ability to model global dependencies and context has led to significant performance improvements across various tasks. In vision, this means that each region of an image can directly attend to other regions, regardless of their spatial distance, enabling a more holistic representation of the scene. The most influential instantiation of this concept is the vision transformer.

Vision Transformer (ViT). The Vision Transformer (ViT) [82] adapts this paradigm to images by representing an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times C}$ as a sequence of N non-overlapping patches. Each patch of size $P \times P$ is flattened and linearly projected into a D-dimensional embedding, resulting in:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{N\times D}, \quad N = \frac{HW}{P^2}. \tag{2.32}$$

Positional encodings $\mathbf{P} \in \mathbb{R}^{N\times D}$ are then added to preserve spatial information:

$$\mathbf{Z}^{(0)} = \mathbf{X} + \mathbf{P}. \tag{2.33}$$

The resulting token sequence $\mathbf{Z}^{(0)}$ is processed by a stack of transformer encoder layers to yield a global representation of the image, which can be used for classification or dense prediction.

ViT has achieved state-of-the-art results in numerous benchmarks, but its performance depends heavily on large-scale pretraining and substantial computational resources. To address these limitations, hierarchical variants such as the *Pyramid Vision Transformer* [83] introduce multi-stage feature hierarchies and spatial reduction attention mechanisms, combining the global reasoning of transformers with the efficiency and multi-scale representation of CNNs.

The ability of Transformers to model global dependencies and contextual relationships across an entire image provides a powerful mechanism for visual matching, far exceeding the capabilities of local, hand-crafted features. Armed with these architectures, the field also revisited the problem of 3D reconstruction, aiming to replace the fragile classical pipeline with robust, end-to-end learned models.

2.6 Modern 3D Reconstruction with Deep Learning

The convergence of deep learning and geometric vision has led to a new generation of 3D reconstruction models. Initially, this involved ”hybrid” approaches, where deep learning (primarily CNNs) replaced specific, fragile components of the classical pipeline, such as feature matching (e.g., SuperPoint/SuperGlue [84, 85]) or the MVS densification step (e.g., MVSNet [86]).

While these hybrid models improved robustness, they still inherited the complexity of a multi-stage process. The latest paradigm shift, enabled by the power of Transformers, is to unify the entire pipeline, from pixels to 3D model, into a single, end-to-end architecture.

2.6.1 DUSt3R and Its Successors

DUSt3R (Dense and Unconstrained Stereo 3D Reconstruction) [3] is a transformer-based architecture that replaces the modular structure of classical 3D reconstruction pipelines. It directly regresses dense 3D geometry from a pair of RGB images without requiring prior knowledge of camera intrinsics or poses at inference time. The central idea is to predict pointmaps: dense, per-pixel 3D coordinates expressed in a shared reference frame. This representation unifies correspondence estimation and 3D reconstruction. Given two input images (I_1, I_2), the network outputs two pointmaps and their corresponding confidence maps:

$$X_{1,1}, X_{2,1} \in \mathbb{R}^{W\times H\times 3}, \quad C_{1,1}, C_{2,1} \in \mathbb{R}^{W\times H}. \tag{2.34}$$

Here, $X^{n,m}$ denotes the pointmap of image I_n expressed in the coordinate frame of image I_m. This formulation allows the network to model the geometric relationship between the views without explicitly solving for camera motion or performing triangulation.

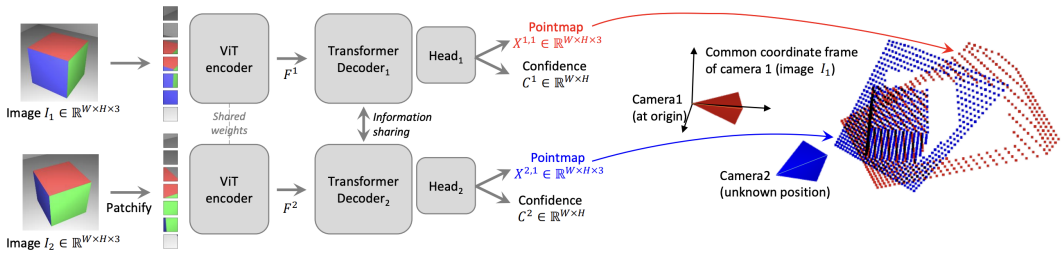


FIGURE 2.13: Architecture of DUSt3R [3]. Two input images (I_1, I_2) are encoded in a Siamese manner using a shared Vision Transformer encoder. The token representations F^1 and F^2 are processed by transformer decoders that exchange information through cross-attention layers, allowing the model to reason about geometric relationships between views. Finally, two regression heads predict the corresponding 3D pointmaps and their confidence maps, both expressed in the same coordinate frame of I_1.

The architecture, shown in Figure 2.13, uses a Siamese Vision Transformer (ViT) encoder to extract dense visual tokens F^1 and F^2. These tokens are processed by two transformer decoders that iteratively exchange information via cross-attention, enabling geometric reasoning across both views. Each branch concludes with a lightweight regression head that outputs a 3D coordinate and a confidence score for every pixel.

During training, the model learns to produce geometrically consistent pointmaps by minimizing a loss function. This loss enforces that corresponding pixels across the two images map to the same 3D location in the coordinate frame of I_1. The confidence maps learn to automatically down-weight occluded or out-of-view regions, allowing the model to handle incomplete observations robustly.

The result for a single pair of images is a dense 3D reconstruction, internally consistent and aligned in a common coordinate frame (that of I_1), though its global scale remains unknown. However, the authors also demonstrate that by processing multiple overlapping pairs, the resulting pointmaps can be robustly stitched together into a single, globally consistent 3D model (via global alignment), resolving the scale ambiguity and achieving large-scale metric reconstruction.

This approach represents a fundamental paradigm shift. Rather than enforcing 3D consistency through explicit projective geometry, DUSt3R learns it implicitly as an emergent property of the network. This allows the model to capture robust geometric priors directly from data, making it effective even in textureless regions, under varying illumination, or in the presence of moderate occlusions—conditions that often challenge classical geometry-based



FIGURE 2.14: Reconstruction example from 4 random frames of a RealEstate10K indoor sequence, after global alignment. On the left-hand side, the 4 input frames are depicted, and on the right-hand side the resulting point-cloud and the recovered camera intrinsics and poses [3].

pipelines. As a result, DUSt3R produces dense 3D reconstructions that are geometrically coherent and well-aligned across views.

MASt3R: Joint 3D Reconstruction and Feature Matching

DUSt3R has inspired a new generation of learning-based reconstruction methods. MASt3R [4] builds upon DUSt3R by extending its architecture with an additional local descriptor prediction head, denoted as $\text{Head}_{\text{desc}}$ as illustrated in Figure 2.15, enabling the network to jointly perform dense 3D reconstruction and robust local feature matching.

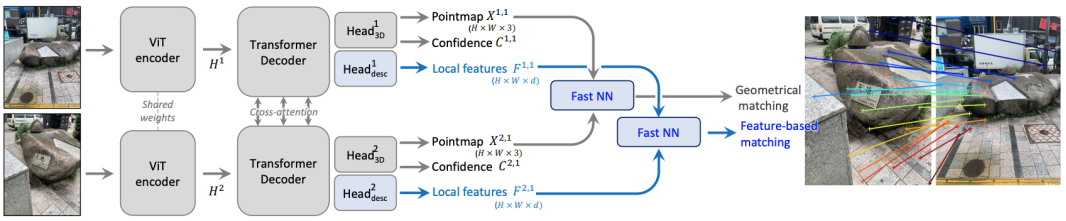


FIGURE 2.15: Architecture of MASt3R [4]. Given two input images (I_1, I_2) to match, MASt3R network regresses for each image and each input pixel a 3D point, a confidence value and a local feature. Plugging either 3D points or local features into a fast reciprocal NN matcher yields robust correspondences. Compared to the DUSt3R framework which it builds upon, the contributions are highlighted in blue.

Given a pair of images depicting the same scene, MASt3R processes them using siamese Vision Transformer (ViT) encoders with shared weights, producing latent representations H^1 and H^2. The encoders are initialized from the pretrained DUSt3R model, which is trained using a CroCo-style self-supervised pretraining strategy to encourage cross-view consistency.

The latent representations are then fed into transformer decoders equipped with cross-attention mechanisms, allowing each view to exchange spatial and geometric information with the other. This cross-view interaction enables the model to reason jointly about correspondences between the two images.

Each decoder outputs predictions through two task-specific heads. The 3D head, denoted as Head^{3D}, regresses, for each input pixel, a pointmap and the associated confidence value. In

parallel, the descriptor head $\text{Head}_{\text{desc}}$ predicts dense local feature maps with spatial resolution $H \times W \times d$, where d denotes the descriptor dimensionality.

Unlike DUSt3R, which relies on a scale-invariant regression loss for supervision, MASt3R introduces a contrastive learning objective based on the InfoNCE loss to supervise descriptor learning. This matching loss explicitly enforces consistent pixel correspondences across views, leading to more robust local image matching while preserving the ability to output metric 3D point maps.

The predicted 3D point maps and local descriptors can then be independently used for geometrical matching or feature-based matching via a fast reciprocal nearest-neighbor search, as illustrated in Figure 2.15.

MonST3R: Extension to Dynamic Scene Reconstruction

MonST3R [87] extends the DUSt3R framework to the reconstruction of dynamic scenes. Rather than explicitly modeling object motion, the authors leverage DUSt3R's pointmap representation to estimate scene geometry independently at each time step. The key insight is that pointmaps can be predicted per frame and expressed in a common camera coordinate system, which remains a meaningful representation even in the presence of dynamic objects.

As illustrated in Figure 2.16, the resulting pointmap sequence forms a time-varying point cloud in which dynamic objects appear at multiple spatial locations according to their motion, while static scene elements remain consistent across frames. Multi-frame alignment is achieved by registering pairs of pointmaps using only static regions of the scene, effectively disentangling camera motion from object dynamics. This formulation can be seen as a natural generalization of DUSt3R to dynamic settings and allows MonST3R to reuse the original network architecture and pretrained weights as initialization.

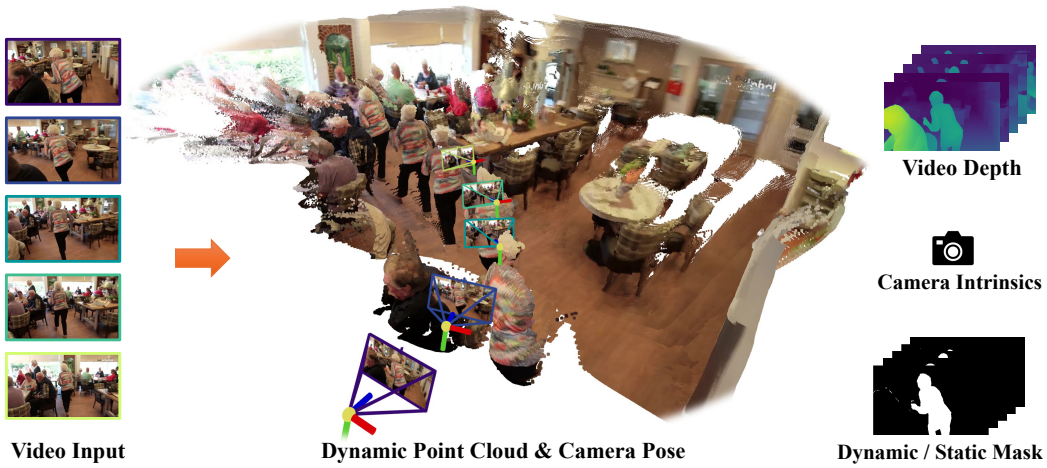


FIGURE 2.16: MonST3R processes a dynamic video to produce a time-varying dynamic point cloud, along with per-frame camera poses and intrinsics, in a predominantly feed-forward manner. This representation then enables the efficient computation of downstream tasks, such as video depth estimation and dynamic/static scene segmentation.

VGGT: Fully Feed-Forward 3D Reconstruction

VGGT [88] represents a further departure from optimization-based reconstruction pipelines by eliminating the need for explicit post-processing or iterative geometric refinement. The model introduces a large feed-forward transformer architecture capable of performing 3D reconstruction from one, a few, or even hundreds of input views in a single forward pass.

VGGT jointly predicts a comprehensive set of 3D attributes, including camera intrinsics and poses, depth maps, point maps, and long-range 3D point tracks. Remarkably, despite its fully feed-forward nature and minimal reliance on explicit geometric constraints, VGGT often outperforms optimization-based methods in both accuracy and efficiency, producing usable reconstructions within seconds. This marks a substantial conceptual shift compared to DUSt3R, MASt3R, and VGGSfM, which still depend on costly global alignment or iterative optimization stages to achieve consistent multi-view reconstructions.



FIGURE 2.17: VGGT is a large feed-forward transformer with minimal 3D-inductive biases trained on a trove of 3D-annotated data. It accepts up to hundreds of images and predicts cameras, point maps, depth maps, and point tracks for all images at once in less than a second, which often outperforms optimization-based alternatives without further processing.

Despite their architectural differences, DUSt3R and its successor models share a fundamental limitation: metric scale cannot be recovered from RGB input alone. As a result, the reconstructed geometry is defined only up to an arbitrary scale and remains unaligned with the robot's own reference frame. This prevents these representations from being directly applicable to robotic tasks, where metric consistency and accurate alignment with the robot coordinate system are essential. Addressing this limitation constitutes one of the central challenges of this thesis. In Chapter 3, we introduce a framework to recover metric scale and produce reconstructions that are directly anchored to the robot's reference frame.

Chapter 3

Robot-Centric Perception

Perception becomes meaningful only when visual information is spatially aligned in the robot's own reference frame. As introduced in Chapter 1, without such alignment, perception remain disconnected from robot action, preventing effective and safe interaction with the environment.

This perception alignment is realized through *camera-to-robot calibration*, a family of methods that estimate the rigid transformation between one or more cameras and the robot's reference frames, as illustrated in Figure 3.1.

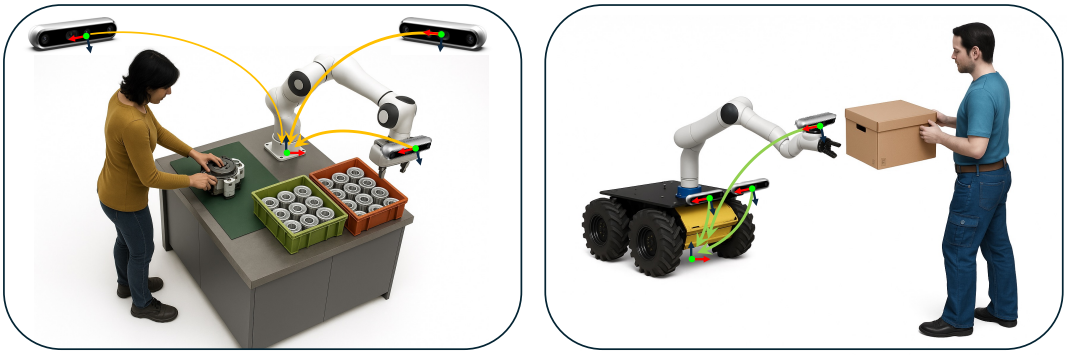


FIGURE 3.1: Illustration of representative human-robot collaboration scenarios highlighting the robot-centric perception problem, which aims to estimate the relative poses between multiple cameras and the robot reference frames denoted with colored arrows.

While classical camera-to-robot calibration techniques provide the mathematical foundations for this estimation, their applicability in real-world Human-Robot Collaboration (HRC) and mobile scenarios remains limited. Robots operate in unstructured environments, where multiple cameras, whether fixed, mounted on manipulators, or attached to mobile platforms, must be consistently calibrated quickly and reliably. This requires methods that minimize production downtime and human intervention, ultimately enabling autonomous, patternless, and self-calibrating systems.

3.1 Related-Works in Camera-to-Robot Calibration

Single-camera hand-eye calibration In the literature, several methods tackle hand-eye calibration in single-camera setups. Early works focused on the eye-in-hand configuration, formalizing it as $AX = XB$ [89, 61]. Subsequent approaches explored different rotation parametrization, including quaternions [90], Lie theory [91], dual quaternions [92], and the Kronecker product [93]. Other methods address the $AX = ZB$ formulation, allowing the simultaneous estimation of both the camera-to-base (X) and end-effector-to-board (Z) transformations [62, 94, 95]. However, all these classical methods typically depend on directly

measuring the board-to-camera transformation A using the Perspective-n-Point (PnP) algorithm [60], see Section 2.3.1. This approach can introduce significant errors, particularly when handling blurred or low-resolution images that hinder precise pattern detection. To overcome this, alternative methods minimize the visual re-projection error, the 2D distance between observed pattern features and their 3D points re-projected onto the image plane. Tabb et al. [96] proposed a method parameterizing the rotation components using Euler angles and then finding the solution using Levenberg-Marquadt iterative approach. Koide et al. [97] formulated this minimization as a pose graph optimization, achieving high accuracy at a high computational cost.

Multi-camera hand-eye calibration In scenarios involving multiple cameras, the spatial transformation between cameras is often determined either by performing hand-eye calibration for each individual camera independently and applying a transformation chain, or by calibrating just one camera using hand-eye calibration followed by stereo camera calibration [26]. However, both approaches carry the risk of error propagation. Only a few works address the simultaneous calibration of a camera network relative to a robot. Wang et al. [98] proposed a multi-camera calibration method to handle non-overlapping camera network setups, relying on an external motion capture system to achieve precise camera position during the calibration process. Tabb et al. [99] introduced a multi-camera hand-eye calibration for a small robotic workcell, relying on the minimization of re-projection error, considering the board-to-end-effector transformation Z to be unique for all cameras. However, a critical limitation of these methods is their reliance on close-proximity detection; they assume cameras are positioned near the calibration pattern (e.g., $<$ 1 meter). This assumption fails in real-world HRC scenarios, where cameras must be placed far from the robot to monitor the entire workspace and human activity. In these settings, detecting the pattern becomes challenging, degrading calibration accuracy and requiring many images to converge. This highlights a critical gap between existing methods and the needs of large-scale HRC environments.

Motion-based Calibration Motion-based calibration for mobile robots is fundamentally related to hand-eye calibration [100], solving $AX = XB$ where A and B represent the relative motions of the camera and the robot base (derived from odometry), respectively [101, 102, 103, 27]. However, it is well-known that these methods fail to recover the full 6-DoF transformation when the robot moves on a planar surface [28]. The lack of vertical motion makes the z-component of the transformation unobservable. To address this, some methods use ground plane detection, but often in a sequential two-step process that can introduce inconsistencies and requires many images [104]. Learning-based methods have emerged, using road features [105] or vanishing points [106], but these rely on strong environmental priors (e.g., lane markings, straight-line motion) that are not always available.

3.2 Camera-to-Robot Calibration Contributions

Building upon the calibration principles from Chapter 2 and addressing the limitations identified in the state-of-the-art, this chapter presents a series of contributions that target camera-to-robot calibration across progressively more complex robotic configurations. The discussion evolves from classical hand-eye calibration toward unified, patternless, and reconstruction-driven paradigms, following a trajectory that mirrors the increasing complexity and autonomy of modern robotic systems.

The first part focuses on the **single-camera hand-eye calibration** problem, proposing a unified iterative formulation that applies seamlessly to both *eye-on-base* and *eye-in-hand* setups. Starting from this theoretical groundwork, the second part extends the concept to

multi-camera hand-eye calibration, addressing the challenges of jointly optimizing several cameras surrounding a robot arm, proposing also a graph-based optimization framework. To rigorously evaluate and benchmark these complex multi-camera systems, a gap in existing research, this stage concurrently introduces the METRIC dataset, a novel benchmark specifically designed for this purpose. The third part focuses on **motion-based calibration** for mobile robots, where the robot-centric perception is achieved by exploiting motion constraints and ground-plane geometry to estimate full 6-DoF extrinsics even under planar motion. As the final step in this progression, the chapter culminates with Calib3R, a foundation-model-based approach that unifies **multi-camera calibration** and **metric 3D reconstruction** into a single, patternless pipeline capable of autonomous alignment in complex environments.

Altogether, these works trace a coherent evolution from classical calibration to unified, patternless, and reconstruction-driven alignment, advancing toward truly embodied perception and seamless visual-motor integration in robotic systems. Portions of this chapter have been published in the following works:

- [51] Daniele Evangelista, **Davide Allegro**, Matteo Terreran, Alberto Pretto and Stefano Ghidoni, "An unified iterative hand-eye calibration method for eye-on-base and eye-in-hand setups." In IEEE 27th International Conference on Emerging Technologies and Factory Automation ETFA 2022, Stuttgart, Germany.
- [50] Daniele Evangelista, Emilio Olivastri, **Davide Allegro**, Emanuele Menegatti and Alberto Pretto, "A Graph-Based Optimization Framework for Hand-Eye Calibration for Multi-Camera Setups." In: IEEE International Conference on Robotics and Automation ICRA 2023, London, UK.
- [44] **Davide Allegro**, Matteo Terreran, and Stefano Ghidoni, "Multi-Camera Hand-Eye Calibration for Human-Robot Collaboration in Industrial Robotic Workcells." IEEE Robotics and Automation Letters R-AL 2024.
- [45] **Davide Allegro**, Matteo Terreran, and Stefano Ghidoni, "METRIC—Multi-Eye to Robot Indoor Calibration Dataset." Information, MDPI, 2023.
- [6] **Davide Allegro**, Matteo Terreran, and Stefano Ghidoni, "MEMROC: Multi-Eye to Mobile RObot Calibration" Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems IROS 2024, Abu Dhabi.
- [42] **Davide Allegro**, Matteo Terreran, and Stefano Ghidoni, "Calib3R: A 3D Foundation Model for Multi-Camera to Robot Calibration and 3D Metric-Scaled Scene Reconstruction." submitted to IEEE Transactions on Robotics T-RO 2025.

3.3 A Unified Iterative Hand-Eye Calibration Method for Eye-on-Base and Eye-in-Hand Setups

As discussed in the state of the art (Section 3.1), classical hand-eye calibration methods are typically fragmented, addressing the *eye-in-hand* and *eye-on-base* configurations separately. However, many real-world applications require both setups, such as collaborative tasks where an eye-in-hand camera locates objects for the robot, while a fixed eye-on-base camera monitors the human operator for safety [107]. This fragmentation requires operators to use different tools and methods, complicating the calibration process. Furthermore, most methods rely on PnP algorithms, which can be unreliable when images of the calibration pattern are not sharp (e.g., due to lighting, motion blur, or low sensor resolution).

This section proposes an hand-eye calibration methodology that addresses both limitations. It provides a single, unified formulation capable of accommodating both camera-to-robot configurations. Critically, the method operates by directly minimizing the 2D reprojection error, the pixel distance between detected 2D corners and their corresponding 3D points reprojected onto the image plane. This avoids the intermediate, error-prone PnP step. Thanks to this formulation, the method can be applied to both eye-in-hand and eye-on-base setups by correctly inverting the known transformations within the same optimization algorithm.

The proposed method solves this as a nonlinear optimization, aiming to find the unknown transformations X and Z that minimize the sum of squared distances. Here, i denotes the i^{th} robot pose (from 0 to $N-1$) and j corresponds to the j^{th} corner (from 0 to $M-1$). The optimization problem is formalized as:

$$\underset{X,Z}{\operatorname{argmin}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left\| p_{ij}^{D} - p_{ij}^{P} \right\|_2^2 \tag{3.1}$$

where $p_{ij}^{D} = \left(u_x, u_y\right)_{ij}^{D}$ is the detected 2D corner and $p_{ij}^{P} = \left(u_x, u_y\right)_{ij}^{P}$ is its reprojected 3D pattern point. The minimization is computed with respect to the two unknown transformations X and Z. The definitions of p_{ij}^{P}, X, and Z depend on the specific hand-eye configuration, as illustrated in Figure 3.2 and detailed below.

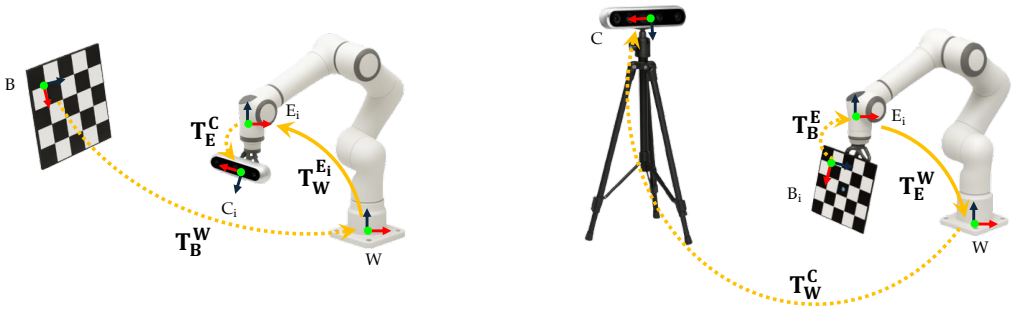


FIGURE 3.2: The two single-camera configurations solved by the unified calibration framework. On the left, the *eye-in-hand* setup, where the camera moves with the end-effector (E_i) and observes a static board B. The unknowns are the end-effector-to-camera pose ($X = T_E^C$) and the board-to-base pose ($Z = T_B^W$). On the right, the *eye-on-base* setup, where a fixed camera C observes a board B moving with the end-effector. The unknowns are the base-to-camera pose ($X = T_W^C$) and the board-to-end-effector pose ($Z = T_B^E$).

To solve the formulated hand–eye calibration problem, we implemented our optimization framework using the Ceres Solver. The cost function is defined via automatic differentiation, directly encoding the reprojection residual given in Eq. (3.1). The resulting non-linear least-squares problem is solved using a Dense Schur linear solver, combined with a Huber robust loss function to mitigate the influence of outliers and improve convergence stability.

3.3.1 Eye-in-Hand

In the *eye-in-hand* configuration (Figure 3.2, left), the camera is mounted on the robot's end-effector and observes a static calibration pattern (the "board") in the environment. The general optimization problem (3.1) applies by defining the two unknown static transformations as:

- $X = T_E^C$: the pose of the robot end-effector E with respect to the camera C.

- $Z = T_B^W$: the pose of the board B with respect to the robot base W.

The known transformation, provided by the robot's kinematics, is $T_{E_i}^W$, which is the reference frame of the i^{th} end-effector E expressed in the base reference frame W. Given the 3D points P_j^B (where $j = 0, \dots, M-1$) expressed in the board's frame B, their reprojection p_{ij}^P onto the image plane is given by the following kinematic chain:

$$p_{ij}^P = \hat{f}\left(T_E^C \cdot (T_{E_i}^W)^{-1} \cdot T_B^W P_j^B\right), \tag{3.2}$$

where $\hat{f}(P)$ is the camera projection function, which maps a 3D point P from the camera's frame to the 2D image plane using the intrinsic matrix K and d. The optimization problem thus becomes finding the X and Z that minimize the following squared re-projection error:

$$\underset{X,Z}{\operatorname{argmin}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left\| p_{ij}^D - \hat{f}\left(T_E^C \cdot T_W^{E_i} \cdot T_B^W \cdot P_j^B\right) \right\|_2^2. \tag{3.3}$$

3.3.2 Eye-on-Base

In the *eye-on-base* configuration (Figure 3.2, right), a fixed camera observes a calibration pattern mounted on the robot's end-effector. The same optimization problem (3.1) is instantiated by redefining the two unknown transformations:

- $X = T_W^C$: the pose of the robot base W with respect to the camera C.
- $Z = T_B^E$, the pose of the board B with respect to the end-effector E.

As in the previous case, the known transformation provided by the robot is $T_{E_i}^W$. The reprojection p_{ij}^P of the P_j^B points onto the image plane is therefore given by this projection model:

$$p_{ij}^P = \hat{f}\left(T_W^C \cdot T_{E_i}^W \cdot T_B^E \cdot P_j^B\right) \tag{3.4}$$

that applied to the general minimization (3.1), gives:

$$\underset{X,Z}{\operatorname{argmin}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left\| p_{ij}^D - \hat{f}\left(T_W^C \cdot T_{E_i}^W \cdot T_B^E P_j^B\right) \right\|_2^2. \tag{3.5}$$

By observing (3.3) and (3.5), it is evident how the same optimization framework (3.1) can solve both configurations, demonstrating the flexibility and generality of the unified reprojection-error-based approach.

3.3.3 Experiments and Evaluation

To rigorously validate the proposed unified calibration method, we designed a comprehensive evaluation protocol. This validation targets two distinct goals: first, a robustness analysis in a controlled simulated environment to test the optimization process's convergence and accuracy under different noise conditions; and second, a practical accuracy evaluation in real-world robotic setups to prove the method's effectiveness and flexibility in both eye-in-hand and eye-on-base configurations.

Simulated Robustness Analysis

The primary goal of the simulation was to quantify the core optimizer's robustness to the two most common sources of error in real-world calibration: visual detection noise and initial

guess perturbation. We adopted the evaluation protocol from Koide [97], generating 18 virtual images from nine camera stations observing a 7×5 pattern. To test robustness against visual noise, we perturbed the 2D corner coordinates by $(\Delta X, \Delta Y) \in [0, 5]$ pixels, mimicking detector inaccuracies. To test the convergence basin, we perturbed the initial guess for the transformations by sampling from a pose distribution with $\|t\| \in \mathcal{N}(\mu = 0, \sigma = 0.25)$ [m] and $\theta \in [0, 10]$ [deg]. This entire simulation was repeated 100 times for each noise setting to ensure statistical significance. The results are reported in Figure 3.3. The classical hand-eye calibration methods, Tsai [61] and Daniilidis [92], proved highly sensitive to visual noise, with their estimation errors degrading by an order of magnitude. This confirms their inherent weakness to the PnP errors we aimed to solve. In contrast, our method and Koide's [97], both of which directly minimize reprojection error, demonstrated high robustness. A detailed comparison, shown in Figure 3.4, further revealed that our method consistently achieved lower estimation errors than Koide's, particularly under high noise conditions. This superior performance stems from our more constrained formulation: our method optimizes only the two static unknowns (X, Z) using the robot poses as hard constraints, whereas the pose-graph in Koide [97] optimizes $N + 2$ variables, including a separate board-to-camera pose (T_B^C) for each of the N images. This simulated study thus confirms our more rigid formulation establishes a new state-of-the-art for robustness and accuracy.

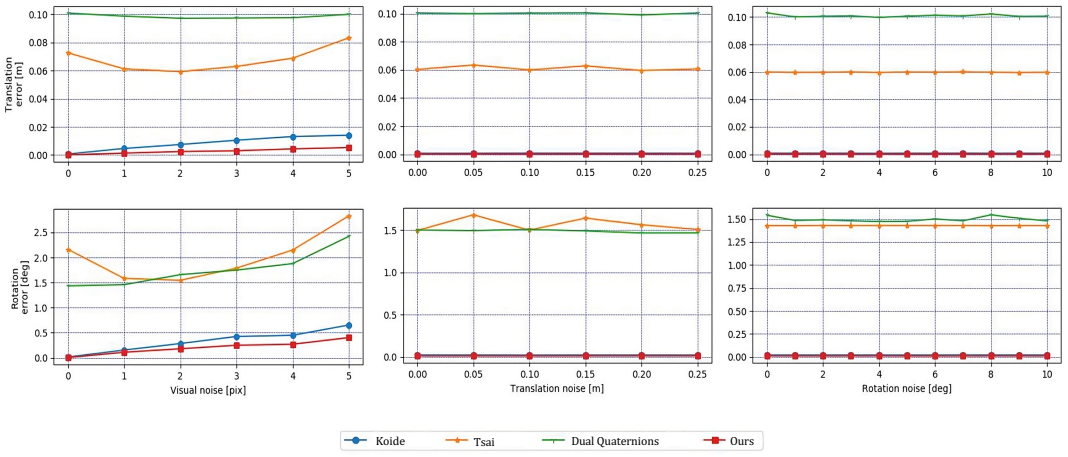


FIGURE 3.3: Robustness evaluation of all calibration methods under increasing visual detection noise (first column) and initial guess perturbations in translation and rotation (second and third columns). The proposed method demonstrates consistent accuracy across all noise levels, while classical hand-eye calibration methods (Tsai, Daniilidis) exhibit order-of-magnitude degradation in both translation and rotation estimation.

Real-World Validation

The second validation phase demonstrated the method's effectiveness in practical, real-world robotic workcells, specifically proving its unified nature. The experiments were conducted using a Franka Emika Panda arm in the two distinct configurations shown in Figure 3.5. The eye-in-hand setup (Figure 3.5, left) used a ZED camera mounted on the end-effector observing a static 9×7 checkerboard. The eye-on-base setup (Figure 3.5, right) used a fixed Microsoft Kinect V2 camera observing a 5×4 checkerboard mounted on the end-effector from a distance of 1.10 m. In both cases, only monocular RGB data was used. Lacking ground truth in the real world, performance was evaluated using the mean Reprojection Error (mRE), see Section 2.3.2.

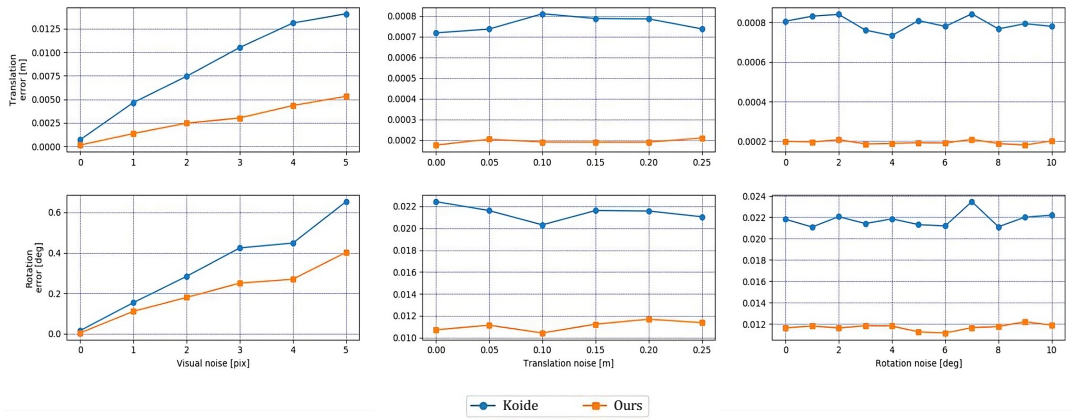


FIGURE 3.4: Focus on the result of the evaluation of the proposed hand-eye calibration method vs. Koide's method only.

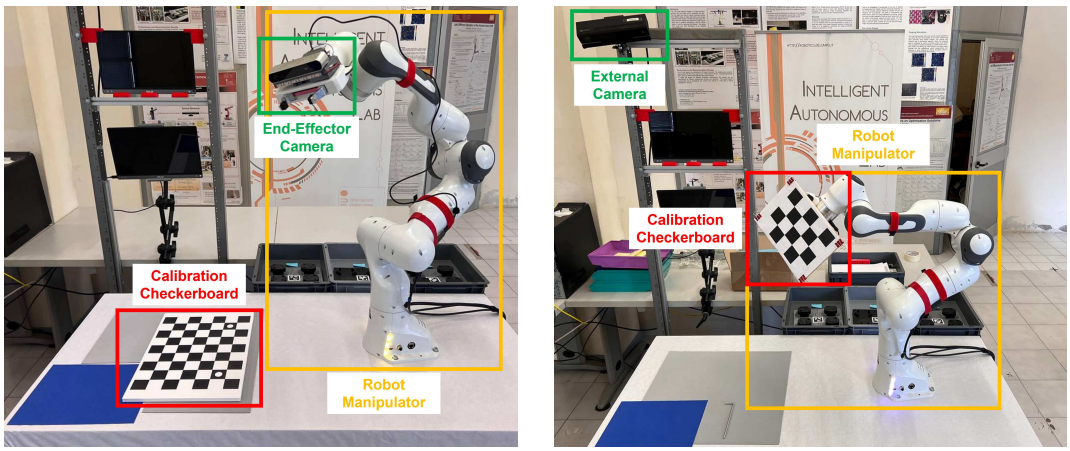


FIGURE 3.5: The two real-world experimental setups used for validation. On the left, the *eye-in-hand* configuration with a ZED camera on the end-effector. On the right, the *eye-on-base* configuration with a fixed Microsoft Kinect V2 observing the robot.

TABLE 3.1: Comparison of hand-eye calibration methods in both configurations.

Eye-in-hand Calibration	
Method	**Reprojection Error [pix]**
Tsai [61]	52.28
Daniilidis [92]	133.77
Koide [97]	13.56
Ours	**6.47**

(A) Results for the eye-in-hand setup.

Eye-on-base Calibration	
Method	**Reprojection Error [pix]**
Tsai [61]	4.51
Daniilidis [92]	7.68
Gwak [108]	5.13
Ours	**0.27**

(B) Results for the eye-on-base setup.

The quantitative results, presented in Table 3.1, demonstrate the superior performance of our unified approach.

In the eye-in-hand test (Table 3.1a), our method achieved a mean Reprojection Error (mRE) of 6.47 pixels. This represents a substantial 52.3% reduction in error compared to the next-best method, Koide (13.56 pixels). Furthermore, it performed multiple orders of magnitude better than the classical solvers (Tsai and Daniilidis), which yielded extremely

high reprojection errors (52-133 pixels mRE), indicating their struggle to find an accurate solution in this setup.

For the eye-on-base setup, Koide's implementation was not applicable, so we replaced it with Gwak's method [108] as a relevant optimization-based baseline alongside the classical solvers. The results from this test (Table 3.1b) revealed an even more dramatic improvement. Our method achieved an exceptionally low mRE of 0.27 pixels, whereas the best-performing classical method, Tsai, only reached 4.51 pixels. This signifies an order-of-magnitude improvement, corresponding to a 94.0% reduction in reprojection error compared to Tsai, demonstrating the significant advantage of our unified approach in this configuration.

The key takeaway from this validation is the confirmation of our method's unified power. The exact same formulation and implementation, without any modification, achieved state-of-the-art accuracy in both distinct physical configurations. This proves that our approach not only simplifies the calibration pipeline by eliminating the need for fragmented, configuration-specific tools but also delivers a more robust and precise perception from the robot perspective.

3.4 Multi-Camera Hand-Eye Calibration

While the unified framework in Section 3.3 addresses the fragmentation between eye-in-hand and eye-on-base setups for a single sensor, modern industrial scenarios introduce a new layer of complexity: multi-camera systems.

Projects such as Sharework[1] and DrapeBot[2] utilize networks of cameras to supervise Human-Robot Collaboration, avoid occlusions, and monitor large workcells. This architecture constitutes a large-scale, multi-camera *eye-on-base* setup (as shown in Figure 3.6), enabling continuous monitoring of the entire workspace.

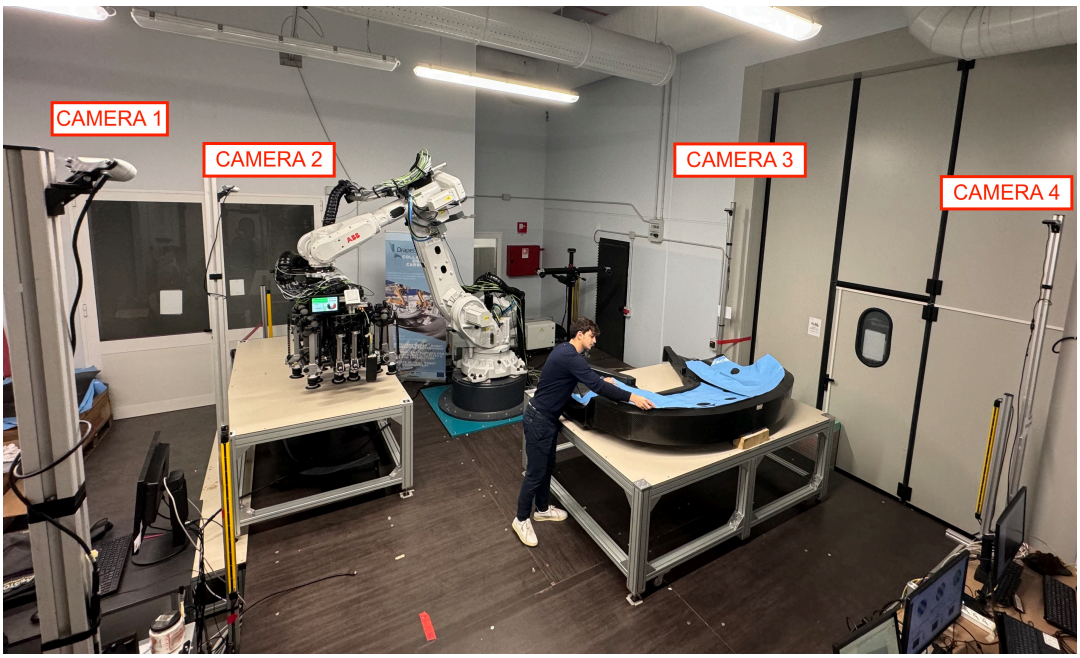


FIGURE 3.6: A large industrial robotic workcell equipped with a camera network around an ABB robot arm, enabling human-robot collaboration task in a carbon fiber draping process as foreseen in the DrapeBot European Project.

[1] https://sharework-project.eu/
[2] https://www.drapebot.eu/

Calibrating such a system presents significant challenges for this *eye-on-base* configuration: i) the calibration pattern must be compact to avoid collisions during the robot's movement; ii) the calibration often involves a large camera network, necessary to monitor the whole robotic workcell, dealing with rather large distances among cameras and robot; iii) the calibration process needed to be performed in short time to reduce as much as possible production line downtime; iv) accurate calibration must be provided even with a limited number of images, which is a common occurrence in industrial environments due to the difficulty of moving the robot arm safely in a cluttered space.

3.4.1 Multi-Camera Hand-Eye Calibration for Human-Robot Collaboration in Industrial Workcells

To address these challenges, this section introduces a methodology that generalizes the optimization framework (3.1) presented in Section 3.3, which is reported here for convenience:

$$\underset{X,Z}{\operatorname{argmin}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left\| p_{ij}^{D} - p_{ij}^{P} \right\|_2^2 . \tag{3.6}$$

We extend the formulation to solve for an entire multi-camera system with L sensors simultaneously, estimating the pose of each camera with respect to all other sensors and to the robot's base reference frame. Unlike conventional methods, which calibrate each camera independently, our approach prevents the accumulation of errors by introducing a joint optimization guided by two critical constraints: (i) a single, common board-to-end-effector transformation T_B^E for all cameras, and (ii) camera-to-camera relative pose constraints $T_{C_t}^{C_k}$.

The first constraint, T_B^E, is based on the fact that all cameras observe the same physical pattern rigidly attached to the robot end-effector. This eliminates the redundancy of estimating this transformation L times. Generalizing (3.5) to a network of L cameras and N robot poses, this first constraint is formulated as a global reprojection error term c_{rpj}:

$$c_{rpj} = \sum_{k=0}^{L-1} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left\| \hat{f}_k \left(T_W^{C_k} T_{E_i}^{W} T_B^E P_j^B \right) - p_{ijk}^{D} \right\|_2^2 , \tag{3.7}$$

where $\hat{f}_k$ is the projection function for camera k, and $T_W^{C_k}$ is the hand-eye transformation to be estimated for that camera.

The second constraint leverages the common case where cameras have overlapping fields of view (Figure 3.7). When multiple cameras (e.g., C_k and C_t) observe the calibration pattern simultaneously, we can define an additional kinematic chain $T_{C_t}^{C_k} T_W^{C_t}$. We can enforce consistency by minimizing the difference between the corners p_{ijk}^{D} detected by camera C_k and the corners reprojected onto C_k through camera C_t. This process is crucial for refining the relative transformations $T_{C_t}^{C_k}$ between the cameras, exploiting the occurrence of cross-detection (or cross-observations)—when the checkerboard is simultaneously detected by more cameras.

To formalize this, we define a cross-detection matrix $\mathbf{X}_i(k,t)$, as follow:

$$\mathbf{X}_i(k,t) = \begin{cases} 0, & \text{if } k = t \text{ or no cross-observation} \\ 1, & \text{otherwise} \end{cases} . \tag{3.8}$$

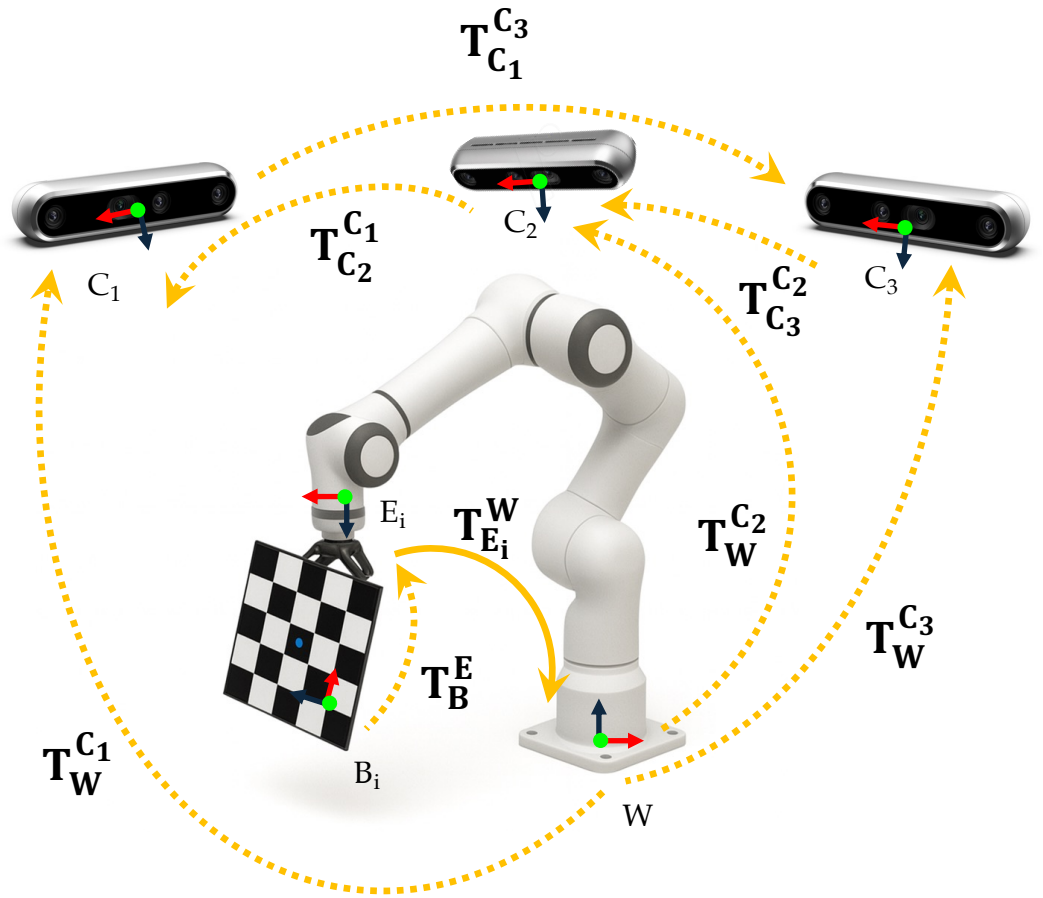


FIGURE 3.7: Multi-camera hand-eye setup, illustrating the geometric transformations optimized through the proposed multi-camera hand-eye calibration method. They include the single board-to-end-effector transformation common to all cameras, the spatial constraints among the cameras, and all the hand-eye transformations.

This matrix activates the constraint only when cross-detection occurs. This leads to the cross-camera cost function c_{cross}:

$$c_{cross} = \sum_{k=0}^{L-1} \sum_{t=0}^{L-1} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \mathbf{X}_i(k,t) \cdot \left\| p_{ijkt}^{cross} - p_{ijk}^{D} \right\|_2^2 \tag{3.9}$$

where

$$p_{ijkt}^{cross} = \hat{f}_k \left(T_{C_t}^{C_k} T_W^{C_t} T_{E_i}^{W} T_B^{E} P_j^{B} \right) . \tag{3.10}$$

Overall, the cost function to be optimized can be summarized as the sum of two main contributions, the term c_{rpj} aiming to minimize the re-projection error for each individual camera and the term c_{cross} to impose constraints on the relative pose of pairs of cameras:

$$\underset{T_{C_t}^{C_k}, T_W^{C_t}, T_B^{E}}{\operatorname{argmin}} \; c_{rpj} + c_{cross} \, . \tag{3.11}$$

Note how this formulation is inherently flexible. If no pair of cameras simultaneously detects the pattern, all entries in $\mathbf{X}_i$ are zero, and the c_{cross} term is naturally neglected. The optimization then gracefully degrades to solving c_{rpj} alone. To our knowledge, this is the first multi-camera hand-eye calibration method to leverage both constraints in a single non-linear

optimization.

The proposed method is implemented using the Ceres Solver and optimized via a Levenberg–Marquardt trust-region strategy. Automatic differentiation is employed to accurately and efficiently compute Jacobians for the reprojection residuals. Given the potentially large number of parameters and residuals involved in multi-camera calibration, the problem is solved using an iterative Schur-based linear solver, which is well suited for the block structure induced by camera and hand–eye parameters. To improve robustness against spurious detections, imperfect corner localization, and occasional tracking failures, a Cauchy robust loss function is applied to all residuals, effectively down-weighting outliers and enhancing convergence stability.

3.4.2 A Graph-Based Optimization Framework for Multi-Camera Hand-Eye Calibration

To further explore the multi-camera hand-eye calibration problem, we developed an algorithm grounded in graph optimization principles, extending the framework introduced by Koide [97]. While Koide's original method was designed for a single-camera, eye-in-hand configuration, our adaptation generalizes this approach to handle the complex interdependencies of a multi-camera, eye-on-base setup. The resulting formulation provides a unified and structured representation of all camera-robot relationships, enabling joint optimization across the entire system. Therefore, we construct a single hyper-graph encompassing all cameras and measurements, illustrated conceptually in Figure 3.8.

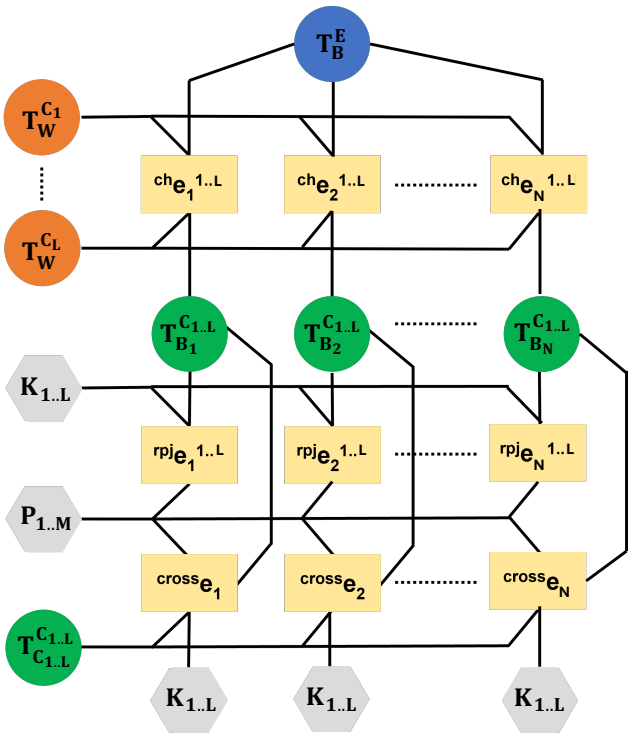


FIGURE 3.8: Graph representation for the multi-camera eye-on-base calibration. **Vertices** (circles) represent the unknown poses to be estimated: the world-to-camera poses $\{T_W^{C_k}\}$, the shared board-to-end-effector pose T_B^E, relative camera poses $\{T_{C_t}^{C_k}\}$ and the board-to-camera poses $T_{B_i}^{C_k}$. **Edges** (rectangles) represent error terms. Fixed parameters (3D board points $\mathbf{P}^B$ and intrinsic parameters K_L) are shown as **hexagons**.

The circular vertices in this graph represent the parameters to be estimated. The set of L world-to-camera transformation: $\{T_W^{C_k}\}_{k=1}^{L}$, the single, shared board-to-end-effector transformation: T_B^E, the set of relative transformations between camera pairs $\{T_{C_t}^{C_k}\}$ and the N board poses $\{T_{B_i}^{C}\}_{i=1}^{N}$.

The calibration is achieved by minimizing a global cost function derived from the edges (rectangles in the graph) connecting these vertices. The edges represent error terms based on measurements (detected 2D corners p_{ijk}^D and known robot poses $T_{E_i}^W$) and enforce geometric consistency according to three principles: kinematic chain, reprojection error and cross-observation error.

First, the **kinematic chain error** (${}^{ch}e$) explicitly enforces the closure of the kinematic loop for each camera k at each pose i. It ensures consistency between the pose of the board B with respect to the robot base W, computed via two paths: the robot path ($T_{E_i}^W \cdot T_B^E$) and the camera path ($T_{C_k}^W \cdot T_{B_i}^{C_k}$). The error ${}^{ch}\mathbf{e}_i^k$ represents the difference between these two paths in the $SE(3)$ manifold space. Stacked over all cameras at pose i, this results in the error vector ${}^{ch}\mathbf{e}_i^{1..L}$ defined in (3.12).

$$ {}^{ch}\mathbf{e}_i^{1..L} = \left[{}^{ch}\mathbf{e}_i^1 \ldots {}^{ch}\mathbf{e}_i^k \ldots {}^{ch}\mathbf{e}_i^L\right]^T, \tag{3.12} $$

Second, the **reprojection error** (${}^{rpj}e$) serves as the core measurement constraint. For each camera k and robot pose i, this term minimizes the pixel distance between the detected 2D corners p_{ijk}^D and the reprojection of the corresponding 3D board corners P_j^B. The reprojection uses the transformation chain from the board frame B to the camera frame C_k, defining the individual error term ${}^{rpj}e_{ijk}$ for the j-th corner as follow:

$$ {}^{rpj}\mathbf{e}_{ij}^k = \hat{f}_k\left(T_W^{C_k} T_{E_i}^W T_B^E P_j^B\right) - p_{ijk}^D, \tag{3.13} $$

where $\hat{f}_k$ is the projection function for camera k. Stacked over all corners j and cameras at pose i, this gives the error vector ${}^{rpj}\mathbf{e}_i^{1..L}$ defined in (3.14).

$$ {}^{rpj}\mathbf{e}_i^{1..L} = \left[{}^{rpj}\mathbf{e}_i^1 \ldots {}^{rpj}\mathbf{e}_i^a \ldots {}^{rpj}\mathbf{e}_i^L\right]^T. \tag{3.14} $$

Third, the **cross-observation error** (${}^{cross}e$) is crucial for leveraging multi-camera information and ensuring global consistency. It activates only when two cameras, C_k and C_t, simultaneously observe the calibration pattern at the same robot pose i. To encode the co-visibility at pose i, we used the cross-detection matrix $\mathbf{X}_i$ defined in (3.8). When $\mathbf{X}_i(k,t) = 1$, this edge minimizes the reprojection error in camera C_k, projecting the 3D point P_j^B through the camera C_t using the estimated relative transformation $T_{C_t}^{C_k}$.

The transformation chain from board B to camera C_k via C_t is defined as $T_B^{C_k} = T_{C_t}^{C_k} \cdot T_W^{C_t} \cdot T_{E_i}^W \cdot T_B^E$. The individual cross-observation error term ${}^{cross}\mathbf{e}_{ijkt}$ is therefore:

$$ {}^{cross}\mathbf{e}_{ijkt} = \hat{f}_k\left(T_{C_t}^{C_k} \cdot T_W^{C_t} \cdot T_{E_i}^W \cdot T_B^E \cdot P_j^B\right) - p_{ijk}^D. \tag{3.15} $$

Adding the relative transformations $T_{C_t}^{C_k}$ as vertices to be estimated allows this edge to explicitly optimize the geometric coherence of the camera network. The total cross-observation error at pose i sums the squared norms of these individual errors, weighted by the co-visibility matrix:

$$ {}^{cross}\mathbf{e}_i = \sum_{k=1}^{L}\sum_{t=1}^{L} \mathbf{X}_i(k,t) \cdot \sum_{j=0}^{M-1} \|{}^{cross}\mathbf{e}_{ijkt}\|_2^2, \quad \text{with } k \neq t \quad . \tag{3.16} $$

The final objective function combines these three error components, weighted by factors λ_1 and λ_2 to balance their influence:

$$\min \sum_{i=1}^{N} \left(\lambda_1 \| {}^{rpj}\mathbf{e}_i \|_2^2 + \| {}^{ch}\mathbf{e}_i \|_2^2 + \lambda_2 \cdot {}^{cross}\mathbf{e}_i \right) . \tag{3.17}$$

This optimization problem is efficiently solved using standard graph optimization libraries like g2o [109]. The weighting factors $\lambda_1 = 10^{-6}$ and $\lambda_2 = 10^{-3}$ were determined experimentally to achieve robust convergence.

3.4.3 The METRIC Dataset for Multi-Camera Calibration Evaluation

A significant challenge in rigorously evaluating the multi-camera calibration methods presented in Section 3.4.1 and Section 3.4.2 is the lack of suitable public benchmarks. While some datasets exist, they often suffer from critical limitations for assessing performance in realistic, large-scale industrial scenarios. Many rely on manual pattern movement, which risks introducing motion blur and inconsistent data acquisition, thereby compromising calibration accuracy assessment [110, 98]. Furthermore, existing datasets are typically captured in small robotic workcells with sensors positioned relatively close to the robot [99]. This fails to represent the challenges encountered in large HRC environments, such as increased sensor-robot distances and the need for compact calibration patterns, which significantly impact pattern detection and overall calibration robustness.

To address this critical gap and facilitate rigorous, comparative evaluation under relevant industrial conditions, we developed and publicly released the METRIC[3] (Multi-Eye To Robot Indoor Calibration) dataset [45]. METRIC is specifically designed to benchmark multi-camera eye-on-base calibration algorithms under varying sensor-robot distances, workcell sizes, and using a compact calibration pattern moved by the robot itself, ensuring consistent and sharp image acquisition. The dataset comprises two complementary components: a synthetic dataset for controlled robustness analysis and a real-world dataset for practical performance evaluation.

Synthetic Dataset

The synthetic data was generated using the Gazebo simulation environment to provide perfectly known ground truth (GT) transformations and allow for controlled experiments assessing robustness. We simulated a robotic workcell featuring a Panda robot arm and a network of Microsoft Kinect V2 cameras (Figure 3.9 left). The intrinsic parameters were predefined, and the environment used diffuse illumination to prevent potential shadows. Crucially, we generated data sequences for three distinct workcell sizes (small, medium, large) by systematically varying the camera height (h), sensor-robot distance (d), and workcell side lengths (l_1, l_2), as illustrated in Figure 3.9 right and detailed in Table 3.2.

TABLE 3.2: Main dimensions for the three workcells considered in the synthetic dataset: small, medium and large.

Workcell sizes	**Colour**	h (m)	d (m)	l_1 (m)	l_2 (m)
Small workcell		1.00	2.05	2.00	3.00
Medium workcell		1.00	2.70	3.00	4.00
Large workcell		1.50	3.50	4.00	5.00

[3] https://zenodo.org/records/7976757, accessed on 27 October 2025

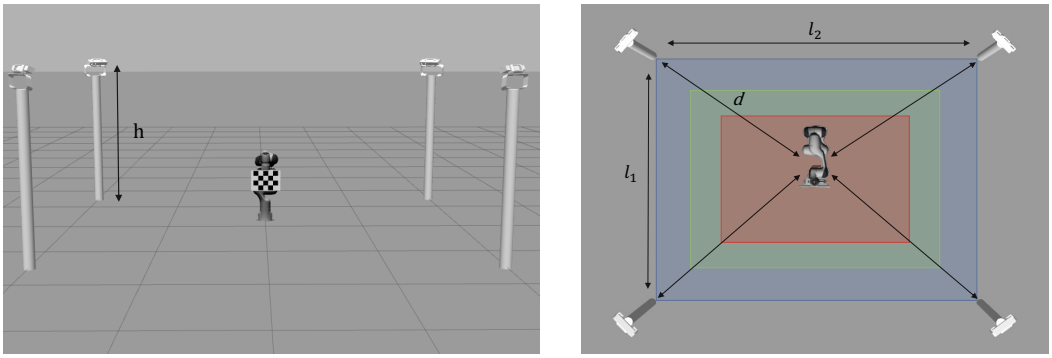


FIGURE 3.9: Simulated METRIC robotic workcell from a side view on the left; robotic workcell from a top view on the right, showing the different crucial dimensions.

This setup allows us to quantify the degradation in algorithm performance as the camera-to-board distance increases. For each acquisition sequence, the dataset provides: the RGB image sequences from each camera and the corresponding robot end-effector poses ($T_{E_i}^{W}$), required by the calibration algorithms, and the perfectly known camera poses ($T_{W}^{C_k}$), relative camera poses ($T_{C_t}^{C_k}$), and the board-to-end-effector pose (T_{B}^{E}), obtained directly from the simulator for reliable accuracy assessment.

Real-World Dataset

The real-world dataset component captures the complexities inherent in physical environments, including variable lighting, surface reflections, and sensor-specific distortions. Data was collected at the Intelligent Autonomous Systems Laboratory (IAS-Lab), University of Padova, using an experimental setup designed to mirror the target industrial scenario (Figure 3.10). This setup featured a Panda robot arm surrounded by a network of four Intel RealSense D455 cameras.

Data was acquired in two workcell configurations (small and large), analogous to the simulation, allowing evaluation under different geometric scales. Table 3.3 details the specific dimensions used for these configurations. This also facilitates studying the impact of real-

TABLE 3.3: Dimensions for the two workcells considered in the real-world dataset: small and large.

Workcell sizes	h (m)	d (m)	l_1 (m)	l_2 (m)
Small workcell	1.25	2.30	2.25	3.25
Large workcell	1.25	2.90	3.00	5.00

world sensor characteristics like resolution, FoV, and distortion on calibration performance, with the relevant sensor specifications provided in Table 3.4.

TABLE 3.4: Sensor specifications.

	Kinect V2	**Depth camera D455**	**LiDAR camera L515**
RGB resolution	1920×1080	1280×800	1920×1080
RGB FoV (**h** $\times$ **v**)	$84.1^\circ \times 53.8^\circ$	$90^\circ \times 65^\circ$	$70^\circ \times 43^\circ$

For acquiring accurate GT in the real world, we employed a fiducial marker-based approach. An AprilTag (15 cm x 15 cm) was rigidly mounted to the robot's end-effector using



FIGURE 3.10: Large-scale real workcell equipped with a Panda robot arm with a checkerboard attached on its end-effector, surrounded by four Intel RealSense Depth D455 cameras.

a precisely manufactured 3D-printed fixture. The pose of the AprilTag relative to the robot base (T^{W}_{Tag}) was computed using the robot's forward kinematics and the known CAD model of the fixture. The pose of the AprilTag relative to each camera ($T^{C_k}_{Tag}$) was estimated using the standard AprilTag detection library [4], which provides reliable results for distances up to 5 meters. Combining these yields the ground truth robot base-camera pose: $T^{C_k}_{W} = T^{C_k}_{Tag} \cdot T^{Tag}_{W}$. The relative poses $T^{C_k}_{C_t}$ were derived accordingly. Similar to the synthetic component, each real acquisition sequence includes the input data, RGB image sequences from all cameras and their synchronized robot end-effector poses ($T^{W}_{E_i}$), alongside the corresponding ground truth obtained via the AprilTag method, providing both absolute camera poses ($T^{C_k}_{W}$) and relative inter-camera poses ($T^{C_k}_{C_t}$) for robust evaluation.

Performance Evaluation on Synthetic METRIC Data

We utilized the synthetic split of the METRIC dataset to rigorously evaluate the performance of our proposed multi-camera calibration methods (Section 3.4.1 and Section 3.4.2, against established single-camera and one multi-camera baseline. Single-camera methods were applied independently to each camera's data stream. Performance was assessed using both ground truth errors (e^{GT}_{t}, e^{GT}_{θ}) and kinematic chain residual errors (e^{res}_{t}, e^{res}_{θ}) across the three simulated workcell sizes (small, medium, large).

The comprehensive results are presented in Table 3.5. A clear trend emerges from the results: the size of the workcell strongly affects calibration accuracy. As the distance between the cameras and the calibration pattern increases, corner detection becomes less reliable, leading to a consistent degradation in both translation and rotation accuracy across all

[4] https://github.com/AprilRobotics/apriltag, accessed on 27 October 2025

TABLE 3.5: Average errors for all cameras achieved by hand-eye calibration techniques in the METRIC simulated workcells. The upper block reports single-camera methods, while the lower block lists multi-camera approaches. Bold and underlined values indicate the 1^{st} and 2^{nd} top-performing calibration methods for each metric, respectively.

Method	Small workcell				Medium workcell				Large workcell			
	Ground truth		AX=ZB		Ground truth		AX=ZB		Ground truth		AX=ZB	
	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t^{res} [mm]	e_θ^{res} [deg]	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t^{res} [mm]	e_θ^{res} [deg]	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t^{res} [mm]	e_θ^{res} [deg]
Ours 3.3	1.93	0.03	1.98	0.05	5.46	0.07	6.43	0.08	19.98	0.29	35.21	0.53
Tsai [61]	427.16	2.06	40.26	0.62	820.90	6.72	47.29	1.12	587.59	0.21	35.47	0.56
Park [91]	3.27	0.04	2.35	0.12	5.27	0.03	4.47	0.18	12.06	0.20	4.97	0.18
Danililidis [92]	42.43	0.58	15.34	0.37	100.09	3.27	39.24	1.08	57.58	0.17	24.9	0.70
Andreff [111]	10.05	0.05	8.72	0.33	91.83	0.38	80.84	1.30	149.53	0.34	116.54	2.89
Shah [95]	1.90	0.05	2.63	0.13	3.03	0.05	4.37	0.19	5.35	0.08	4.51	0.16
Li [94]	1.93	0.05	2.71	0.14	3.15	0.05	4.55	0.19	5.87	0.08	4.58	0.16
Koide [97]	1.68	0.03	1.46	0.07	3.94	**0.02**	2.31	0.11	1236.30	0.04	773.11	5.38
Tabb [99]	2.52	0.20	2.86	0.32	5.79	0.36	7.35	0.38	13.52	0.89	17.45	0.65
Ours 3.4.1	**0.71**	**0.02**	**0.45**	**0.03**	**0.75**	**0.02**	**0.83**	**0.05**	**1.08**	**0.01**	**2.02**	**0.09**
Ours 3.4.2	1.02	**0.02**	1.14	0.05	3.38	**0.02**	2.02	0.10	1.43	0.02	4.02	0.15

evaluated metrics.

This degradation is particularly evident in single-camera methods applied independently. Well-established approaches such as Park [91], Daniilidis [92], and even our own unified single-camera method [51] exhibit a noticeable decline in accuracy as the workspace enlarges. The effect becomes extreme for Tsai [61], Andreff [111], and Koide [97], which produce dramatically higher errors, sometimes diverging completely, in the large-workcell scenario. These results clearly highlight the failure of independently applied single-camera approaches to maintain robustness under large-scale, real-world conditions, emphasizing the need for a joint optimization framework that enforces global consistency across multiple sensors.

In contrast, both proposed multi-camera methods demonstrate remarkable robustness and consistently superior accuracy across all workcell sizes. They exhibit minimal sensitivity to the increasing sensor-robot distance. Notably, even in the most challenging large workcell, both methods maintain low residual errors ($e_t^{res}, e_\theta^{res}$) and achieve exceptional accuracy relative to ground truth: translation errors e_t^{GT} remain around 1-2 mm, and rotation errors e_θ^{GT} are as low as 0.01-0.02 degrees. This highlights the critical advantage of jointly optimizing the entire camera network and leveraging cross-camera constraints (c_{cross} or e^{cross} terms), which effectively mitigates the impact of noise and detection difficulties inherent in large-scale configurations.

To further assess robustness, we analyzed performance of our direct joint optimization method (Section 3.4.1) while varying the number of calibration images per camera (from 2 to 50, randomly selected). As shown in Figure 3.11, the proposed multi-camera methods achieve near-optimal performance even with a limited number of images. Accuracy stabilizes rapidly, with results obtained using just 5 images per camera being comparable to those using 50 images, where performance saturates. This is a crucial finding for industrial applications where minimizing data acquisition time (and thus robot downtime) is crucial.

Performance Evaluation on Real-World METRIC Data

The same methods were tested also in the real-world data of METRIC in order to assess their applicability to real-world industrial settings. As illustrated in Table 3.6, the two proposed multi-camera hand-eye calibration methods consistently outperform other methods and it ensures convergence to an optimal solution, not achievable by some single-camera methods [92], [95], [94]. This can primarily be attributed to the incorporation of two additional constraints, which enhance the algorithm's robustness in real-world scenarios where corner detection may be imprecise. In particular, our method of Section 3.4.1 achieves a translation error e_t^{GT} lower than 52 mm and a rotation error e_θ^{GT} lower than 0.3 deg, demonstrating

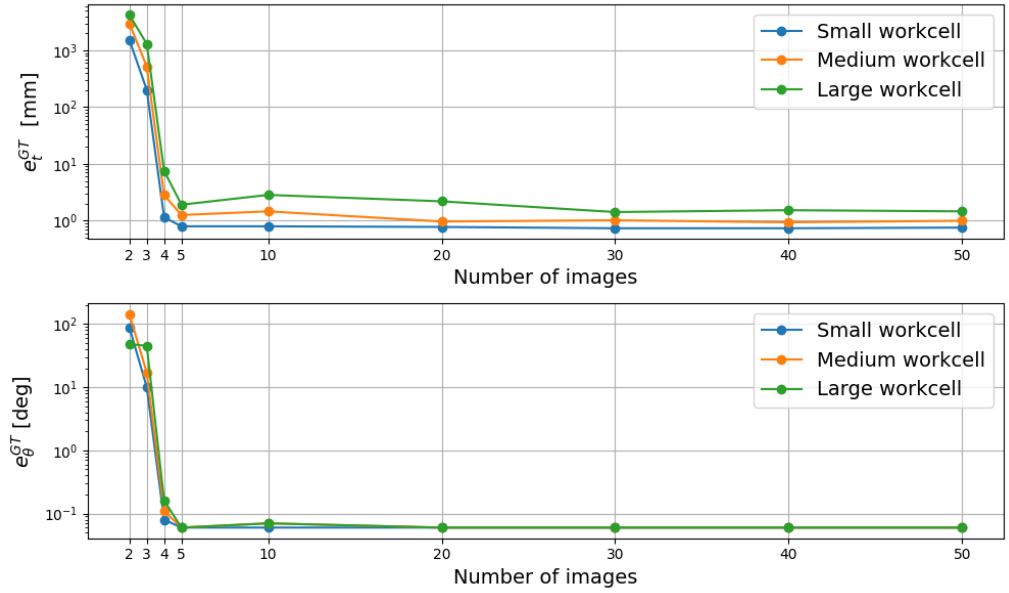


FIGURE 3.11: Analysis of translation error e_t^{GT} and rotation error e_θ^{GT} for the METRIC simulated workcells as the number of images per camera used in the calibration process varies. Errors are expressed in a logarithmic scale.

effectiveness even within the larger workcell. Notably, even with the lower-resolution Intel RealSense D455 sensor, the proposed method achieves an average error comparable to that obtained with the other sensors, guaranteeing superior performance with respect to all other state-of-the-art methods. In general, multi-camera calibration methods demonstrate greater accuracy compared to running N times the single-camera methods, which do not have the possibility of mitigating the impact of inaccurate pattern detection through other points of view that are simultaneously considered.

The availability of ground truth on real data in METRIC makes it possible to verify the reliability of the kinematic chain residual error metric (Section 2.3.2) in a real-world scenario, as shown in Table 3.6 where they follows the same trend of GT error based metrics (Section 2.3.2) for the methods tested.

3.4.4 Results on Real-World Industrial Scenarios

To comprehensively evaluate the robustness of our calibration method even in real industrial environments, we performed calibration experiments in two industrial robotic workcells developed as use cases in the DrapeBot project [112]. Both workcells have been designed for human-robot collaboration and collaborative transportation of flexible materials, covering an area of about 20 m^2. The former, illustrated in Figure 3.12 left and identified as the ABB robotic workcell, was equipped with four Intel RealSense Depth D455 sensors surrounding an ABB industrial manipulator at an average distance of 5 meters from the base of the robot. The second workcell, designated as the Kuka industrial workcell, was equipped with a Kuka industrial manipulator surrounded by three RealSense Depth D455 sensors at an average distance of 6 meters from the robot base (Figure 3.12 right).

Within these two robotic workcells, the image collection comprised 8 images per camera for the first workcell and 15 images per camera for the second one. This limited number was attributed to the challenges associated with moving the robot to various locations and orientations in front of the sensors, due to the presence of large objects, such as the mold depicted in Figure 3.12 near the robot, which hinders its movement. The calibration pattern

TABLE 3.6: Average errors achieved by hand-eye calibration techniques on the METRIC real workcells. The upper block reports results from single-camera methods, while the lower block corresponds to multi-camera approaches. Bold and underlined values indicate the 1^{st} and 2^{nd} top-performing calibration methods, respectively ("—" denotes non-convergence).

Method	**Small real workcell**											
	Microsoft Kinect V2				**Intel RealSense Depth D455**				**Intel RealSense LiDAR L515**			
	Ground truth		AX=ZB		Ground truth		AX=ZB		Ground truth		AX=ZB	
	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]
Ours 3.3	42.79	0.57	13.21	0.81	45.14	0.43	72.91	1.22	26.20	0.39	50.33	3.58
Tsai [61]	75.13	0.15	12.11	0.90	1215.45	14.32	224.68	10.65	396.83	0.59	53.51	**0.86**
Park [91]	56.61	0.14	10.42	0.86	167.70	3.48	69.20	2.13	72.40	0.12	40.32	1.79
Daniilidis [92]	49.62	0.14	10.32	0.94	1425.42	0.43	541.38	1.69	316.34	0.30	71.11	1.77
Andreff [111]	235.17	0.58	127.61	5.51	1101.90	11.95	759.66	11.76	596.57	6.78	443.67	4.75
Shah [95]	27.22	0.75	10.37	0.94	23.90	0.54	65.90	2.31	18.51	0.34	30.93	1.68
Li [94]	66.67	0.73	13.89	0.93	51.03	0.72	100.29	2.40	23.70	0.31	34.27	1.64
Koide [97]	46.52	0.12	12.84	0.68	72.87	0.28	22.30	2.91	36.67	0.78	82.95	2.96
Tabb [99]	51.57	0.73	11.20	0.81	63.98	1.36	81.32	2.76	34.59	0.94	67.23	3.56
Ours 3.4.1	**22.01**	**0.09**	**6.54**	**0.55**	**13.21**	**0.07**	**9.21**	**0.66**	**13.68**	**0.02**	**24.95**	1.32
Ours 3.4.2	42.42	**0.09**	7.31	0.56	41.80	0.34	11.10	1.13	24.11	0.44	33.75	1.70

Method	**Large real workcell**											
	Microsoft Kinect V2				**Intel RealSense Depth D455**				**Intel RealSense LiDAR L515**			
	Ground truth		AX=ZB		Ground truth		AX=ZB		Ground truth		AX=ZB	
	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]	e_t^{GT} [mm]	e_θ^{GT} [deg]	e_t [mm]	e_θ [deg]
Ours 3.3	77.26	0.77	123.12	3.21	136.39	1.33	57.32	4.32	60.59	0.43	54.36	1.66
Tsai [61]	1924.88	18.78	318.14	15.30	2759.10	14.91	461.62	15.25	1881.75	11.53	222.23	7.50
Park [91]	336.86	1.08	178.91	4.69	355.10	5.33	226.85	5.10	233.57	3.98	70.63	1.60
Daniilidis [92]	1757.97	13.23	629.15	6.07	–	–	–	–	8359.06	0.32	4176.45	1.60
Andreff [111]	2370.61	12.34	1639.04	12.20	2554.29	35.89	1680.34	7.25	1881.23	12.79	1310.25	10.94
Shah [95]	54.92	0.73	75.34	2.62	–	–	–	–	26.15	0.31	38.12	2.02
Li [94]	129.39	0.72	111.44	2.61	–	–	–	–	26.39	0.30	66.42	2.49
Koide [97]	69.38	0.35	20.72	1.04	65.39	0.32	45.21	2.13	59.58	0.11	15.05	1.05
Tabb [99]	105.01	0.75	154.21	5.32	153.20	1.74	167.34	6.35	75.33	0.77	45.23	1.21
Ours 3.4.1	**51.18**	0.30	**12.17**	**0.65**	**45.18**	**0.16**	**35.98**	2.12	**19.34**	**0.09**	12.45	**0.85**
Ours 3.4.2	63.54	**0.24**	15.79	0.71	59.33	0.22	48.50	**2.10**	50.18	**0.09**	**11.63**	0.89

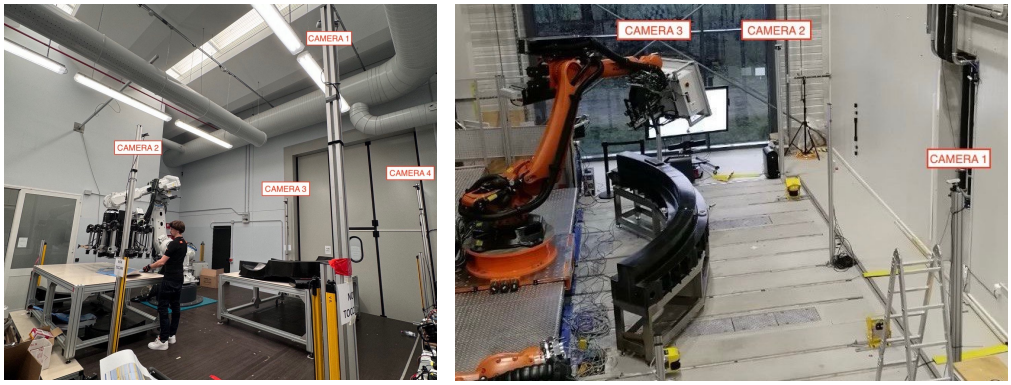


FIGURE 3.12: On the left, the ABB industrial workcell equipped with four Intel RealSense Depth D455 cameras positioned around the manipulator at an average distance of 5 m. On the right, the KUKA industrial workcell equipped with three Intel RealSense Depth D455 cameras surrounding the manipulator at an average distance of 6 m.

was composed of a checkerboard with 4×3 inner corners, spaced approximately 6 cm apart. The results of the calibration process are reported in Table 3.7. These results are assessed using the metrics defined in kinematic chain residual metric, due to the absence of ground truth data.

Table 3.7 highlights that our proposed multi-camera hand-eye calibration method achieves high accuracy with a translation error of approximately 1 cm and a rotation error lower than 1 deg. In contrast, some single-camera calibration methods (e.g., ours of Section 3.3 and the works in [94] and [97]) struggle to converge to an optimal solution, proving to be unsuitable for such challenging scenarios. Notably, our method shows exceptional accuracy with a limited number of images (8 per camera for the ABB workcell and 15 for the Kuka workcell). This efficiency is significant, especially in the context of industrial robotic workcells, where

TABLE 3.7: Average error on industrial workcells.

Method	ABB industrial workcell (8 images per camera)		Kuka industrial workcell (15 images per camera)	
	AX=ZB		AX=ZB	
	e_t [mm]	e_θ [deg]	e_t [mm]	e_θ [deg]
Ours 3.3	718.94	7.52	485.92	6.15
Tsai [61]	172.07	10.15	30.28	1.27
Park [91]	83.54	11.93	27.82	2.21
Daniilidis [92]	77.09	11.14	36.33	1.42
Andreff [111]	1378.25	6.06	71.37	1.66
Shah [95]	106.99	10.76	72.94	1.52
Li [94]	–	–	652.44	1.23
Koide [97]	131.08	9.60	1737.68	10.23
Tabb [99]	53.76	3.18	22.65	1.23
Ours 3.4.1	**6.31**	**0.98**	**14.12**	**0.67**
Ours 3.4.2	11.95	2.65	17.38	0.78

acquiring images is challenging due to obstacles that restrict flexible robot movement. Consequently, this approach significantly reduces calibration times and minimizes interruptions in production lines.

3.5 MEMROC: Multi-Eye to Mobile Robot Calibration

Shifting focus from manipulator arms to mobile robots, robot-centric perception remains critical for navigation [113, 114]. As discussed in Section 3.1, motion-based calibration offers a practical solution by leveraging the robot's egomotion, typically solving the $AX = XB$ formulation where A relates camera motion and B relates robot base motion (derived from odometry). However, the core challenge, also highlighted previously, is the unobservability of the full 6-DoF camera pose due to the predominantly planar motion of ground robots. While existing approaches attempt to overcome this using ground plane detection or learning-based methods, they often require large amounts of data, exhibit sensitivity to odometry noise, or rely on specific environmental features, limiting their robustness and ease of use. To address these specific limitations, this section introduces MEMROC (Multi-Eye to Mobile RObot Calibration) [6].

MEMROC is a novel motion-based method designed to efficiently and robustly determine the full 6-DoF extrinsic parameters for single or multiple cameras relative to a mobile robot's base frame. It incorporates a robust mechanism for ground plane estimation that overcomes the planar motion degeneracy using minimal data (fewer than 20 viewpoints including ground plane observations) and exhibits robustness to noisy odometry measurements, making frequent recalibration practical in dynamic environments. An overview of the proposed framework is illustrated in Figure 3.13.

3.5.1 MEMROC Method

MEMROC follows a motion-based approach where calibration is obtained by minimizing the general $\|AX - XB\|_2^2$ cost function, introducing two main elements: a known calibration pattern fixed in the scene and a ground plane estimation module. The use of a known pattern (e.g. checkerboard) allows to overcome the inherent scale ambiguity in motion-based

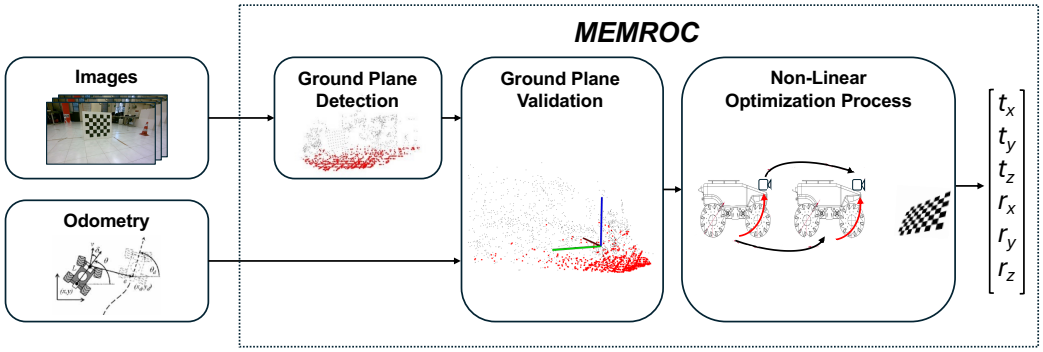


FIGURE 3.13: MEMROC Overview. The system is structured into three primary components: the Ground Plane Detection module, responsible for identifying the ground plane; the Ground Plane Validation module, which ensures the ground plane's alignment with the mobile robot; and the Motion-Based Calibration Process, the concluding stage that calibrates the system.

approaches that uses only RGB data, leading to several advantages: (i) more accurate camera motion estimation and (ii) a lower number of images needed in the optimization process compared to methods relying solely on visual features. Ground plane estimation, on the other hand, allows to address z-coordinate unobservability and achieve full 6-DoF calibration. It provides a set of measurements of the camera's height relative to the ground which are then used as constraints in the optimization process.

A schematic overview of MEMROC is shown in Figure 3.13, representing its main modules. First, a set of pairs of robot poses and images is collected by moving the robot in front of a calibration pattern as depicted in Figure 3.14; robot poses A_i are obtained from odometry, while images allow to estimate camera poses B_i with respect to the calibration pattern by means of PnP (Perspective-n-Point) algorithm. While the accuracy of PnP was noted as a potential concern in previous sections, particularly with the small patterns or large distances common in manipulator setups, its application here benefits from using a sufficiently large and clearly detectable checkerboard pattern. This makes the estimation of the camera's pose (B_i) relative to this stable reference via PnP a well-conditioned problem, yielding accurate estimates of the camera's motion.

Images are also used to compute 3D correspondences based on feature matching for ground plane detection and validation based on RANSAC and Singular Value Decomposition. Finally, robot and camera poses are used in the motion-based optimization process with ground plane measurements as constraint.

Ground Plane Detection

The general idea is to estimate the ground plane equation from camera measurements, in order to compute its height from the ground and use such information in the optimization process. In particular, assuming a 3D representation of the scene in the camera frame is available (e.g., pointcloud), we identify the largest plane in such representation using the RANSAC (RANdom SAmple Consensus) method [115].

When considering RGB-D cameras, the 3D representation of the scene can be easily obtained from the pointcloud provided by the camera. However, the quality of the pointcloud can be very inaccurate in outdoor scenarios. Therefore, we consider an alternative approach to obtain 3D points of the scene based on feature matching between consecutive frames.

In particular, given two consecutive images I_i and I_{i+1}, for each one of them a set of keypoints and descriptors is extracted (e.g., SIFT or SuperPoint [84]).

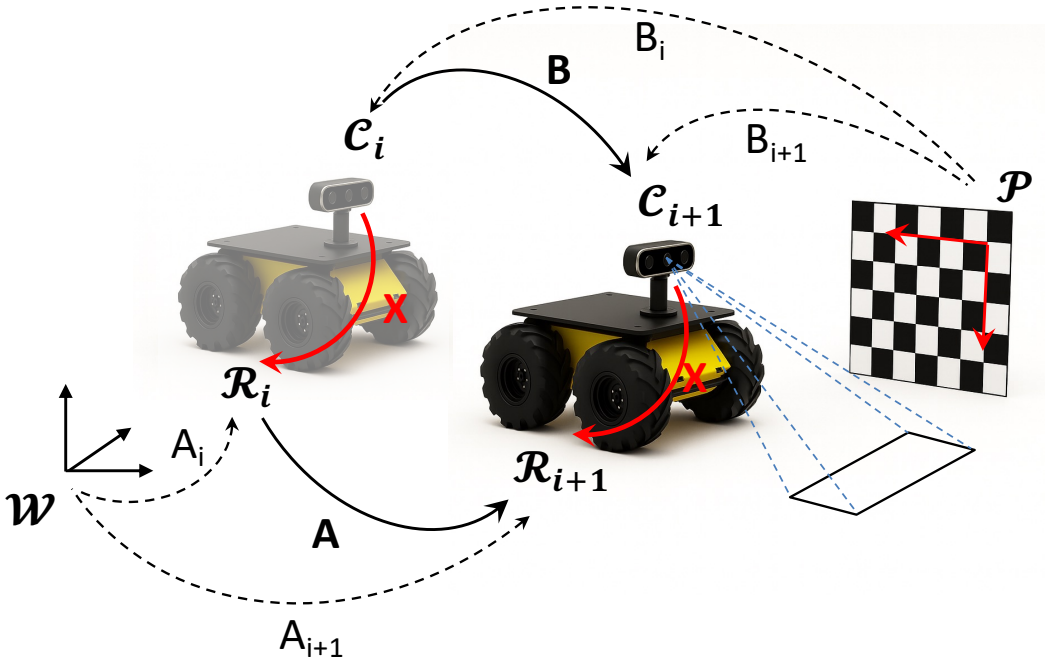


FIGURE 3.14: Formulation of the motion-based calibration method. A mobile robot R equipped with a camera C moves from pose i to $i+1$ towards a calibration pattern P in the scene. A_i represents the odometry of the mobile robot with respect to the fixed frame W (i.e., the robot starting position), while B_i represents the camera pose with respect to the calibration pattern. X denotes the camera to robot transformation to be estimated.

Let $\mathcal{M} = \{(p_{1,i}, p_{1,i+1}), \ldots, (p_{M,i}, p_{M,i+1})\}$ the set of matches between the two images computed by a feature matcher (e.g., LightGlue [116]). Based on the matches $\mathcal{M}$ and the absolute camera poses B_i, B_{i+1} provided by the PnP algorithm, it is possible to compute the 3D coordinates of the matches by means of stereo triangulation using direct linear transform (DLT) [117]. Note that by using the absolute camera poses, the final result is a set of 3D points expressed with respect to the pattern reference frame $\mathcal{P}$, as depicted in Figure 3.13. Such points constitutes the 3D scene representation used for estimating the ground plane with RANSAC (red points in Figure 3.13), which provides the equation of the largest plane in the scene expressed in the general equation form

$$ax + by + cz + d = 0 \tag{3.18}$$

where a, b and c are the components of the normal vector $\vec{n} = (a, b, c)$ perpendicular to the plane and d is the minimum distance from the plane, which also represent the height of the camera from the ground plane to be used in the constrained calibration process.

Ground Plane Validation

The ground plane detection algorithm described above aims to compute the plane equation of the largest plane in the reconstructed scene. However, such plane could it be different from the ground plane, especially in crowded indoor scenarios (e.g., vertical walls, large flat objects) so a validation procedure to filter out wrong detections is needed. When the robot moves, the rigidly attached camera follows the same trajectory, expressed in a different reference frame (i.e., world and pattern respectively). Let $a = \{a_1, a_2, \ldots, a_n\}$ and $b = \{b_1, b_2, \ldots, b_n\}$ two sets of corresponding 3D points, obtained from the translation part of matrices A_i and B_i respectively. By means of Singular Value Decomposition (SVD) [118], it

is possible to find a rigid transformation that optimally aligns the two sets in the least squares sense:

$$(R,t) = \underset{R\in SO(3),t\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{i=1}^{n} \|(Rb_i + t) - a_i\|^2 \tag{3.19}$$

The obtained rigid transformation describes the roto-translation (R_P^W, t_P^W) between the world and the pattern reference frame. Although the estimated translation is affected by a bias term (it assumes that the 3D points are coincident, while a and b are only parallel planes which differ for an offset term), the estimated rotation can be used to transform the 3D points of the scene obtained Section 3.5.1 in the world reference frame, $p^W = R_P^W\, p^P$. By introducing such operation before the RANSAC-based plane detection, the estimated plane equation is then expressed in world coordinates and can be easily verify the correct alignment of the normal of the plane with the z-axis of the world reference frame. If the estimated plane is not parallel to the ground, it is rejected.

Camera-to-Robot Calibration

As illustrated in Figure 3.14, for each robot movement, we collect transformations A_i and B_i representing the robot's absolute pose at time step i relative to its initial position (i.e., world frame W), and the pose of the camera with respect to the fixed calibration pattern reference frame.

However, robot poses obtained by odometry are generally noisy and affected by drift errors. This makes such data unsuitable for the calibration process, making it preferable to consider relative movements of the robot to limit the effect of drift in odometry. Therefore, in MEMROC, we aim to minimizes the cost function defined as

$$c = \sum_{i=0}^{N-1} \|AX - XB\|_2^2, \tag{3.20}$$

where A represents the robot's incremental transformation within a 2D plane, B denotes the camera's incremental motion, and X is the camera to robot transformation matrix we aim to determine through the calibration process. Given N samples of incremental motion, we define our calibration methodology through the objective of minimizing a specifically formulated cost function, detailed in equation (3.22).

$$c_{Zi} = \chi_i \left\| [T_C^R]_z - \overline{z_i} \right\|_2^2 \tag{3.21}$$

$$c = \sum_{i=0}^{N-1} \left(\left\| T_{R_{i+1}}^{R_i} T_C^R - T_C^R T_{C_{i+1}}^{C_i} \right\|_2^2 + c_{Zi} \right) \tag{3.22}$$

In this formulation, $T_{R_{i+1}}^{R_i}$ and $T_{C_{i+1}}^{C_i}$ symbolize the incremental motions of the robot and the camera, respectively, capturing their transition from one state to the next. Our objective is to accurately determine the transformation T_C^R, representing the rototranslation between the camera and the robot. An additional term c_{Zi} in the cost function aims to reduce the difference between the z-axis coordinate of the desired transformation and the camera's elevation, $\overline{z_i}$, as identified through ground plane detection (i.e., $\overline{z_i} = d$). The variable χ_i, which can either be 0 or 1, signifies whether the i^{th} pose yields a valid measure of the camera's height above the ground plane. This approach allows for the precise estimation of the camera's 6-DoF pose relative to the robot, moving beyond simple ground plane detection to integrate a comprehensive set of spatial data into the calibration effort. Remarkably, a single z_i measurement

suffices for this calibration. Moreover, this optimization process calibrates each camera independently with respect to the robot, thereby eliminating the necessity for overlapping fields of view among sensors. This strategy not only simplifies the calibration process but also expands its versatility to support various sensor arrangements without sacrificing precision.

Multi-Sensor Joint Optimization

When multiple cameras are mounted on the mobile platform, each can be calibrated independently through the optimization process described in Section 3.5.1. However, if the cameras have overlapping fields of view, cross-detections, as introduced in Section 3.4.1, may occur, since multiple cameras can simultaneously observe the calibration pattern, providing additional information about their relative poses. In such a scenario, it is possible to include mutual relations between cameras in the cost function to optimize not only the extrinsic parameters of each camera relative to the robot frame, but also the relative poses between pairs of cameras.

Consider the scheme in Figure 3.14 and assume that two cameras C_1 and C_2 are mounted on the vehicle in the front direction. When the robot moves from pose i to pose $i+1$, if both cameras can detect the calibration pattern, the relation between their incremental motion can be expressed as:

$$T^{C_{1,i}}_{C_{1,i+1}} T^{C_1}_R T^R_{C_2} = T^{C_1}_R T^R_{C_2} T^{C_{2,i}}_{C_{2,i+1}} \tag{3.23}$$

which can be easily reformulated in the general $AX = XB$ equation by assuming $X = T^{C_1}_R T^R_{C_2} = (T^R_{C_1})^{-1} T^R_{C_2}$. Note that such unknown matrix is expressed as the product of the two camera to robot transformations $T^R_{C_1}$ and $T^R_{C_2}$ to be estimated in the calibration process, therefore providing an additional constraint to be exploited in the optimization.

Let denote by $\mathcal{S}_i$ the set of camera pairs (j,k) which detects the calibration pattern at step i, and $\mathbb{1}_{\mathcal{S}_i}$ the corresponding indicator function. With such notation, the constraint describing the joint detection by multiple M cameras at step i can be expressed as:

$$c_{Ji} = \sum_{j=1}^{M} \sum_{k=j}^{M} \mathbb{1}_{\mathcal{S}_i} \left\| T^{C_{j,i}}_{C_{j,i+1}} T^{C_j}_R T^R_{C_k} - T^{C_j}_R T^R_{C_k} T^{C_{k,i}}_{C_{k,i+1}} \right\|^2 \tag{3.24}$$

Finally, including the joint constraint (3.24) in the cost function (3.22), the overall cost function can be expressed as:

$$c = \sum_{i=0}^{N-1} \left(\sum_{j=1}^{M} \left(\left\| T^{R_i}_{R_{i+1}} T^R_C - T^R_C T^{C_{j,i}}_{C_{j,i+1}} \right\|^2 + c_{Zi} \right) + c_{Ji} \right) \tag{3.25}$$

In the following, Joint-MEMROC will be used to denote the calibration process with joint optimization. By exploiting the mutual constraints between camera poses introduced by overlapping fields of view, our extensive experiments will demonstrate that Joint-MEMROC significantly enhances calibration accuracy. This improvement comes from incorporating additional geometric information derived from overlapping areas, enriching the calibration process by providing a more detailed understanding of the sensor configuration.

Both the aforementioned optimizations are implemented using a trust-region formulation based on the Levenberg–Marquardt algorithm and solved with a Dense Schur linear solver. To improve numerical stability and robustness in the presence of noisy measurements and occasional outliers, mixed-precision solves and a Cauchy robust loss function are employed throughout the optimization.



FIGURE 3.15: Set of 4 images representing 4 different scenarios where the dataset was collected. The first two on the left show two different real-world scenarios, the other two simulated scenarios.

3.5.2 MEMROC Dataset

To evaluate this proposed method, we collected a dataset[5] comprising 3000 synthetic images and 3000 real images, to comprehensively validate the proposed method for calibrating a multi-camera system fixed on a mobile robot. The synthetic images were generated using the Gazebo simulation toolbox, wherein we utilized a Husky Unmanned Ground Vehicle (UGV) from Clearpath Robotics[6] equipped with three simulated Intel RealSense Depth D455 sensors. We conducted 10 independent image acquisition runs across diverse environments, each featuring distinct background and floor textures, as shown in Figure 3.15.

For each run, the mobile robot was moved in 100 different poses, at varying distances and orientations in front of a 5×4 checkerboard pattern with 10 cm squares. During each pose, all three sensors simultaneously captured images, their corresponding 3D point clouds, and the robot's odometry (i.e., its position and orientation relative to the starting position).

The synthetic dataset includes ground truth information consisting of:

- The geometric transformations of each sensor relative to the mobile robot's reference frame;
- The relative rototranslations among the various sensors of the multi-camera system.

The real part of the dataset consists of images captured by the mobile robot's three onboard cameras, all front-facing, from Robotnik[7]. The robot is equipped with two Intel RealSense LiDAR L515 sensors and one Intel RealSense Depth D455 sensor, each mounted in order to have overlapping fields of view (FoV). While D455 cameras can be used both outdoor and indoor, L515 cameras are designed especially for indoor applications and provides very low-quality point cloud when used outside; in the real dataset we consider both sensor to prove the robustness and generality of our approach.

The real-world data consists of images captured in 10 diverse scenarios: 6 indoor and 4 outdoor environments. We collected the data by driving the mobile robot in 100 poses at various distances and orientations with respect to a 6×5 checkerboard with 11 cm squares. During data acquisition, all three sensors captured images simultaneously together with their corresponding 3D point clouds and the robot's odometry.

The indoor data encompasses two distinct environments showcasing variations in ground surfaces, background textures, and the inclination and positioning of the calibration pattern. This diversity mainly aims to challenge the calibration methods with different visual features. The outdoor data focuses on capturing images with diverse backgrounds and varying checkerboard inclinations. This data is particularly valuable for testing the robustness of calibration methods against potential lighting variations and when the robot's odometry becomes very noisy due to the inherent drift experienced on uneven terrains.

[5] https://github.com/davidea97/MEMROC.git
[6] https://clearpathrobotics.com/
[7] https://robotnik.eu/

Crucially, for both the synthetic and real datasets, we maintained the relative positions of the cameras throughout the entire data collection process. This allows us to evaluate the consistency of calibration methods across multiple runs.

This dataset is valuable for comparing various multi-camera calibration methods. The availability of ground truth information enables to assess how well they perform with or without the calibration patterns, their accuracy with varying image quantities and their effectiveness in different scenarios.

Furthermore, in our dataset a checkerboard pattern is visible in all images: it can be used not only for calibration but also as ground truth for the evaluation of visual odometry methods [119, 120]. To the best of our knowledge, no other publicly available dataset offers this combination of features, making it a beneficial resource for the research community.

3.5.3 Experiments and Results

We conducted several experiments on the collected data to demonstrate the effectiveness and robustness of our calibration method in practice. First, we validated our method using synthetic data, whose ground truth parameters are known (Section 3.5.3). Next, we assessed the calibration accuracy in the real-world environments (Section 3.5.3). We considered both datasets for comparing the proposed methods against the state-of-the-art calibration approaches, namely the motion-based method proposed by Zuñiga et al. [104] and the learning-based method proposed by Yan et al. [106]. They both calibrate sensors with respect to the vehicle reference frame, but the latter calibration method only provides rotation calibration parameters, therefore only rotation errors will be reported in the resulting tables. To compare calibration performance, we also assessed the accuracy of estimating the transformations between the various sensors in the multi-camera system. To establish a reliable baseline for our measurements in real-world calibration parameter experiments, we utilized the stereo calibration method developed by Li et al. [121] as the ground truth reference. This method will also be used as a comparison in the simulated dataset to show its correctness and accuracy.

Experiments on Synthetic Dataset

The primary goal of using synthetic data is to evaluate the correctness and accuracy of our calibration method by directly comparing it to the ground truth. This allows us to compute the translation error across the three axes and the rotation error around the three rotation axes, expressed in terms of roll, pitch, and yaw. This metric was used to evaluate first the precision of the estimated camera-to-robot transformation and then we assess the consistency of the transformations among cameras by evaluating the camera-to-camera rototranslation error. This helps us identify any potential cumulative errors that might arise during the calibration process.

Tables 3.8 and 3.9 report the average error for each calibration parameter of camera-to-robot and camera-to-camera transformations, respectively. These errors represent the mean values calculated across all 10 synthetic datasets.

TABLE 3.8: Average error in Camera-to-Robot transformation

Method	**Translation parameters**			**Rotation parameters**		
	t_x [cm]	t_y [cm]	t_z [cm]	r_x [deg]	r_y [deg]	r_z [deg]
MEMROC	**0.61**	0.25	0.15	**0.08**	**0.13**	0.16
Joint-MEMROC	0.66	**0.16**	**0.14**	0.14	0.14	**0.15**
RobotAutocalib [104]	0.92	0.79	0.92	1.87	1.32	1.84
SensorX2Car [106]	–	–	–	0.26	0.82	1.22

TABLE 3.9: Average error in Camera-to-Camera transformation

Method	Translation parameters			Rotation parameters		
	t_x [cm]	t_y [cm]	t_z [cm]	r_x [deg]	r_y [deg]	r_z [deg]
MEMROC	0.42	**0.23**	0.87	**0.11**	0.18	0.15
Joint-MEMROC	**0.20**	0.25	0.78	0.22	0.23	**0.14**
RobotAutocalib [104]	2.58	4.99	3.72	2.52	3.09	1.68
SensorX2Car [106]	–	–	–	0.44	0.21	2.40
Li [121]	0.26	0.27	**0.34**	0.12	**0.14**	0.17

Our method achieves higher calibration accuracy for both translation and rotation errors compared to existing state-of-the-art approaches. Notably, RobotAutocalib struggles with camera-to-camera transformations, possibly due to its disregard for relative constraints existing among cameras. Conversely, the Li stereo calibration method performs well in this specific scenario. Yan's method, despite being learning-based, demonstrates strong accuracy and generalizability across various scenarios, particularly for rotation parameters.

Analysis on image quantity the synthetic dataset enables us to evaluate the impact of the number of images on the effectiveness of our multi-sensor calibration method.

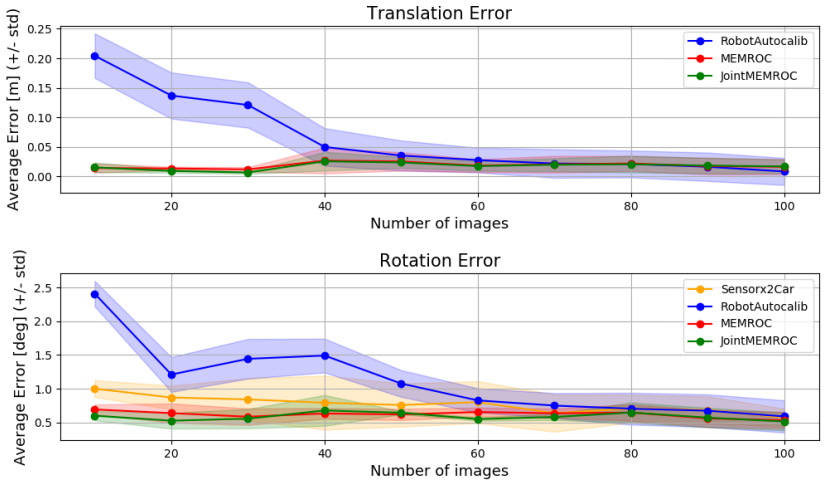


FIGURE 3.16: Impact of image quantity on calibration methods, where the lines represent the average errors and the shaded areas indicate the standard deviation, reflecting the variability and consistency of each method.

Figure 3.16 illustrates the substantial impact of image quantity on the effectiveness of calibration methods. From this point of view MEMROC shows remarkable robustness and accuracy even with minimal data available for the calibration, outperforming the motion-based method by Zuñiga's algorithm and the learning-based Yan's approach.

Analysis with noisy odometry the synthetic dataset enabled us to test the robustness and efficiency of our method in challenging scenarios, specifically where odometry measurements are compromised by noise. To thoroughly evaluate robustness in such conditions, we introduced Gaussian noise $N(0,\sigma_t)$ for translation, with $\sigma_t = \lambda \cdot 0.2\,\text{cm}$ and Gaussian noise $N(0,\sigma_r)$ for rotation, with $\sigma_r = \lambda \cdot 0.01\,\text{rad}$ where $\lambda \in [0,10]$. Noise primarily affects odometry measurements in the x and y axes for translation, and in the z-axis for rotation,

reflecting the 2D nature of the motion estimates. Figure 3.17 illustrates how increasing noise levels in odometry measurements affect the average translation and rotation errors in camera-to-robot transformations. As a result, the comparison in this figure is limited to motion-based calibration methods, since they rely on odometry data.

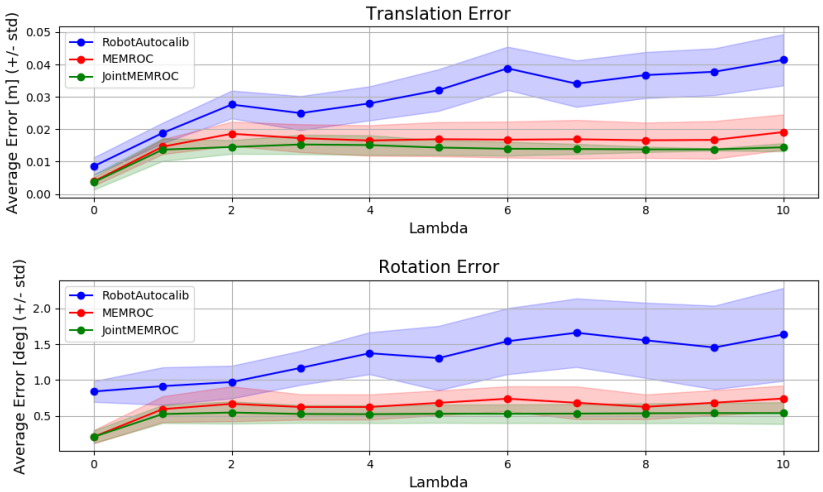


FIGURE 3.17: Evaluation of motion-based calibration method with incremental noise in measurements provided by odometry, with $\lambda \in [0, 10]$. The lines represent the average errors and the shaded areas indicate the standard deviation.

Figure 3.17 demonstrates that the proposed motion-based method exhibits significantly higher accuracy and robustness compared to the RobotAutocalib method under noisy conditions. Notably, our method achieves translation errors consistently below 2 cm and rotation errors below 1°. This highlights its suitability for real-world scenarios with noisy measurements. Our experiments with synthetic data demonstrate the efficiency of the proposed method in two key ways. First, they enable the calibration of all 6-DoF for both camera-to-robot and camera-to-camera transformations. Second, they achieve high accuracy, surpassing the motion-based method by Zuñiga and the learning-based by Yan. The ability to calibrate with just a few images highlights that the system is particularly suited for real-world scenarios where a fast and user-friendly approach is crucial.

Experiments on Real-World Scenarios

To comprehensively assess the effectiveness of our proposed calibration methods, we conducted a series of experiments in real-world indoor and outdoor scenarios. These experiments allowed us to analyze not only the precision of the calibration but also its robustness under non-ideal conditions. This includes situations where odometry measurements suffer from drift errors. Notably, these real-world tests were crucial for demonstrating the consistency of the results across diverse environments with varying visual-features and floor types, which can significantly impact odometry accuracy. We evaluated our calibration method against state-of-the-art approaches on 10 real-world datasets, each containing 300 images. Figure 3.18 shows for simplicity only the average value and standard deviation obtained by MEMROC and state-of-the-art methods across all runs for each calibration parameter (t_x, t_y, t_z for translation and r_x, r_y, r_z for rotation) of the Intel RealSense LiDAR L515 sensor with respect to the robot.

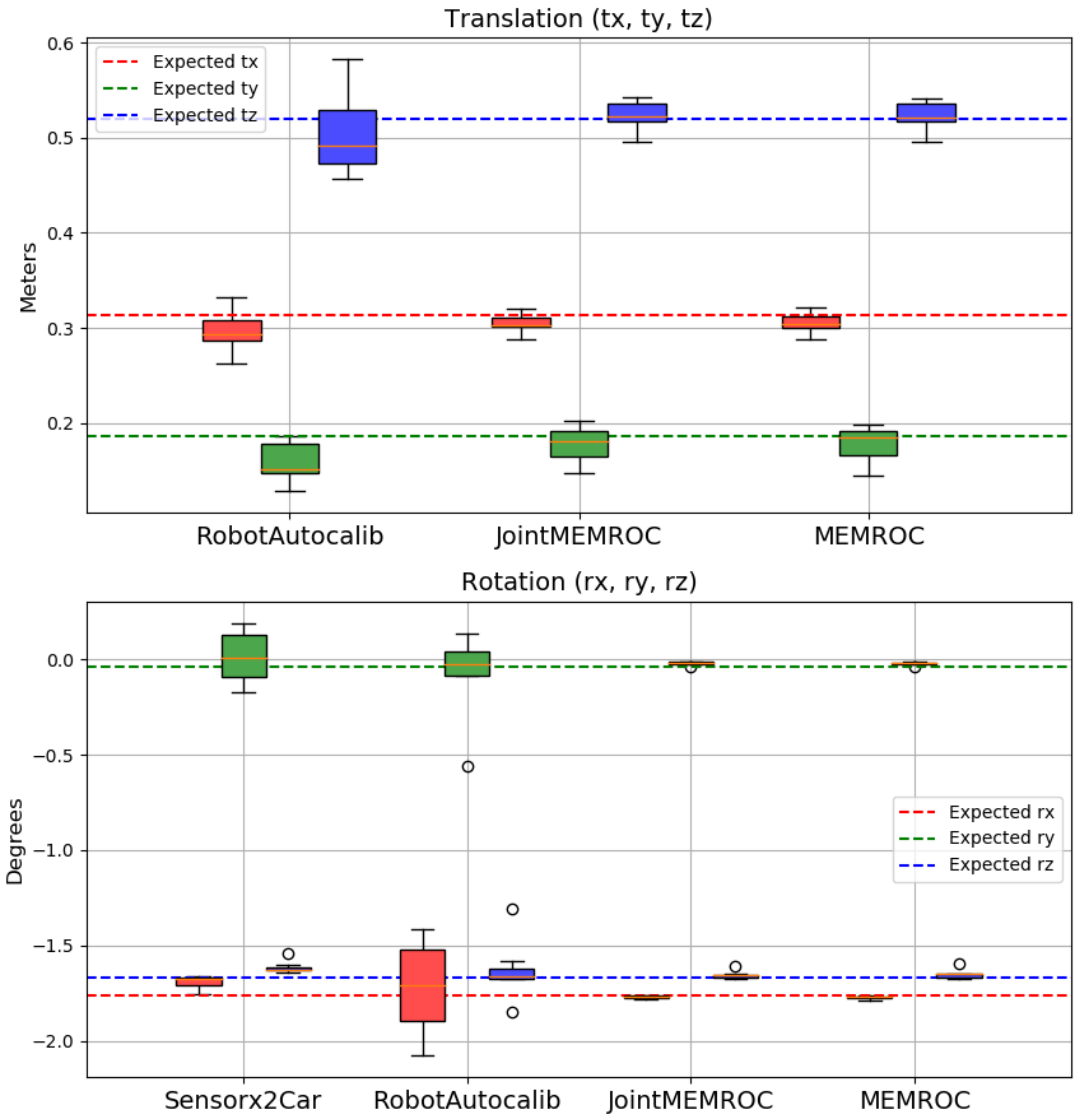


FIGURE 3.18: Calibration parameters with average and standard deviation of the Intel RealSense LiDAR L515 sensor estimated by different calibration methods. The expected values were measured by using an AprilTag fixed at a precise distance from the mobile robot in front of the sensors.

Our experiments demonstrate that MEMROC consistently outperform existing approaches. It achieves higher accuracy on average, and it exhibits greater consistency across multiple runs of the algorithm, resulting in reliable and precise measurements.

Table 3.10 presents the average errors for relative calibration parameters between the three cameras. Our method achieves superior calibration accuracy compared to both state-of-the-art motion-based and learning-based approaches. It should be emphasized that the MEMROC achieves even higher precision through joint optimization of all sensors, leveraging their overlapping fields of view. This technique is particularly effective for multi-sensor calibration.

3.5.4 Discussion

In this work we presented MEMROC, a novel method for calibrating multiple sensors with respect to a mobile robot. Utilizing robust ground plane detection, MEMROC addresses z-unobservability in planar motion, enabling the estimation of cameras' full 6-DoF pose with

TABLE 3.10: Average error in Camera-to-Camera transformation

Method	Translation parameters			Rotation parameters		
	t_x [cm]	t_y [cm]	t_z [cm]	r_x [deg]	r_y [deg]	r_z [deg]
MEMROC	1.97	1.15	0.34	0.79	1.02	0.72
Joint-MEMROC	**0.48**	**0.56**	**0.33**	**0.74**	**0.83**	**0.66**
RobotAutocalib [104]	2.41	4.42	2.43	3.83	4.82	4.22
SensorX2Car [106]	-	-	-	2.95	1.99	2.75

just a few images of a simple calibration pattern. Moreover, when sensors have overlapping fields of view, MEMROC exploits this condition optimizing also the geometric constraints existing between sensors. A comprehensive analysis of our method against state-of-the-art methods demonstrates that it is highly accurate and more robust than other methods in non-ideal conditions, when the odometry is affected by noise. Moreover, MEMROC proves to be suitable for daily use and for real-world scenarios as it requires only minimal data to achieve higher calibration accuracy. Nevertheless, its primary limitation remains the requirement of observing a calibration pattern.

3.6 Calib3R: A 3D Foundation Model for Multi-Camera to Robot Calibration and Metric-Scaled Reconstruction

The methods presented so far, unified single-camera calibration (Section 3.3), multi-camera extensions for manipulators (Section 3.4), and motion-based techniques for mobile robots like MEMROC (Section 3.5), represent significant advancements in camera-to-robot calibration. However, a persistent limitation, particularly for achieving robust calibration accuracy in diverse scenarios, has been the need on calibration patterns. This necessity is crucial for previous approaches. For instance, MEMROC effectively used a pattern crucially to overcome the inherent scale ambiguity in camera motion estimation from RGB data alone; observing a known-size pattern allows estimating the camera's metric motion (B_i), which is essential for solving the $AX = XB$ problem. While effective, the need for these physical targets hinders seamless deployment, automation, and frequent recalibration in unstructured or dynamically changing environments where placing or ensuring visibility of a pattern is impractical. This motivates exploring pattern-less approaches.

Interestingly, the core challenge mirrors that of metric 3D reconstruction from RGB images, such as Structure-from-Motion (see Section 2.4). Both calibration and reconstruction fundamentally rely on estimating camera motion and establishing geometric correspondences between views. However, traditional RGB-based reconstruction typically yields geometry only up to an unknown scale factor and in an arbitrary coordinate frame, disconnected from the robot. The key insight is that these two problems might be solved jointly. If we can leverage the known metric motion of the robot (from kinematics or odometry) alongside the relative camera motion and scene geometry derived from image correspondences (even if initially scale-ambiguous), we could potentially formulate a unified optimization. Such a process could simultaneously determine the correct metric scale for the reconstruction and the camera-to-robot transformation required to align that reconstruction with the robot's reference frame, all without needing a calibration pattern. This unified perspective paves the way for the next advancement presented in this chapter.

To exploit this opportunity, we propose Calib3R, the first method to jointly perform

camera-to-robot calibration and dense 3D reconstruction using only RGB images and corresponding robot poses, all within a single comprehensive optimization process. While previous works have explored patternless calibration using Structure-from-Motion or even recent foundation models, they typically decouple reconstruction and calibration into separate steps [122], often failing to resolve metric scale correctly or requiring post-processing alignment [123, 124], especially when relying on hand-crafted features such as SIFT [63] for point correspondences, can lead to suboptimal calibration accuracy and reduced robustness. Calib3R addresses these shortcomings through its unified formulation. Images captured by a single- or multi-camera system mounted on a robot are first processed by the 3D foundation model MASt3R [4], which produces local pointmaps. These pointmaps, combined with the robot's poses, are used in a unified optimization that reconstructs the global scene geometry, performs camera-to-robot calibration, and, in the multi-camera case, refines the relative alignment between cameras. The result is a metric-scaled 3D reconstruction aligned with the robot's reference frame, as illustrated in Figure 3.19.

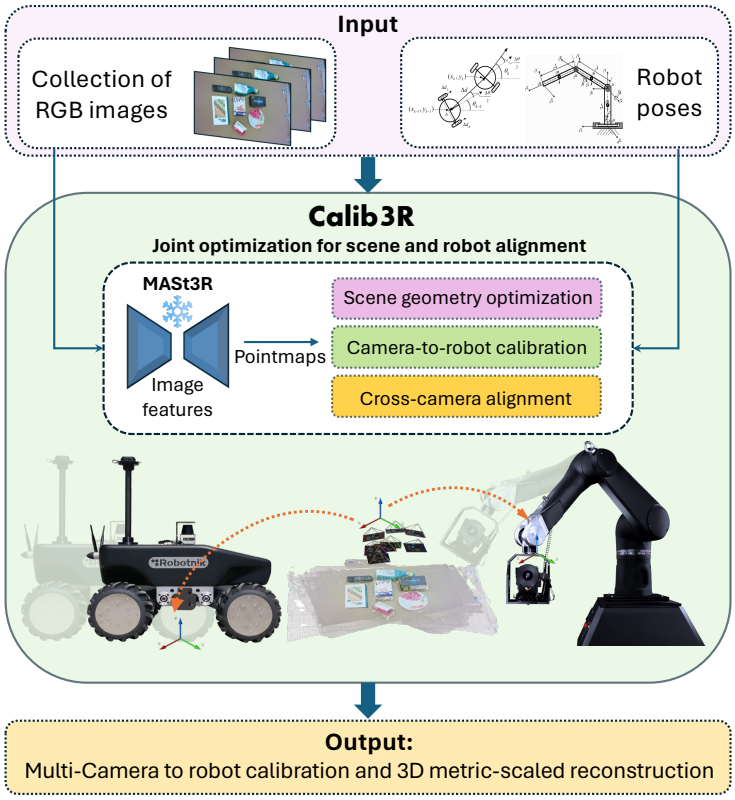


FIGURE 3.19: Overview of the Calib3R method. A set of RGB images and their corresponding robot poses are used as input. The images are processed by the 3D foundation model MASt3R to generate dense local pointmaps. These pointmaps, together with the robot poses, are then used in a joint optimization that estimates the global scene geometry, performs camera-to-robot calibration, and refines cross-camera alignment. The result is a metric-scaled 3D reconstruction aligned with the robot reference frame.

In summary, this section presents the following key contributions:

- Calib3R, a unified framework for joint camera-to-robot calibration and metric-scaled 3D scene reconstruction in robotic systems with multiple cameras, including both mobile platforms and manipulators;

- a joint loss function that integrates relative local reconstructions from a 3D foundation model with robot motion constraints, enabling the estimation of all main camera parameters in a single optimization step without requiring any calibration pattern;
- an extensive performance evaluation on diverse datasets, demonstrating that Calib3R achieves a precision level comparable to or better than pattern-based methods, while being more general;
- a systematic study on data efficiency, showing that Calib3R remains robust and outperforms both targetless and pattern-based baselines even in low-data scenarios.

3.6.1 Preliminaries

Our method takes inspiration from the recent MASt3R-SfM framework [125], which performs Structure-from-Motion optimization based on pointmaps output by the 3D foundation model MASt3R [4]. In this preliminaries section, these models are briefly described highlighting their main mathematical concepts and notations, that is similar to the ones introduced in Section 2.6.1.

MASt3R is a recent 3D foundation model which computes joint local 3D reconstruction and dense pixel-wise matching between a pair of input images. Formally, given two RGB images of size $W \times H$, denoted as $\mathcal{I}^i$, $\mathcal{I}^j \in \mathbb{R}^{W\times H\times 3}$, the model predicts two corresponding pointmaps, $\mathbf{X}^{i,i}$, $\mathbf{X}^{i,j} \in \mathbb{R}^{W\times H\times 3}$. These pointmaps are dense 2D maps with the same structure as the input images, where each pixel, instead of storing an RGB color, contains its corresponding 3D coordinate expressed in the reference frame of $\mathcal{I}^i$. Each pointmap is associated with its own confidence map, $\mathbf{C}^{i,i}$ and $\mathbf{C}^{i,j} \in \mathbb{R}^{W\times H\times 1}$ which indicate the reliability of the predicted 3D points. Additionally, the model outputs two d-dimensional features maps $\mathbf{D}^{i,i}, \mathbf{D}^{i,j} \in \mathbb{R}^{W\times H\times d}$, capturing local appearance features. To establish reliable correspondences between the two images, the fast nearest-neighbor (NN) algorithm from [4] is applied to the feature maps $\mathbf{D}^{i,i}$ and $\mathbf{D}^{i,j}$. This algorithm initializes a set of points on a regular pixel grid and iteratively refines them to obtain the correspondences $\mathcal{M}^{i,j} = \{y^i_p \leftrightarrow y^j_p\}_{p=1,\dots,|\mathcal{M}^{i,j}|}$ between $\mathcal{I}^i$ and $\mathcal{I}^j$, where y^i_p, y^j_p denotes a pair of matching pixels.

MASt3R-SfM builds upon the MASt3R model to develop an unconstrained SfM pipeline for multi-view reconstruction from a collection of images. It exploits MASt3R's encoder as an image matcher to identify pairs of overlapping images, and its decoder to perform pairwise local 3D reconstruction in the form of pointmaps, along with dense pixel-wise correspondences for the selected pairs. Based on the resulting local pointmaps, it performs a two-stage global optimization: first aligning local reconstructions in 3D space via a matching loss, then refining the alignment by minimizing 2D reprojection errors.

Specifically, given a collection of N images, MASt3R-SfM first constructs a sparse scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each vertex in $\mathcal{V}$ is an image and each edge $e = (n, m) \in \mathcal{E}$ is a pair of overlapping images [125]. To select this subset of informative image pairs, it adopts a scalable pairwise image matcher which predicts co-visibility scores between image pairs to retain only those with sufficient overlap. The matcher builds on the ASMK (Aggregated Selective Match Kernels) image retrieval method [126], leveraging the token features produced by MASt3R's encoder as local features, to predict an approximate co-visibility score $s \in [0, 1]$ that measures the likelihood of overlapping content between two images, $\mathcal{I}^i$ and $\mathcal{I}^j$. By evaluating all possible image pairs, a symmetric co-visibility score matrix $S \in [0, 1]^{N\times N}$ is constructed. MASt3R-SfM then builds a sparse, connected co-visibility graph by sampling a minimal yet connected set of image pairs from $\mathcal{E}$, via farthest point sampling and nearest-neighbor connections. For each image pair (i, j) in the co-visibility graph, MASt3R's decoder is used for joint local 3D reconstruction and pixel-wise matching, obtaining raw

pointmaps $\mathbf{X}^{i,i}$, $\mathbf{X}^{i,j}$. Since each pair involving a same image $\mathcal{I}^n$ provides a different estimate of $\mathbf{X}^{n,n}$ and $\mathbf{C}^{n,n}$ for all $(n,j) \in \mathcal{E}^n$ where $\mathcal{E}^n = \{(a,b) \in \mathcal{E} | a = n \vee b = n\}$, the various estimates obtained for an image $\mathcal{I}^n$ are aggregated into a canonical pointmap $\tilde{\mathbf{X}}^n$ using a confidence-weighted average defined as:

$$\tilde{\mathbf{X}}^n = \frac{\sum_{e \in \mathcal{E}^n} \mathbf{C}^{n,n}(e)\, \mathbf{X}^{n,n}(e)}{\sum_{e \in \mathcal{E}^n} \mathbf{C}^{n,n}(e)} \tag{3.26}$$

where $\mathbf{X}^{n,n}(e)$ is the pointmap $\mathbf{X}^{n,n}$ obtained from the image pair $e = (n,j) \in \mathcal{E}^n$. MASt3R-SfM aims to compute global scene reconstruction, which strongly depends on parameters such as the camera intrinsics K and the extrinsic pose of each view, denoted as $T^{\mathcal{W}}_{\mathcal{C}_n}$, for $n = 0, \ldots, N-1$. To ensure that the pointmaps precisely satisfy the pinhole projection model, it introduces a *constrained pointmap* χ^n defined explicitly as a function of the camera parameters:

$$\chi^n = \pi_n^{-1}(\sigma_n, K, T^{\mathcal{W}}_{\mathcal{C}_n}, \tilde{Z}^n) \tag{3.27}$$

where each 3D point $\chi^n_{i,j}$ at pixel (i,j) in the constrained pointmap is obtained by means of the inverse reprojection function π_n^{-1} depending on 3D points from the depth map $\tilde{Z}^n = \tilde{\mathbf{X}}^n_{:,:,3}$, the camera intrinsics K, the camera pose of view n, $T^{\mathcal{W}}_{\mathcal{C}_n}$, and a scale factor σ_n. These parameters are treated as optimization variables for each pointmap to enforce relative consistency across views through a two-step optimization process that addresses the global alignment.

First, the optimization coarsely aligns each local pointmap χ^n within a common world coordinate system $\mathcal{W}$ by minimizing a 3D matching loss $\mathcal{L}_{3D}$ via gradient descent. This loss reduces the distance between corresponding 3D points across multiple views and is defined as:

$$\mathcal{L}_{3D} = \sum_{\substack{p \in \mathcal{M}^{(n,m)} \\ (n,m) \in \mathcal{E}}} q_p \left|\left| \chi^n_p - \chi^m_p \right|\right|_2 \tag{3.28}$$

where $p \in \mathcal{M}^{n,m}$ denotes the matching pixels in each respective image, weighted by their respective confidence q_p. Then, a second optimization stage refines the scene geometry by minimizing the 2D reprojection error of 3D points in all image pairs, represented by the reprojection loss $\mathcal{L}_{2D}$ defined as:

$$\mathcal{L}_{2D} = \sum_{\substack{p \in \mathcal{M}^{(n,m)} \\ (n,m) \in \mathcal{E}}} q_p \left(\left|\left| y^n_p - \pi_n(\chi^m_p) \right|\right|_2 + \left|\left| y^m_p - \pi_m(\chi^n_p) \right|\right|_2 \right) \tag{3.29}$$

ensuring that projected 3D points $\pi_n(\chi^m_p)$ from camera m accurately correspond to their original 2D observations y^n_p on camera n and viceversa, with each term weighted by the confidence q_p of the pixel correspondence between y^n_p and y^m_p.

However, this two-step sequential approach is prone to error accumulation, making it less reliable than joint optimization methods. Moreover, MASt3R-SfM exhibits physical scale inconsistencies [122], producing 3D representations that lack metric accuracy and are not aligned with the robot's reference frame, posing significant challenges for deployment in real-world robotics scenarios.

3.6.2 Calib3R

Calib3R jointly performs camera-to-robot calibration and 3D scene reconstruction, supporting a wide range of robotic systems equipped with RGB cameras, including both mobile platforms and manipulators, as illustrated in Figure 3.19.



FIGURE 3.20: Formulation of camera-to-robot calibration. Let $\mathcal{R}_i$ denote the robot's reference frame at time step i, either the robot base for a mobile platform (left) or the end-effector for a manipulator arm (right). The robot is equipped with one or more cameras $\mathcal{C}_j$ and moves from pose i to $i+1$ within the environment, capturing images of a static 3D scene during its motion. A_i and $B_{j,i}$ represent the relative motions of the robot and the j^{th} camera, while X denotes the unknown rototranslation matrix describing the pose of the camera with respect to the robot reference frame.

Calib3R follows the approach presented in MASt3R-SfM, making use of the foundation model MASt3R to compute pairwise local 3D pointmaps, which are then optimized and aligned through a global optimization process into a common reference frame. The proposed method supports both single- and multi-camera configurations mounted on a robot, employing a unified optimization framework that leverages robot poses from kinematics or odometry to ensure metric-scaled and globally consistent 3D reconstructions aligned with the robot's coordinate frame.

As illustrated in Figure 3.20, consider a mobile robot navigating an environment or a manipulator operating within its workspace, equipped with M RGB cameras. While moving through a sequence of N robot poses, the system captures images of a generic static 3D scene. Given this collection of $N \times M$ images, the objective is to estimate the transformation $T^{\mathcal{R}}_{\mathcal{C}_j}$ between each camera $\mathcal{C}_j$ and a robot reference frame $\mathcal{R}$ (e.g., robot end-effector or robot base) while simultaneously producing a metric 3D reconstruction of the observed scene, expressed in the initial robot reference frame ($\mathcal{W} = \mathcal{R}_0$). To achieve this, Calib3R first computes pairwise local 3D pointmaps as intermediate representations, exploiting the co-visibility graph strategy introduced in [125]. As described in Section 3.6.1, a subset of informative image pairs $\mathcal{E}$ is selected using a pairwise image matcher based on MASt3R's encoder, potentially including pairs from different cameras if they have overlapping content. MASt3R then processes these pairs to generate local 3D reconstructions and computes canonical pointmaps $\tilde{\mathbf{X}}^i_{(j)}$ for each image $i = 0, \ldots, N-1$ from each camera $j = 0, \ldots, M-1$.

These local reconstructions are then aligned in the robot reference frame through a single optimization process that aims to minimize a loss function involving three main contributions: (i) a scene geometry term to enforce 3D reconstruction quality, (ii) a camera-to-robot calibration term to incorporate robot motion and metric-scaled information, and (iii) a cross-camera term to model rigidity constraints between sensors when multiple cameras are available.

The **scene geometry** term estimates the rigid transformations $T^{\mathcal{W}}_{\mathcal{C}_{j,i}}$ that align each camera view $\mathcal{C}_{j,i}$ to a common reference frame $\mathcal{W}$, based on 3D correspondences between local pointmaps. This term is composed of the two losses introduced in Section 3.6.1: the 3D matching loss $\mathcal{L}_{3D}$ defined in (3.28) to minimize distance of any pair of matching 3D point in the pointmaps, and the 2D reprojection loss $\mathcal{L}_{2D}$ in (3.29) which enforce consistency

between different viewpoints.

Since Calib3R is designed for general multi-camera setups, these constraints are explicitly modeled through per-camera losses $\mathcal{L}_{3D,j}$ and $\mathcal{L}_{2D,j}$ for each generic j^{th} camera.

The **camera-to-robot calibration** term is a loss defined according to the classic hand-eye calibration formulation:

$$\mathcal{L} = \sum_{i=0}^{N-1} ||A_i X - X B_i||_2^2 \tag{3.30}$$

As illustrated in Figure 3.20, A_i and B_i are the robot and camera motions between poses i and $i+1$, while X is the unknown rigid transformation from the camera to the robot reference frame.

In traditional hand-eye calibration, both A_i and B_i are assumed to share the same metric scale, typically ensured by deriving B_i from observations of a calibration pattern. In contrast, Calib3R does not rely on any patterns and directly estimates camera motion during the optimization process, where the relative motion of the j^{th} camera from pose i to pose $i+1$ is computed as:

$$B_{j,i} = T_{\mathcal{C}_{j,i+1}}^{\mathcal{C}_{j,i}} = T_{\mathcal{W}}^{\mathcal{C}_{j,i}} \cdot T_{\mathcal{C}_{j,i+1}}^{\mathcal{W}} \quad . \tag{3.31}$$

Since such estimations of the camera motion lack a consistent metric scale, the standard hand-eye calibration formulation in (3.30) cannot be applied directly.

To overcome this, we explicitly introduce an unknown scale factor $\lambda_j > 0$ for each camera j mounted on the robot, and incorporate it into the optimization. A separate scale factor per camera is essential to handle cases where no visual overlap exists between images captured by different cameras. Without such shared content, the relative scale between their independently reconstructed scenes cannot be inferred from visual correspondences. The resulting calibration loss for the j^{th} camera is defined as:

$$\mathcal{L}_{cal,j} = \sum_{i=0}^{N-1} \left|\left|A_i X - X B_{j,i}(\lambda_j)\right|\right|_2^2 \tag{3.32}$$

where

$$B_{j,i}(\lambda_j) = \begin{bmatrix} R_{B_{j,i}} & \lambda_j t_{B_{j,i}} \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.33}$$

represents the camera motion with its translation components scaled by the λ_j factor. This formulation ensures alignment between each camera's motion and the robot's trajectory, enforcing metric consistency and enabling all camera observations to be expressed in the initial robot reference frame.

When multiple cameras are mounted on the same robotic system, for instance, in mobile platforms to enhance environmental perception [6, 104], it is possible to exploit the relative rigid transformation between cameras as additional constraints in the optimization process. To this end, Calib3R introduces a dedicated **cross-camera** consistency loss that jointly optimizes the relative poses between multiple cameras when available. By leveraging synchronized robot motion and enforcing rigidity constraints, it preserves the fixed inter-camera geometry throughout the trajectory, enabling consistent multi-camera calibration within a unified optimization framework.

As shown in Figure 3.21 for a multi-camera robot setup, the motion of two cameras $\mathcal{C}_n$ and $\mathcal{C}_m$ between poses i and $i+1$ can be expressed using the general hand-eye formulation $AX = XB$, as follows:

$$T_{\mathcal{C}_{n,i+1}}^{\mathcal{C}_{n,i}} T_{\mathcal{C}_m}^{\mathcal{C}_n} = T_{\mathcal{C}_m}^{\mathcal{C}_n} T_{\mathcal{C}_{m,i+1}}^{\mathcal{C}_{m,i}} \tag{3.34}$$

where $X = T_{\mathcal{C}_m}^{\mathcal{C}_n}$ is the unknown rigid transformation between cameras $\mathcal{C}_n$ and $\mathcal{C}_m$, while

$A_i = T^{\mathcal{C}_{n,i}}_{\mathcal{C}_{n,i+1}}$ and $B_i = T^{\mathcal{C}_{m,i}}_{\mathcal{C}_{m,i+1}}$ denote their respective motions between consecutive robot poses.

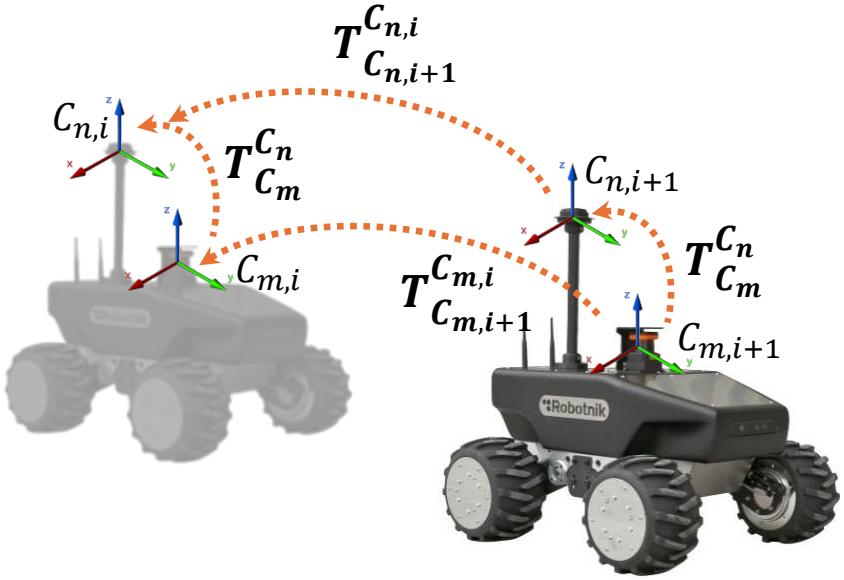


FIGURE 3.21: Relation of incremental motion of multiple cameras onboard of the same robot formulated as an hand-eye calibration problem where the unknown matrix $X = T^{\mathcal{C}_n}_{\mathcal{C}_m}$ describes the relative pose among cameras $\mathcal{C}_n$ and $\mathcal{C}_m$.

.

Applying this constraint across all images for a camera pair (C_n, C_m) mounted on the robot yields the following cross-camera loss:

$$\mathcal{L}_{cross} = \sum_{i=0}^{N-1} \left\| T^{\mathcal{C}_{n,i}}_{\mathcal{C}_{n,i+1}}(\lambda_n) \cdot T^{\mathcal{C}_n}_{\mathcal{C}_m} - T^{\mathcal{C}_n}_{\mathcal{C}_m} \cdot T^{\mathcal{C}_{m,i}}_{\mathcal{C}_{m,i+1}}(\lambda_m) \right\|^2_2 \tag{3.35}$$

where $T^{\mathcal{C}_{n,i}}_{\mathcal{C}_{n,i+1}}(\lambda_n)$ and $T^{\mathcal{C}_{m,i}}_{\mathcal{C}_{m,i+1}}(\lambda_m)$ represent the relative transformation between consecutive poses with their translation components scaled by the corresponding factors λ_n and λ_m. This cross-camera loss enforces geometric and temporal consistency across cameras by ensuring that their relative motions, remain coherent throughout the robot trajectory.

In summary, Calib3R jointly optimizes scene geometry, camera-to-robot calibration parameters, and inter-camera relationships through the following unified loss function:

$$\mathcal{L}_{Calib3R} = \sum_{j=0}^{M-1} \left(\underbrace{\mathcal{L}_{3D,j} + \mathcal{L}_{2D,j}}_{\text{scene}} + \mathcal{L}_{cal,j} \right) + \sum_{\forall(n,m)} \mathcal{L}_{cross}\,. \tag{3.36}$$

The unified loss function is minimized via gradient descent, simultaneously estimating in a single step all key camera parameters to ensure a consistent, metric-scaled, and physically-constrained reconstruction aligned with the robot's reference frame. Specifically, the optimization yields: (i) the pose $T^{\mathcal{W}}_{\mathcal{C}_{j,i}}$ for each image i of each camera j relative to the initial robot frame, (ii) a scale factor λ_j for each camera's trajectory, and (iii) the rigid camera-to-robot transformations $T^{\mathcal{R}}_{\mathcal{C}_j}$ for all cameras mounted on the robot.

When only a single camera is mounted on the robot ($M = 1$), the joint loss $\mathcal{L}_{cross}$ vanishes, since there are no camera pairs to compare. In this case, it has no influence on the overall optimization.

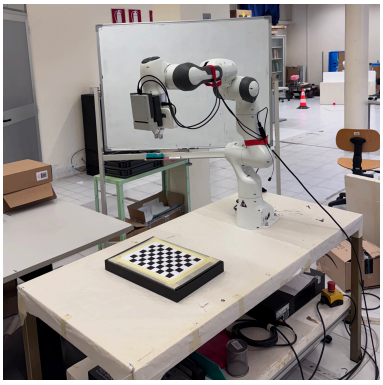


FIGURE 3.22: Franka Pattern dataset: Experimental setup with a 7×10 checkerboard (3 cm squares) placed within the robot's workspace for calibration.

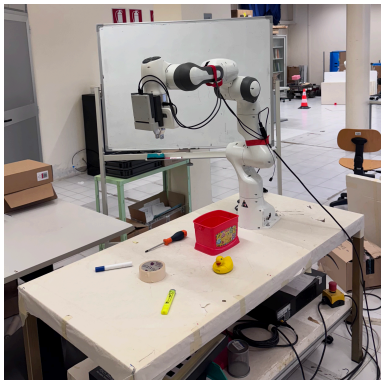


FIGURE 3.23: Franka Object dataset: Generic objects are placed in the robot's workspace to support calibration and scene reconstruction without relying on patterns.

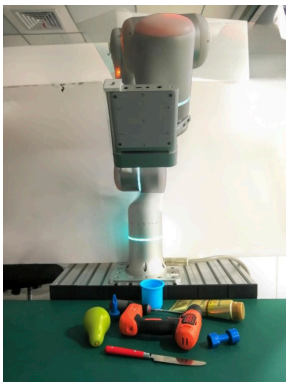


FIGURE 3.24: GraspNet-1Billion dataset: Robot setup with generic objects placed in its workspace [5].

3.6.3 Experimental Evaluation Protocol

To validate the effectiveness of the proposed Calib3R method, we designed a comprehensive evaluation protocol encompassing both robotic arms and mobile robot scenarios, demonstrating its applicability across different robotic platforms. In both contexts, we selected datasets that include at least one scene with a standard calibration pattern, enabling direct and fair comparisons with both traditional calibration methods that explicitly rely on such patterns and more recent pattern-free approaches based on generic image features. In addition, we also assess Calib3R on datasets that do not contain any pattern, to highlight its general applicability in unconstrained and realistic environments. Our evaluation is focused on two core aspects, as introduced in Section 3.6.2: the accuracy of camera-to-robot calibration, and the metric scale consistency of the scene geometry.

Metrics

We adopt two quantitative metrics to comprehensively assess the calibration accuracy and the metric scale accuracy of the 3D reconstruction.

Calibration accuracy is measured in terms of translation and rotation errors of the estimated camera-to-robot transformation, as the average of the Ground Truth Error, detailed in Section 2.3.2, obtained across all cameras.

Metric scale accuracy is evaluated on datasets that include a checkerboard. All checkerboard corners detected across N views from M cameras are reprojected into the reconstructed pointcloud and arranged based on the known checkerboard layout. We compute the mean Euclidean distance $d^{(k)}$ between adjacent corners (both row- and column-wise) for each checkerboard detection k, and evaluate the mean reconstructed square size m_s in comparison with the ground-truth square size s. We report the absolute scale error $m_s - s$, the standard deviation σ_s, and the relative scale error δ_s as follows:

$$m_s = \frac{1}{K}\sum_{k=0}^{K-1} d^{(k)}, \quad \delta_s(\%) = \frac{|m_s - s|}{s} \times 100\,. \tag{3.37}$$

This evaluation provides a robust indication of Calib3R's ability to produce metrically accurate reconstructions, an essential capability for robotic perception and interaction.



FIGURE 3.25: Real-world experimental setup of the MEMROC dataset [6].

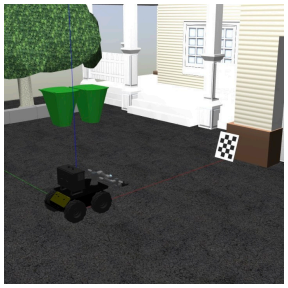


FIGURE 3.26: Experimental configuration of the synthetic MEMROC dataset [6].



FIGURE 3.27: Robot from the OpenLORIS dataset [7].

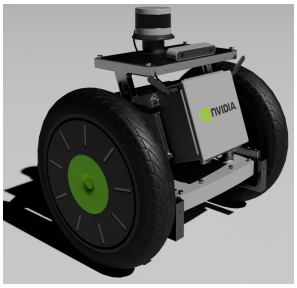


FIGURE 3.28: Experimental settings employed in the synthetic CSE Benchmark dataset [8].

Manipulator-Mounted Camera Validation

Calib3R is validated on three different robotic arm datasets: two collected using our own manipulator setup and one public benchmark. The Franka Pattern and Franka Object datasets were acquired using a Kinect Azure camera rigidly mounted on a Franka Emika Panda robot arm (see Figures 3.22 and 3.23). The Franka Pattern dataset consists of 25 robot poses captured in front of a 7×10 checkerboard with 3 cm squares, while the Franka Object dataset replicates the same poses but replaces the checkerboard with generic objects to simulate a pattern-free setting.

To extend our evaluation to realistic manipulation scenarios, we also include the GraspNet-1Billion benchmark [5], a large-scale public dataset for robotic grasping (Figure 3.24). We randomly select 25 images from scene_0100 to maintain consistency with our experimental protocol.

Mobile Robot Validation

For mobile robots, we evaluate Calib3R on multiple datasets collected across various platforms and diverse environments.

The MEMROC dataset [6] includes both real and synthetic sequences captured using a tri-camera setup mounted on a mobile robot observing a 7×6 checkerboard with 10 cm squares (see Figure 3.25 and Figure 3.26). This dataset enables direct comparison between pattern-based and pattern-free calibration methods.

The OpenLORIS dataset [7] provides single-camera sequences recorded in realistic indoor environments (Figure 3.27), while the CSE Benchmark [8] offers synthetic data generated in NVIDIA Isaac Sim using a mobile robot equipped with a dual-camera configuration (Figure 3.28).

Computational Setup

Experiments with Calib3R and the 3D foundation model-based method were conducted on an NVIDIA A40 GPU with 48 GB of GDDR6 memory. Classical calibration methods were executed on a standard workstation equipped with an Intel Core i7-1280P CPU, 16 GB of RAM, and running Ubuntu 20.04.

3.6.4 Results on Hand-Eye Calibration

We conducted extensive experiments on Franka Pattern dataset (Figure 3.22) to compare classical pattern-based hand-eye calibration methods [61, 91, 100, 111, 92, 51], against recent reconstruction-based methods inspired by Zhi [122]. Specifically, reconstruction-based

methods, including COLMAP [1] and 3D foundation model such as DUSt3R [3], MASt3R-SfM [125], and VGGT [88], were used to estimate camera poses purely through visual geometry. Subsequently, these poses are used to perform hand-eye calibration using the formulation proposed by Zhi [122].

TABLE 3.11: Average error in the estimated camera-to-robot (hand-eye) transformation, computed using images acquired from 25 robot poses in the Franka Pattern dataset (Figure 3.22), Franka Object dataset (Figure 3.23) and GraspNet-1Billion (Figure 3.24). The Pattern column indicates whether the corresponding method requires a calibration pattern. Traditional method can not be used in Franka Object dataset and GraspNet-1Billion benchmark since it is not involved a calibration pattern.

Method	**Pattern**	**Franka Pattern dataset**		**Franka Object dataset**		**GraspNet-1Billion** [5]	
		e_t [cm]	e_θ [rad]	e_t [cm]	e_θ [rad]	e_t [cm]	e_θ [rad]
Tsai [61]	✓	2.631	0.041	-	-	-	-
Park [91]	✓	3.610	0.038	-	-	-	-
Horaud [100]	✓	2.743	0.038	-	-	-	-
Andreff [111]	✓	11.341	0.033	-	-	-	-
Daniilidis [92]	✓	2.472	0.035	-	-	-	-
Evangelista [51]	✓	**0.781**	0.041	-	-	-	-
COLMAP [1] + Calib	✗	28.210	0.122	18.212	0.092	22.324	0.122
DUSt3R [3] + Calib	✗	24.641	0.043	16.431	0.031	18.212	0.042
MASt3R SfM[125] + Calib	✗	1.812	0.023	1.742	0.027	3.427	0.032
VGGT [88] + Calib	✗	1.734	0.034	1.674	0.038	2.342	0.031
Calib3R	✗	1.127	**0.014**	**0.415**	**0.011**	**1.744**	**0.023**

The quantitative results of our experiments, summarized in Table 3.11, present a thorough comparison of classical pattern-based (✓) and modern pattern-free calibration strategies (✗) by evaluating average translation and rotation errors in the estimated camera-to-robot transformations. Notably, Calib3R significantly outperforms all reconstruction-based methods that follow a two-step calibration pipeline, such as COLMAP + Calib and DUSt3R + Calib, reducing the translation error by an order of magnitude (1.127 cm compared to 28.210 cm and 24.641 cm, respectively). Calib3R also outperforms MASt3R-SfM + Calib, achieving nearly a 40% reduction in both translation and rotation errors. Importantly, although both methods are built on MASt3R, the joint optimization strategy of Calib3R clearly demonstrates that a unified pipeline yields more accurate and consistent results. This underlines the inherent limitations of decoupled approaches, which are more vulnerable to intermediate reconstruction errors and misalignments between stages. Furthermore, Calib3R surpasses even the strongest reconstruction-based competitor, VGGT + Calib, reducing translation error by over 30%. Among traditional pattern-based methods, Calib3R outperforms all except Evangelista's reprojection-minimization approach, which achieves a slightly lower translation error of 0.781 cm. However, Calib3R achieves the best rotation accuracy overall, including compared to Evangelista's method, reaching a remarkably low error of 0.014 radians. This robustness in rotation estimation stems from two key factors: first, rotation is inherently unaffected by scale ambiguity, which remains a challenging component of calibration; and second, unlike checkerboard- or pattern-based methods that rely on a limited set of locally accurate points, our approach leverages 3D foundation models that extract and match visual features densely and globally distributed across the image, resulting in more stable and redundant geometric constraints. To demonstrate Calib3R's broad applicability, we evaluated it not only on scenes with calibration patterns, where it performs comparably or better than the state of the art, but also on patternless environments. Specifically, we tested it on Franka Object dataset (Figure 3.23), replacing the checkerboard with arbitrary objects to simulate a realistic scenario without explicit calibration targets. In this more challenging setting, traditional pattern-based methods are no longer applicable, underscoring the importance of robust pattern-free calibration approaches. Calib3R clearly outperforms all reconstruction-based methods, achieving exceptionally low translation and rotation errors (0.415 cm and 0.011

radians, respectively). Remarkably, these results are even superior to Evangelista's method tested on Franka Pattern dataset, which had the advantage of using a dedicated calibration pattern. This highlights a significant advancement in the accuracy and reliability of pattern-free calibration approaches.

Moreover, since 3D foundation model-based reconstruction methods are typically more computationally demanding, we restrict our comparison to other foundation model-based approaches and analyze how calibration accuracy varies with the number of input images in the Franka Object dataset, as shown in Figure 3.29.

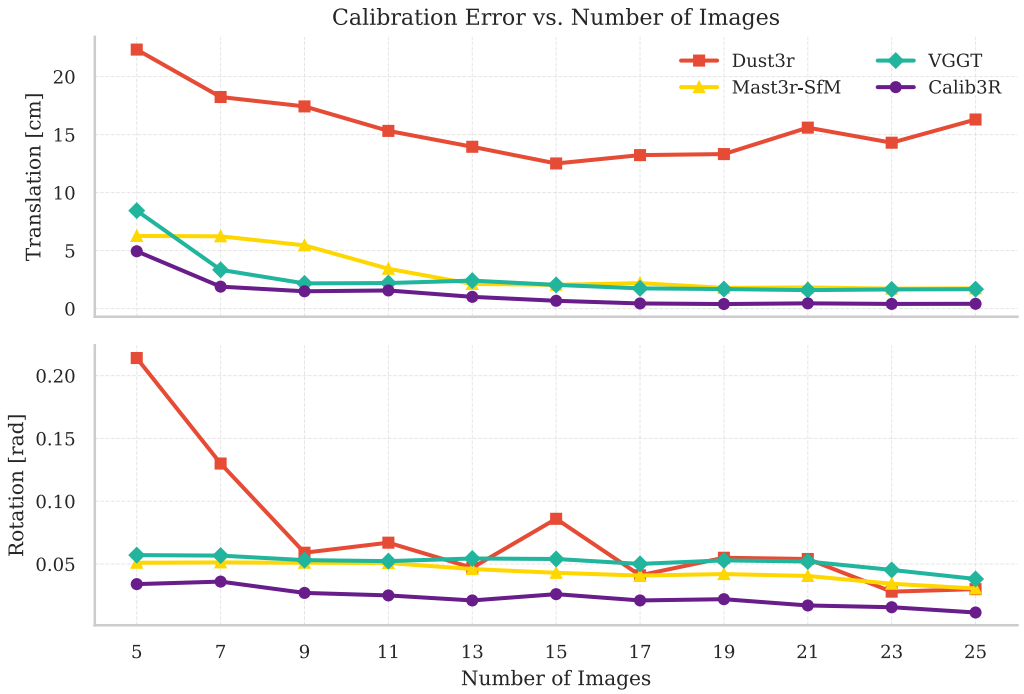


FIGURE 3.29: Camera-to-robot transformation accuracy as a function of the number of input images. Starting from 25 images, 2 random images were removed at each step down to a minimum of 5.

In this experiment, we started with the initial set of 25 images captured by the single-camera system and progressively removed 2 random images at each step until reaching a minimum of 5 images, a particularly low number for calibration. The results show that Calib3R consistently outperforms all other methods in both rotation and translation accuracy. Specifically, it achieves a translation error below 5 cm and a rotation error under 0.04 radians with as few as 5 images. Notably, with just 15 images, Calib3R maintains a translation error below 1 cm.

To further assess the robustness of the proposed method, we evaluated its calibration accuracy on the publicly available GraspNet-1Billion dataset [5]. Similarly to the Franka Object dataset we limited the comparison to pattern-free methods. The resulting hand-eye calibration accuracy is reported in the last column of Table 3.11. Calib3R achieves superior performance even on the challenging GraspNet dataset, although the results are slightly less accurate than those obtained on the Franka Object dataset. This gap may be attributed to the limited variability in robot motion within GraspNet sequences, which primarily involve grasping tasks. The resulting camera viewpoints are predominantly top-down, with minimal rotation, particularly around the roll and pitch axes. This lack of diverse and distinctive motion trajectories likely limits the observability of the unknown transformation, thereby reducing the effectiveness of calibration methods that rely on motion diversity to achieve high

accuracy. A qualitative example of the metric-scaled 3D reconstruction using a scene from the GraspNet-1Billion dataset is shown in Figure 3.30. The reconstruction was generated using 15 RGB images, and it clearly shows how all elements are accurately aligned with the robot's base reference frame.

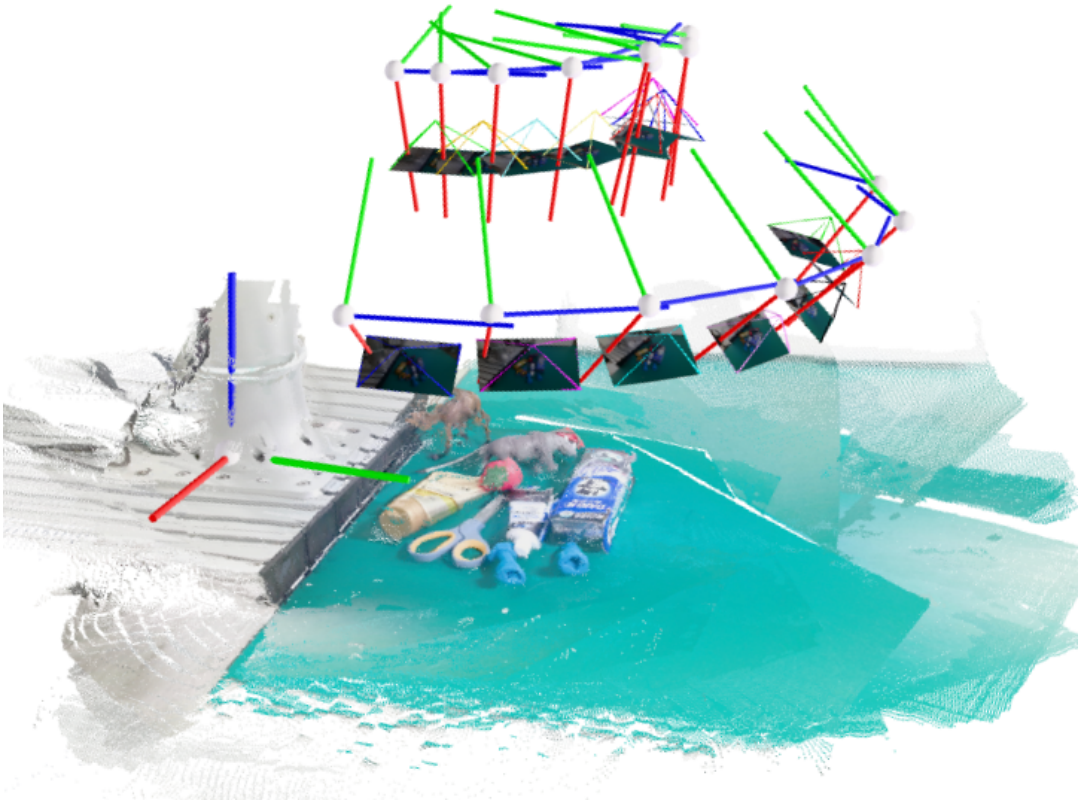


FIGURE 3.30: Metric-scaled 3D reconstruction of a scene from the GraspNet dataset [5], generated using only 15 RGB images. Both the estimated camera poses and the corresponding robot poses are expressed in the robot's base frame by applying the kinematics from the initial end-effector pose (at step 0).

As a final step, we assessed the metric scale accuracy of the reconstructed 3D scene on the Franka Pattern dataset, focusing on Calib3R and recent dense reconstruction-based approaches. Note that COLMAP was excluded from this analysis, as it does not produce reconstructions that are sufficiently dense or complete to reliably project the checkerboard corners into 3D space. The evaluation of this metric is reported in Table 3.12. As clearly

TABLE 3.12: Metric scale accuracy of 3D reconstruction on Franka Pattern dataset. The table reports the mean scale error, the standard deviation and the percentage scale error relative to the true checkerboard square size of 3 cm. Lower values indicate higher metric consistency.

Method	**Franka Pattern dataset**		
	$m_s - s$ [cm]	σ_s [cm]	ϵ_s [%]
DUSt3R [3] + Calib	0.79	0.091	26.33
MASt3R SfM [125] + Calib	0.41	0.027	13.67
VGGT [88] + Calib	<u>0.17</u>	<u>0.008</u>	<u>5.67</u>
Calib3R	**0.11**	**0.005**	**3.67**

shown, Calib3R achieves the highest metric consistency among all methods, significantly reducing the relative scale error of its closest competitor VGGT + Calib (3.67% vs. 5.67%).

3.6.5 Results on Motion-Based Calibration

To evaluate the generalizability and robustness of Calib3R in mobile robot scenarios, we performed extensive experiments with several state-of-the-art calibration methods. These approaches are designed exclusively for calibration and do not include 3D reconstruction, which is typically handled by separate systems. To the best of our knowledge, no existing method jointly performs both calibration and reconstruction in mobile robot setups. Therefore, we evaluate Calib3R against the following calibration-only baselines:

- **Motion-based methods:** Zuniga et al. [104] (pattern-free ✗) and MEMROC and its variant Joint-MEMROC [6] (pattern-based ✓);
- **Learning-based method:** SensorX2Car by Yan et al. [106] (pattern-free ✗).

Note that, a well-known limitation in mobile robot calibration is the z-unobservability problem [28]. Since the robot moves on a planar surface, the z-component of the transformation cannot be inferred from motion alone. In accordance with existing literature [104], we estimate this z-translation component post-calibration using the camera's height from the ground.

To this end, we propose a VLM-based ground plane estimation pipeline. Specifically, we employ Grounding DINO[127] and SAM 2[128] to segment the floor in each RGB image using natural language prompts. These segmentations are back-projected to the final 3D pointcloud and the final z-component of the camera-to-robot transformation is estimated as the average height of the camera across N all frames:

$$z = \frac{1}{N} \sum_{i=0}^{N-1} d_i \,. \tag{3.38}$$

This prompt-based approach proves more robust than traditional geometric methods (e.g., RANSAC [115]), which are prone to misidentifying other planar surfaces such as tables or walls as the ground.

Table 3.13 summarizes the results across the following datasets: real-world and synthetic MEMROC datasets [6], OpenLORIS[7], and the CSE benchmark[8]. The MEMROC datasets feature visible checkerboard calibration patterns, allowing the use of pattern-based methods such as MEMROC and Joint-MEMROC. In contrast, OpenLORIS and CSE do not contain such patterns, making pattern-based methods inapplicable.

TABLE 3.13: Average error in the estimated camera-to-robot transformation across the real-world and synthetic MEMROC datasets, as well as the OpenLORIS and CSE benchmarks, computed over 25 robot poses. The Pattern column indicates whether the method requires a visible calibration pattern.

Method	**Pattern**	**Real MEMROC** [6]		**Synthetic MEMROC** [6]		**OpenLORIS** [7]		**CSE benchamrk** [8]	
		e_t [cm]	e_θ [rad]	e_t [cm]	e_θ [rad]	e_t [cm]	e_θ [rad]	e_t [cm]	e_θ [rad]
MEMROC [6]	✓	3.412	0.031	0.712	**0.004**	–	–	–	–
Joint-MEMROC [6]	✓	**3.325**	0.031	**0.709**	0.005	–	–	–	–
Zuniga [104]	✗	12.231	0.123	7.172	0.113	35.126	0.292	17.123	0.189
SensorX2Car [106]	✗	–	0.043	–	0.026	–	0.039	–	0.028
Calib3R	✗	3.725	**0.021**	2.432	0.006	**3.419**	**0.033**	**2.723**	**0.005**

Calib3R demonstrates strong and consistent performance on both the real-world and synthetic versions of the MEMROC dataset. On the real-world data, it achieves results comparable to the best pattern-based method, Joint-MEMROC, with only a 12% increase in translation error (3.725 cm vs. 3.325 cm), while improving rotation accuracy by approximately 32%, as discussed in Section 3.6.4. As expected, MEMROC and Joint-MEMROC achieve superior performance on the synthetic dataset, thanks to their use of highly accurate relative

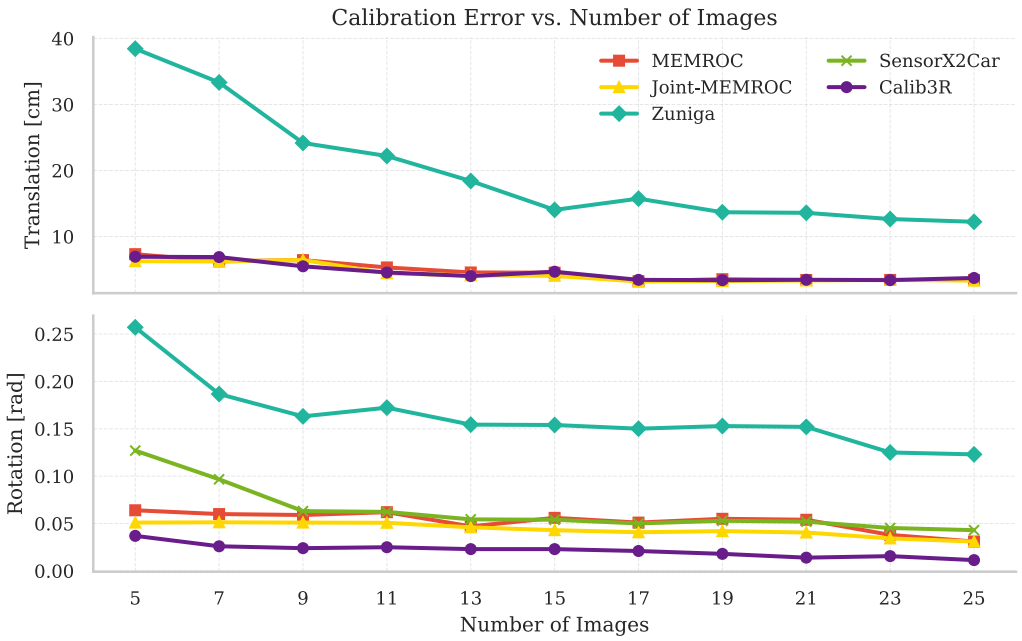


FIGURE 3.31: Camera-to-robot transformation accuracy as a function of the number of input images. Starting from 25 images, 2 random images were removed at each step down to a minimum of 5.

camera poses obtained via the calibration pattern. In such ideal conditions, free of noise, pattern detection is nearly perfect, resulting in metrically precise camera poses.

Among pattern-free methods, Calib3R clearly stands out. The motion-based method by Zuniga et al. [104] shows significantly higher errors, particularly on the real-world MEMROC, with translation errors exceeding 10 cm and rotation errors above 0.1 rad. The learning-based approach SensorX2Car [106] does not estimate translation and achieves low rotation accuracy.

We further evaluated Calib3R on the OpenLORIS [7] and CSE benchmark [8] datasets (last two columns of Table 3.13). On both datasets, Calib3R substantially outperforms the alternatives, especially in translation, achieving errors of 3.419 cm and 2.723 cm, an order of magnitude lower than those of Zuniga's method (35.126 cm and 17.123 cm), which is impractical for real-world robotic applications. SensorX2Car remains limited by its lack of translation estimation. In rotation as well, Calib3R achieves strong results outperforming all other state-of-the-art pattern-free approaches.

As in the robotic arm scenario, we evaluated the calibration methods on the real MEMROC dataset by progressively reducing the number of images per camera, from 25 down to 5. The results, shown in Figure 3.31, reveal that Calib3R significantly outperforms other pattern-free methods, maintaining high accuracy even with fewer than 10 images per camera.

A qualitative example of a 3D reconstruction using the MEMROC dataset is shown in Figure 3.32, generated from only 15 RGB images per camera. The figure illustrates the estimated camera poses alongside the corresponding mobile robot poses at each acquisition step. The reconstruction is metric-scaled and aligned with the initial position of the robot's base reference frame, accurately reflecting the robot's pose at each point in the trajectory.

The real-world MEMROC dataset was also used to evaluate the metric scale accuracy of the 3D scene reconstructed exclusively by Calib3R, since other methods focus solely on camera calibration and do not produce any reconstruction. The evaluation follows the same protocol adopted for robotic arm setups (see Section 3.6.3). Specifically, we compute the

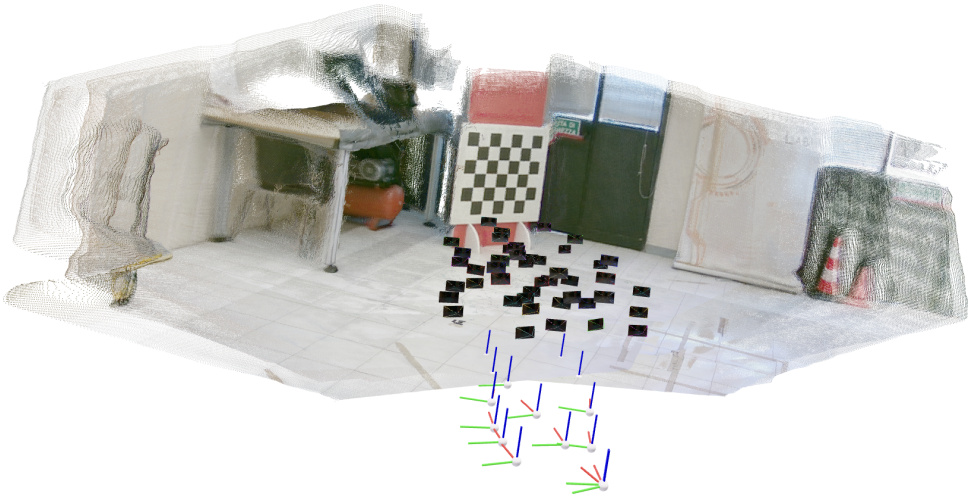


FIGURE 3.32: Metric-scaled 3D reconstruction of the environment using 15 images per camera from a MEMROC dataset sequence [6]. The reconstruction includes the estimated camera poses and the corresponding 15 robot poses, all expressed with respect to the first robot pose reference frame.

mean reconstructed square size m_s, its standard deviation σ_s, and the relative percentage error across all detectable checkerboard corners.

Calib3R achieves a highly accurate metric reconstruction, with a mean scale error of $m_s - s = 0.48\,\text{cm}$, a standard deviation of $\sigma_s = 0.025$, and a percentage scale error $\epsilon_s = 4.8\%$.

3.6.6 Discussion

In this section, we presented Calib3R, a unified calibration method for both single- and multi-camera setups mounted on robotic arms or mobile robots. Calib3R leverages the 3D foundation model MASt3R to estimate the relative pose of one or more cameras with respect to the robot's reference frame.

Our method requires only a set of RGB images acquired from the robot-mounted cameras and the corresponding robot poses obtained via kinematics or odometry. Unlike traditional approaches that treat calibration and 3D reconstruction as separate tasks, Calib3R performs both simultaneously, producing accurate camera-to-robot calibration and a metric scale 3D reconstruction of the environment aligned to the robot reference frame. To the best of our knowledge, this is the first unified approach that generalizes across both robotic arms and mobile platforms, as well as across single- and multi-camera configurations.

We carried out extensive experiments using both real-world and synthetic datasets, demonstrating that Calib3R is highly accurate, robust, and effective even in low-data conditions with fewer than 10 images per camera. Notably, we found that traditional calibration patterns can be replaced by generic task-relevant objects, achieving camera-to-robot transformation errors below 0.5 cm in robotic arm setups, outperforming conventional methods.

Limitations. Despite these advantages, Calib3R presents several limitations that point toward important directions for future work. Since the method relies on a foundation model and a global optimization over dense point maps extracted from all input images, its computational complexity grows rapidly as the number of images increases. In particular, both memory consumption and optimization cost scale nonlinearly with the number of views involved in the joint optimization.

In practice, the current implementation can efficiently handle up to approximately 50 images in total on a single GPU with 24 GB of memory, which is sufficient for a wide range of calibration scenarios considered in this thesis. However, scaling beyond this regime requires increased computational resources, such as higher-memory GPUs or distributed optimization strategies. Addressing this limitation will require future research on more scalable optimization formulations, view selection strategies, or hierarchical and incremental calibration schemes that preserve accuracy while reducing computational overhead.

Furthermore, as in classical extrinsic calibration problems, the success of the optimization critically depends on the quality and diversity of the collected data. Accurate calibration requires camera images and corresponding robot poses that provide sufficient geometric changes, particularly in terms of rotational and translational variation. If the motion of the robot exhibits limited variation, especially along certain rotational axes, the calibration problem may become ill-conditioned or degenerate, leading to poor convergence or ambiguous solutions.

At the same time, in the case of mobile robots, excessive separation between consecutive poses may introduce additional challenges due to odometry drift, which can negatively affect the consistency of the optimization. As a result, there exists an inherent trade-off between ensuring sufficiently diverse viewpoints to avoid geometric degeneracies and maintaining pose proximity to limit the accumulation of drift. Careful trajectory design and data collection strategies are therefore essential to balance these competing requirements and ensure reliable convergence of the calibration process.

Chapter 4

Human Understanding

Having established the robot-centric perception framework in Chapter 3—the essential prerequisite enabling the robot to interpret visual data within its own reference frame and connect perception to action—we now shift focus to the content derived from this aligned perception. Building upon this robust geometric foundation, the next critical step towards effective human-robot collaboration (HRC) moves beyond mere coexistence to true partnership. This requires endowing the robot with the ability to perceive, interpret, and anticipate the actions and intentions of its human counterparts within the shared environment. This capability for human understanding is crucial, particularly in dynamic industrial settings where safety and efficiency are critical [129].

Understanding human behavior serves multiple fundamental functions in HRC. Gestures can form an intuitive communication channel, allowing operators to command or guide the robot naturally. More fundamentally, continuously monitoring an operator's actions allows the robot to detect potential errors, anticipate the next steps in a collaborative task, and adapt its own behavior accordingly, thereby enhancing both safety and productivity [130].

Achieving this level of understanding hinges on tackling two core, interdependent challenges. First, the robot needs a precise representation of the human's physical state in 3D space. Accurate 3D Human Pose Estimation (HPE) is therefore a foundational prerequisite. Knowing the exact 3D position and configuration of the human body enables reliable collision avoidance, facilitates ergonomic assessments, and provides the essential input for interpreting movements. Second, building upon these pose estimates, the robot must then perform Human Action Recognition (HAR) to infer the meaning behind the observed motion [131, 132, 133].

This chapter addresses these two fundamental components of human understanding. Recognizing the limitations of existing methods, particularly their fragility in real-world HRC scenarios prone to occlusions and clutter, we first introduce a robust multi-view 3D HPE framework. This method fuses independently predicted 3D skeletons from multiple viewpoints and refines them through reprojection-based optimization, enhancing geometric consistency and accuracy even under partial visibility.

Subsequently, leveraging these enhanced pose estimates, we tackle the challenge of HAR in continuous, real-world interactions, moving beyond models trained on short, isolated action clips. We present an online HAR framework operating directly on the estimated 3D skeletons. This system employs an ensemble of lightweight binary classifiers, each specialized for a specific action, coupled with a causal temporal buffer to enable real-time, frame-by-frame inference in ongoing collaborative tasks.

Through these contributions, this work aims to equip robots with the perceptual capabilities needed to continuously recognize and appropriately respond to human actions in dynamic, unstructured environments, paving the way for more natural, safe, and context-aware human-robot interaction.

Portions of this chapter have been published in the following works:

- [48] Laura Bragagnolo, Matteo Terreran, **Davide Allegro** and Stefano Ghidoni, "Multi-view Pose Fusion for Occlusion-Aware 3D Human Pose Estimation", European Conference on Computer Vision ECCV 2024, Milan, Italy.
- [47] Matteo Terreran, Laura Bragagnolo, **Davide Allegro** and Stefano Ghidoni "A Lightweight Ensemble Framework for Online Skeleton-Based Human Action Recognition in Industrial Environments", In European Conference on Mobile Robots ECMR 2025, Padua, Italy.

4.1 3D Human Pose Estimation

As established in the chapter introduction, accurate 3D Human Pose Estimation (HPE) is a foundational prerequisite for effective HRC. However, achieving this accuracy in typical industrial scenarios presents unique and significant challenges. The typical setting includes a multi-camera setup that frames the human-robot collaboration area—cameras are usually very tilted and often not synchronized. Moreover, the human body is often only partially visible, as surrounding obstacles and equipment often occlude the subject. Such conditions are very under-represented in 3D human pose estimation datasets, which are usually collected in very controlled environments.

As multiple viewpoints are often available in human-robot collaboration workcells, most commonly adopted solutions are multi-view human pose estimation methods, that can accurately estimate 3D human joint position when camera calibration parameters are known. Most state-of-the-art multi-view approaches are, however, based on triangulation of 2D keypoints [30, 29, 31], so they cannot handle significant amounts of occlusions and clutter, as 2D pose estimation quality quickly degrades under these conditions and good predictions cannot be recovered. Recent monocular 3D human pose estimation methods have achieved remarkable results on public benchmarks [134, 135] and exhibit strong generalization to new and diverse scenarios [136, 137]. On the other hand, when huge parts of the body are not visible, they tend to struggle with precise absolute scale prediction and reliable 3D joint configuration. For these reasons, current human pose estimation systems are unsuitable for real industrial scenarios [138].

In this section, we propose a novel multi-view 3D human pose estimation method based on the fusion of 3D keypoints from multiple viewpoints to achieve robust and reliable human pose prediction in human-robot collaboration settings, characterized by the presence of heavy occlusions and few camera viewpoints.

While many current approaches fuse information at the 2D feature level and thus suffer significantly when joints are obscured, our method departs from this paradigm by taking as input full 3D human pose predictions derived from state-of-the-art absolute monocular methods.

To mitigate joints position artifacts produced in presence of large occlusions, we leverage multiple viewpoints and introduce a novel multi-view 3D keypoint fusion module to merge single-view 3D pose predictions. Considering per-joint reprojection errors on all camera views, we can effectively correct body pose hallucinations and scale recovery errors. Accurate 3D pose localization is then achieved thanks to an optimization procedure on the fused pose, considering reprojection errors on 2D views. The final prediction is further refined considering body symmetry constraints during optimization. With respect to previous work [139, 29] we do not require any prior information or statistics about human limbs length.

We validate our method on data collected from real human-robot collaboration workcells and on Human3.6M-Occluded, a novel benchmark derived from the public dataset Human3.6M [140] adding synthetic occlusions on camera views.

To summarise, the main contributions are:

1. a multi-view 3D human pose estimation method based on per-joint multi-view fusion and optimization, designed to tackle common problems in complex real scenarios, such as occlusions;
2. a novel refinement step using limb length symmetry constraints that operate without requiring prior anthropometric information or statistics;
3. an analysis on robustness and generalization capabilities of multi-view 3D human pose estimation methods in real human-robot collaboration workcells;
4. the novel benchmark Human3.6M-Occluded, to test pose estimation algorithms under severe occlusions;
5. extensive evaluation and ablation studies considering multi-view 3D human pose estimation methods on the public dataset Human3.6M and on Human3.6M-Occluded.

4.1.1 Related Work

Multi-view 3D Human Pose Estimation

Multi-view human pose estimation methods leverage multiple camera viewpoints to provide accurate human pose predictions in 3D space. They typically include two steps: first they extract 2D features from each view independently, and then reconstruct 3D information via triangulation [30, 31, 141]. If triangulation-based approaches are very precise when provided with good quality 2D predictions, occluded environments are traditionally challenging for these methods, as 2D estimated poses tend to be very noisy. Some approaches try to refine 2D predictions and triangulation incorporating epipolar geometry constraints: the epipolar transformer [141] learns attention weights to fuse pixels along the epipolar line in neighboring views; Ma et al. [142] introduce Transfusion, a transformer framework for multi-view 3D pose estimation, aiming at directly improving 2D predictions integrating information from different views via geometry position encoding. Similarly, [143, 144] propose to leverage geometric correlations between views to refine 2D pose estimations. Zhang et al. [31] present AdaFuse, which tackles human pose estimation in presence of occlusions by learning an adaptive fusion weight to reduce the impact of poor quality views. In [139] and [29] constraints on human limbs length are exploited in the human pose estimation process in order to boost accuracy and robustness. They, however, require knowledge about subject's bones length or bone length statistics computed on the dataset of interest, making it difficult to apply such methods in unconstrained environments.

These methods typically do not generalize well to new scenarios and camera arrangements without extensive re-training and still struggle when dealing with strong occlusion. Therefore, they are not suitable to industrial multi-camera environments, which are prone to significant occlusions and in which collecting huge amounts of data for training may be impractical or even impossible.

Absolute 3D Human Pose Estimation

Recently, many 3D human pose estimation approaches have focused on single-view images or videos [134, 145, 135]. This is an ill-posed problem, affected by depth ambiguities and occlusions. Most works [146, 147] mainly focus on root-relative human pose estimation, which provide human pose predictions with respect to a reference frame centered in the pelvis joint (root). Nevertheless, for real-world applications such as human-robot collaboration, it is crucial to accurately estimate the human body pose in the global 3D space [133, 131].

Pavlakos et al. [148] used a simple anthropometric approach, optimizing the absolute person distance based on 2D pixel positions and root-relative depth estimates from volumetric heatmaps, ensuring that the back-projected skeleton's bone lengths matched the average bone lengths from the training set. This method, however, has limited generalization power, as it requires pre-calculated statistics about human bone lengths. Zhan et al. [149] use an intrinsic-parameter-invariant representation to convert the 2D keypoints from pixel space to 3D rays in a normalized 3D space; such rays are then fused from consecutive frames using temporal convolution. Sárándi et al. introduce MeTRAbs [150, 151], tackling scale and distance estimation of 3D poses while also addressing body occlusions. They train a fully-convolutional network to output location and depth heatmaps for 2D joints, providing accurate metric-scale depth information. Still, when huge parts of the body are not visible, predicted scale and pose may be unreliable.

Occlusion-Aware Human Pose Estimation

Occlusion-aware human pose estimation has been primarily addressed by root-relative single-view methods. To boost resilience to missing joints, synthetically occluded data can be exploited during training, as suggested in [152]. Cheng et al. [153] make use of 2D confidence heatmaps of keypoints and introduce an optical-flow consistency constraint, to filter out the unreliable estimations. More recently, generative models have been successfully employed to retrieve occlusion-robust 3D pose, as in [147]. Several single-view datasets representing scenarios with occlusions, such as 3DPW-Occ [137] and 3DOH50K [137] have also been proposed. Such works tackle root-relative 3D human pose estimation from monocular images and videos and mainly focus on everyday life scenarios.

Our work addresses the gaps in the current literature by proposing an occlusion-aware 3D human pose estimation method from multi-view images, targeting human-robot collaboration environments. Differently from current multi-view approaches, mostly relying on 2D feature triangulation, we aggregate single-view 3D predictions from absolute human pose estimation algorithms, to achieve robustness against strong occlusions.

4.1.2 Multi-view Pose Fusion for Occlusion-Aware 3D Human Pose Estimation

The proposed method consists of three main parts: (i) single-view 3D pose prediction; (ii) multi-view 3D pose aggregation; (iii) 3D pose refinement based on reprojection error minimization. In the first step, for each camera in the network, a candidate 3D pose is estimated by means of a monocular 3D human pose estimation algorithm, such as [151, 149]. The 3D poses obtained from each view are then aggregated together by means of a dedicated multi-view fusion module, which aims to combine all the independent single-view predictions in a same 3D pose. Finally, the aggregated 3D pose is further refined in an optimization module based on constrained reprojection error minimization. An overview of the proposed method is illustrated in Figure 4.1.

Single-view 3D Pose Estimation

As a first step, our approach takes as input multiple images captured simultaneously by C cameras with known calibration parameters. Three-dimensional human poses are extracted from 2D images following a top-down approach: (i) a people detector finds all the possible patches containing a person, (ii) for each patch, a CNN predicts the 3D location of the human body joints with respect to the camera reference frame. Our method assumes that all the cameras in the network are calibrated with respect to the same reference frame (e.g., robot base frame), so that the rotation R_k and translation t_k of each camera k with respect to the

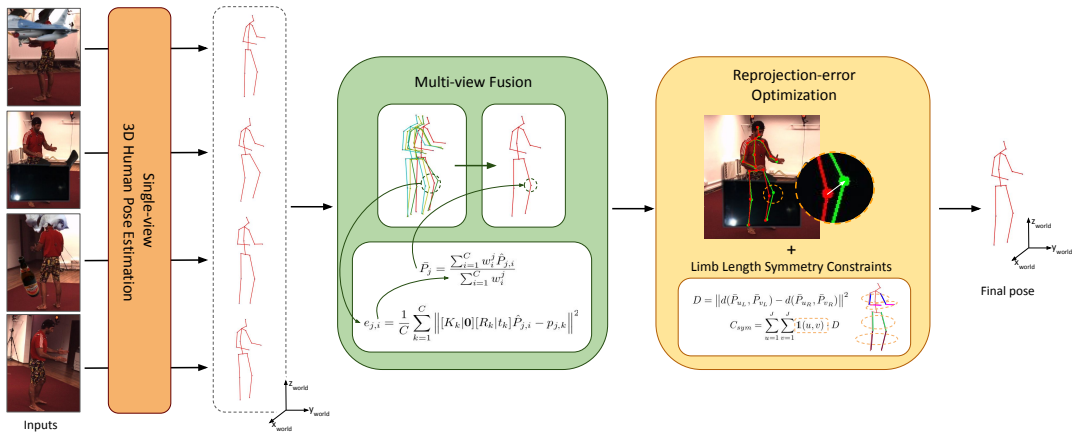


FIGURE 4.1: Overview of the proposed framework for occlusion-aware 3D human pose estimation. The method first predicts a 3D pose from each single view, then fuses information from multiple views, and finally refines the 3D pose through reprojection error minimization.

reference frame are known (see Section 3.4.1). For each view, the final output is a tuple of 3D poses, 2D bounding boxes and confidence scores.

Multi-view Fusion Module

Monocular 3D human pose estimation methods typically provide accurate and plausible body joint predictions when considering relative joint positions. However, they typically struggle in estimating the absolute distance of each body joints from the camera, especially in presence of large body occlusions. Moreover, when portions of the human body are not visible, there exists an ambiguity between distance and size: small objects close to the camera may look similar to large objects farther away from the camera [150], leading to incorrect 3D pose prediction in metric space. In order to alleviate this problem, we propose to leverage information from multiple views, from the intuition that averaging the absolute positions of the single-view estimations should give a good approximation of the true location of the human body.

Consider a network of C cameras and denote as $\{\hat{P}_{j,i}\}$ the set of 3D joints predicted from camera i. The j^{th} joint of the fused skeleton can be expressed as the weighted average of the j^{th} joint for all single-view detections:

$$\bar{P}_j = \frac{\sum_{i=1}^{C} w_i^j \hat{P}_{j,i}}{\sum_{i=1}^{C} w_i^j} \,. \tag{4.1}$$

The use of weights w_i allows to penalize the views framing the subject from a higher distance or the detections with lower confidence. When the human body is partially or heavily occluded, in fact, the 3D pose obtained from monocular 3D pose estimation methods is not fully reliable: if some parts of the body are not visible, the pose estimation algorithm uses clues from the visible parts to estimate the positions of the missing joints. Significant occlusions reduce evidence for the obscured joints' positions, leading to incorrect final predictions compared to the true pose. For instance, if a person's legs are occluded by a table, the model lacks evidence to determine whether the person is standing with crossed, bent or straight legs.

To better incorporate all the available information, we propose the use of per-joint weights: if a 3D joint (e.g. the right ankle) is incorrectly predicted in one view, the error given by projecting that joint over all the other views will be higher with respect to the reprojection error

computed for that same joint correctly predicted from another view.

In particular, given the set $\{\hat{P}_{j,i}\}$ of 3D joints predicted from camera i, a single joint $\hat{P}_{j,i}$ can be easily projected onto the image plane of a camera k by means of the camera intrinsic parameters K_k and extrinsic parameters R_k and t_k. Averaging the reprojection error between the 2D joints and the predictions on each frame $\{p_{j,i}\}$, allows to define the per-joint weight $w_{j,i}$ as follows:

$$w_{j,i} = \frac{1}{e_{j,i}} , \qquad e_{j,i} = \frac{1}{C} \sum_{k=1}^{C} \left\| K_k [R_k | t_k] \hat{P}_{j,i} - p_{j,k} \right\|^2 . \tag{4.2}$$

Optimization Module

The fusion module merges multi-view 3D predictions, leveraging multiple viewpoints to mitigate artifacts in joint positions caused by occluded body parts. However, when significant occlusions occur in multiple camera views, averaging predictions may not suffice, as estimated pose and camera distance can be slightly incorrect in all views. To address this issue, we formulate an optimization problem to refine the fused 3D pose $\bar{P}_j$, minimizing the per-joint reprojection error across all C cameras available:

$$\underset{\bar{P}_j}{\operatorname{argmin}} \sum_{k=1}^{C} \sum_{j=1}^{J} \left\| K_k [R_k | t_k] \bar{P}_{j,i} - p_{j,k} \right\|_2^2 . \tag{4.3}$$

Without additional constraints, the optimization process can change the position of all body joints; however, there are symmetries in the human body that can be exploited to guide optimization. In particular, we design a limb length symmetry constraint considering four pairs of bones: left and right upper arms, lower arms, upper legs and lower legs. For each pair we constrain the limb length to be the same, that is, we enforce left and right limbs to have consistent measures. In particular, we compute the left (L) and right (R) 3D limb lengths as the euclidean distance $d(\hat{P}_i, \bar{P}_j)$ between the corresponding left and right joints. If u and v are a pair of corresponding limb joints, the final cost for the symmetry constraint is given by:

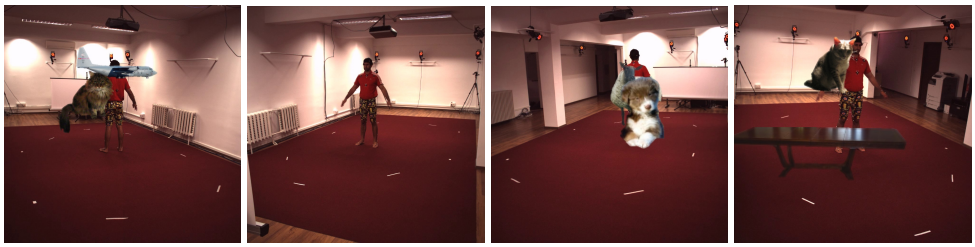
$$C_{sym} = \sum_{u=1}^{J} \sum_{v=1}^{J} \mathbb{1}(u,v) \cdot \left\| d(\bar{P}_{u_L}, \bar{P}_{v_L}) - d(\bar{P}_{u_R}, \bar{P}_{v_R}) \right\|_2^2 , \tag{4.4}$$

$$\text{where } \mathbb{1}(u,v) = \begin{cases} 1 & \text{if } (u,v) \text{ is upper/lower arm, upper/lower leg} \\ 0 & \text{otherwise .} \end{cases}$$

Note that, with respect to previous work [148, 29, 139], we do not require any prior information on human limbs length or pre-computed statistics on datasets' 3D annotations. Finally, the optimization problem is defined as follows:

$$\underset{\bar{P}_j}{\operatorname{argmin}} \sum_{k=1}^{C} \sum_{j=1}^{J} \left\| K_k [R_k | t_k] \bar{P}_{j,i} - p_{j,k} \right\|_2^2 + C_{sym} . \tag{4.5}$$

The optimization is initialized from the fused 3D pose $\bar{P}_j$ obtained by multi-view aggregation and is solved as a global non-linear least-squares problem over all joints. We employ a Levenberg–Marquardt trust-region strategy implemented in the Ceres Solver, combined with an iterative Schur-based linear solver and a robust Cauchy loss function, while enforcing limb-length symmetry constraints across left and right body parts.



(A) (B) (C) (D)

FIGURE 4.2: Sample sequence from the Human3.6M-Occluded dataset, with occlusions on three of four views.

4.1.3 Experiments

In order to evaluate the capabilities of our method to handle significant occlusions and clutter, following the work of [152] and [137], we propose Human3.6M-Occluded, a novel benchmark derived from Human3.6M [140], adding synthetic occlusions on camera views. The robustness and reliability of our method in challenging human-robot environments is then inspected considering data sequences of human-robot collaboration tasks acquired in a real industrial workcell and in a laboratory environment. We finally conduct ablation studies to analyse the behaviour of our framework in presence of few occluded viewpoints and camera synchronization errors, typical in industrial multi-camera setups.

Experimental Settings and Metrics

Human3.6M and Human3.6M-Occluded The Human3.6M dataset is currently one of the largest multi-view datasets providing 3D human pose annotations. It consists of 3.6 million frames captured by 4 synchronized 50 Hz cameras. 3D and 2D keypoint ground truth is obtained by a marker-based motion capture system composed of 10 IR-cameras. The dataset includes 11 different subjects performing 15 daily actions, such as eating, walking, and giving directions. Each scene features only one subject. Following the traditional protocol, subjects S9 and S11 are considered for evaluation. To provide fair comparisons with previous work [30], [142], sequences of subject S9 with incorrect 3D annotations (parts of 'Greeting', 'SittingDown' and 'Waiting') were ignored for final results calculations.

Motivated by the lack of human body occlusions in Human3.6M and of public multi-view human pose estimation datasets including real occlusions in cluttered scenarios, we introduce the novel benchmark Human3.6M-Occluded. This is derived from the Human3.6M dataset by synthetically adding significant body occlusions, by taking objects and animals images from the Pascal VOC 2012 dataset [154] as occluders, as done in [152]. Considering the four available camera views for each scene in the dataset, we partially cover the subject's body on three out of four views. This kind of scenario is very difficult to deal with, as each body joint is not guaranteed to be visible by at least two views. An occluded view includes two random objects overlapping the human bounding box. Object size and location inside the box are chosen at random. Sample images from the Human3.6M-Occluded benchmark are shown in Figure 4.2.

Differently from previous works presenting similar approaches, we make reproducible code available[1] to generate the dataset, to encourage further research on occlusion-aware human pose estimation and to facilitate comparison with other methods.

[1] https://github.com/laurabragagnolo/human3.6m-occluded



(A) Workcell for collaborative draping (B) Workcell for collaborative assembly

FIGURE 4.3: Sample images from the sequences acquired in real human-robot collaboration scenarios. a) Industrial workcell; b) Laboratory workcell.

Human-Robot Workcell To test the robustness of our human pose estimation method in real industrial environments, we conducted experimental validation on data acquired in two different human-robot collaboration scenarios: an industrial workcell developed as use case for the DrapeBot project [155, 112], and a laboratory collaborative workcell. The former has been designed for collaborative transportation of flexible materials and covers an area of about 20 m^2. It was equipped with four Intel RealSense Depth D455 sensors surrounding an ABB industrial manipulator at an average distance of 4 m from the base of the robot. The latter, equipped with four Intel RealSense Depth D455 cameras and a Franka Emika Panda collaborative manipulator, has been designed for the collaborative assembly of furniture, (e.g., wooden chair). The industrial workcell provides a very challenging testing scenario, with a lot of clutter in the scene and very tilted cameras. The laboratory workcell represents a simpler scenario, with less occluders. In both settings cameras are not perfectly synchronized. Sample images extracted from acquired sequences are shown in Figure 4.3.

In all sequences collected, participants were equipped with the Xsens MVN Awinda motion capture system [156] to record whole-body kinematics. The system consists of 17 wearable inertial measurement units (IMUs) placed all over the body, providing an accurate full-body 3D pose not affected by occlusions that has been used as a ground truth information. Since the 3D pose obtained from inertial sensors suffers from drifting errors in the absolute position over long period of time, we correct this effect using the information provided by the multi-camera pose tracking solution OpenPTrack [157], by aligning the inertial-based 3D pose with the 3D position of the the neck joint detected by OpenPTrack, which has been empirically proven to be the most robust to occlusions among all body joints.

Evaluation Metrics The 3D pose estimation accuracy is measured considering the Mean Per Joint Position Error (MPJPE), that is, the Euclidean distance between predicted 3D position and ground truth, for each joint. We consider both the absolute and the root-relative MPJPE, that requires aligning the pelvis joints coordinates of the estimated and ground truth pose. While the absolute error takes into account the accuracy of the global localization of the prediction, the relative error focuses on the accuracy of the human keypoints configuration with respect to the pelvis joint, also known as the root joint. The MPJPE is expressed in millimeters, where lower values represent higher accuracy.

Implementation Details The single-view 3D predictions for multi-camera fusion are obtained using MeTRAbs [151]. The considered model has been trained considering multiple skeleton conventions, e.g. H36M [140], 3DHP [158], SMPL-X [159], and several 3D human

TABLE 4.1: The results of evaluation on the Human3.6M dataset. The table presents the MPJPE relative to pelvis for state-of-the-art multi-view human pose estimation methods.

MPJPE relative to pelvis, mm																
Methods	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
RANSAC (as in [31])	18.7	19.5	19.5	20.9	22.1	17.2	19.6	24.8	31.6	22.5	21.7	22.0	**17.4**	22.3	17.3	21.7
T-Iskakov et al. [30]	20.4	22.6	20.5	19.7	22.1	20.6	19.5	23.0	25.8	33.0	23.0	21.6	20.7	23.7	21.3	22.6
V-Iskakov et al. [30]	19.9	20.0	18.9	18.5	20.5	19.4	18.4	22.1	**22.5**	28.7	21.2	20.8	19.7	22.1	20.2	20.8
Zhang et al. [31]	**17.0**	**18.1**	**17.3**	21.1	**19.1**	**16.4**	18.1	**19.2**	26.3	**19.4**	**19.1**	21.0	16.8	**19.6**	**16.9**	**19.4**
Ma et al. [142]	25.2	27.8	24.8	24.3	27.2	28.8	24.3	25.9	63.4	32.6	27.4	25.1	27.0	25.2	24.7	29.4
Wan et al. [144]	19.5	20.9	19.5	**18.3**	21.1	20.0	**17.9**	21.3	23.9	30.1	21.6	**19.9**	18.9	22.8	19.5	21.1
Ours	25.7	27.2	27.2	27.2	31.4	26.1	29.3	33.8	36.2	29.3	29.7	28.5	25.7	30.6	26.0	29.0

TABLE 4.2: The results of evaluation on the Human3.6M dataset. The table presents the absolute MPJPE for state-of-the-art multi-view human pose estimation methods.

MPJPE absolute, mm																
Methods	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
RANSAC (as in [31])	18.3	19.8	20.0	18.4	22.1	17.3	19.8	23.6	28.1	22.8	23.9	17.9	18.0	21.2	17.6	21.5
T-Iskakov et al. [30]	21.7	23.7	22.2	20.4	26.7	24.2	19.9	22.6	31.2	35.6	26.8	21.2	20.9	24.6	21.1	24.5
V-Iskakov et al. [30]	18.0	**18.3**	**16.5**	**16.1**	**17.4**	18.2	**16.5**	**18.5**	**19.4**	20.1	**18.2**	17.4	**17.2**	**19.2**	**16.6**	**17.9**
Zhang et al. [31]	**17.4**	**18.3**	17.8	17.8	19.4	**16.8**	18.5	19.2	22.1	**19.4**	21.4	**16.7**	17.4	19.6	17.4	19.4
Ma et al. [142]	24.4	26.4	23.4	21.1	25.2	23.2	24.7	33.8	29.8	26.4	26.8	24.2	23.2	26.1	23.3	25.8
He et al. [141]	25.7	27.7	23.7	24.8	26.9	31.4	24.9	26.5	28.8	31.7	28.2	26.4	23.6	28.3	23.5	26.9
Moliner et al. [143]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	26.0
Ours	23.8	24.6	23.4	24.1	26.9	24.0	25.3	28.3	31.7	25.9	26.9	24.2	23.0	27.3	24.2	25.6

pose estimation datasets, including Human3.6M [140], CMU Panoptic [160] and TotalCapture [161]. For each of the experiments presented below, we do not perform any fine-tuning.

Results on Human3.6M and Human3.6M-Occluded

The results of the evaluation of our method and state-of-the-art multi-view human pose estimation algorithms on the Human3.6M dataset is reported in Table 4.1 and Table 4.2. Considering the root-relative MPJPE (4.1), in scenarios with very few occlusions and in which the human body is well visible from all the four viewpoints, our method performs slightly worse with respect to current approaches. For multi-view methods, however, the absolute position error is much more interesting to look at, since it represents the quality of the 3D global localization of the human body prediction. Results for the absolute MPJPE are reported in Table 4.2. Our framework surpasses most recent approaches and gives comparable results to strong baselines such as algebraic triangulation [30]. Looking the state of the art, it is worth noticing that our method and TransFusion [142] yield very similar results, while AdaFuse [31] produces the strongest performance on this benchmark, considering both relative and absolute errors.

The loss of accuracy with respect to the state of the art is mainly due to the fact that competing approaches rely on heavy geometrical constraints, such as triangulation and epipolar lines, to perform 2D feature fusion. When the human body is fully visible and 2D features are very reliable, exploiting geometry gives extremely precise results that, however, poorly adapt to scenarios with occlusions, in which 2D joint predictions are not as reliable. Differently, our method uses information about the geometry of the scene, i.e., rotation and translation between cameras, for the sole 3D joint reprojection and then finds the optimal pose that minimizes reprojection errors on multiple views. This enforces looser geometrical constrains on the final pose, which lead to slightly worse results on the standard Human3.6M with respect to the other methods, but more robust performance in presence of occlusions.

To validate this, we now test our approach in presence of significant occlusions, on the proposed Human3.6M-Occluded benchmark. For comparisons we select AdaFuse [31]

TABLE 4.3: The results of evaluation on the proposed Human3.6M-Occluded benchmark. The table presents the MPJPE relative to pelvis for our method and state-of-the-art multi-view human pose estimation algorithms.

MPJPE relative to pelvis, mm																
Methods	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
RANSAC (as in [31])	302.8	69.7	57.8	48.1	49.2	94.1	67.5	132.4	137.7	58.0	66.3	54.8	38.1	267.4	56.9	97.0
T-Iskakov et al. [30]	69.4	75.1	83.3	65.4	76.2	91.0	106.3	192.9	632.4	117.7	91.9	105.0	49.5	101.9	70.7	134.9
Zhang et al. [31]	**32.5**	41.3	**33.9**	**35.0**	**37.7**	38.0	**36.4**	**50.5**	83.0	47.4	47.9	38.7	**28.9**	**40.3**	37.6	43.5
Ma et al. [142]	48.8	65.8	51.9	48.7	61.7	57.1	62.3	75.1	384.8	224.8	74.1	50.6	122.9	97.6	35.9	97.3
Ours	34.1	**40.6**	36.2	35.7	42.9	**35.9**	40.0	60.8	**68.1**	**42.7**	**41.3**	**35.5**	30.8	43.0	**34.3**	**41.9**

TABLE 4.4: The results of evaluation on the proposed Human3.6M-Occluded benchmark. The table presents the absolute MPJPE for our method and state-of-the-art multi-view human pose estimation algorithms.

MPJPE absolute, mm																
Methods	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
RANSAC (as in [31])	305.5	61.3	51.9	47.2	46.6	130.0	62.2	117.5	117.5	53.7	65.5	55.0	35.2	60.2	51.6	80.7
T-Iskakov et al. [30]	67.1	71.3	80.4	63.4	74.8	88.5	100.9	188.7	611.2	111.7	92.2	110.6	47.1	99.7	69.8	127.4
Zhang et al. [31]	**31.5**	39.5	**32.4**	32.3	**36.3**	36.6	**34.7**	**46.4**	71.7	45.4	47.8	37.0	28.0	**37.8**	35.3	41.1
Ma et al. [142]	48.1	65.2	50.9	47.9	60.8	56.1	61.6	74.1	384.2	224.2	73.4	49.9	122.3	96.5	35.8	96.5
Ours	32.0	**36.9**	32.9	**31.5**	37.9	**32.9**	36.2	53.7	**64.9**	**37.5**	**38.9**	**32.8**	**27.8**	38.5	**31.4**	**37.8**

and TransFusion [142], which have a specific focus on human pose estimation in unconstrained environments with occlusions and provide official codes and training weights on Human3.6M. We also compare with RANSAC (implementation from [31]) and the algebraic triangulation [30], taken as baselines. For this experiment all the models are trained on the original Human3.6M dataset, without any fine-tuning on the Human3.6M-Occluded data.

Results for the relative and absolute MPJPE on this benchmark are shown in Table 4.3 and Table 4.4. Our system provides the best performances with respect to both evaluation metrics, proving its superior capabilities in handling challenging scenes with occlusions with respect to competing methods relying on 2D features and strong geometrical constraints.

Results on the Human-Robot Workcell Scenario

We now inspect the robustness of our human pose estimation approach on real industrial environments. For this, we consider the data sequences collected in the industrial workcell and in our laboratory, presented earlier in this section. Results are shown in Table 4.5 and Table 4.6. Sequences acquired in the industrial workcell include the participation of two different subjects (performing two sequences each) and are numbered from 1 to 4; sequences collected in our laboratory include only one subject and are numbered from 5 to 7.

Our framework demonstrates a great advantage with respect to the other approaches, producing reliable predictions even on such challenging scenes, with an average absolute joint position error of about 14 cm. On the other hand, the performance of the other methods dramatically decreases in the same context, leading to errors of several meters with respect to the ground truth reference, as shown in 4.4 and 4.5. Differently, our framework produces consistently better results on both scenarios, displaying excellent generalization power and adaptability to real multi-camera setups.

While results on Human3.6M-Occluded already confirmed the superior performance of our method in presence of significant occlusions, results on the real human-robot collaboration scenario really highlight the shortcomings of current methods and the true advantage of our framework. Nevertheless, the discrepancy that we observe between our results and those provided by the literature is due to conditions that characterize industrial multi-camera setups, such as camera synchronization errors and large occlusions on all viewpoints. These

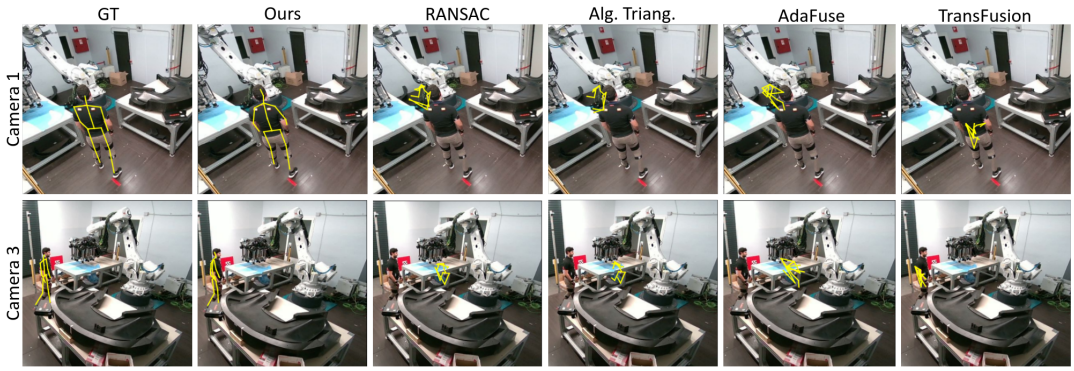


FIGURE 4.4: Qualitative comparison between our approach and the state of the art on the industrial workcell scenario.

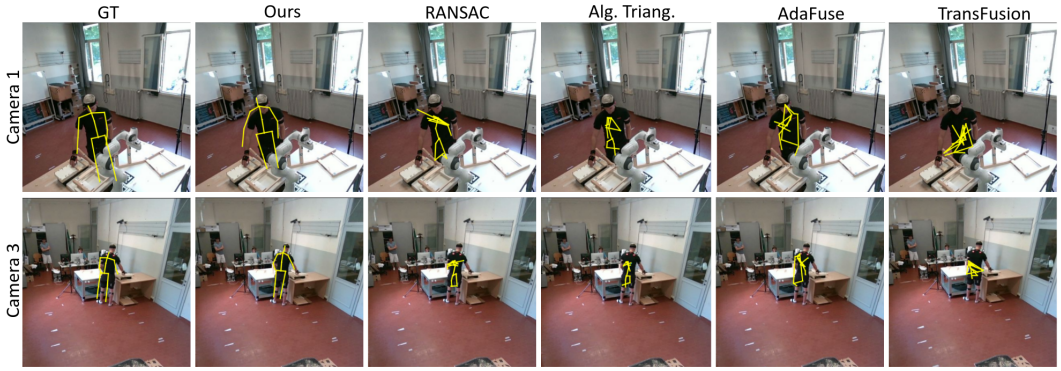


FIGURE 4.5: Qualitative results of our approach and state-of-the-art multi-view methods on the laboratory collaborative workcell.

factors are not considered in public datasets [140, 160], and thus, not handled by most approaches in the literature.

Ablation Studies

In the remainder of this section we consider the Human3.6M and the Human3.6M-Occluded benchmarks to carry out ablation studies on different aspects typical of real industrial scenarios: (i) imperfect synchronization of cameras and (ii) heavy occlusions with few viewpoints available.

TABLE 4.5: The results of evaluation on data sequences collected in real human-robot workcells. The table presents the MPJPE relative to pelvis for our method and state-of-the-art multi-view human pose estimation algorithms.

MPJPE relative to pelvis, mm								
Methods	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6	Seq 7	**Avg**
RANSAC (as in [31])	541.0	265.2	328.7	312.9	261.1	287.7	260.4	322.4
T-Iskakov et al. [30]	439.5	257.5	315.8	307.0	270.7	288.1	257.8	305.2
Zhang et al. [31]	580.7	370.0	402.1	447.0	347.2	344.9	321.7	401.9
Ma et al. [142]	505.7	453.7	446.9	481.4	412.5	430.5	410.9	448.8
Ours	**90.4**	**98.7**	**87.9**	**88.8**	**100.9**	**114.0**	**109.8**	**98.6**

TABLE 4.6: The table presents the absolute MPJPE on data collected in real human-robot workcells for our method and state-of-the-art multi-view human pose estimation algorithms.

MPJPE absolute, mm								
Methods	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6	Seq 7	**Avg**
RANSAC (as in [31])	8981.5	7295.3	8109.3	7336.4	2800.5	2817.2	2813.5	5736.2
T-Iskakov et al. [30]	8784.7	7291.9	8112.4	7327.2	2796.3	2813.9	2808.4	5705.0
Zhang et al. [31]	9000.7	7276.1	8116.3	7308.6	2754.0	2765.7	2773.4	5713.5
Ma et al. [142]	8907.5	7370.7	7296.0	8211.2	2788.6	2785.8	2786.8	5735.2
Ours	**110.7**	**106.1**	**171.7**	**154.8**	**116.7**	**198.1**	**140.9**	**142.7**

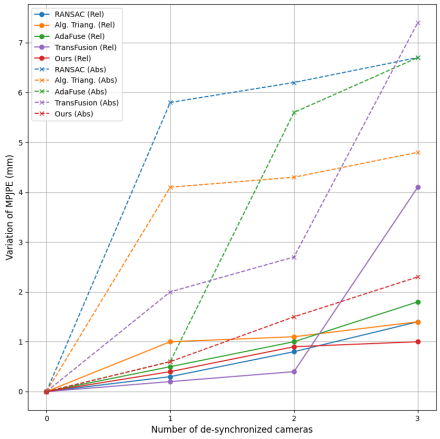


FIGURE 4.6: Variation of relative and absolute MPJPE with increasing number of de-synchronized cameras. Solid lines represent variation of the relative error, dashed lines represent variation of the absolute errors.

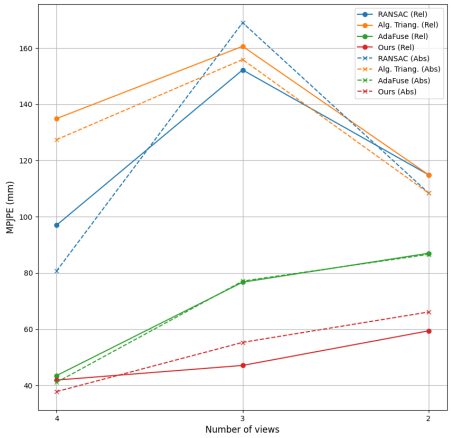


FIGURE 4.7: Relative and absolute MPJPE gradually removing the number of viewpoints, with occlusions. Solid lines represent variation of the relative error, dashed lines represent variation of the absolute errors.

Camera Synchronization While public datasets are acquired in very constrained and controlled environments employing reliable camera synchronization mechanisms, such conditions are hardly met in industrial environments. To assess the advantages of our method, we now isolate the effect of slight errors in camera synchronization on multi-view human pose estimation methods. Considering the Human3.6M dataset, we study the variation of the mean joint position error as we simulate synchronization errors on an increasing number of cameras. At first, three of the four cameras will give the correct frame corresponding to time t while we simulate a synchronization error on the fourth one. The de-synchronized camera will give a frame corresponding to a wrong timestamp t_e, defined according to (4.6).

$$t_e = t + e \;\text{ with }\; e \sim \text{Uniform}(\{-2,-1,+1,+2\}) \tag{4.6}$$

Then, we simulate a synchronization error on two of the four cameras and, finally, on three out of four cameras. Figure 4.6 shows how the absolute and relative MPJPE vary over the three experiments. Our method proves to be the most robust to camera synchronization errors in terms of both relative and absolute MPJPE, which increase of only 1 and 2 mm when three cameras out of four are unsynchronized.

Number of Occluded Viewpoints To study the robustness of multi-view pose estimation approaches in presence of few occluded viewpoints, we consider Human3.6-Occluded and progressively reduce the number of available cameras to three and to two. It is worth noticing

that in Human3.6M-Occluded, for each scene, three out of four views contain occlusions and the occluded cameras are selected at random. This means that when removing a camera, it can happen that for some scenes all views are occluded.

Figure 4.7 shows the relative and absolute MPJPE with reduced number of viewpoints for the compared methods. For a fair comparison, we do not consider TransFusion [142] for this ablation study, as this approach requires to be re-trained when the number of available cameras change. One can easily notice that our framework is the most robust to significant occlusions with few viewpoints, yielding a relative and absolute joint position error of about 6 cm when tested on scenes with only two cameras and occlusions on at least one view.

4.1.4 Discussion

In this section, we proposed a novel multi-view 3D pose fusion approach [48] for robust and reliable 3D human pose estimation in human-robot collaboration scenarios, generally characterized by significant occlusions and limited camera viewpoints. While most methods rely on 2D keypoints triangulation, we aggregate single-view 3D joint predictions, proposing a per-joint weighting depending on the mean reprojection error on all views, to mitigate the effect of incorrect predictions. The fused 3D pose is further refined via reprojection error optimization, introducing limb length symmetry constraints to enhance estimation accuracy.

Our framework was evaluated on challenging scenarios with sever occlusions, considering data sequences from real human-robot collaboration workcells and the proposed benchmark Human3.6M-Occluded. Our method outperforms state-of-the-art techniques on heavy occlusions and exhibits strong generalization capabilities to unseen environments, making it a suitable solution for 3D human pose estimation in challenging human-robot collaboration settings.

It is important to note that the proposed fusion and refinement process operates on a per-frame basis. Single-view 3D pose predictions are assumed to be independent across views within the same time step, and no temporal constraints or filtering are enforced across consecutive frames. Accordingly, the reprojection error is modeled independently for each frame, without assuming temporal correlations in the observation noise.

4.2 Online Action Recognition

With the ability to accurately estimate 3D human pose, even under occlusion as addressed previously, the focus now shifts to interpreting that pose information. As introduced in the first part of Section 4, recognizing human actions becomes crucial for an effective and safe collaboration between humans and robots, especially in industrial scenarios.

Although many works in the literature have proposed highly accurate action recognition systems based on video [162] or skeleton [34] data, these approaches are typically evaluated on short sequences, trimmed to contain a single isolated action [32, 33]. In contrast, real-world applications involve continuous, unsegmented streams where operators perform multiple actions in succession, often without clear temporal boundaries between them. Furthermore, collaborative applications require the robot to provide timely and context-aware responses to human actions in order to deliver meaningful assistance. This necessitates the development of online action recognition techniques that generate reliable predictions based only on past and current observations, without having access to future frames. Industrial applications present further challenges in terms of generality, adaptability, and robustness. Industrial processes often involve highly specific actions and gestures related to particular production workflows that are not represented in standard publicly available datasets. Acquiring and annotating new, dedicated datasets is typically time-consuming and impractical in fast-paced production environments. This underscores the need for recognition systems

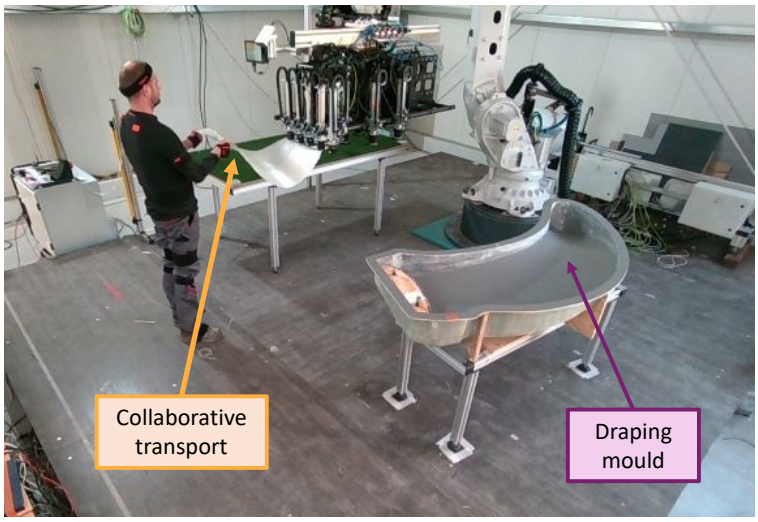


FIGURE 4.8: Example of a collaborative draping task in an industrial setting. The human operator and robot jointly transport the material from a pick-up table to a mould, where it is manually draped. Human actions are continuously monitored throughout the process to trigger and coordinate robot operations.

that are lightweight, easy to train with limited data, and capable of adapting to different processes with minimal effort. Furthermore, these systems must guarantee high accuracy and robustness, as misclassifications or false detections could lead to incorrect robot behaviors, especially when recognized actions serve as direct commands or control signals within the collaborative process. In this section, we propose a simple but effective online action recognition framework based on skeleton data to tackle such challenges.

The framework is composed of two main components: (i) an ensemble of binary classifiers, each specialized in detecting a specific action of interest, and (ii) a causal temporal buffer that accumulates the most recent skeleton data over a fixed number of frames. Unlike conventional sliding-window approaches, which often require window-wise normalization relative to the first frame of each segment, our method integrates a lightweight preprocessing strategy that removes this dependency, enabling efficient and consistent online inference without the need to reprocess frames at each window update. The core of the framework is an ensemble strategy combining multiple classifiers' predictions to improve overall recognition performance. Specifically, we consider a set of binary classifiers where each one is specialized in detecting a specific action of interest, enabling the system to better handle unknown or irrelevant actions—which are frequent in industrial environments. While conventional multiclass models are forced to classify every input into one of the known categories, our ensemble structure naturally rejects unrelated gestures, resulting in more precise and reliable predictions in dynamic, unstructured scenarios. The proposed framework has been validated considering a real industrial application, such as the collaborative draping of carbon fiber parts proposed in the DrapeBot project [155]. In this process, illustrated in Figure 4.8, a human operator and a robot jointly transport multiple carbon fibre plies from a pick-up table to a mould, where the operator manually shapes each layer. Throughout the process, the operator's actions are continuously monitored and used to trigger robot behaviours at key stages. The online action recognition system has been developed using a small dataset of draping-related actions acquired in a laboratory setting, and subsequently evaluated in an industrial environment involving four participants performing the collaborative draping task across multiple trials. The results demonstrated the framework's effectiveness and its ability to generalize from controlled laboratory conditions to a real-world industrial scenario. To

summarise, this work presents the following main contributions: (i) an online action recognition framework based on a ensemble of lightweight binary classifiers designed for ease of training with limited data; (ii) a modified sliding-window approach incorporating a per-frame preprocessing strategy to minimise computational overhead during online inference; and (iii) extensive experimental validation involving multiple participants in both laboratory conditions and a real industrial collaborative draping process, demonstrating the framework's effectiveness and generality. The dataset of draping-related actions acquired for this study is also publicly released to support future research.

4.2.1 Related Work

Human action recognition (HAR) is generally defined as the process of identifying and analyzing the movements of one or several parts of the human body from sensor data [163]. Most works in this field rely on video-based approaches, extracting spatiotemporal features from RGB streams to model motion patterns and recognize actions [162, 164]. These methods typically employ convolutional, recurrent, or transformer-based architectures to model appearance and temporal dependencies in video sequences. For instance, SlowFast [165] uses a dual-pathway design to capture semantic context and motion dynamics at different frame rates, while VideoSwin [166] adopts a hierarchical transformer with shifted 3D windows to effectively model local and global motion patterns. In recent years, skeleton-based action recognition has emerged as a lightweight and effective alternative, particularly suited for applications where robustness to environmental variations is critical [34, 167]. By representing human poses as a sequence of 2D or 3D joint coordinates, skeleton-based approaches offer several advantages over video-based methods, including reduced computational demands and greater resilience to lighting and background changes [168]. Moreover, to further enhance generalization across diverse scenarios, common approaches apply preprocessing techniques that normalize joint coordinates with respect to the subject's position and orientation, typically aligning the skeleton to a reference frame based on the first frame of each sequence [168, 169]. Despite significant progress, the majority of existing HAR methods are designed for trimmed video segments, where each sequence contains a single, well-defined action [32, 33]. This assumption simplifies both model training and evaluation but limits the applicability of such methods in real-world contexts, where actions occur as part of a continuous, unsegmented stream. In practical procedural activities, such as manual assembly [170] or cooking [171], individuals perform sequences of diverse actions in rapid succession, often without clear temporal boundaries between them. To manage this, a common strategy involves applying a sliding window technique over the input sequence, enabling action recognition models trained on trimmed data to operate on continuous data streams [172, 173]. Online action recognition further constrains the problem by requiring frame-by-frame predictions without access to future frames. To address these limitations, recent approaches have explored methods to compensate for the absence of future information. For example, MiniROAD [174] introduces temporal weighting mechanisms that emphasize recent observations within a window, while InfoGCN++ [175] integrates a human motion prediction module to estimate forthcoming poses. Moreover, having only past information available makes it much more difficult to accurately distinguish between very similar or overlapping actions, especially in industrial environments where task-specific gestures not found in common public datasets are generally used. To improve both accuracy and responsiveness, recent works have explored the use of ensemble-based frameworks, where multiple action classifiers are used in parallel aggregating then their predictions. Such classifiers can be implemented with different architectures [176], or with a same architecture trained to focus on different aspects of the motion, such as distinct body parts or specific temporal patterns [132]. In this work,

we extend this concept by proposing an ensemble framework of binary classifiers, each dedicated to recognize a specific action, aimed at improving accuracy and robustness against unexpected actions.

4.2.2 HAR Framework

The proposed framework adopts a skeleton-based action recognition approach, which takes as input a sequence of 3D human poses over time and predicts the action performed at each corresponding instant, based on a predefined set of actions of interest. A schematic representation of the system is shown in Figure 4.9, highlighting the main steps involved. First, skeletal information about the human operator is obtained by means of state-of-the-art 3D pose estimation methods [177, 150] and normalized with respect to the hip joint, obtaining a general representation of his/her movements. A fixed-size buffer is then updated by inserting the new information about the person's pose and discarding the older one, allowing for a small window of the person's latest movements at each instant. Finally, that window is used as a common input to a set of specific binary models for the recognition of individual actions, and the prediction of the model with the highest confidence is chosen as the actual action recognized at the current instant.

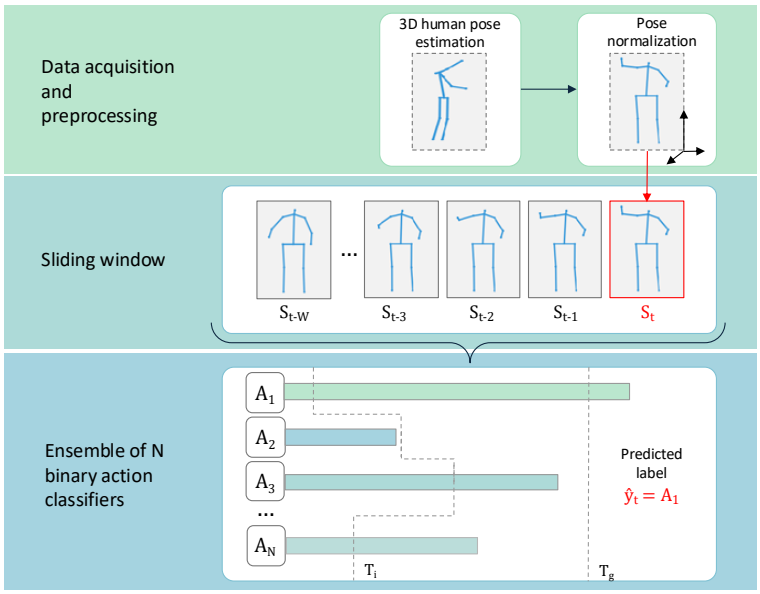


FIGURE 4.9: Framework overview. Normalized skeletons are stored in a fixed-size buffer of length W to capture current and recent movements. The buffer is processed by a set of N lightweight binary classifiers. Classifier predictions are validated using a per-classifier threshold T_i, selecting the final prediction based on the highest score exceeding a global threshold T_g.

Sliding Window and Pose Normalization

The sliding window technique is often adopted to exploit action recognition models designed for trimmed sequences and use them on untrimmed sequences, dividing the input data stream in overlapping temporal segments of fixed length. However, this technique has several limitations in terms of efficiency and robustness that make it inefficient for an online context. The length of the temporal segments is determined by the input required by the particular action recognition network used, which typically uses an input of about 100 frames; in a continuous stream of data such a large temporal segment may contain more than one action performed, confounding the model in its prediction. In addition, the information contained in the time

segment must be pre-processed to make it suitable as input for the network, typically involving various normalization and padding operations. As described in [169], a common practice is to preprocess skeleton sequences by aligning the center of the person in the first frame to the origin and reorienting the skeletons based on the person's orientation in that frame. This dependency on the initial frame requires re-adjusting all skeletons within each sliding window relative to its first frame, resulting in redundant computations across overlapping buffers.

Our framework takes inspiration from the sliding window technique by proposing several modifications to make it more suitable for an online context, namely: (i) the use of a very small window and (ii) skeleton pre-normalization based on single frame information. For each new skeleton frame available, our pre-processing aligns the center point of the skeleton by subtracting the hip joint coordinates from all 3D joints. The skeleton is then rotated so that the spine is parallel with the z-axis in the 3D-Cartesian coordinate system and the segment between the hips joint is parallel with the y-axis. This results in a skeleton that is always front-facing, regardless of the path made by the person, and thus better suited to recognize particular movements required in industrial processing. Normalized skeletons are accumulated in a fixed-size buffer of length W, updated by discarding the oldest frame with each new input. By using a window of length $W = 30$ is possible to retain a short history of recent movements, ensuring recognition is mostly based on relevant motion for a single action.

Ensemble of Binary Classifiers

For the action recognition component of our framework, we adopt a strategy based on an ensemble of binary classifiers, each specialized in recognizing a specific action of interest. This design choice is particularly important in industrial scenarios, where the set of relevant actions is typically narrow (e.g., specific production workflows) and the majority of operator movements are unrelated gestures that should be ignored. In such contexts, inputs may frequently deviate from the distribution observed during training, and multiclass models tend to misclassify these out-of-distribution instances by forcing them into one of the known categories. Our ensemble strategy, instead, improves precision and robustness by reducing false positives in the presence of unexpected or atypical operator behaviors. Each binary classifier is implemented as a lightweight neural network composed of two sequential blocks, each consisting of a 1D convolutional layer, batch normalization, and ReLU activation. The convolutional layers use 64 and 128 filters respectively, both with a kernel size of 7, to capture local temporal patterns within the input skeleton sequences. The feature maps produced by the final convolutional layer are aggregated using a Global Max Pooling operation, followed by a sigmoid activation function to produce the final binary prediction. The ensemble is trained following a "One-vs-All" strategy [178], where each binary classifier is specialized to detect a specific action of interest, treating all other actions as negative samples. To address the class imbalance inherent in this formulation, each classifier is trained using a weighted binary cross-entropy loss defined as:

$$\mathcal{L} = -w^{+} y \log(\hat{y}) - w^{-} (1 - y) \log(1 - \hat{y}) \tag{4.7}$$

where $y \in \{0, 1\}$ is the ground-truth label, $\hat{y}$ is the predicted probability, and w^{+} and w^{-} are the weights associated with the positive and negative classes, respectively. These weights are computed based on the inverse class frequencies within the training set. At inference time, the framework collects the output confidence scores from all binary classifiers. Each score represents the predicted probability of the presence of its corresponding action in the current input window, and the final predicted action is selected as the one with the highest

score. To improve robustness in industrial contexts where unexpected actions may occur, we introduce a dual-threshold mechanism: (i) a global threshold T_g to accept the prediction only if the highest score exceeds T_g, otherwise assigning an unknown class; and (ii) per-classifier thresholds T_i to validate individual predictions before selection. This allows fine-grained control over the sensitivity and precision of individual classifiers, making it possible to tune the ensemble's behavior to the specific application context and operational requirements.

4.2.3 Collaborative Draping HAR Dataset

To demonstrate the effectiveness of the proposed framework in a real industrial setting, a collaborative draping process is considered as a case study: a human operator and a robot work together to transport multiple layers of carbon fiber from a pick-up table to a mould where the material is manually shaped, using human actions as intuitive communication cues to coordinate the process workflow. A set of 8 main actions of interest has been identified as relevant for such process, including both gesture-based commands and natural movements performed by the operator: *Next*, *Point*, *Pick*, *Place*, *Transport*, *Drape*, *Rest*, *Walk*. Gesture-based commands enable intuitive communication with the robot—for example *Next*, performed by raising one hand, signals the robot to proceed to the next operation, while *Point* indicates a request to the robot to inspect a target location on the mould. Natural movements includes short-duration actions marking the start and end of collaborative transport tasks, such as *Pick* and *Place* of the material, as well as long-duration actions like *Transport* (carrying the ply) and *Drape* (manually shaping the material on the mould). In addition, general operator movements such as *Rest* and *Walk* are included to better represent the variability of real industrial workflows. Finally, an *Unknown* class was added to capture unstructured, random movements that occur naturally during the task but are not relevant to the process.

A dedicated dataset, named Collaborative Draping HAR dataset[2], was collected in a laboratory setting equipped with a markerless motion caption system based on OpenPTrack [177] and composed of four RGB-D sensors (Intel Realsense D455) positioned to capture the scene from multiple viewpoints. Unlike OpenPTrack, which primarily relies on depth information, 3D skeletons from each view were extracted using Metrabs [150], a state-of-the-art monocular pose estimator that provides human poses in absolute 3D coordinates. The dataset includes recordings from seven participants (two female and five male). Each participant performed 6 repetitions of each of the 8 actions related to the collaborative draping process, and 18 repetitions of random, unconstrained movements for the unknown class. This resulted in a total of 462 trimmed samples, with each sample consisting of a 3-second sequence of 3D skeletons corresponding to a single action instance. For all samples, the collected data includes: (i) individual 3D skeleton detections from each RGB-D sensor and (ii) a fused 3D skeleton generated via a multi-camera tracking algorithm [177]. This fused representation enhances robustness to occlusions and tracking interruptions. Each skeleton is composed of 15 joints, with 3D coordinates expressed with respect to the camera network reference frame.

4.2.4 Experimental Evaluation

The framework has been validated considering several collaborative draping tasks in both a controlled laboratory environment and a real industrial robotic workcell. The ensemble classifiers have been trained on the Collaborative Draping HAR dataset acquired in laboratory, and then applied to a real industrial process to demonstrate the generalization capabilities of the proposed framework. In both scenarios, skeleton- and video-based action recognition approaches have been considered for comparison.

[2]The Collaborative Draping HAR dataset is publicly available at: https://doi.org/10.5281/zenodo.11352768

Collaborative Draping HAR Dataset

Based on the set of actions relevant to the collaborative draping process described in Section 3.5.2, we implemented our framework as an ensemble of 8 binary classifiers, each specialized in detecting a specific target action. A cross-subject validation protocol was adopted, using data from subjects 1 to 6 for training and subject 7 for testing. Each binary classifier was trained to recognize a given action A_i as the positive class, while treating all other actions in the dataset as negative examples. Given the limited size of the dataset, we augmented the training data by exploiting both the 3D fused skeletons obtained through multi-camera tracking and the individual skeletons extracted from each camera view. Additionally, a data augmentation strategy based on adding random noise to joint positions and applying left-right reflections was employed to further increase the variability of the training samples.

Ensemble performance was evaluated using standard multiclass classification metrics: (i) *class accuracy*, the ratio of correctly classified sequences per class; (ii) *Top-N accuracy*, the percentage of sequences where the correct label appears within the top N predictions; and (iii) *F1-score*, the harmonic mean of precision and recall, accounting for class imbalance. The results are reported in Table 4.7, alongside a comparison with state-of-the-art video-based [165, 166] and skeleton-based [34, 167, 169] action recognition methods. The video-based models were pretrained on the Kinetics-400 [179] dataset, whereas the skeleton-based models were trained from scratch using the augmented data.

TABLE 4.7: Action recognition results on the Collaborative Draping HAR dataset. Best results are highlighted in **bold**, and second-best results are underlined.

Method	Class accuracy								Top-1 Accuracy	Top-3 Accuracy	F1
	Next	Point	Pick	Place	Move	Drape	Rest	Walk			
Slowfast [165]	**100.00**	83.33	45.83	87.50	75.00	66.67	62.50	**100.00**	81.44	95.08	80.17
Swin-video [166]	95.83	58.33	37.50	50.00	95.83	87.50	45.83	**100.00**	70.08	**98.11**	70.93
2s-AGCN [34]	**100.00**	87.50	58.33	**100.00**	87.50	83.33	25.00	91.67	73.48	94.32	73.78
ST-GCN [167]	**100.00**	79.17	50.00	**100.00**	95.83	87.50	41.67	**100.00**	73.86	94.32	74.80
ST-GCN++ [169]	66.67	70.83	75.00	**100.00**	91.67	**100.00**	54.17	**100.00**	80.30	96.21	81.16
Ours	98.18	**94.24**	**93.33**	99.70	**99.09**	97.88	**93.64**	97.27	**84.85**	89.39	**88.18**

The proposed ensemble achieved higher Top-1 accuracy and F1-score compared to state-of-the-art methods, demonstrating its capability to reduce false positives while effectively detecting true actions. In particular, the ensemble demonstrated higher accuracy on challenging action classes such as *Point*, *Pick*, and *Drape*, where most other approaches struggled. This result highlights the framework's capability to capture more effective and discriminative representations of action-specific movements, made possible by leveraging specialized binary classifiers focused on individual classes.

Online Action Recognition in Laboratory Environment

To evaluate the ability of the proposed framework to accurately discriminate actions of interest within a continuous stream of data resembling a real process, the same experimental setup used for dataset acquisition was used to record 8 continuous sequences of various subjects mimicking a collaborative draping process. As illustrated in Figure 4.10, such process begins with the operator approaching the robot and signaling the start of the task by raising an arm (*Next* gesture). The operator then picks up the carbon fiber ply and moves with it (*Transport*) while the robot follows. The ply is then placed on the table (*Place*), and the operator manually shapes it (*Drape*). Finally, the operator steps away and signals the end of the process by raising an arm (*Next*).

All sequences were manually annotated frame by frame with the corresponding action label. To evaluate the online HAR framework, we used frame-wise accuracy, measuring the percentage of correctly classified frames, and segment-based metrics like *F1 score* and *Edit score* to assess action segmentation quality. In this context, an action segment is defined as a contiguous sequence of frames sharing the same label. The F1 score is computed over action segments at multiple IoU thresholds (10%, 25%, and 50%), where a predicted segment is considered correct if its overlap with the ground-truth exceeds the selected threshold. The Edit score evaluates the quality of predicted action sequences by penalizing over-segmentation (fragmented predictions), under-segmentation (merged actions), and incorrect action ordering.

The proposed framework is compared against online action recognition approaches [173, 175], as well as the previous methods for trimmed sequences, adapted for online use via a sliding window of 30 frames. The results, summarized in Table 4.8, consistently show superior performance of the proposed ensemble framework across all considered metrics. In particular, the segment-based metrics reveal a substantial performance gap between our method and the baseline approaches. As reported in Figure 4.10 for some baseline methods, state-of-the-art approaches tend to produce highly fragmented predictions, with numerous short false positive detections that interrupt action segments and degrade temporal coherence. In contrast, our ensemble framework generates much smoother and temporally consistent predictions, with action segments closely aligned in duration and timing with the ground truth. The system successfully recognizes all main actions performed by the operator while exhibiting robustness to minor noisy movements, which tend to confuse other methods and lead to spurious transitions. This improved temporal stability and reduced over-segmentation directly contribute to the higher F1 scores and Edit scores achieved by the ensemble approach. Among the baselines, skeleton-based models adapted with a sliding window approach outperformed dedicated online action recognition methods, revealing a limitation of the latter when applied to new scenarios with limited data. In particular, InfoGCN++, which relies on future pose prediction, performed the worst—likely due to the difficulty of adapting its prediction module to action-specific motion patterns with limited training data.

TABLE 4.8: Online action recognition results on continuous sequences from a lab-simulated collaborative draping task.

Method	**Accuracy**	**F1@10**	**F1@25**	**F1@50**	**Edit-score**
Slowfast [165]	36.72	15.98	14.50	8.88	11.56
Swin-video [166]	34.37	12.67	10.72	8.04	15.16
2s-AGCN [34]	48.72	20.42	18.92	9.61	20.02
ST-GCN [167]	<u>54.59</u>	26.13	<u>24.46</u>	12.40	<u>22.08</u>
ST-GCN++ [169]	51.51	<u>26.96</u>	23.96	<u>14.64</u>	20.80
STGCN-SWMV [173]	18.77	14.09	9.21	2.71	15.19
InfoGCN++ [175]	30.31	3.03	0.00	0.00	14.34
Ours	**64.41**	**53.42**	**48.63**	**39.04**	**45.60**

Online Action Recognition in Industrial Environment

The developed framework was finally deployed and validated in a real industrial scenario, during an actual collaborative draping process involving a robot manipulator as proposed in

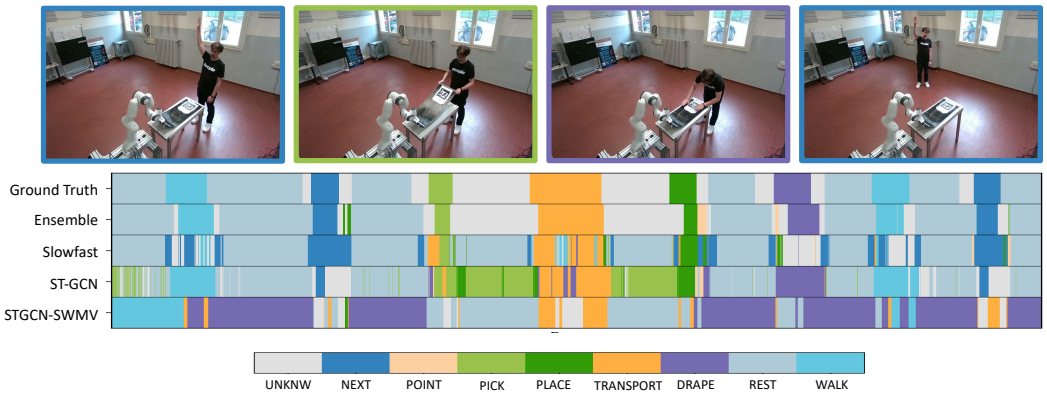


FIGURE 4.10: Qualitative results for a lab-simulated collaborative draping task. Main steps of the process are shown at the top: (i) initial Next gesture to start the process; (ii) Pick and (iii) Drape of the ply; (iv) Next gesture to signal end of the process. Per-frame action predictions are reported for the proposed ensemble framework and baselines, compared with manually annotated ground-truth.

the DrapeBot project [155]. In this setting, human actions are constantly monitored, controlling the robot's behavior in real time based on explicit or implicit gestures. Specifically, the process consists of the following main steps. The human operator raises one hand to trigger the start of the process (*Next*) and approaches the pick-up table to lift one end of the ply (*Pick*). This triggers the collaborative transport phase, with the robot grasping the other end and assisting by following the operator's movements. Once over the mould, the ply is deposited, and the robot moves aside to allow the operator to perform the manual draping (*Drape*). Once finished, the operator signals completion with a final *Next* gesture, restarting the process for the next ply.

An example of the process is shown in Figure 4.11. Compared to the laboratory setup, the industrial environment presents a more dynamic and complex scenario, with additional elements in the scene (e.g., tables, moving robot) increasing the likelihood of occlusions. The overall process duration is also longer, mainly due to extended transport operations and the time required for the robot to approach or move away from the mould. During these waiting periods, the human operator often remains still or performs casual movements, adding further variability to the action recognition task.

The robotic cell was equipped with the same camera system described in Section 3.5.2 and used in the laboratory setup, enabling the acquisition of both images and skeletal data during the execution of 27 trials involving multiple participants. In each trial, the robot's actions were directly triggered by the operator's actions recognized in real time by the proposed framework. The recorded data were subsequently manually annotated to enable a detailed performance evaluation and comparison against baseline methods.

The results are shown in Table 4.9, confirming the higher performance of the ensemble approach over the baselines in both frame-level accuracy and segment-based metrics. In particular, two versions of the ensemble framework were considered: one using standard values for all classifiers' thresholds ($T_i = 0.5$), and a second with manually tuned thresholds for actions that directly trigger robot operations, namely $T_{Next} = 0.8$ and $T_{Pick} = 0.6$. Results show that while both configurations perform similarly overall, the tuned version achieves slight improvements on segment-based metrics, reflecting more precise predictions with fewer false positives. Frame-wise accuracy, however, remains largely unchanged, as minor gains on a few frames have a negligible effect when averaged over all sequences, principally because of the large number of frames in each sequence (about 3000 frames). Compared to the laboratory scenario, a general decline in segment-based metrics is observed, mainly due to the increased difficulty in recognizing long, continuous action segments. As shown

in Figure 4.11, while the models generally manage to identify most of the actions, longer actions such as *Transport* are often fragmented by false predictions, which penalizes both the F1 score and the Edit score. This behavior can be attributed to several factors: slight variations in the way participants perform certain actions (e.g., transporting the ply at a higher position than in the training data), noisier skeleton tracking caused by occlusions, and more unpredictable movements by the operator while waiting for the robot to complete its tasks.

Among the baselines, ST-GCN adapted with a sliding window mechanism proves to be the best approach, successfully recognizing many relevant actions in Figure 4.11. In contrast, both online and video-based approaches struggle to recognize short actions such as *Pick* and *Place*. Especially online methods demonstrate insufficient performance for a real application, failing to detect any of the main robot interaction actions (*Next*, *Pick*) and misclassifying them as longer and more generic actions (*Walk*, *Transport*), as shown in Figure 4.11. This highlights a key limitation of the online methods considered, which tend to require large amounts of data to reliably distinguish between actions. In industrial settings, where collecting extensive datasets is often impractical, the proposed ensemble framework offers a practical alternative, being easy to train on limited data and adaptable through its reusable binary classifiers and thresholds.

TABLE 4.9: Online action recognition results on continuous sequences from a real industrial collaborative draping task.

Method	**Accuracy**	**F1@10**	**F1@25**	**F1@50**	**Edit-score**
Slowfast [165]	35.73	2.31	1.56	0.87	2.11
Swin-video [166]	47.25	4.86	2.89	1.12	5.34
2s-AGCN [34]	74.10	8.45	5.21	1.58	7.67
ST-GCN [167]	<u>74.70</u>	<u>11.46</u>	<u>9.27</u>	<u>3.19</u>	10.28
ST-GCN++ [169]	71.77	9.93	7.32	2.39	8.36
STGCN-SWMV [173]	35.75	6.92	4.72	2.52	10.80
InfoGCN++ [175]	24.02	8.57	3.00	0.86	<u>23.81</u>
Ours - with T_i fixed	76.98	28.54	26.67	14.46	26.12
Ours - with T_i tuned	**76.98**	**28.57**	**26.69**	**14.47**	**26.14**

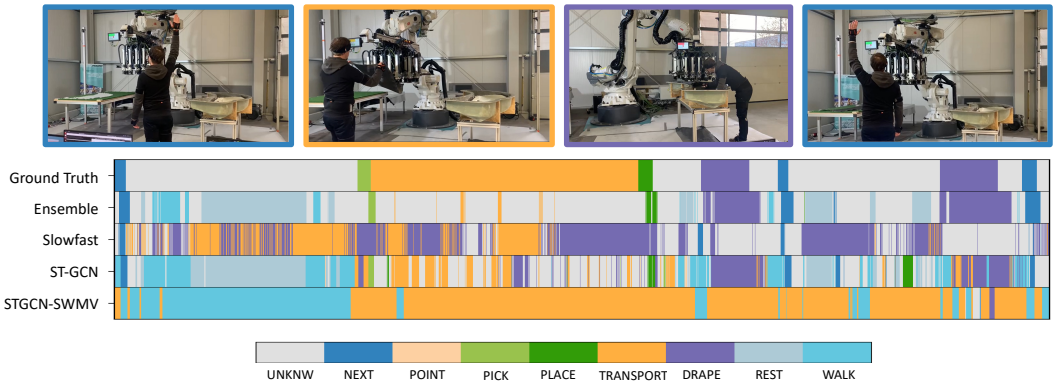


FIGURE 4.11: Qualitative results for a real industrial collaborative draping task. Main steps of the process are shown at the top: (i) initial Next gesture to start the process; (ii) collaborative transport and (iii) manual draping of the ply; (iv) Next gesture to signal end of the process. Per-frame action predictions are reported for the proposed ensemble framework and baselines, compared with manually annotated ground-truth.

Runtime Analysis

Since industrial processes typically impose strict response time requirements, we compared the proposed framework and baselines also in terms of inference time and action recognition delay. Given a ground-truth action segment a_{gt} of length H starting at instant t_{gt}, the delay of a predicted action segment a_p starting at instant t_p, is defined as $d = |t_p - t_{gt}|$. We also considered latency, defined as $l = d/H$, to quantify the fraction of an action's duration that must elapse before it is recognized [173]. Due to the high number of false positives and missed detections from the baseline models, for better clarity the recognition delay analysis was restricted to the only *Next* action, which demands particularly fast and reliable recognition as it directly triggers robot operations.

As reported in Table 4.10, the ensemble-based framework achieved lower inference time and fewer unrecognized actions (higher recall), thanks to the use of multiple lightweight binary classifiers running in parallel. In general, skeleton-based models confirm their suitability for real industrial applications, offering faster inference times than video-based models.

TABLE 4.10: Action recognition performance in terms of runtime and recognition delay on the industrial scenario.

Method	**Inference Time [s]**	**Recall ↑**	**Delay [s] ↓**	**Latency ↓**
Slowfast [165]	1.2506	<u>90.12</u>	<u>0.40</u>	<u>0.30</u>
Swin-video [166]	1.2227	32.10	0.60	0.46
2s-AGCN [34]	0.0539	60.49	0.83	0.65
ST-GCN [167]	0.0604	41.28	0.42	0.32
ST-GCN++ [169]	0.5524	79.01	0.84	0.64
STGCN-SWMV [173]	**0.0036**	0.00	-	-
InfoGCN++ [175]	0.0965	0.00	-	-
Ours	<u>0.0318</u>	**98.77**	**0.39**	**0.29**

In terms of delay and latency, our framework performs very well, detecting actions that require rapid response—like *Next*— with an average delay of 0.39 seconds and a latency of 29%, meaning the action is typically recognized well before the gesture is completed. Interestingly, none of the evaluated online action recognition methods successfully detected the *Next* action in the industrial scenario data, as reported in Figure 4.11. This highlights a significant limitation of such approaches in generalizing to scenarios where available data is scarce or entirely absent. In contrast, our framework can be efficiently trained on a small set of lab-collected data and immediately deployed in a real industrial environment. Experiments were performed on a PC equipped with an Intel Core i7-1180 16-core CPU, 16 GB of DDR4 RAM and a Nvidia GeForce RTX 3060 GPU with 6 GB.

4.2.5 Discussion

This work presented a lightweight and adaptable framework for online human action recognition in industrial environments. By combining an ensemble of binary classifiers with a computationally efficient sliding-window strategy, our approach enables accurate real-time action recognition while requiring minimal training data. The use of dedicated classifiers for individual actions improves robustness to irrelevant or unpredictable movements, a key

advantage in dynamic industrial settings. Validated in both laboratory and industrial environments, the framework showed strong generalization across scenarios and outperformed state-of-the-art baselines, particularly for time-critical actions triggering robot operations. Its modular design and per-classifier thresholds allow easy tuning for precision and robustness, making it a practical, adaptable solution for industrial applications requiring fast and reliable action recognition.

Chapter 5

Robot-Object Interaction

Having established the robot's spatial self-awareness (Chapter 3) and its ability to understand human partners (Chapter 4), we now turn to the final pillar supporting embodied intelligence: purposeful interaction with the physical world. While understanding humans is vital for collaboration, true robotic autonomy demands the capability to engage effectively with the broader environment, particularly with the objects it contains. This involves not just executing pre-programmed tasks but adapting actions to novel objects, dynamic configurations, and unpredictable changes.

Imagine a logistics robot encountering a slightly different shaped package than usual, how does it adjust its grasp? Or consider an assistive robot in a home needing to fetch an item partially obscured by clutter, how does it reason about the object's properties and plan its manipulation? Similarly, picture an autonomous drone inspecting a construction site where vehicles are moving, how does it build an accurate map while also tracking and understanding these dynamic elements? These scenarios highlight the critical need for robots that can generalize skills and perceive dynamically.

This chapter delves into this challenge of adaptive robot-object interaction, exploring solutions within two complementary yet fundamental robotic domains: object manipulation by robotic arms and active exploration by mobile platforms.

In the context of manipulation, achieving generalization from limited human demonstrations remains a key hurdle. How can a robot learn a task like "insert peg A into slot B" from a few examples and then successfully perform it with a slightly different peg C and slot D, without requiring extensive retraining? We address this by proposing a framework that integrates photorealistic 3D scene reconstruction (using Gaussian Splatting [35]), powerful visual understanding from foundation models [36], and the validation capabilities of physics-based simulation. By reasoning about scene geometry and leveraging equivariant transformations, our approach enables robots to adapt demonstrated skills to unseen objects and configurations with few demonstrations available.

Turning to mobile robots, the challenge shifts to exploration and mapping in dynamic environments. While navigating static scenes is well-studied [38, 39, 40], the real world is rarely static. How can a robot efficiently explore an unknown area while simultaneously building accurate models of objects that are moving within it? Addressing this gap, we introduce an active mapping framework. This system utilizes the representational power of Gaussian Splatting and leverages Fisher Information to select information-rich viewpoints [41]. Crucially, it incorporates motion prediction to anticipate object dynamics, enabling proactive trajectory planning to efficiently reconstruct moving targets alongside the environment.

Together, these contributions push towards robots that not only perceive and understand their surroundings but can also act adaptively and effectively upon the objects within them, paving the way for more autonomous and versatile robotic systems.

5.1 Dream to Manipulate: Compositional World Models Empowering Robot Imitation Learning with Imagination

Imitation learning offers a significant paradigm for teaching robots new skills by leveraging human demonstrations, enabling flexible adaptation and task expansion [180, 181, 182, 183]. However, the practical application of imitation learning is often hindered by its significant data requirements. Learning robust, generalizable policies typically demands numerous demonstrations [182], yet acquiring extensive real-world data through methods like teleoperation is labor-intensive and time-consuming.

Furthermore, augmenting the limited available real data presents its own challenges in robotics. Naive data augmentation techniques often fail because they disregard the physical causality inherent in manipulation tasks; simply perturbing trajectories does not guarantee physically plausible outcomes [184].

This section proposes an innovative approach to alleviate the data bottleneck in imitation learning by employing world models. These models learn internal representations of the environment dynamics, allowing the agent to simulate or "imagine" the consequences of its actions. By generating diverse, causally consistent trajectories in imagination, world models can effectively augment sparse real demonstration datasets.

Specifically, we introduce a novel world model architecture that integrates a physics simulator with the high-fidelity rendering capabilities of Gaussian Splatting [35]. The physics engine ensures that imagined interactions adhere to physical laws, preserving causal consistency. Simultaneously, Gaussian Splatting provides an explicit 3D representation capable of rendering photorealistic images from novel viewpoints. This combination is crucial for bridging the reality gap: the generated data is not only physically plausible but also visually highly realistic, minimizing the domain shift between real and imagined experiences.

We demonstrate that leveraging this world model to generate supplementary training data significantly enhances the performance of imitation learning agents, particularly in few-shot scenarios where only a handful of real demonstrations are available. The experiments confirm that our physically-grounded, photorealistic world model enables more data-efficient learning of manipulation skills.

Part of the content of this section is based on:

[46] Leonardo Barcellona, Andrii Zadaianchuk, **Davide Allegro**, Samuele Papa, Stefano Ghidoni, Efstratios Gavves, "Dream to Manipulate: Compositional World Models Empowering Robot Imitation Learning with Imagination", In: The Thirteenth International Conference on Learning Representations ICLR 2025, Singapore.

5.1.1 A Compositional World Model for Robotic Manipulation

Designing an effective world model specifically for robotic manipulation presents unique challenges. Beyond simply predicting future states, such a model must provide a visually realistic representation of the environment, accurately simulate physical dynamics to ensure causal consistency, and allow the agent to interactively query the outcomes of potential actions [185, 186, 187, 188]. Current approaches often face trade-offs: large generative models trained on internet-scale video can achieve photorealism but may hallucinate physically implausible interactions outside their training data [188, 189], while physics-based simulations guarantee consistency but often lack visual fidelity or require costly manual scene creation [190].

Based on these requirements, we identify three key principles for a successful manipulation world model:

- A compositional framework that supports the creation of new and valid combinations [189];

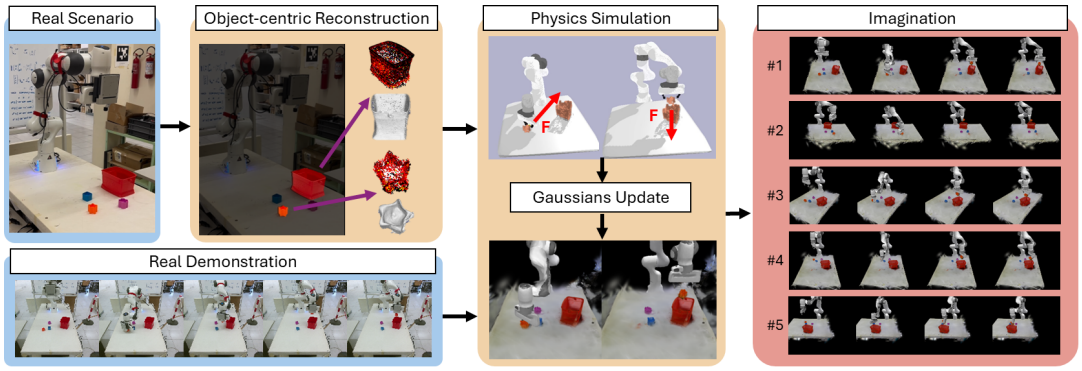


FIGURE 5.1: Overview of imagination with DREMA, which builds a compositional manipulation world model from environment images using object-centric Gaussian Splatting to generate novel demonstrations by transforming real ones.

- Object-centric representation, as objects play a crucial role in the majority of tasks [191, 192];
- Ease of manipulation to guarantee that actions and their outcomes remain controllable [190].

To embody these principles, we introduce DREMA, our compositional world model framework, illustrated in Figure 5.1. DREMA constructs its representation by first performing object-centric 3D reconstruction using Gaussian Splatting (GS) [35]. Foundational models are employed to segment the scene and group the optimized Gaussians into distinct object representations, achieving both object-centricity and high visual fidelity. Subsequently, this compositional GS scene is embedded into a physics simulator (PyBullet [193]), linking each object's visual representation (its Gaussians) to a corresponding rigid body simulator entity. This hybrid GS-Physics approach ensures both manipulability and causal consistency.

This structure allows robot policies to be executed within the DREMA environment, generating physically valid outcomes while rendering visually realistic observations closely mimicking the real world. Crucially, DREMA serves as a powerful tool for imagination-based data generation to augment sparse imitation learning datasets. We demonstrate that applying techniques like equivariant transformations to initial trajectories within the DREMA model allows the creation of diverse, high-quality, and physically plausible new demonstrations. These generated trajectories augment the original limited dataset, enhancing policy generalization even in few-shot scenarios, similar to how humans use imagination.

5.1.2 Related Work

World Models

World models were originally conceived to improve policy learning in reinforcement learning by predicting the consequences of actions on the environment [194, 185, 195, 196]. State-of-the-art approaches often leverage generative models to build these world models [197, 198, 199, 200]. While initially applied to game simulations [194, 185] and navigation tasks [198, 197], world models are now being extended to robotics [199, 201, 189]. For example, UniSim [199] uses diffusion to generate possible future frames conditioned on robot actions. DayDreamer [201] demonstrates that skills such as locomotion and manipulation can be learned using Dreamer [185], while RoboDreamer [189] uses compositionality to generate novel actions. The main limitation of these approaches is their inability to comprehensively capture the motion dynamics and physics specific to the robot's environment, particularly when faced with scenarios outside their training domain [199]. By contrast, we

propose a compositional manipulation world model that integrates cutting-edge photorealism with Gaussian Splats and realistic dynamics with physics engines, showing we can tailor them for efficient imitation learning in real-world scenarios.

Simulation-based Reconstructions

Thanks to recent advances in reconstruction [202] and in physics simulations [203, 204], in the last year many have proposed to combine both for realistic simulations of actual environments [205, 206, 207, 208, 37, 209, 210]. Particle-based simulators are capable of modeling more complex dynamics than traditional rigid body simulators [204], making them beneficial for robotics [211]. Recently, neural radiance fields have been applied to implicitly learn dynamic systems [212, 213]. The introduction of Gaussian Splatting [202] has opened avenues for novel approaches that bridge these simulation techniques with high-quality rendering methods [205, 206]. However, they are not compositional [205] or object-centric [205, 206]. PEGASUS [207] combines Gaussian representations with mesh simulations to improve the 6D pose estimation, while [37] and [208] show that Gaussian representations can be adapted to track moving objects. Both approaches do not allow agents to predict future states, rather focusing more on tracking in observed past trajectories. In contrast, we empirically demonstrate that our approach predicts useful estimation of future states. While simulating a replica of the environment can benefit policy evaluation [190], two recent studies showed that scene reconstruction can enhance robot policies [209, 210]. [209] reconstruct the environment to deform robot trajectories, while our object-centric approach focuses on the interaction between objects and the agent. Although being close to our approach, [210] requires to *manually model* the object-centric representation and needs adaptation between the simulation and the real world. Instead, we leverage the zero-shot capabilities of foundational models [36, 214, 215] to automatically create a compositional world model and show the effectiveness of using transformed demonstrations for task learning without requiring extensive interaction with the world model.

Imitation Learning and Data Augmentation

Imitation learning involves acquiring task knowledge from a dataset of demonstrations [216]. Vision-based imitation learning focuses on deriving policies directly from RGB or RGB-D images [180, 9, 181, 217]. PerAct [9], for instance, voxelizes the environment to infer actions in 3D. However, despite advancements in generalization, many approaches [218, 217] still require hundreds of demonstrations to learn robust policies. MIRA [219] employs NeRF's novel view synthesis to simulate new camera views of the environment, while [220] applied Gaussian Splatting for action inference. In contrast, our approach leverages Gaussian Splatting to generate not only novel views but also new environmental configurations, enhancing generalization.

Data augmentation is an established technique [73, 75]. However, its application in robotics is challenging, as altering robot actions can unpredictably affect environments and tasks [184]. Despite this, [221] demonstrated that even simple augmentations can improve robot policies. Approaches such as generating counterfactual transitions [184] and applying invariant transformations [222] enhance learning by expanding training datasets. However, these methods primarily recombine existing trajectories rather than generate entirely new configurations. In contrast, our approach imagines novel equivariant configurations of objects and actions, enabling more diverse and effective data augmentation. A related method, MimicGen [223], uses object-centric segments for compositional data generation but requires multiple demonstrations. Our method achieves comparable benefits while working effectively with as few as one demonstration. Whereas some authors argue that imagining various

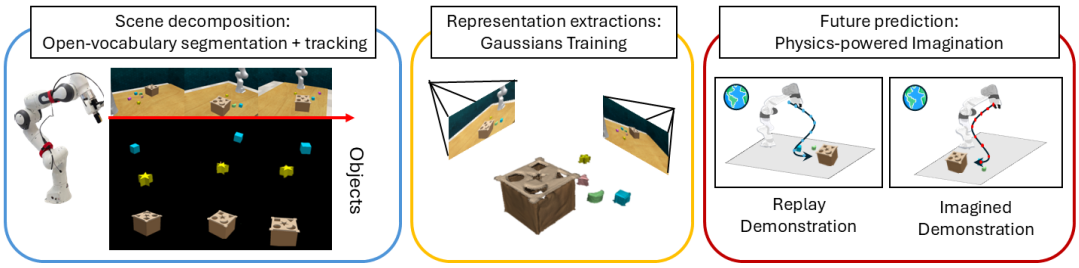


FIGURE 5.2: Steps to create the compositional world model with DREMA: observation of the environment and scene decomposition, representation extraction and future predictions.

object configurations is unrealistic [180], we demonstrate that this is achievable in real-world scenarios.

5.1.3 Dream-to-Manipulate with Gaussian Splats & Physics Engines

We want to construct a *compositional manipulation world model* $\mathcal{M}$, that is (i) explicitly compositional, (ii) object-centric, and (iii) controllable, allowing the agent to control all the objects inside $\mathcal{M}$. The model $\mathcal{M} = (\mathcal{O}_t, \mathcal{A}_t, D, \mathcal{T})$ comprises a set of K object assets in 3D, $\mathcal{O}_t = \{o_{k,t}\}_{k=1}^{K}$ at time t; the model of the agent $\mathcal{A}_t$ that manipulates the objects in the world; the dynamics D of the world in the form of an operator function, $(\mathcal{O}_{t+1}, \mathcal{A}_{t+1}) = D \cdot (\mathcal{O}_t, \mathcal{A}_t)$, which —given the actions of the agent $\mathcal{A}$ at time t— transforms the states of object assets $\mathcal{O}_t$ and agent's state $\mathcal{A}_t$; the manipulation tasks $\mathcal{T}$ expected from the agent. For clarity, we drop the time index t from $o_{k,t}$ when this can be inferred from the context. The input to the algorithm that constructs the compositional manipulation world model is simply an RGB-D video $\mathbf{X} = (x_1, ..., x_N)$ of N frames.

In the following, we first discuss our compositional 3D scene representation based on Gaussian Splats, which forms the basis for our world model. Afterward, we detail how we obtain each component of the world model given our 3D scene representation.

Representing 3D Objects with Gaussian Splatting

Central to the compositional manipulation world model is the inverse graphics function $\mathcal{G} = h(\mathbf{X})$ using Gaussian Splatting [202]. The inverse graphics function returns a 3D representation of the world $\mathcal{G}$ given a sequence of n images $\mathbf{X} = (x_1, ..., x_n)$, collected as a video [202]. The key idea behind Gaussian Splatting is to ground many Gaussian blobs $g = (p, r, s, \alpha, c), \forall g \in \mathcal{G}$ physically in the 3D space. Per Gaussian, we have $p = [p_x, p_y, p_z]$ as its center , its orientation r and scale $s = (s_u, s_v)$ (the covariance), the encoded color c, and the opacity α of the color. That is, given any arbitrary 3D location and angle in space (corresponding to a 2D camera plane), Gaussian Splats can render a projected 2D image that resembles the true observed object (or scene). For rendering the appearance onto the 2D image, one simply α-blends the colors of the Gaussians along each of the rays that connect the 3D point to the points on the 2D projected image. In the end, the Gaussian Splatting algorithm optimizes the Gaussian parameters in g to render the objects and the scene photorealistically for both known and unknown 3D locations (named novel view synthesis). Gaussian Splats have taken the graphics community by storm [224, 205, 207] not only for their exceptional photorealism but also for their real-time computation. For further details, we refer to the seminal work of [202].

Object-centric Gaussian Splats. Gaussian Splats generate photorealistic renderings of the 3D scene that are unstructured in terms of object semantics. Instead, we are interested in

object-centric representations. To obtain object-centric Gaussian Splats, together with the input 2D images in $\mathbf{X}$ for our K objects in our world, we introduce sets of segmentation masks, $\mathbf{Y}_k \in \mathcal{Y}, k = 1, \ldots, K$. We consider k to be varying so that not constrain in advance the number of objects in our world. Each object-set of segmentations $\mathbf{Y}_k = \left(y_{1,k}, ..., y_{n,k}\right)$ contains the 2D masks for the k-th object in all N images in $\mathbf{X}$. With segmentation masks $\mathbf{Y}_k = \left(y_{1,k}, ..., y_{n,k}\right)$ for each object k and for all n frames, we can simply mask the respective video frames, $y_{n,k} \odot x_{n,k}$, to obtain the object-centric instances, which we can then feed to the Gaussian Splatting. Since in our setting the robot is also equipped with depth sensors, we further strengthen the original Gaussian Splatting optimization by including the depth measurements $x^{\text{depth}} \in \mathbf{X}_{\text{depth}}$ for additional supervision, $\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{n}}\mathcal{L}_{\text{n}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}$.

Here $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_n$ are the reconstruction and normal losses defined in the original 2DGS. The extra depth regularization, $\mathcal{L}_{\text{depth}} = \|\hat{x}^{\text{depth}}_{GS} - x^{\text{depth}}\|_1$, uses the ℓ_1 norm to penalize Gaussians that have outlier depth. We set the hyperparameters λ_{normal} and λ_{depth} for all experiments empirically as in [224]. Our object-centric inverse graphics function takes the form $\mathcal{G}_k = h\left(\mathbf{X}, \mathbf{X}_{\text{depth}}, \mathbf{Y}_k\right)$.

Deformable and articulated objects. It is important to note that since the standard Gaussian Splats only work with (static) scenes, we cannot account for deformable or articulated objects. An exciting future direction is to bootstrap on the significant progress in Dynamic Gaussian Splatting [225] to automatically model also articulation and deformation.

Next, we describe how to decompose the scene into k sets of regions defined by segmentation masks $\mathcal{Y}$ to obtain object assets and model dynamics.

Decomposing the Scene into Object-centric Regions

Decomposing the high-dimensional scene representation into objects [226, 227, 228] that are independently controllable greatly simplifies control and the manipulation of the dynamics in the environment [229, 230]. Inspired by recent work on embeddable 3D assets [231, 228], we ideally want to learn Gaussian Splats that belong to the same object. We take advantage of the zero-shot capabilities of open-vocabulary tracking models [215, 232, 128] to ensure that we obtain consistent predictions. Our compositional world model uses the DEVA open-vocabulary tracker to extract the objects and their segmentation masks $\mathcal{Y}_k$ in the video $\mathcal{X}$ of the scene. Since DEVA discovers objects given prompts, we use general and minimal prompts: `object` for all the objects and `table` for segmenting the table of our table-top robot. Having computed the segmentation masks $\mathcal{Y}_k = \left(y_{1,k}, ..., y_{n,k}\right)$ object k and for all n frames, we can learn object-centric Gaussian Splats as described in 5.1.3.

Advances in the object-centric [233, 234, 228] and open-vocabulary [128, 235] literature can be seamlessly integrated into our compositional manipulation world model.

Modelling Robot-Object Dynamics with Object Assets

To be able to predict the consequences of the actions of the robot, we must model the dynamics D. We can directly use external physics engines [236, 203] for the dynamic operator D. For our model, we rely on `PyBullet` [203] for its simplicity.

While object-centric Gaussian Splat representations $\mathcal{G}_k$ are good for photorealistic renderings of objects, they are not designed for physics simulations. Instead, physics engines —including `PyBullet`— typically operate on object meshes. We thus need to convert $\mathcal{G}_k$ into object mesh grids, $M_k = \left(\mu_{k,\cdot}, \rho_{k,\cdot}, u_{k,\cdot}\right)$, with mesh centers $\mu_{k,\cdot}$, orientations $\rho_{k,\cdot}$, and appearances $u_{k,\cdot}$, to obtain our world object assets, $o_k = \left(\mathcal{G}_k, M_k, \omega\right)$. With ω we explicitly denote the relevant physical parameters of the world, such as the masses of objects or the friction coefficient. In the robot simulator, these are in the URDF files [237]. During learning

the policies, the parameters ω can be either given or inferred [238]. The physics engine acts as an operator on the positional and orientation parameters of the mesh grid of all objects in the world, that is

$$\begin{pmatrix} \mu_{\cdot,t+1} \\ \rho_{\cdot,t+1} \end{pmatrix} = \begin{pmatrix} \mu_{\cdot,t} + \Delta\mu_{1\ldots K} \\ \Delta\rho_{1\ldots K} \cdot \rho_{\cdot,t} \end{pmatrix}, \tag{5.1}$$

using standard Newtonian mechanics as the dynamics operator. Note that any object in the world, including the robot arm, can interact with any other object. This is why in 5.1 we drop the object subscript k, and why in the differential Δ we add all objects $1 \ldots K$. Knowing the forces that the robot exerts on the environment at any moment, our compositional manipulation world model can easily *imagine* the future states of the world and the consequences of the robot's actions.

Finally, we apply the change in mesh center $\Delta\mu_k$ and mesh orientation $\Delta\rho^k$ to all Gaussians $g \in \mathcal{G}_k$ by updating their centers $p' = p + \Delta\mu_k$ and their orientations $r' = \Delta\rho_k \cdot r$.

Modelling the agent. In the above, we focused on the objects or our world. The robot is also an object that is self-visible, thus it can be just as easily represented as an object-centric mesh grid from the corresponding Gaussian Splat reconstruction. A difference of the robot arm compared to other objects is that it is articulated. However, the robot joints are known in advance, and we can use their associated segmentation masks to link the different robot parts, ensuring accurate Gaussian representation for rendering purposes. Regarding the dynamics of the robot arm, our assumption is that the robot can model its own dynamics perfectly, similarly as humans have an accurate internal model [239]. This is a reasonable assumption, since most manipulation robots have the corresponding URDF file, allowing for the instantiation of a simulator with the robot model.

World model verification from demonstrations. So far we assumed that the physical parameters are fixed. With this assumption, we cannot guarantee that the manipulation world model will lead to correct predictions for arbitrary actions, as errors in the Gaussian Splat reconstructions and their mesh grids, or misalignment in the physical parameters, could result in cumulative errors when applying the dynamics. That said, our undertaking in this work is that by adopting explicitly grounded appearance and dynamics models for the world and its objects, we can still obtain a useful model. Suppose that the model predictions for replaying the sequence of actions in the demonstration result in a final state similar to the final state of the demonstration. In that case, we can still use the world model as an interactive and compositional representation of the demonstration itself. In the next section, we show that such an interactive representation allows us to augment the original demonstrations with imagined ones to train a more robust imitation learning policy.

Engineering Considerations for Real-World Deployment

The proposed compositional manipulation world model is a significant undertaking, requiring extensive engineering to ultimately be effective with real-world robots. However, one of the key advantages of this approach is that it enables *a real robot* to learn novel policies using only a single example. While Gaussian Splatting is primarily designed for novel view synthesis, we observed unexpected behavior at low resolutions, leading to incorrect action predictions by the robot. Additionally, depth information rendered was inaccurate, as the original Gaussian Splatting method does not ensure that Gaussians align with surfaces. To address this, we implemented 2DGS Gaussian Splatting [224]. For successful real-world

deployment of DREMA, properly aligning the reconstruction and the simulator is crucial, which we achieved using a calibrated camera mounted on the robot [44]. Also, estimating physical parameters is a complex field in its own right [240, 241, 242, 243], and methods such as ASID [238] or AdaptSim [244] can be applied for this purpose. In this paper, for simplicity we pick constant physical parameters that seemed to work well in the validation sets.

5.1.4 Equivariant Transformations for Imitation Learning

In computer vision, many data augmentations preserve the semantic information of an image, enabling robust recognition and self-supervised representation learning. However, augmenting demonstration data for imitation learning presents a greater challenge [187, 245]. When altering an observation sequence (*e.g.*, by changing an object's initial position), it is necessary to also modify the corresponding action sequence to maintain a semantically consistent demonstration—one that still solves the intended task. Therefore, generating valid augmentations in imitation learning requires *equivariant transformations* that simultaneously adjust both the high-dimensional observations and the associated actions.

In imitation learning, the agent learns from a dataset $\mathcal{Z} = \{\zeta_1, \ldots, \zeta_M\}$ consisting of M demonstrations, each paired with a corresponding task encoding $\mathcal{T} = \{t_1, \ldots, t_M\}$. When tasks are described by language-based goals, the task t_m is represented by a language embedding (e.g., pre-trained CLIP embeddings) corresponding to the phase that describes the task. Each demonstration ζ_m consists of a sequence of continuous actions $\mathbf{A} = (a_1, \ldots, a_t)$, which are encoded as end-effector poses and gripper states. These actions are paired with observations $\mathbf{Q} = (q_1, \ldots, q_t)$, which in our case can be the object assets from the compositional manipulation world model.

Equivariant transformations for imagining novel demonstrations. Finding equivariant transformations for high-dimensional observations $\mathbf{Q}$ and actions $\mathbf{A}$ is typically challenging, but DREMA simplifies this by using explicit representations for object assets $\mathcal{O}$. In our world model, both the robot's actions $\mathbf{A}$ (the 3D positions of the end-effector and gripper) and the 3D positions of the objects, including initial and target locations, are transformed equivariantly under any rigid transformation. As a result, these transformations preserve task success in the imagined demonstrations. This allows us to generate novel "imaginations" by applying geometric transformations to the objects' positions and the robot's actions, without changing the robot's internal state. By executing these transformed actions $\hat{\mathbf{A}}$ and rendering the corresponding observations $\hat{\mathbf{Q}}$ in DREMA, we generate new demonstrations $\hat{\zeta} \in \mathcal{Z}_{new}$ not present in the original dataset $\mathcal{Z}$. In the following, we describe several such transformations applicable to manipulation tasks (see 5.3).

Roto-translation of the environment and objects. Given the roto-translation R_κ and the point of application P_κ, we transform each object asset o_k at time $t = 0$ (the beginning of the demonstration) and the Gaussians of the environment. To adapt the same demonstration to the transformed environment R_κ is applied to the end-effector translation et_i and the end-effector rotation er_i of each action $a_i = (et_i, er_i, ga_i)$. Consequently, the agent (fixed) is required to execute the transformed set of actions in the new environment.

Object rotation. This transformation involves rotating the objects around the final position of the action sequence $\mathbf{A}$. The transformation R is applied to both the object poses and the action trajectory $\mathbf{A}$, keeping the demonstration consistent with the original task. In this second transformation we fixed $P_\kappa = et_t$ (the last position of the end-effector). We then rotate the objects assets o_k at time $t = 0$ and the actions a_i similarly to the previous approach. In

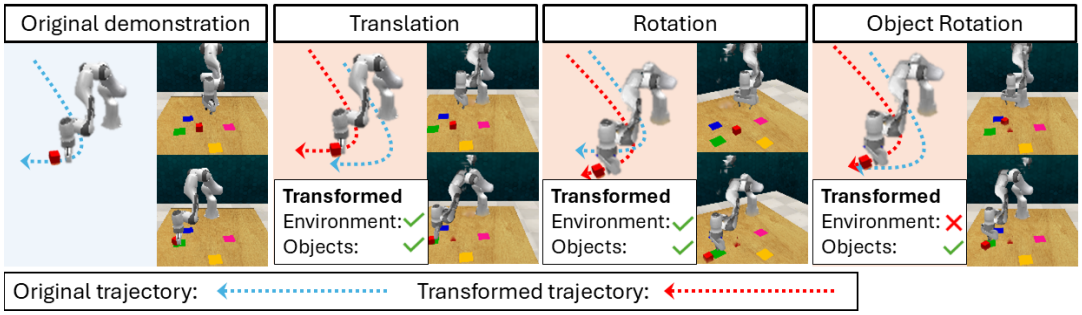


FIGURE 5.3: The effect of equivariant translation, equivariant rotation, and the object rotation transformations. Top row: start of demonstration. Bottom row: target of demonstration.

this transformation we do not change the environment since the trajectory brings the object to a similar position.

Verification of transformed demonstrations. Although these transformations can generate novel demonstrations, there is a risk of producing invalid outcomes due to factors like the robot's inability to execute the entire sequence or inaccuracies in the world model dynamics. Therefore, it is crucial to verify that the generated demonstrations still solve the intended task. In a general setting, a success detector would be required to ensure task completion. However, for goal-based tasks where equivariant transformations are applied, it is sufficient to confirm that the final positions of the objects in the transformed trajectory remain close to their expected transformed goal positions $\hat{l}_i$: $\|\hat{l}_i - R_i(l_i - P_{\kappa,i}) + P_{\kappa,i}\| < \tau$, where R_i represents the transformation applied, $P_{\kappa,i}$ is the center of the transformation, and $\tau = 0.015$ meters is the empirical threshold for positional accuracy.

5.1.5 Experiments

We evaluate the performance of DREMA in both simulated and real-world environments, comparing the effectiveness of imagination-generated data to imitation learning agents trained only on original demonstrations. For this evaluation, we use PerAct [9], a common baseline for imitation learning. Our study addresses the following key questions: (i) How does DREMA perform with only a few demonstrations available, in single- and multi-task settings? Does our manipulation world model still provide an advantage when more demonstrations are available? (ii) How do the different components of DREMA contribute to overall performance? (ii) How well does DREMA scale to real-world tasks, where both imitation learning and world model training are more complex?

One-shot Policy Learning: Single- and Multi-task

We test DREMA with a minimal number of demonstrations per task, i.e., one example per task variation up to five maximum examples (five examples for each task except for *slide block* and *place wine* that have 4 and 3 variations). We compare PerAct trained with original demonstrations, PerAct trained with DREMA's imagined ones, and PerAct trained with imagined and original combined. All simulation experiments are conducted on 50 environment configurations, with the test repeated five times to account for variability introduced by the motion planner.

Data collection in simulation. We replicate the experimental setup of PerAct [9] using RLBench [246], which involves a Franka Emika Panda robot with a parallel jaw gripper,

TABLE 5.1: Comparison of trained on original, augmented, and DREMA with imagined and all demonstrations, reporting mean ± std and max success over 5 runs on 50 test environments.

	Single-task									
	close jar		*insert peg*		*lift*		*pick cup*		*sort shape*	
	mean ± std	max	mean ± std	max	mean ± std	max	mean ± std	max	mean ± std	max
PerAct *(Original data)*	38.4 ± 0.80	40	0.0 ± 0.00	0.0	22.8 ± 1.6	**26**	13.2 ± 2.04	16	6.4 ± 1.50	8
Random patches [221]	45.2 ± 3.49	48	2.0 ± 1.79	4	20.4 ± 2.65	24	37.2 ± 3.25	42	3.6 ± 2.94	8
Random table color [247]	45.0 ± 1.73	46	1.6 ± 1.50	4	23.6 ± 0.80	24	25.6 ± 2.94	30	8.0 ± 1.26	10
Distractors [248]	36.4 ± 0.80	38	0.4 ± 0.80	2	22.8 ± 1.60	24	**41.2** ± 2.99	**44**	4.4 ± 2.33	8
DREMA *(imagination)*	41.2 ± 2.40	46	1.2 ± 0.98	2	17.2 ± 1.6	20	28.0 ± 4.20	36	9.6 ± 1.50	**12**
DREMA + Original *(All data)*	**51.2** ± 1.60	**54**	**2.4**± 2.33	**6**	**23.6** ± 1.50	**26**	34.4 ± 3.88	40	**11.2** ± 1.60	**12**
	place wine		*put in cupboard*		*slide block*		*stack blocks*		*avg single-task*	
	mean ± std	max	mean ± std	max	mean ± std	max	mean ± std	max	mean	max
PerAct *(Original data)*	10.0 ± 1.79	12	2.0 ± 0.00	2	48.4 ± 3.20	50	2.8 ± 0.98	4	16.0	17.6
Random patches [221]	12.8 ± 2.40	16	2.4 ± 1.50	**4**	36.8 ± 0.98	38	0.0 ± 0.0	0.0	17.8	20.4
Random table color [247]	10.0 ± 2.83	14	1.6 ± 0.80	2	40.8 ± 4.32	46	7.2 ± 2.04	10	18.2	20.7
Distractors [248]	14.8 ± 4.49	22	**3.2** ± 0.98	**4**	49.2 ± 0.98	50	8.4 ± 2.94	12	20.1	22.7
DREMA *(imagination)*	16.0 ± 2.19	18	0.4 ± 0.80	2	54.4 ± 2.15	62	4.0 ± 0	4	19.1	22.4
DREMA + Original *(All data)*	**26.8** ± 4.30	**32**	2.8 ± 0.98	**4**	**62.0** ± 2.19	**66**	**11.6** ± 1.96	**14**	**25.1**	**28.2**

placed on a table with objects that vary in color and position. We focus on tasks excluding articulated objects, as our current object-centric priors do not handle articulations. The experiments cover 1) non-prehensile tasks and 2) pick-and-place tasks (nine tasks in total). Following [218], we collect a sequence, $\mathbf{X}$, of 200 images by simulating a rotating camera around the scene. PerAct supports multi-task training, where a single policy adapts to multiple tasks based on prompts. We train both PerAct and DREMA on three tasks using the same demonstrations as in the single-task setup. Hyperparameters are consistent with the original paper [9], except for the training iterations and batch size. To prevent overfitting, we validate all models performance on a separate set for each 10k training iterations for multi-task and 5k for single-task and select the best-performing model.

TABLE 5.2: Comparison of PerAct [9] trained on original demonstrations to DREMA trained on only imagination demonstrations and the combination of both in the multi-task setting. The table reports the mean ± std and maximum success rate over 5 test runs.

	Multi-task							
	close jar		*sort shape*		*slide block*		*avg multi-task*	
	mean ± std	max	mean ± std	max	mean ± std	max	mean	max
PerAct *(Original data)*	26.0 ± 3.10	28	7.2 ± 1.60	10	34.0 ± 5.06	38	22.4	25.3
DREMA *(PerAct with imagined data)*	28.0 ± 3.35	32	**18.0** ± 2.83	**22**	48.0 ± 1.79	50	31.3	34.7
DREMA *(PerAct with all data)*	**46.0** ± 3.58	**52**	6.4 ± 3.20	12	**54.0** ± 2.19	**58**	**35.5**	**40.7**

One-shot single- and multi-task results (5.1 & 5.2). Since the limited number of demonstrations is insufficient to learn a general policy, the PerAct model trained solely on original demonstrations performs the worst.

In contrast, DREMA, exploiting around 800 demonstrations, significantly outperforms PerAct with the original data in single-task settings (5.1), achieving an average accuracy improvement of +9.1%. This result highlights that additional imagination-based demonstrations help create more robust policies, even when the original dataset is small. When DREMA is compared to invariant augmentations—namely, random patches, random table color, and random distractors—it achieves the best performance, with a 5.0% average improvement over

the best invariant augmentation. Additionally, DREMA improves performance in all tasks compared to PerAct, differently from invariant augmentations. In multi-task settings (5.2), DREMA is tested on *close jar*, *slide block*, and *sort shape* simultaneously. It achieves an average accuracy of 35.5%, representing a +13.1% improvement over PerAct, demonstrating that even this setting can significantly benefit from imagination-based demonstrations. DREMA trained on *only imagination demonstrations* and evaluated in the original environment still performs significantly better than PerAct (+3.1 on single task +9.1 on multi-task).

Analysis and Ablations

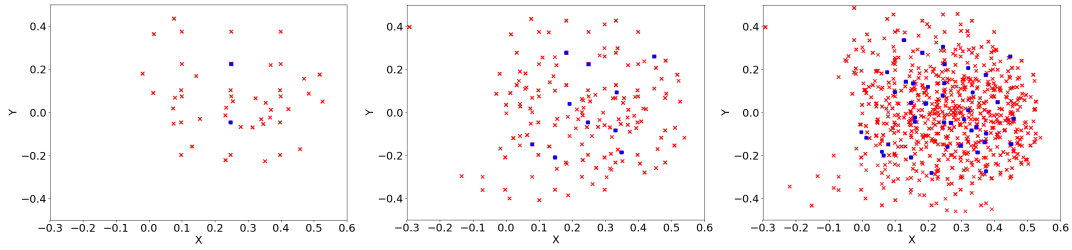


FIGURE 5.4: Keypoints distribution of original (blue dots) and generated data (red cross) for the *sort shape* task. Data is generated from one examples (left), five examples (middle) and twenty examples (right).

Varying dataset size (5.5 & 5.4). We analyze the effectiveness of DREMA's imagined demonstrations based on the number of original demonstrations provided. Specifically, we vary the dataset size $|\mathcal{D}| \in \{4, 10, 20\}$ for the *slide block* and and $|\mathcal{D}| \in \{5, 10, 20\}$ *sort shape* tasks.

Figure 5.5 shows the average success rate of DREMA and PerAct as a function of the original dataset size. Increasing the training examples to 20 improves performance in both tasks. PerAct trained on 20 examples matches DREMA trained with imagined data from just 5 demonstrations (81 and 61 imagined demos for *sort shape* and *slide block*, respectively). However, when using the same 20 examples to generate imagined data (437 and 391 imagined demonstrations for *sort shape* and *slide block*, respectively), the model improves by 6.4% and 10.4%.

This shows that generating novel configurations and actions remains beneficial, even with more demonstrations. Figure 5.4 in the appendix shows how augmented data cover a wider portion of the robot's workspace, enabling the agent to handle configurations beyond those in the original demonstrations.

Ablating transformations (5.3). We compare the equivariant transformations and include a baseline called *Replay*, which replays the original demonstrations. The *Object Rotation* and *Roto-translation* augmentations generate entirely imagined data. Both *Object Rotation* and *Roto-translation* increase the coverage of the state space and improve the generalization of the learned policy, with *Roto-translation* having the greatest impact. Combining these transformations results in even more general policies, yielding a 10.4% improvement in mean accuracy. This highlights the importance of learning a good world model and using it to explore and apply valid data augmentations.

Experiments with a Real Robot

Real robot setup. We verify the effectiveness of our approach in real-world environments. Our setup includes a Franka Emika Panda on a table, with a calibrated Kinect Azure mounted

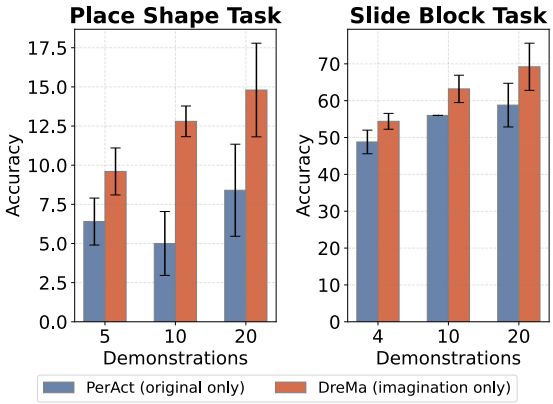


FIGURE 5.5: Imagined demonstrations keep improving imitation learning even with increasing number of original data.

TABLE 5.3: The mean ± std and maximum success rate of DREMA trained on single-task demonstrations from different types of transformations over 5 test runs on 50 examples.

	close jar		*sort shape*		*slide block*	
	mean ± std	max	mean ± std	max	mean ± std	max
Replay	10.0 ± 3.10	16	1.2 ± 0.98	2	26.0 ± 0.00	26
Object Rotation	25.2 ± 0.98	26	10.4 ± 2.65	**14**	42.0 ± 0.00	42
Roto-translation	41.2 ± 2.40	44	10.0 ± 2.53	**14**	50.4 ± 1.50	52
DREMA + Original *(All data)*	**51.2** ± 1.60	**54**	**11.2** ± 1.60	12	**62.0** ± 2.19	**66**

on the end-effector and a Kinect V2 positioned in front of the table, replicating the setup used in PerAct. We collect five tasks (number of demonstrations in 5.5). We train two models per task —one using only the original demonstrations and another combining original and imagined demonstrations— and do model selection as in PerAct.

The models are tested in 20 different environment configurations: 10 with object positions similar to the training set and 10 with positions outside the distribution (OOD), to assess generalization.

Verification of the learned world model (5.4). To assess the accuracy of DREMA in modeling demonstrations, we compare the final object positions at the end of the demonstration with those predicted by the world model. Results show that the world model aligns well with the collected data for simpler demonstrations. For more complex examples, errors could increase. This discrepancy could be addressed using real demonstrations to adapt the world model's parameters.

Real robot results (5.5 & 5.6). The imagined demonstrations double the model's task performance (62.9% for DREMA vs 31.7% for PerAct on average). Moreover, PerAct, trained only on original demonstrations, struggled when the object distribution differed from the training data.

In contrast, DREMA performs robustly in- and out-of-distribution environment configurations. The *pick shape* task had the lowest accuracy, likely due to the small size of the shapes and the model's difficulty in accurately recognizing them. Finally, in 5.6, we also confirm that DREMA with *only imagined trajectories* can outperform original PerAct training (+25% on *pick block* task) when more examples are available, showing the usefulness of the imagined trajectories on their own.

TABLE 5.4: Localization errors.

Task	**Error** (m)
pick block	0.010
pick shape	0.050
push	0.049
Average	0.038

TABLE 5.5: In- and out-of-distribution evaluation with real robots.

	pick block 4 examples		*pick shape* 5 examples		*push* 3 examples		*place obj* 4 examples		*erase* 4 examples		*Average*	
	In distr.	OOD	In distr.	OOD	In distr.	OOD	In distr.	OOD	In distr.	OOD	avg.	OOD
PerAct	55	50	30	10	40	10	20	10	30	20	31.7	25.0
DREMA (All)	**90**	**90**	**35**	**30**	**80**	**60**	**65**	**40**	**50**	**50**	**62.9**	**58.3**

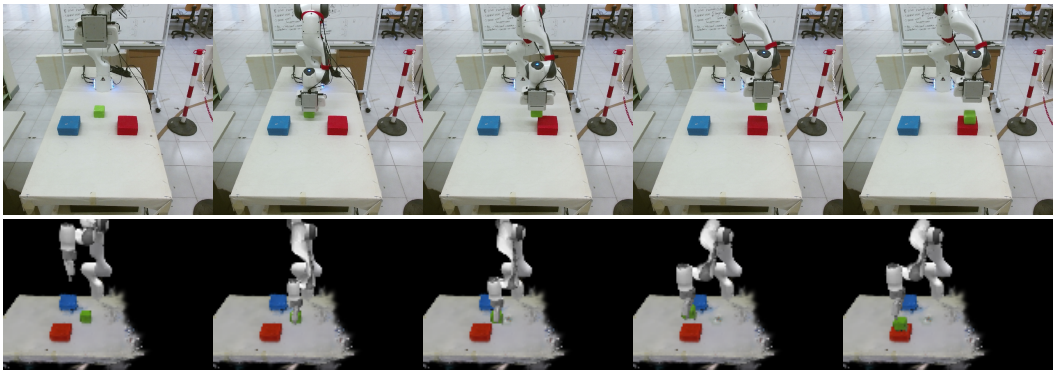


FIGURE 5.6: Original (top) and imagined demonstration (bottom) after a 90° rotation transformation.

TABLE 5.6: Results of the *pick block* task with original and imagination demonstrations.

	4 examples	**8 examples**
Real	55	55
Imagination	50	80
Imagination + Real	**90**	**85**

5.1.6 Discussion

We introduce a novel framework for constructing manipulation world models. By integrating real-time photorealism with Gaussian Splatting and physics simulators, DREMA enables robots to accurately predict the consequences of their actions and learn from minimal data. Our evaluations show significant improvements in policy learning, demonstrating the model's robustness and efficiency in real-world applications. We believe DREMA can influence other research areas, such as augmenting task variations with VLMs or enabling policy evaluation through imagination.

Limitations. Although DREMA demonstrates several strengths, it has limitations. First, it requires full observability of the environment to model physics and appearances. Second, it currently handles only simple objects, but recent advances in automatic URDF estimation could address this issue. Third, we construct it with an open-loop approach without feedbacks; incorporating them into the world model construction could improve both prediction accuracy and policy learning. Finally, scaling the world model to large and diverse environments may increase simulation complexity. In addition, the task demonstration pipeline assumes clean and accurately segmented demonstrations, which can be time-consuming to obtain and could be relaxed through more automated or weakly supervised data acquisition strategies. Overall, we believe that future research can address these issues, particularly the second and the third. Another interesting future direction is to study the effect of DREMA with more recent models (such as RVT2 [217] or Diffusion Policy [249]) or approaches that directly exploit RGB images to show the benefit of Gaussian Splatting.

5.2 Active Mapping for Dynamic Object Exploration

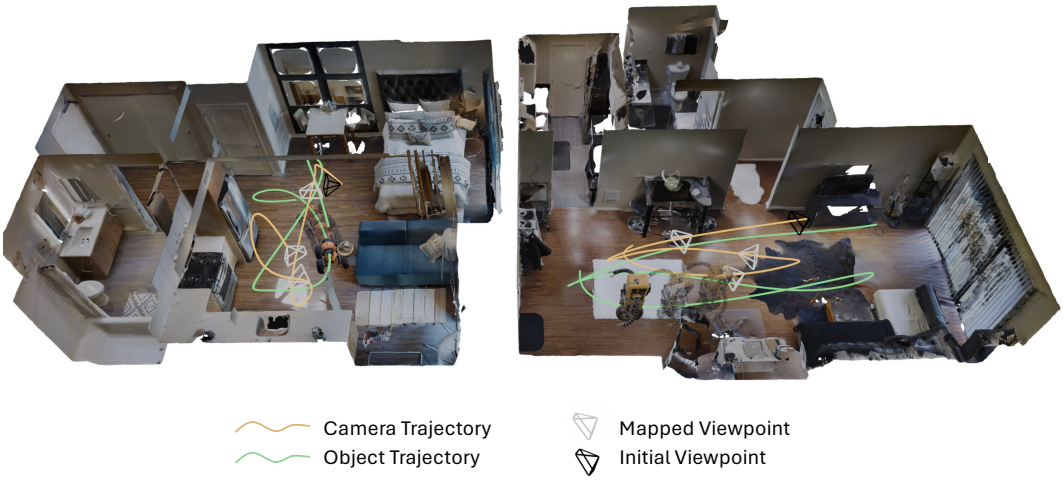


FIGURE 5.7: We introduce the novel task of active mapping of moving objects, requiring agents to plan observation trajectories while compensating for target motion. We also propose a method to solve this task and a benchmark for evaluation.

Extending the idea of adaptive perception from manipulation to mobile robotics, this section addresses the challenge of building scene representations when objects are in motion.

Scene exploration and mapping have been extensively studied in computer vision and robotics [250, 251, 252], with renewed interest driven by autonomous systems like drones and potential applications such as automated digital-twin generation. Existing exploration methods, however, rely on the assumption that the scene is static. This assumption fails in many real settings where moving objects constitute essential components of the environment. For example, in a construction site, trucks and mobile equipment are key elements of the scene that continuously reshape the workspace, yet the site cannot be halted to capture them in static conditions.

This is why we introduce a new task: active mapping of moving objects. An autonomous agent must reconstruct the 3D geometry of a non-cooperative object that moves independently of the mapping activity. This task is challenging because the agent must gather views that reveal new parts of the object while compensating for the object's future motion during its own navigation. As a result, viewpoint quality depends jointly on geometric informativeness and the feasibility of reaching that view at the right time.

To solve this new task, we propose "Paparazzo", a learning-free framework for active 3D reconstruction of dynamic objects. Paparazzo considers a set of viewpoints distributed in a foveal configuration around the target object and moving with it over time. To select the most informative viewpoints, we rely on Fisher Information computed from a 3D Gaussian Splatting model [253], while, to predict the object trajectory and the future positions of these viewpoints, we leverage an Extended Kalman Filter [254]. Crucially, the viewpoint with the highest expected information gain is not always the optimal choice: because viewpoints move with the target object, a view with slightly lower information gain but significantly shorter travel time may be more beneficial and efficient. For this reason, we explore multiple strategies that jointly account for information gain and motion feasibility when selecting the next best viewpoint. Since Paparazzo requires no training data, it generalizes immediately to new scenes and previously unseen objects.

A first advantage of the Extended Kalman Filter (EKF) is that it can combine past observations of the object to accurately predict its trajectory. We also use it to detect when the trajectory prediction is not reliable, e.g. when the object changes moving direction abruptly.

When the EKF novelty indicates unreliable prediction, Paparazzo switches to a mode where it continuously adjusts the agent position to keep the object centered in the camera's field of view and within an optimal range, prioritizing observations that enhance motion estimation.

For evaluation, we introduce a comprehensive benchmark and protocol for active mapping in dynamic environments, measuring reconstruction fidelity, spatial coverage, and temporal consistency across several baselines. We assume access to the target object's mask whenever it is visible. In practice, this can be obtained by background subtraction when the static scene is already known or by segmenting moving objects [255, 256]. For this benchmark, we developed a simulator based on the Habitat simulator [257] generating complex motions of target objects within different environments.

Our experiments show that Paparazzo significantly improves 3D reconstruction fidelity and mapping efficiency compared to several baselines, marking a key step toward intelligent scene understanding in dynamic environments. We will make the code of our benchmark publicly available for future comparison against Paparazzo.

Our key contributions can be summarized as:

- We introduce the novel task of active mapping of moving objects, where an agent must efficiently reconstruct the 3D geometry of non-cooperative, independently moving targets.
- We propose Paparazzo, a learning-free dual-mode framework combining 3D Gaussian Splatting–based information gain with EKF-based motion prediction.
- We present the first benchmark for this task and demonstrate large performance gains across multiple dynamic scenarios.

5.2.1 Related Works

Active Mapping for Static Scenes

The goal of active mapping is usually to determine how an agent should move to efficiently explore and reconstruct an unknown 3D environment. The exploration must be exhaustive: by the end of the task, the agent should have covered the entire scene while keeping its trajectory as short as possible.

Works on active mapping can be broadly categorized into traditional and learning-based approaches. Traditional methods primarily rely on heuristic strategies, such as frontier-based exploration [250, 258] and next-best-view (NBV) selection [259, 260], or a combination of both [261, 262], to guide the robot's exploration process. They often employ voxel grids or point clouds to represent the scene.

Learning-based approaches have recently emerged to leverage deep neural networks and more expressive scene representations. For example, MACARONS [263] uses neural networks to predict the coverage gain of candidate camera poses, effectively guiding NBV selection. NextBestPath [264] learns to predict the piece of trajectories that maximizes the cumulative coverage gain along the path.

With the advent of NeRF [265] and 3D Gaussian representations [35], recent works emerged, such as ANM [38], NARUTO [266], and ActiveGS [267], that train such models as the intermediate scene state, using measures such as confidence or Fisher information to determine the next-best pose. Combined with traditional path planning algorithms, these methods achieve impressive performance in producing high-quality 3D reconstructions.

To the best of our knowledge, all active mapping works consider a static scene. In this paper, we are interested in mapping a non-cooperative mobile object, which is much more challenging as we need to estimate the object motion and compensate for it when planning the next move of our agent.

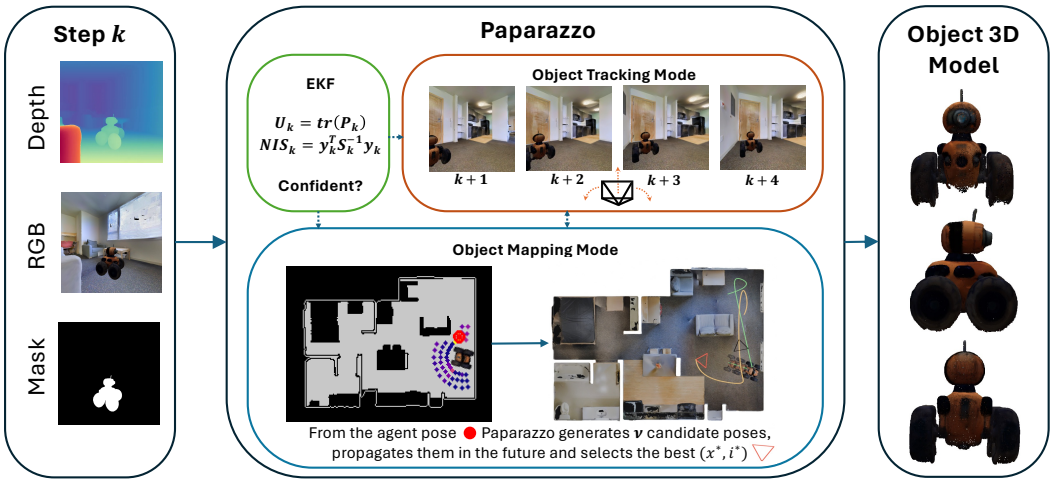


FIGURE 5.8: Paparazzo alternates between Object Tracking Mode and Object Mapping Mode based on the confidence of the EKF motion estimate. When the filter is uncertain, the agent prioritizes acquiring stabilizing observations; once confident, it predicts future object motion, generates and propagates candidate viewpoints, and selects the optimal one $(\mathbf{x}^*, i^*)$ that minimizes the final cost function.

Mobile Object Passive Reconstruction

Our work is related to the reconstruction of mobile objects, such as in-hand scanning, where a target object is moved in front of one or several cameras [268, 269, 270, 271, 272, 273]. Like us, they aim to reconstruct an object while estimating its motion within a scene. In particular, [273] also uses Gaussian primitives to represent the object as we do. The key difference is that in the case of in-hand scanning, a user moves the object aiming to improve the reconstruction, which means the object motion is intended to support the task, so it is "cooperative". In our case, we need to plan how to move the agent in the environment to capture new relative poses between the object and the agent, in addition to track and reconstruct the object. In such a scenario, the object moves in a "non-cooperative" manner, meaning that the object does not move in a way that facilitates its reconstruction.

5.2.2 Paparazzo

As shown in Figure 5.8, our "Paparazzo" method alternates between two operating phases depending on the confidence of its estimate of the target object's motion:

- *Object Tracking Mode* (Section 5.2.6): Paparazzo switches to this mode when its estimate of the object's motion is uncertain. It then keeps the target object in its field of view to improve this estimate.
- *Object Mapping Mode* (Section 5.2.7): Paparazzo switches to this mode when it is confident enough of the object's motion. It then plans motions to informative viewpoints for efficient reconstruction and executes them.

5.2.3 Problem Formulation

Let an agent equipped with a fixed, front-facing RGB-D camera operate in a 3D world frame W that contains a dynamic object O.

At each discrete time step k, the camera pose $T_{C_k}^W \in SE(3)$ is assumed to be known from the agent's localization system, while the object pose $T_{O_k}^W \in SE(3)$ is unknown and must

be estimated. When the object is detected, its segmentation mask $\mathcal{M}_k$ allows extracting the corresponding 3D points $\mathcal{P}_{O_k}^{C_k} = \{p_i^{C_k}\}_{i=1}^{P}$ in the camera frame, where each $p_i^{C_k}$ is the 3D point obtained by back-projecting a pixel inside the mask using the depth information.

The objective is to determine informative viewpoints that can be reached by the agent and enable it to efficiently observe and reconstruct the complete surface of the moving object with minimal views, producing a consistent 3D model in the object's local reference frame while predicting and adapting to its motion.

5.2.4 Initialization

At the first detection time t_d, we initialize the object pose from the 3D object points $\mathcal{P}_{O_{t_d}}^{C_{t_d}}$. The object translation $t_{O_{t_d}}^{C_{t_d}}$ is defined as the centroid of these points. The object rotation $R_{O_{t_d}}^{C_{t_d}}$ is constructed by aligning its z-axis with the world vertical direction, while the x–y axes are obtained by performing PCA on the points projected onto the ground plane. The initial object pose expressed in the world coordinate system therefore is:

$$T_{O_{t_d}}^{W} = T_{C_{t_d}}^{W} T_{O_{t_d}}^{C_{t_d}}, \quad \text{with} \quad T_{O_{t_d}}^{C_{t_d}} = \begin{bmatrix} R_{O_{t_d}}^{C_{t_d}} & t_{O_{t_d}}^{C_{t_d}} \\ 0 & 1 \end{bmatrix}. \tag{5.2}$$

We initialize Gaussian primitives $\mathcal{G}_O$ from the object's segmented RGB-D observation, as in the SplaTAM backbone [274]. Although SplaTAM was originally designed for static scenes, we expressed $\mathcal{G}_O$ in the object reference frame by means of the estimated object pose that remains consistent across time as the object moves.

5.2.5 EKF-Based Motion Prediction

We rely on an Extended Kalman Filter (EKF) defined on $SE(3)$ to estimate the object state $T_{O_k}^{W}$ and its uncertainty P_k. We assume a constant-velocity motion model, so the object state is composed of the object pose and its linear and angular velocities.

In particular, The EKF used in Paparazzo estimates the object pose in the world reference frame and its body-frame twist:

$$(\, T_k,\, \omega_k \,), \qquad T_k \in SE(3),\ \ \omega_k \in \mathbb{R}^6 . \tag{5.3}$$

Since $SE(3)$ is a nonlinear manifold, the EKF operates on a minimal error state expressed in its tangent space:

$$\delta x_k = \begin{bmatrix} \delta\xi_k \\ \delta\omega_k \end{bmatrix} \in \mathbb{R}^{12}, \qquad \delta\xi_k \in \mathfrak{se}(3),\ \delta\omega_k \in \mathbb{R}^6 . \tag{5.4}$$

Here, $\delta\xi_k$ represents a 6-DoF perturbation of the pose expressed in the Lie algebra $\mathfrak{se}(3)$, while $\delta\omega_k$ encodes the deviation in the object twist. With this representation, the EKF proceeds through the standard prediction and update phases using Lie-group consistent linearizations.

Prediction. Under the constant-twist motion model adopted in Paparazzo, the pose evolves via the exponential map:

$$T_{k|k-1} = T_{k-1}\, \exp(\omega_{k-1}\, \Delta t) . \tag{5.5}$$

The covariance follows the linearized dynamics:

$$P_{k|k-1} = F_k P_{k-1} F_k^{\top} + Q , \tag{5.6}$$

where F_k is the Jacobian of the motion model and Q is the block-diagonal process noise matrix defined in Sec. 5.2.5.

Update. Whenever the object is visible, Paparazzo provides an absolute pose estimate $T_{O_k}^{W,\text{meas}} \in SE(3)$, obtained from the pose initialization and the ICP refinement (as described in Sec. 5.2.6). The innovation is computed on the Lie algebra:

$$y_k = \log(T_{\text{err}}) \in \mathfrak{se}(3)\,. \tag{5.7}$$

where $T_{\text{err}} = (T_{k|k-1})^{-1} T_{O_k}^{W,\text{meas}}$.

Since the measurement constrains only the object pose, the Jacobian is

$$H = \begin{bmatrix} I_6 & 0 \end{bmatrix}\,, \tag{5.8}$$

and the innovation covariance and Kalman gain are respectively

$$S_k = H P_{k|k-1} H^\top + R, \qquad K_k = P_{k|k-1} H^\top S_k^{-1}\,. \tag{5.9}$$

The error state and covariance update follow the standard EKF equations:

$$\delta x_k = K_k\, y_k, \qquad P_k = (I - K_k H)\, P_{k|k-1}\,, \tag{5.10}$$

and the nominal state is corrected by retracting the pose increment on $SE(3)$:

$$T_k = T_{k|k-1}\, \exp(\delta\xi_k)\,, \qquad \omega_k = \omega_{k|k-1} + \delta\omega_k\,. \tag{5.11}$$

This formulation explicitly shows how the process noise Q, measurement noise R, and innovation y_k contribute to the filtering process.

EKF Parameters and Confidence Criteria The process noise Q is block–diagonal, separating uncertainty in pose and twist:

$$Q = \begin{bmatrix} Q_t & 0 \\ 0 & Q_\omega \end{bmatrix}, \qquad Q_t = 10^{-4} I_6\,, \tag{5.12}$$

$$Q_\omega = \text{diag}\big(10^{-4}, 10^{-4}, 10^{-4}, 5\times10^{-4}, 5\times10^{-4}, 5\times10^{-4}\big)\,.$$

These values are deliberately small: the object moves smoothly from frame to frame and receives frequent and accurate pose corrections from ICP. Slightly larger rotational entries in Q_ω keep the filter responsive to small angular variations while maintaining stability. All parameters were tuned empirically to balance prediction smoothness with robustness against occasional ICP imperfections (e.g., partial views or local registration jitter). The measurement noise covariance is set to $R = 10^{-3} I_6$, modeling the uncertainty of the absolute 6D pose provided by ICP. Since the filter starts with no prior knowledge of the object state, the initial covariance is set to $P_0 = I_{12}$, representing an uninformative prior over pose and twist.

Given these noise models, the key question becomes determining when the EKF is sufficiently reliable to switch from Object Tracking Mode to Object Mapping Mode. To this end, Paparazzo monitors two complementary quantities derived from the EKF state: (i) the uncertainty U_k, and (ii) the innovation consistency metric NIS_k. In particular, the innovation quantifies the consistency of a new measurement $T_{O_k}^{W,\text{meas}}$ of the target object pose with the current state estimate:

$$\text{NIS}_k = \mathbf{y}_k^\top S_k^{-1} \mathbf{y}_k\,, \tag{5.13}$$

where $\mathbf{y}_k = \log((T^W_{O_{k|k-1}})^{-1} T^{W,\text{meas}}_{O_k})^\vee$ is the innovation on $SE(3)$ and $S_k = HP_{k|k-1}H^\top + R$ the innovation covariance.

These values are compared against their respective thresholds, $\tau_u = 0.1$ and $\tau_n = 0.5$. In particular, the choice of the NIS threshold is guided by empirical observations. Although a 6D pose measurement would theoretically call for a $\chi^2(6)$ statistical test, in practice the innovations are orders of magnitude smaller. This is a direct consequence of three conditions in our setting: the object exhibits smooth frame-to-frame motion, the process noise is intentionally low to preserve rigid-motion coherence, and the ICP refinement provides accurate local pose corrections. Therefore, classical chi-squared thresholds (e.g., 12.59 at the 95th percentile) are too large to be meaningful. Instead, we adopt a data-driven threshold: during steady motion the NIS remains consistently below $0.1-0.2$, whereas genuine motion changes or ICP inconsistencies yield significantly larger values. Based on this empirical separation, we set $\tau_n = 0.5$. Thus, values above this threshold indicate that the observed motion is no longer compatible with the current state estimate. However, using these thresholds at a single time step is not sufficient, rather, they serve as indicators for assessing long-term EKF stability.

Confidence Condition. The EKF is considered confident when both the uncertainty and innovation remain below their thresholds for $N_s = 4$ consecutive frames:

$$U_k < \tau_u, \qquad \text{NIS}_k < \tau_n\,. \tag{5.14}$$

This temporal stability requirement prevents spurious mode switches caused by short-term ICP mismatches or partial occlusions. Only after N_s stable frames the system transitions from the Object Tracking Mode to the Object Mapping Mode. Conversely, if either quantity exceeds its threshold for N_s consecutive frames, the system reliably detects a genuine motion change and switches back to the reactive Object Tracking Mode.

Table 5.7 summarizes all EKF parameters.

TABLE 5.7: EKF parameters used in all experiments.

Parameter	Value
τ_u (uncertainty threshold)	0.1
τ_n (NIS threshold)	0.5
N_s (stability window)	4
Q_t	$10^{-4} I_6$
Q_ω	$\text{diag}(10^{-4}, 10^{-4}, 10^{-4}, 5{\cdot}10^{-4}, 5{\cdot}10^{-4}, 5{\cdot}10^{-4})$
R	$10^{-3} I_6$
P_0	I_{12}

5.2.6 Object Tracking Mode

The goal of this mode is to prioritize frequent observations of the target object to refine motion estimates: the agent actively keeps the object within the camera's field of view, while continuously reconstructing it and updating its motion estimate. At each time step, the agent rotates to move the segmentation mask toward the image center, and translates to adjust its distance to the object so that the object's apparent size remains approximately half of the image.

Object Point Cloud Accumulation Whenever the object is visible and the mask $\mathcal{M}_k$ is available, we extract the corresponding object point cloud $\mathcal{P}_{O_k}^{C_k}$ in the current camera frame C_k. To bring this point cloud into the object reference frame O_{t_d}, we compute the following initial alignment:

$$\hat{\mathcal{P}}_{O_k}^{O_{t_d}} = T_W^{O_{t_d}} \, T_{C_k}^{W} \, \mathcal{P}_{O_k}^{C_k}, \tag{5.15}$$

where $T_W^{O_{t_d}}$, is the inverse of the rototranslation computed during the initialization phase (Sec. 5.2.4). However, the point cloud $\hat{\mathcal{P}}_{O_k}^{O_{t_d}}$ does not yet account for the motion that occurred between the detection time t_d and the current step k. As a result, it remains misaligned with the previously accumulated 3D model $\mathcal{P}$. To recover this missing relative motion and correctly integrate the new observations, we apply the alignment procedure described in Sec. 5.2.6. We first compute a coarse but globally consistent registration using KISS-Matcher [275], which is robust to outliers and large inter-frame displacements, estimating an initial transformation $\hat{T}_{\text{align},k}$. This estimate is then refined with Colored ICP [276] to obtain the final transformation $T_{\text{align},k}$, that is applied to $\hat{\mathcal{P}}_{O_k}^{O_{t_d}}$ before merging it into the current object model:

$$\mathcal{P} \leftarrow \mathcal{P} \cup (T_{\text{align},k} \, \hat{\mathcal{P}}_{O_k}^{O_{t_d}}). \tag{5.16}$$

The same alignment also provides the absolute 6D pose measurement used by the EKF

$$T_{O_k}^{W,\text{meas}} = T_{O_{t_d}}^{W} \, T_{\text{align},k}. \tag{5.17}$$

Object Gaussian Optimization The Gaussian Splatting model $\mathcal{G}_O$ is concurrently refined using the SplaTAM optimization process, which incrementally densifies and updates the dynamic object representation using the new RGB-D observation transformed into the object frame.

In particular, Whenever the object is detected and its mask $\mathcal{M}_k$ is available, we update the dynamic Gaussian model $\mathcal{G}_O$. At time k, all Gaussians are expressed in the current object reference frame O_k, which evolves over time according to the relative motions $T_{O_k}^{O_{t_d}}$ obtained during the alignment stage. Before optimizing the model, Paparazzo performs densification, which requires determining which regions of the object surface are actually visible at time k and already represented in $\mathcal{G}_O$. To do this, all Gaussian centers g^{O_k} are first transformed into the current camera frame C_k using the latest estimated object pose:

$$g^{C_k} = T_W^{C_k} \, T_{O_k}^{W,\text{meas}} \, g^{O_k}. \tag{5.18}$$

The projection enables silhouette rendering: pixels in the mask $\mathcal{M}_k$ that are not covered by any projected Gaussian indicate unexplained regions of the object surface, where new Gaussians must be inserted. However, mask-based visibility alone is insufficient. Gaussian centers corresponding to the back side of the object may still fall inside $\mathcal{M}_k$ when projected, even though they are not physically visible in the current view. To ensure geometric correctness, Paparazzo applies a depth-consistency filter. For each projected Gaussian at pixel u, let$z_{g_k}(u)$ be its depth and $z_k(u)$ the depth observed in the current frame. A Gaussian is considered visible and already accounted for in $\mathcal{G}_O$ only if:

$$\left| z_{g_k}(u) - z_k(u) \right| \leq \tau_d, \qquad \tau_d = 0.02 \text{ m}. \tag{5.19}$$

This constraint ensures that only the physically observable surface contributes to determining which parts of the object require densification.

Keyframe Selection and Optimization For the gaussian model optimization, Paparazzo leverages past keyframes that still observe the same surface region visible at time k. Each past keyframe $i < k$ stores an object point cloud $\mathcal{P}_{O_i}^{C_i}$ extracted at time i. During the accumulation procedure, this point cloud is first expressed in the object reference frame O_{t_d} (using Eq. 5.15), and then registered via the estimated alignment transformation $T_{\text{align},i}$:

$$\hat{\mathcal{P}}_i = T_{\text{align},i}\, \hat{\mathcal{P}}_{O_i}^{O_{t_d}} . \tag{5.20}$$

To determine whether keyframe i observes the same part of the object currently visible at time k, we transform $\hat{\mathcal{P}}_i$ into the current camera frame C_k:

$$\mathcal{P}_{O_i}^{C_k} = T_W^{C_k}\, T_{O_{t_d}}^{W}\, \hat{\mathcal{P}}_i . \tag{5.21}$$

Then, we reproject $\mathcal{P}_{O_i}^{C_k}$ into the current image plane, obtaining a set of pixel locations $\Omega_{k,i}$. We retain only the pixels that fall inside the current mask:

$$\Omega_{k,i} \leftarrow \Omega_{k,i} \cap \mathcal{M}_k . \tag{5.22}$$

Even in this case, mask agreement alone is insufficient, since points from the back side of the object may still project inside $\mathcal{M}_k$. Therefore, we apply the same depth-consistency filter used during densification: a reprojected point at pixel u is kept only if its depth is consistent with the current observation,

$$\left| z_{i\to k}(u) - z_k(u) \right| \;\le\; \tau_d, \qquad \tau_d = 0.02 \text{ m}, \tag{5.23}$$

where $z_{i\to k}(u)$ is the depth of the reprojected 3D point in the camera frame C_k from keyframe i.

Let $\Omega'_{k,i}$ denote the set of points passing both mask and depth checks. We compute the visibility overlap

$$\eta_i = \frac{|\Omega'_{k,i}|}{|\mathcal{M}_k|} . \tag{5.24}$$

Only keyframes with $\eta_i \ge 0.5$ are retained, and at most the top 20 highest-overlap ones are selected for the gaussian optimization. Because the object is rigid, this 3D-aware visibility test ensures that only keyframes observing the same surface region as the current view meaningfully contribute to the update.

We then jointly optimize the Gaussians associated with the selected past keyframes and the current one, using RGB and SSIM losses [274]. This produces a spatially localized, temporally consistent, and computationally efficient refinement of the dynamic object model.

Once the motion uncertainty stabilizes, Paparazzo shifts its focus from re-localization to information-driven exploration, leveraging the learned object GS model to guide active reconstruction.

5.2.7 Object Mapping Mode

When the EKF stabilizes, Paparazzo transitions to the *Object Mapping Mode*. The goal of this mode is to move the agent to poses that will significantly improve its reconstruction of the object, while taking into account the object motion as predicted by the EKF.

As shown in Figure 5.8, we sample candidate poses $\mathcal{V}$ with respect to the object, therefore these poses move together with the object. The camera centers for these poses are distributed around the object in a foveated configuration, and the cameras point toward the object.

If the object was static, we could simply select the most informative poses in $\mathcal{V}$ according to FisherRF [253] applied to the object GS model $\mathcal{G}_O$. However, since the object is moving, we must trade off between (i) the informativeness of a viewpoint and (ii) the time required for the agent to reach it. To quantify this trade-off, we introduce the following criterion:

$$B(\mathbf{x}, i) = -w_{\text{eig}}\text{EIG}(\mathbf{x}) + w_{\text{sync}}C_{\text{sync}}(\mathbf{x}, i)\,, \tag{5.25}$$

where $\text{EIG}(\mathbf{x})$ is the FisherRF informativeness measured for the viewpoint $\mathbf{x} = T_{C_{\text{cand}}}^{O_i}$ with $C_{\text{cand}} \in \mathcal{V}$, and $C_{\text{sync}}(\mathbf{x}, i)$ is a criterion we introduce to measure how well the agent can reach pose $\mathbf{x}$ (which moves together with the target object) at time step i. Weights w_{sync} and w_{eig} balance the contribution of the two terms.

The FisherRF criterion works by quantifying how much a new pose contributes to refining the parameters $\boldsymbol{\theta}$ of our current Gaussian Splatting representation $\mathcal{G}_O$ of the object. It can be computed analytically and efficiently from $\mathcal{G}_O$. More details can be found in [253].

The term $C_{\text{sync}}(\mathbf{x}, i)$ accounts for both the predicted motion of the object and the agent's effort to reach $\mathbf{x}$, and is defined as:

$$C_{\text{sync}}(\mathbf{x}, i) = \left|\hat{s}_{\text{agent}}(\mathbf{x}) - (i - k)\right|. \tag{5.26}$$

Here $\hat{s}_{\text{agent}}(\mathbf{x})$ denotes the number of motion steps required for the agent to reach the viewpoint $T_{O_i}^{W} \cdot \mathbf{x}$, where $T_{O_i}^{W}$ is the object pose predicted by the EKF for time step i. Thus, $T_{O_i}^{W} \cdot \mathbf{x}$ represents the viewpoint $\mathbf{x}$ in the world coordinate system. We compute $\hat{s}_{\text{agent}}(\mathbf{x})$ using an A* motion planner [277]. The term $i - k$ is the predicted number of steps for $\mathbf{x}$ to move from its current location $T_{O_k}^{W} \cdot \mathbf{x}$ (in the world coordinate system) to its predicted location $T_{O_i}^{W} \cdot \mathbf{x}$. We finally pick the pose that leads the best tradeoff over a chosen number N_h of future time steps:

$$(\mathbf{x}^*, i^*) = \underset{\mathbf{x}\in\mathcal{V},(i-k)\leq N_h}{\text{argmin}} \; B(\mathbf{x}, i)\,. \tag{5.27}$$

While moving the agent to pose $T_{O_i^*}^{W} \cdot \mathbf{x}^*$, Paparazzo continuously integrates new RGB-D observations into the object point cloud $\mathcal{P}$, updates $\mathcal{G}_O$, and monitors the EKF consistency. If the NIS or state uncertainty exceed confidence thresholds, mapping is halted and the system reverts to *Tracking Object Mode*, maintaining a reactive loop between mapping and tracking.

This dynamic coupling of information-driven mapping and motion-aware prediction constitutes the core novelty of Paparazzo, enabling 3D reconstruction of moving objects without assuming static scenes.

5.2.8 Experimental Results

To evaluate our Paparazzo method, we introduce a dedicated benchmark and evaluation protocol designed to assess both reconstruction fidelity and spatial coverage over time. Experiments are conducted within Habitat 3.0 [257], a high-performance 3D simulator that provides realistic indoor environments and robot displacements.

We selected ten photorealistic indoor scenes—three from the Matterport3D dataset (M) [278] and three from the Gibson dataset (G) [279]—commonly used for static active mapping [38]. To extend these static scenes to dynamic scenarios, we introduce a synthetic moving target object into each environment. We consider the four target objects shown in Figure 5.9. Each object is evaluated independently across all scenes in separate runs. This setup ensures statistical diversity across both objects and environments. The agent is equipped with an RGB-D sensor, initialized at random navigable positions.



(A) Object 1

(B) Object 2

(C) Object 3

(D) Object 4

FIGURE 5.9: The four target objects used in our experiments, featuring different shapes and colors. Each object is evaluated independently across all environments.

Object Motion Protocol. To comprehensively assess reconstruction performance under diverse object motion dynamics, we consider three motion patterns for the target:

- **Bouncing Ball:** upon collision, the object randomly changes orientation and continues along this new direction.
- **Forward & Backward:** the object moves strictly along a straight line, without rotation, moving forward until it collides with an obstacle and then reversing direction repeatedly.
- **Stop & Go:** similar to the *Bouncing Ball* motion but introduces intermittent stops—pausing every S steps and resuming after G steps—to simulate non-uniform velocity profiles.

Each motion pattern introduces distinct challenges related to motion predictability, visibility continuity, and temporal occlusions. Each translation of the object is 5 cm per step, each rotation is 10° per frame.

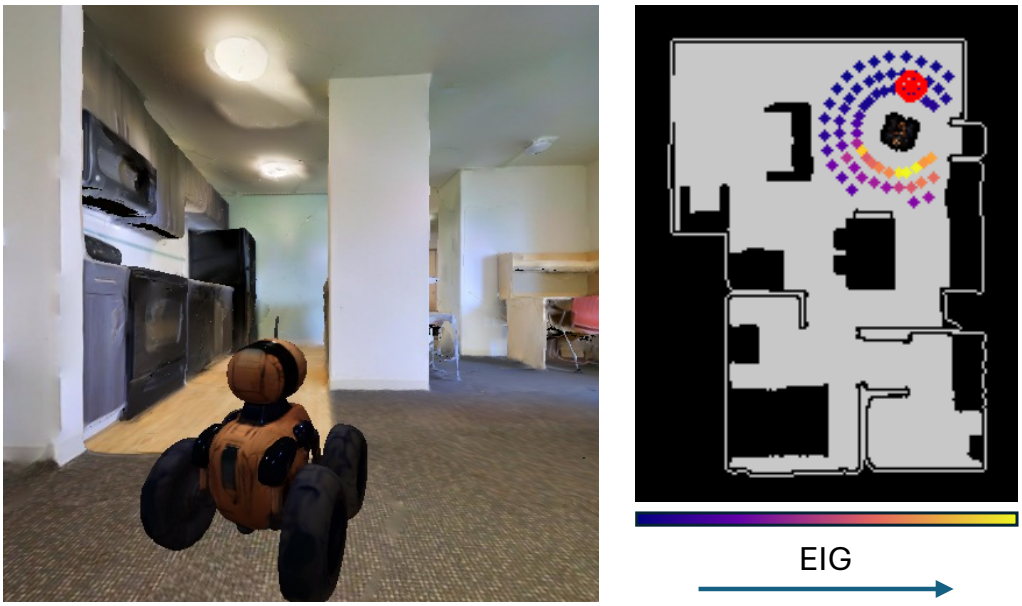


FIGURE 5.10: **Generation of candidate viewpoints for Object Mapping Mode.** When the EKF becomes confident, Paparazzo switches from tracking to mapping and evaluates a set of candidate viewpoints $\mathcal{V}$ distributed around the object. The expected information gain (EIG) of each pose, computed using the FisherRF criterion, is visualized here with a color gradient: darker tones correspond to low informativeness, while brighter tones highlight more informative poses for reconstructing the object.

Agent Motion Model. The agent navigates using a discrete action space made of standard motion primitives:

- **Forward:** move ahead by 10 cm,
- **Rotate Left / Right:** yaw rotation by 10°.

These control primitives follow the canonical Habitat discretization for active perception tasks, allowing reproducible and physically plausible exploration behavior.

During Object Tracking Mode, the agent also maintains a distance of 1–1.5 m from the object. This emulates a realistic "follow-and-observe" behavior and allows reliable short-term tracking even in compact environments.

To construct the viewpoint set $\mathcal{V}$ for Object Mapping Mode, candidate camera poses are sampled around the object using a foveated, object-centered distribution. Camera centers lie on three concentric circular rings on a common horizontal plane, with radii between 0.5 and 2.0 m. Samples are placed every 12° in azimuth, and all camera orientations point toward the object (see Figure 5.10). This design provides dense and uniform surface coverage while restricting viewpoints to geometrically feasible, navigable regions of the scene. Each candidate viewpoint is then evaluated using the FisherRF criterion, which quantifies its expected information gain (EIG).

Because the object is dynamic, Paparazzo must reason not only about the informativeness of a viewpoint at the current time, but also about how informative it will remain as the object moves. To do this, Paparazzo propagates all candidate viewpoints $\mathcal{V}$ across the next N_h predicted object poses obtained from the confident EKF (see Figure 5.11).

This results in a set of $|\mathcal{V}| \times N_h$ future-consistent candidate viewpoints, each synchronized with the anticipated motion of the object and equipped with its predicted EIG.



FIGURE 5.11: **EKF-based prediction of future object poses.** The EKF predicts the object pose, denoted in orange, over the next N_h steps. The figure shows three examples at steps k+10, k+20, and k+30. These predicted poses are then used to propagate all candidate viewpoints $\mathcal{V}$ into the future, producing a set of $|\mathcal{V}| \times N_h$ future-aligned viewpoints that are subsequently evaluated by the final cost function in Eq. (5.25).

Finally, Paparazzo selects the target viewpoint using the cost function defined in Eq. (5.25), which jointly accounts for information gain, reachability, and temporal synchronization between the agent trajectory and the object's predicted motion.

5.2.9 Evaluation Protocol

We report both geometric accuracy and exploration efficiency across all experiments. Metrics are computed by comparing the reconstructed point cloud $\mathcal{P}$ against the ground-truth object model $\mathcal{P}_{GT}$ at the final time step and throughout exploration. We considered three metrics:

- **3D Coverage [%]**: the percentage of ground-truth points that have at least one reconstructed point within a distance threshold τ (higher = more complete reconstruction);
- **Completeness [cm]**: the mean distance from each ground-truth point to its nearest reconstructed point (lower = fewer missing regions).
- **AUC**: the area-under-curve of coverage with respect to the agent steps, reflecting how efficiently coverage increases during exploration (higher = faster coverage).

All metrics are tested across all the scenes, objects, and motion patterns detailed above.

Baselines. We compare Paparazzo against three baselines designed to isolate the contributions of viewpoint selection, motion prediction, and temporal feasibility:

- Random Walk (RW): a classical baseline in active mapping for static scenes. The agent moves randomly across the environment, accumulating object point clouds whenever the object falls within its field of view, without considering the object motion.
- Random Informative Selection (RIS): an ablation of our method that selects, at each mapping iteration, a random feasible pose among the $N_h \times B$ informative candidate viewpoints, ignoring both the synchronization cost and the predicted feasibility of observing the object from that position.

- Tracking-Only (TO): we keep the agent in Object Tracking Mode, a purely passive strategy that continuously tracks the object's motion using the EKF but performs no active viewpoint selection or mapping, serving as a lower bound for reconstruction completeness.

These baselines allow us to disentangle the impact of Paparazzo's key components—motion-aware viewpoint selection and temporal feasibility reasoning—on overall reconstruction quality and mapping efficiency.

5.2.10 Results

Table 5.8 gives the quantitative results across six different indoor scenes for our three dynamic motion types. All reported values are averaged over all test objects within each scene and across five runs, with each run consisting of 500 agent steps.

For all experiments, we configured Paparazzo to predict up to $N_h = 50$ future steps using the EKF.

Over all configurations, Paparazzo consistently outperforms the baselines in terms of coverage, completeness, and AUC, demonstrating its superior efficiency in reconstructing dynamically moving objects. Its adaptive strategy, alternating between object tracking and mapping modes—allows it to handle different motion complexities more effectively than static or passive baselines.

TABLE 5.8: Quantitative results across 6 scenes for our three dynamic motion types (Bouncing Ball (BB), Forward & Backward (FB) and Stop & Go (SG)). Reported values are averaged over all test objects and 5 runs. Each entry shows Coverage (%), Completeness (cm), and Accuracy (cm).

Motion / Method		Denmark (G)			Ribera (G)			Greigsville (G)			PuKPg4mmafe (M)			GdvgFV5R1Z5 (M)			pLe4wQe7qrG (M)		
		Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC
BB	RW	37.03	3.61	0.12	42.52	1.59	0.11	49.77	2.83	0.19	50.47	3.10	0.21	50.21	3.15	0.31	55.03	1.94	0.34
	RIS	53.75	1.82	0.22	56.95	1.43	0.27	60.85	1.25	0.30	61.25	2.40	0.25	74.61	0.97	0.48	66.95	1.33	0.37
	TO	79.78	**0.66**	0.50	83.77	0.64	0.52	80.48	0.75	0.48	81.01	0.88	0.45	81.72	0.71	0.56	79.64	0.74	**0.47**
	Paparazzo	**80.18**	0.72	**0.52**	**86.50**	**0.57**	**0.56**	**82.43**	**0.63**	**0.51**	**81.40**	**0.85**	**0.48**	**82.87**	**0.65**	**0.58**	**79.89**	**0.71**	0.45
FB	RW	33.86	3.64	0.13	41.56	3.54	0.10	43.21	2.70	0.19	39.65	5.44	0.17	45.85	2.63	0.20	48.02	2.31	0.15
	RIS	36.72	2.59	0.18	45.36	2.89	0.18	57.37	1.05	0.20	58.59	2.32	0.18	61.73	1.58	0.36	50.57	2.13	0.17
	TO	68.99	1.30	0.35	**71.81**	1.03	0.39	**76.39**	**0.84**	0.38	52.48	2.98	0.22	62.77	1.37	0.37	54.05	1.86	0.20
	Paparazzo	**70.31**	**1.03**	**0.41**	71.23	**1.02**	**0.40**	76.04	0.87	**0.40**	**67.02**	**1.61**	**0.32**	**73.06**	**0.91**	**0.47**	**57.10**	**1.31**	**0.23**
SG	RW	40.24	2.81	0.10	37.01	3.34	0.06	42.31	1.90	0.09	40.56	5.25	0.19	61.25	1.85	0.36	43.02	3.14	0.09
	RIS	40.07	3.36	0.19	32.12	3.68	0.21	48.75	1.46	0.25	51.93	2.69	0.16	50.10	2.36	0.29	34.03	4.19	0.05
	TO	52.69	1.91	0.20	52.78	2.01	0.20	47.96	2.60	0.18	74.74	1.08	0.22	73.96	0.95	0.43	48.80	2.12	0.15
	Paparazzo	**57.52**	**1.29**	**0.28**	**61.82**	**1.47**	**0.29**	**60.33**	**1.39**	**0.28**	**76.52**	**1.07**	**0.34**	**74.20**	**0.91**	**0.50**	**65.50**	**1.22**	**0.29**

Bouncing Ball (BB) motion. This could be expected as more complex due to its unpredictable motion when bouncing, reconstruction is generally easier. The object's frequent bounces against walls allow it to be seen from multiple viewpoints, benefiting all methods. In this case, the Tracking-only (TO) baseline achieves reasonably good results, on average 80% of coverage; however, Paparazzo consistently surpasses all baselines, improving coverage compared to TO.

The advantage is even more evident when considering the AUC, where Paparazzo outperforms every baseline across almost all scenes, confirming its efficiency in dynamic object reconstruction. The only exception is the pLe4wQe7qrG scene from MP3D (0.45 vs. 0.47 AUC), primarily because its narrow environment limits the agent's ability to anticipate and reposition before the next bounce. Nonetheless, averaged across all scenes, Paparazzo remains clearly superior for BB motion.

In contrast, the Random Walk (RW) baseline performs poorly, as it disregards the object's position entirely. The Random Informative Selection (RIS) baseline fares slightly better but still falls short of TO and Paparazzo since it lacks synchronization with the object's trajectory.

This causes frequent failures to reach feasible observation points before the object passes, resulting in incomplete reconstructions.

Forward & Backward (FB) motion. This condition is more challenging because the object never rotates, therefore the agent must actively reason about the scene to reposition and capture the object from new viewpoints. Consequently, performance metrics are generally lower. Nevertheless, Paparazzo still achieves higher coverage than all baselines, except in a few specific cases such as Ribera and Greigsville, where it achieves similar results but consistently delivers a higher AUC across all scenes. Notably, in the PuKPg4mmafe scene from MP3D, one of the largest environments of our benchmark, Paparazzo improves AUC by more than 45% over the best baseline. The larger space allows for more frequent mapping and re-planning, amplifying Paparazzo's advantage over simpler or reactive exploration methods. Again, RW and RIS perform poorly due to their lack of object-awareness and temporal anticipation.

Stop & Go (SG) motion. This represents the most challenging scenario, as the object intermittently pauses during motion, introducing unpredictable interruptions that hinder motion estimation. This motion pattern best reflects realistic conditions, where objects may temporarily stop or slow down. Consequently, all methods exhibit a general performance drop compared to the previous motion types.

While Paparazzo still achieves substantially higher coverage and completeness than all baselines, its performance also decreases in this setting due to the intrinsic complexity of handling unexpected stops. In particular, difficulties arise when the agent has already planned to move toward an informative pose where it expects the object to appear, but the object stops beforehand in a region that is not visible. In such cases, the agent continues its motion without any visual confirmation, since the object is no longer observable, and the subsequent re-localization of the object becomes considerably more complex.

Despite this limitation, Paparazzo's adaptive behavior allows to exploit stopping phases effectively whenever the object remains visible, in which case the system continues mapping and refining the reconstruction, fully utilizing the available observation time. This advantage is clearly reflected in the AUC, which still increases by at least 15% across all scenes compared to the best baseline. In contrast, the Tracking-only (TO) mode remains idle, waiting for object motion to resume, highlighting the limitations of a purely passive strategy, while RIS again performs poorly due to its lack of temporal synchronization and anticipation.

Overall, these results highlight that Paparazzo's adaptive alternation between Object Tracking Mode and Object Mapping Mode enables it to robustly handle diverse dynamic scenarios and motion complexities. By effectively balancing tracking accuracy with active exploration, Paparazzo consistently outperforms all baselines across motion types, leveraging motion variations and stop phases to achieve superior reconstruction performance.

To further assess the robustness and generality of the proposed method with respect to object variability, we report in Table 5.9 the quantitative results obtained by Paparazzo and the baseline methods for each target object. This analysis is crucial to demonstrate that our approach offers strong performance over diverse objects; its generality is strengthened by the fact that Paparazzo does not rely on any object-specific training or fine-tuning.

Across all motion types, Paparazzo consistently demonstrated higher performance and greater stability than all baselines. The improvement is particularly significant in Forward & Backward (FB) and Stop & Go (SG) motions, where Paparazzo achieves on average more than 5% higher coverage than the best-performing baseline, along with comparable or better completeness and AUC values. This gain reflects Paparazzo's ability to dynamically balance

TABLE 5.9: Quantitative comparison of different motion types averaged across all scenes. We evaluate three dynamic behaviors (Bouncing Ball (BB), Forward&Backward (FB), and Stop&Go (SG)) and report results for four test objects. Each cell shows Coverage (%), Completeness (cm), and Accuracy (cm).

	Motion / Method	**Object 1**			**Object 2**			**Object 3**			**Object 4**		
		Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC	Cov	Comp	AUC
BB	RW	55.85	2.64	0.26	58.11	1.57	0.21	38.54	3.35	0.20	37.51	3.26	0.25
	RIS	67.50	1.98	0.42	64.84	1.39	0.37	49.34	1.62	0.32	67.89	1.16	0.46
	TO	88.99	0.52	**0.56**	**87.56**	0.55	**0.50**	71.46	0.87	0.49	76.24	0.99	0.49
	Paparazzo	**90.17**	**0.50**	**0.56**	86.93	**0.54**	0.49	**73.82**	**0.83**	**0.52**	**77.93**	**0.88**	**0.53**
FB	RW	48.42	3.25	0.19	50.49	2.36	0.14	34.64	4.34	0.14	34.54	3.54	0.12
	RIS	55.87	2.66	0.24	59.42	1.42	0.25	42.31	2.53	0.21	49.29	1.76	0.26
	TO	68.84	1.46	0.38	69.23	1.26	0.33	57.36	2.01	0.34	62.23	1.52	0.37
	Paparazzo	**73.89**	**0.97**	**0.40**	**70.16**	**1.06**	**0.34**	**64.15**	**1.32**	**0.41**	**64.15**	**1.32**	**0.40**
SG	RW	48.86	3.14	0.22	56.08	1.82	0.18	34.23	3.87	0.12	37.07	3.38	0.18
	RIS	48.11	2.77	0.36	46.93	2.13	0.18	39.55	3.61	0.31	36.74	3.32	0.11
	TO	61.71	1.95	0.31	62.98	1.36	0.24	50.90	2.05	0.27	58.35	1.75	0.32
	Paparazzo	**75.03**	**1.04**	**0.41**	**70.17**	**1.16**	**0.29**	**59.28**	**1.41**	**0.33**	**59.45**	**1.31**	**0.37**

exploration and reconstruction during less predictable or partially observable object trajectories (see Figure 5.12).

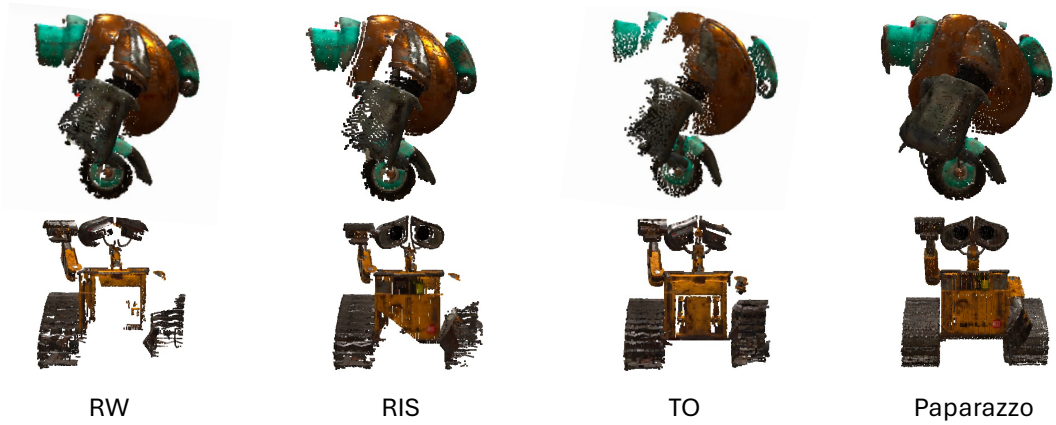


FIGURE 5.12: **Visualization of the 3D reconstruction of Object 3 and Object 4 under Stop & Go motion.** We compare the RW, RIS, and TO baselines against our Paparazzo method. Paparazzo produces significantly more complete and geometrically consistent reconstructions.

In contrast, the Bouncing Ball (BB) motion exhibits a smaller performance margin, with Paparazzo achieving on average +1.41% improvement in coverage over the second-best baseline, TO. As discussed previously, the repetitive and highly predictable bouncing pattern makes this motion type easier for passive methods, particularly Tracking-only (TO). In a single case (Object 2), TO slightly outperforms Paparazzo, likely due to the object's front-facing high-frequency texture details that attract the informative viewpoint selection process toward similar orientations, reducing the effective diversity of captured views. Nonetheless, Paparazzo still maintains overall superior average performance across all objects and metrics.

All these findings confirm that Paparazzo's adaptive policy, alternating between object tracking and active mapping, generalizes effectively across both shape and appearance variations. Its design allows robust handling of heterogeneous objects and motion behaviors without any task-specific training, demonstrating strong potential for deployment in real-world dynamic reconstruction scenarios.

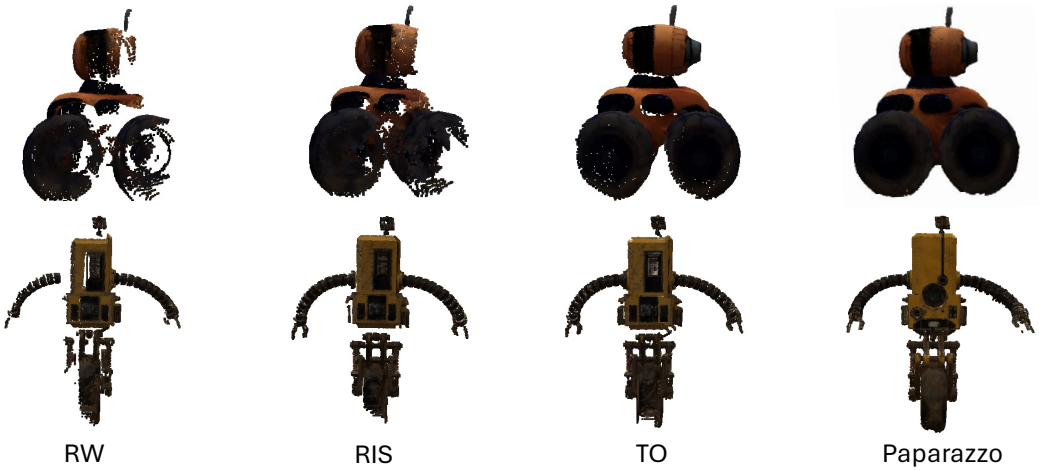


FIGURE 5.13: **Qualitative 3D reconstructions of Object 1 and Object 2 under Stop & Go motion.** Paparazzo consistently produces reconstructions that are significantly more complete, coherent, and stable across both objects. In contrast, the baselines, especially on Object 2, fail to recover the entire frontal surface, leaving substantial portions missing.

5.2.11 Additional Qualitative Results

To further highlight the advantages of Paparazzo over the RW, RIS, and TO baselines, additional qualitative reconstructions are shown in Figure 5.13.

Across all tested objects, Paparazzo produces reconstructions that are both more complete and geometrically more consistent, particularly under challenging motion patterns. This is especially evident in the Stop & Go setting, where baseline methods frequently fail to recover large portions of the object due to motion discontinuities or prolonged static intervals.

In the examples of Object 1 and Object 2 reported in Figure 5.13, Paparazzo achieves substantially higher completeness, reaching 87.23% and 77.45%, respectively. In contrast, all baseline methods exhibit severe missing regions. Notably, for Object 2 the RW, RIS, and TO baselines consistently fail to reconstruct the entire frontal surface of the object, demonstrating their difficulty in handling partial observations and non-smooth trajectories. Paparazzo, instead, successfully recovers these surfaces thanks to its dynamic viewpoint selection and continuous information-driven mapping strategy, which remain effective even when the object stops or follows irregular motions.

The coverage values obtained in this experiment for all methods are summarized in Table 5.10.

TABLE 5.10: Reconstruction coverage (%) for Object 1 and Object 2 under the *Stop & Go* motion pattern. Paparazzo significantly outperforms all baselines.

Method	**Object 1**	**Object 2**
RW	37.24	39.42
RIS	51.12	49.23
TO	78.25	53.24
Paparazzo	**87.23**	**77.45**

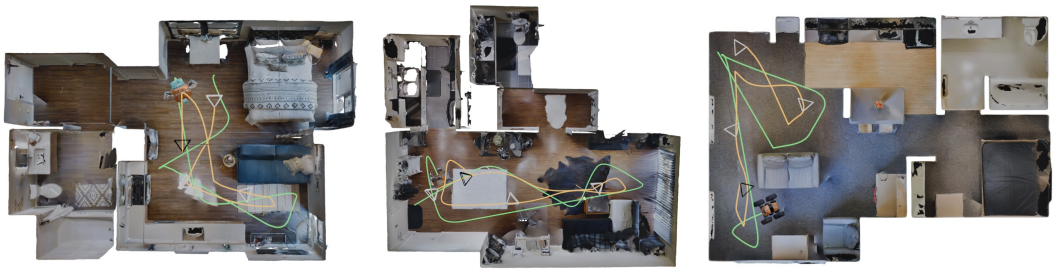


FIGURE 5.14: **Examples of object trajectories**. The moving target is performing the Bouncing Ball motion in the Denmark, Greigsville, and Ribera scenes (left to right). The agent executes the Paparazzo framework while continuously adapting its motion to track and map the moving object.



(A) TO (B) Paparazzo

FIGURE 5.15: **Illustration of Stop & Go motion.** Both panels show the same Stop & Go trajectory executed by the moving object. Across 300 steps, the object pauses once. (a) TO: the agent remains passive during the stop phase, losing valuable time and collecting no new viewpoints, which prevents further progress in the reconstruction. (b) Paparazzo: the agent continues to actively reposition and capture informative views even while the object is stationary, as visible from the additional exploratory camera frustums (in gray). This allows Paparazzo to maintain reconstruction progress during motion interruptions, unlike the TO baseline.

5.2.12 Additional Analysis

To provide a deeper understanding of Paparazzo's behavior during reconstruction, in Figure 5.14 we show three examples where the object follows a Bouncing Ball (BB) motion across three distinct scenes—Denmark, Greigsville, and Ribera. The black frustum marks the initial camera pose, after which the agent alternates between Object Tracking Mode and Object Mapping Mode, continuously adapting its trajectory to maximize reconstruction quality. These examples highlight the robustness of our approach: Paparazzo consistently produces purposeful and informative motion across different environments and object types, without requiring any object-specific tuning or prior scene knowledge.

For completeness, Figure 5.15 presents an example of a Stop & Go motion sequence in the Denmark scene, visualized over the first 300 steps for clarity. The blurred instance of the object marks its initial position, whereas the sharper instance indicates the location where it remained stationary for roughly 150 steps.

This qualitative comparison highlights a key limitation of the Tracking-Only (TO) baseline (Figure 5.15a): although TO operates effectively when the object is continuously moving, it fails when the object stops. In these situations, TO simply remains static, illustrated by the grey camera frustum, waiting passively for motion to resume.

In contrast, Paparazzo (Figure 5.15b) continues reconstructing autonomously even during long standstills. It keeps evaluating informative viewpoints, navigating toward them, and acquiring new observations. As a result, Paparazzo maintains reconstruction progress and achieves a more complete and temporally consistent model of the object, despite the absence of motion.

5.2.13 Discussion and Limitations

In this work, we introduced the task of active mapping of a rigid moving object, a problem setting that departs from the long-standing assumption of static scenes underlying most exploration and reconstruction approaches. By explicitly considering object motion as a first-class component of the mapping process, this work opens a new perspective on active perception in dynamic environments.

We proposed Paparazzo, a learning-free framework that integrates motion prediction, information-driven viewpoint selection, and adaptive agent behavior conditioned on the target object's dynamics. Experimental results demonstrate that Paparazzo consistently outperforms existing baselines, highlighting the importance of explicitly reasoning about where to look, when to look, and how to adapt the agent's motion in response to object dynamics. By continuously balancing viewpoint informativeness, motion feasibility, and field-of-view maintenance, Paparazzo enables efficient and accurate reconstruction of moving objects and facilitates a form of dynamic scene understanding that has remained largely unexplored.

Despite these promising results, several limitations remain. First, the framework assumes that the Extended Kalman Filter (EKF) provides sufficiently accurate motion predictions over a finite planning horizon, which implicitly requires object motion to be smooth and moderately predictable. In scenarios involving abrupt, discontinuous, or highly non-linear trajectories, the EKF may yield unreliable predictions, thereby degrading planning performance. Future work will explore more expressive and uncertainty-aware motion models, such as particle filters or learned dynamics predictors, to better accommodate complex and irregular motion patterns.

Second, the proposed system depends on reliable initial object detection and segmentation. Failures at this stage—caused by occlusions, adverse lighting conditions, or ambiguous textual prompts—can propagate through the pipeline, resulting in incorrect tracking or failure to trigger the active reconstruction process. Incorporating more robust multi-frame tracking, object re-identification mechanisms, and uncertainty-aware re-acquisition strategies would significantly enhance robustness in real-world deployments.

Overall, these findings establish a strong foundation for future research on predictive active mapping in dynamic environments. Extending the proposed framework to handle multiple moving objects, continuous real-world sensing, and richer motion priors represents a promising direction toward general-purpose active perception systems in robotics.

In conclusion, this work demonstrates that active viewpoint planning coupled with motion-aware behavior is a key ingredient for extending 3D reconstruction beyond static scenes and toward truly dynamic, real-world environments.

Chapter 6

Conclusion

This thesis has addressed one of the central challenges in modern robotics: enabling robots to perceive, understand, and interact intelligently within dynamic human-robot collaboration scenarios, advancing the transition from rigid automation towards flexible collaboration. To realize this vision, we contributed along three tightly connected pillars: Robot-Centric Perception, Human Understanding and Robot-Object Interaction.

Robot-Centric Perception The foundation of any intelligent robotic system is the ability to establish a precise and consistent geometric link between perception and the robot's reference frame. To address this, we developed a versatile calibration framework validated across a variety of robotic configurations, from fixed robot arms to mobile platforms.

- For fixed robot manipulators, we introduced both joint optimization and graph-based strategies for multi-camera calibration. These methods achieved superior accuracy and global consistency compared to independent, single-camera approaches. Our contributions, validated on the METRIC dataset and against other multi-camera methods, enabled the accurate calibration of camera networks surrounding robot arms.

- For mobile platforms, we developed MEMROC, a motion-based approach that exploits planar motion and robust ground-plane constraints to recover full 6-DoF camera poses, even under significant odometry noise. This effort culminated in Calib3R, a novel, patternless calibration method that integrates robot motion with geometric cues from a 3D foundation model (MASt3R), enabling metric 3D reconstruction and calibration without requiring calibration patterns. Calib3R is applicable to both mobile robots and manipulators, and it outperforms traditional pattern-based methods, marking a notable advancement in the field.

Human Understanding To support effective HRC, robots must interpret human body posture and behavior in real-world scenarios. We developed a multi-view 3D human pose estimation method robust to clutter and occlusions. By fusing monocular predictions with reprojection constraints, our method outperformed both traditional 2D triangulation and state-of-the-art learned methods, especially under occlusion-heavy conditions. Building on this, we introduced an online Human Action Recognition (HAR) framework capable of real-time operation. Using an ensemble of lightweight binary classifiers, the system continuously recognizes specific industrial actions from a live 3D skeleton stream while filtering irrelevant or unknown motions.

This line of work culminated in a successful real-world deployment. Specifically, the integration of our human understanding framework with the multi-camera joint calibration system was validated in an actual industrial human-robot collaboration scenario, where a robot arm worked alongside humans during carbon fiber production. The robot-centric perception framework enabled the system to accurately estimate 3D human poses with respect to

the robot reference frame and adapt robot behavior in real time, showcasing safe, intelligent, and context-aware interaction in a complex, real-world environment.

Robot-Object Interaction The third pillar focuses on enabling adaptive and safe robot interaction with the physical environment. In manipulation tasks, our DREMA framework integrates hybrid Gaussian Splatting with a learned physics-based world model to simulate photorealistic, causally consistent interactions. These high-fidelity simulations were used to generate synthetic data that substantially enhanced the performance of few-shot imitation learning policies. This demonstrates that synthetic data, when grounded in physically plausible interactions, can effectively teach robots to adapt to new object configurations, reducing the need for time-consuming manual data collection by human operators. For mobile robots, we introduced an active mapping framework focused on the reconstruction of dynamic objects. By combining an Extended Kalman Filter (EKF) for motion prediction with Fisher Information-based viewpoint selection, the system proactively plans future trajectories to intercept and observe moving targets. This approach resulted in drastically improved 3D reconstruction quality, achieving over 80% object completeness, significantly outperforming both static and reactive mapping baselines.

6.1 Take-Home Messages and Scientific Impact

The synergy between the three pillars explored in this thesis constitutes its core scientific contribution. Robot-centric perception provides the geometric grounding required for consistent observation in multi-camera HRC scenarios. Human understanding builds upon this foundation by adding semantic interpretation of human behavior, enabling safe and intuitive interaction. Robot-object interaction completes the loop by allowing robots to act effectively in the physical world, either by generalizing manipulation skills through learned world models (DREMA) or by actively exploring dynamic scenes via information-driven active mapping. Together, these contributions promote an integrated view of robotic intelligence, in which calibration, perception, and action are not treated as independent modules, but as tightly coupled components of a single perceptual and decision-making pipeline.

Beyond the specific technical results, this thesis reflects a broader insight into the evolution of robotic perception. Robust and adaptable robotic intelligence cannot be achieved through loosely connected perception modules, but requires representations that directly link spatial geometry, semantic understanding, and physical interaction, all grounded in the robot's own reference frame.

From this perspective, robot-centric perception should be understood not merely as a calibration task, but as a fundamental principle of robotic intelligence. Anchoring perception to the robot's reference frame is essential for reliable reasoning about humans, objects, and environmental dynamics, and supports transferability across platforms, tasks, and deployment conditions.

Within this framework, foundation models are reshaping the way calibration and perception are approached. Their robustness and generalization capabilities reduce the dependence on carefully designed calibration patterns, which are often impractical in real-world settings. In particular, recent 3D foundation models allow geometric scene representations to be integrated directly into the calibration process, avoiding separate reconstruction stages. In this sense, Calib3R represents an initial step toward self-supervised extrinsic calibration, where spatial alignment emerges naturally from motion, interaction, and perception.

Looking ahead, a key long-term direction is the transition toward task-driven and continual calibration. In such a paradigm, calibration becomes an implicit and ongoing process, driven by the robot's need to act effectively while remaining aware of its three-dimensional

environment, human partners, and dynamic objects. Achieving this level of autonomy in unstructured, human-centered environments remains one of the central challenges in robotics, and, in my view, a critical step toward truly intelligent and collaborative robotic systems.

6.2 Limitations and Future Directions

While the proposed approaches mark significant progress, several limitations remain, opening up promising future research directions:

- Patternless calibration for external cameras: currently, Calib3R is limited to onboard cameras (on manipulators or mobile platforms). Extending it to external camera networks (e.g., surrounding a robot arm) remains an open problem. Our joint optimization framework with calibration patterns partially addresses this, but a fully patternless approach for external networks is a key next step.
- Open-set human understanding: the HAR framework relies on a predefined dictionary (one binary classifier per action). As such, it cannot handle novel or "open-set" behaviors without retraining. A promising direction is to explore Visual Language Models (VLMs) or generative architectures that enable zero-shot or open-set recognition of new actions and intentions.
- Learned predictive world models: the motion prediction in our active mapping framework currently uses a simple EKF with a constant-velocity assumption. This limits performance in scenarios involving complex object dynamics. Future work should explore learned generative world models, building upon the ideas in DREMA, to predict scene evolution and object behavior more accurately over long horizons.

This thesis presented a unified framework for advancing robotic perception and interaction in dynamic, human-centric environments. By grounding perception in the robot's own reference frame, combining spatial and semantic understanding, and enabling adaptive, predictive behaviors, this work moves us closer to the vision of autonomous systems capable of safe, intelligent, and meaningful collaboration with humans.

Through the release of open-source tools and the definition of clear next steps, this work lays a solid foundation for future collaborations, extensions, and real-world applications, and we hope it serves as both a contribution and an inspiration for the broader research community.

Bibliography

[1] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[2] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[3] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[4] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[5] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020.

[6] Davide Allegro, Matteo Terreran, and Stefano Ghidoni. Memroc: Multi-eye to mobile robot calibration. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 884–891. IEEE, 2024.

[7] Xuesong Shi, Dongjiang Li, Pengpeng Zhao, Qinbin Tian, Yuxin Tian, Qiwei Long, Chunhao Zhu, Jingwei Song, Fei Qiao, Le Song, et al. Are we ready for service robots? the openloris-scene datasets for lifelong slam. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 3139–3145. IEEE, 2020.

[8] Harin Park, Inha Lee, Minje Kim, Hyungyu Park, and Kyungdon Joo. A benchmark dataset for collaborative slam in service environments. *IEEE Robotics and Automation Letters*, 2024.

[9] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[10] Piotr Barosz, Grzegorz Gołda, and Adrian Kampa. Efficiency analysis of manufacturing line with industrial robots and human operators. *Applied Sciences*, 10(8):2862, 2020.

[11] Martin Hägele, Klas Nilsson, J Norberto Pires, and Rainer Bischoff. Industrial robotics. In *Springer handbook of robotics*, pages 1385–1422. Springer, 2016.

[12] S Saravana Perumaal and N Jawahar. Automated trajectory planner of industrial robot for pick-and-place task. *International Journal of Advanced Robotic Systems*, 10(2):100, 2013.

[13] Isak Karabegović and Riaz Mirza. Automation of the welding process by use of industrial robots. In *International Conference "New Technologies, Development and Applications"*, pages 3–17. Springer, 2018.

[14] Tamas Czimmermann, Marcello Chiurazzi, Mario Milazzo, Stefano Roccella, Marco Barbieri, Paolo Dario, Calogero Maria Oddo, and Gastone Ciuti. An autonomous robotic platform for manipulation and inspection of metallic surfaces in industry 4.0. *IEEE Transactions on Automation Science and Engineering*, 19(3):1691–1706, 2021.

[15] Dimitris Mourtzis, John Angelopoulos, and Nikos Panopoulos. A literature review of the challenges and opportunities of the transition from industry 4.0 to society 5.0. *Energies*, 15(17):6276, 2022.

[16] Alessio Baratta, Antonio Cimino, Maria Grazia Gnoni, and Francesco Longo. Human robot collaboration in industry 4.0: a literature review. *Procedia Computer Science*, 217:1887–1895, 2023.

[17] AS Mohamed Sahan, S Kathiravan, M Lokesh, and R Raffik. Role of cobots over industrial robots in industry 5.0: a review. In *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, pages 1–5. IEEE, 2023.

[18] Ya A Shchenikov, OS Gromova, and EA Smeshchuk. The human-cobot collaboration in mounting and assembly operations in instrument making. In *Journal of Physics: Conference Series*, volume 1889, page 042055. IOP Publishing, 2021.

[19] Natanael Magno Gomes, Felipe Nascimento Martins, José Lima, and Heinrich Wörtche. Reinforcement learning for collaborative robots pick-and-place applications: A case study. *Automation*, 3(1):223–241, 2022.

[20] Alena Pauliková, Zdenka Gyurák Babel'ová, and Monika Ubárová. Analysis of the impact of human–cobot collaborative manufacturing implementation on the occupational health and safety and the quality requirements. *International Journal of Environmental Research and Public Health*, 18(4):1927, 2021.

[21] Uchenna Emeoha Ogenyi, Jinguo Liu, Chenguang Yang, Zhaojie Ju, and Honghai Liu. Physical human–robot collaboration: Robotic systems, learning methods, collaborative strategies, sensors, and actuators. *IEEE transactions on cybernetics*, 51(4):1888–1901, 2019.

[22] Seung Jae Moon, Jinsol Kim, Hongsik Yim, Yeeun Kim, and Hyouk Ryeol Choi. Real-time obstacle avoidance using dual-type proximity sensor for safe human-robot interaction. *IEEE Robotics and Automation Letters*, 6(4):8021–8028, 2021.

[23] Jianfeng Jiang, Xiao Luo, Qingsheng Luo, Lijun Qiao, and Minghao Li. An overview of hand-eye calibration. *The International Journal of Advanced Manufacturing Technology*, 119(1):77–97, 2022.

[24] Zachary Taylor and Juan Nieto. Motion-based calibration of multimodal sensor extrinsics and timing offset estimation. *IEEE Transactions on Robotics*, 32(5):1215–1229, 2016.

[25] Justinas Miseikis, Kyrre Glette, Ole Jakob Elle, and Jim Torresen. Automatic calibration of a robot manipulator and multi 3d camera system. In *2016 IEEE/SICE International Symposium on System Integration (SII)*, pages 735–741. IEEE, 2016.

[26] Aquib Rashd, Wolfram Hardt, Alexey Kolker, Mohamad Bdiwi, and Matthias Putz. Open-box target for extrinsic calibration of lidar, camera and industrial robot. In *2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA)*, pages 121–125. IEEE, 2020.

[27] Bartolomeo Della Corte, Henrik Andreasson, Todor Stoyanov, and Giorgio Grisetti. Unified motion-based calibration of mobile multi-sensor platforms with time delay estimation. *IEEE Robotics and Automation Letters*, 4(2):902–909, 2019.

[28] Chao X Guo, Faraz M Mirzaei, and Stergios I Roumeliotis. An analytical least-squares solution to the odometer-camera extrinsic calibration problem. In *2012 IEEE International Conference on Robotics and Automation*, pages 3962–3968. IEEE, 2012.

[29] Zhuo Chen, Xu Zhao, and Xiaoyue Wan. Structural triangulation: A closed-form solution to constrained 3d human pose estimation. In *European Conference on Computer Vision*, pages 695–711. Springer, 2022.

[30] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7718–7727, 2019.

[31] Zhe Zhang, Chunyu Wang, Weichao Qiu, Wenhu Qin, and Wenjun Zeng. Adafuse: Adaptive multiview fusion for accurate human pose estimation in the wild. *International Journal of Computer Vision*, 129(3):703–718, 2021.

[32] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[33] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019.

[34] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019.

[35] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[36] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[37] Jad Abou-Chakra, Krishan Rana, Feras Dayoub, and Niko Sünderhauf. Physically embodied gaussian splatting: A realtime correctable world model for robotics. *arXiv preprint arXiv:2406.10788*, 2024.

[38] Zike Yan, Haoxiang Yang, and Hongbin Zha. Active neural mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10981–10992, 2023.

[39] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *European conference on computer vision*, pages 400–418. Springer, 2020.

[40] Georgios Georgakis, Bernadette Bucher, Anton Arapin, Karl Schmeckpeper, Nikolai Matni, and Kostas Daniilidis. Uncertainty-driven planner for exploration and navigation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11295–11302. IEEE, 2022.

[41] Wen Jiang, Boshu Lei, and Kostas Daniilidis. Fisherrf: Active view selection and uncertainty quantification for radiance fields using fisher information. *arXiv preprint arXiv:2311.17874*, 2023.

[42] Davide Allegro, Matteo Terreran, and Stefano Ghidoni. Calib3r: A 3d foundation model for multi-camera to robot calibration and 3d metric-scaled scene reconstruction. *arXiv preprint arXiv:2509.08813*, 2025.

[43] Niccolò Turcato, Giulio Giacomuzzo, Matteo Terreran, Davide Allegro, Ruggero Carli, and Alberto Dalla Libera. Data efficient robotic object throwing with model-based reinforcement learning. *arXiv preprint arXiv:2502.05595*, 2025.

[44] Davide Allegro, Matteo Terreran, and Stefano Ghidoni. Multi-camera hand-eye calibration for human-robot collaboration in industrial robotic workcells. *IEEE Robotics and Automation Letters*, 2024.

[45] Davide Allegro, Matteo Terreran, and Stefano Ghidoni. Metric—multi-eye to robot indoor calibration dataset. *Information*, 14(6):314, 2023.

[46] Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. *arXiv preprint arXiv:2412.14957*, 2024.

[47] Matteo Terreran, Laura Bragagnolo, Davide Allegro, and Stefano Ghidoni. A lightweight ensemble framework for online skeleton-based human action recognition in industrial environments. In *2025 European Conference on Mobile Robots (ECMR)*, pages 1–8. IEEE, 2025.

[48] Laura Bragagnolo, Matteo Terreran, Davide Allegro, and Stefano Ghidoni. Multi-view pose fusion for occlusion-aware 3d human pose estimation. In *European Conference on Computer Vision*, pages 117–133. Springer, 2024.

[49] Alberto Bacchin, Davide Allegro, Stefano Ghidoni, and Emanuele Menegatti. Sood-imagenet: a large-scale dataset for semantic out-of-distribution image classification and semantic segmentation. In *European Conference on Computer Vision*, pages 80–97. Springer, 2024.

[50] Daniele Evangelista, Emilio Olivastri, Davide Allegro, Emanuele Menegatti, and Alberto Pretto. A graph-based optimization framework for hand-eye calibration for multi-camera setups. *arXiv preprint arXiv:2303.04747*, 2023.

[51] Daniele Evangelista, Davide Allegro, Matteo Terreran, Alberto Pretto, and Stefano Ghidoni. An unified iterative hand-eye calibration method for eye-on-base and eye-in-hand setups. In *2022 IEEE 27th international conference on emerging technologies and factory automation (ETFA)*, pages 1–7. IEEE, 2022.

[52] Haolin Fei, Ziwei Wang, Stefano Tedeschi, and Andrew Kennedy. Boosting visual servoing performance through rgb-based methods. *Robotic Intelligence and Automation*, 43(4):468–475, 2023.

[53] Heng Li, Xiaodong Gu, Weihao Yuan, Luwei Yang, Zilong Dong, and Ping Tan. Dense rgb slam with neural implicit maps. *arXiv preprint arXiv:2301.08930*, 2023.

[54] Zitian Wang, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Object as query: Lifting any 2d object detector to 3d detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3791–3800, 2023.

[55] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer graphics forum*, volume 37, pages 625–652. Wiley Online Library, 2018.

[56] Vassileios Balntas, Andreas Doumanoglou, Caner Sahin, Juil Sock, Rigas Kouskouridas, and Tae-Kyun Kim. Pose guided rgbd feature learning for 3d object pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3856–3864, 2017.

[57] Ahmad Jalal, Yeonho Kim, and Daijin Kim. Ridge body parts features for human pose estimation and recognition from rgb-d video data. In *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–6. IEEE, 2014.

[58] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2002.

[59] Jean-Yves Bouguet and Pietro Perona. Camera calibration from points and lines in dual-space geometry. In *Proc. 5th European Conf. on Computer Vision*, pages 2–6. Citeseer, 1998.

[60] Xiao Xin Lu. A review of solutions for perspective-n-point problem in camera pose estimation. In *Journal of Physics: Conference Series*, volume 1087, page 052009. IOP Publishing, 2018.

[61] Roger Y Tsai, Reimar K Lenz, et al. A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration. *IEEE Transactions on robotics and automation*, 5(3):345–358, 1989.

[62] Hanqi Zhuang, Zvi S Roth, and Raghavan Sudhakar. Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form ax= yb. *IEEE Transactions on Robotics and Automation*, 10(4):549–554, 1994.

[63] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[64] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[65] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[66] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[67] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[68] Shiqi Li, Chi Xu, and Ming Xie. A robust o (n) solution to the perspective-n-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1444–1450, 2012.

[69] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[70] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International journal of computer vision*, 35(2):151–173, 1999.

[71] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[72] Santiago A Cadena, George H Denfield, Edgar Y Walker, Leon A Gatys, Andreas S Tolias, Matthias Bethge, and Alexander S Ecker. Deep convolutional models improve predictions of macaque v1 responses to natural images. *PLoS computational biology*, 15(4):e1006897, 2019.

[73] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[74] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[75] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[76] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[77] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[78] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[79] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 750–767, 2018.

[80] Sungheon Park, Jihye Hwang, and Nojun Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *European Conference on Computer Vision*, pages 156–169. Springer, 2016.

[81] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[82] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[83] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[84] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[85] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[86] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[87] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

[88] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[89] Yiu Cheung Shiu and Shaheen Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax= xb. *IEEE journal of robotics and automation*, 1987.

[90] Jack CK Chou and M Kamel. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *The international journal of robotics research*, 10(3):240–254, 1991.

[91] Frank C Park and Bryan J Martin. Robot sensor calibration: solving ax= xb on the euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5):717–721, 1994.

[92] Konstantinos Daniilidis and Eduardo Bayro-Corrochano. The dual quaternion approach to hand-eye calibration. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 318–322. IEEE, 1996.

[93] Rong-hua Liang and Jian-fei Mao. Hand-eye calibration with a new linear decomposition algorithm. *Journal of Zhejiang University-SCIENCE A*, 9(10):1363–1368, 2008.

[94] Aiguo Li, Lin Wang, and Defeng Wu. Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product. *Int. J. Phys. Sci*, 5(10):1530–1536, 2010.

[95] Mili Shah. Solving the robot-world/hand-eye calibration problem using the kronecker product. *Journal of Mechanisms and Robotics*, 5(3):031007, 2013.

[96] Amy Tabb and Khalil M Ahmad Yousef. Parameterizations for reducing camera reprojection error for robot-world hand-eye calibration. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3030–3037. IEEE, 2015.

[97] Kenji Koide and Emanuele Menegatti. General hand–eye calibration based on reprojection error minimization. *IEEE Robotics and Automation Letters*, 4(2):1021–1028, 2019.

[98] Yifu Wang, Wenqing Jiang, Kun Huang, Sören Schwertfeger, and Laurent Kneip. Accurate calibration of multi-perspective cameras from a generalization of the hand-eye constraint. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1244–1250. IEEE, 2022.

[99] Amy Tabb and Khalil M Ahmad Yousef. Solving the robot-world hand-eye (s) calibration problem with iterative methods. *Machine Vision and Applications*, 28(5):569–590, 2017.

[100] Radu Horaud and Fadi Dornaika. Hand-eye calibration. *The international journal of robotics research*, 14(3):195–210, 1995.

[101] Andrea Censi, Antonio Franchi, Luca Marchionni, and Giuseppe Oriolo. Simultaneous calibration of odometry and sensor parameters for mobile robots. *IEEE Transactions on Robotics*, 29(2):475–492, 2013.

[102] Rainer Kümmerle, Giorgio Grisetti, and Wolfram Burgard. Simultaneous calibration, localization, and mapping. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3716–3721. IEEE, 2011.

[103] Lionel Heng, Bo Li, and Marc Pollefeys. Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1793–1800. IEEE, 2013.

[104] David Zuñiga-Noël, Jose-Raul Ruiz-Sarmiento, Ruben Gomez-Ojeda, and Javier Gonzalez-Jimenez. Automatic multi-sensor extrinsic calibration for mobile robots. *IEEE Robotics and Automation Letters*, 4(3):2862–2869, 2019.

[105] Stephanie W Meyer, Howard Chen, and David M Bevly. Automatic extrinsic rotational calibration of lidar sensors and vehicle orientation estimation. *IFAC-PapersOnLine*, 54(20):424–429, 2021.

[106] Guohang Yan, Zhaotong Luo, Zhuochun Liu, and Yikang Li. Sensorx2car: Sensors-to-car calibration for autonomous driving in road scenarios. *arXiv preprint arXiv:2301.07279*, 2023.

[107] Luca Tagliapietra, Elisa Tosello, and E Menegatti. Curami: human-robot collaboration for intelligent assembly tasks. In *Seconda Conferenza Italiana di Robotica e Macchine Intelligenti, I-RIM*, 2020.

[108] Seungwoong Gwak, Junggon Kim, and Frank Chongwoo Park. Numerical optimization on the euclidean group with applications to camera calibration. *IEEE Transactions on Robotics and Automation*, 19(1):65–74, 2003.

[109] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *2011 IEEE international conference on robotics and automation*, pages 3607–3613. IEEE, 2011.

[110] Amy Tabb, Henry Medeiros, Mitchell J Feldmann, and Thiago T Santos. Calibration of asynchronous camera networks: Calico. *arXiv preprint arXiv:1903.06811*, 2019.

[111] Nicolas Andreff, Radu Horaud, and Bernard Espiau. On-line hand-eye calibration. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No. PR00062)*, pages 430–436. IEEE, 1999.

[112] Matteo Terreran, Stefano Ghidoni, Emanuele Menegatti, Enrico Villagrossi, Nicola Pedrocchi, Nicola Castaman, Alberto Gottardi, Christian Eitzinger, Luca Vescovi, Giuseppe Salemi, et al. A smart workcell for cooperative assembly of carbon fiber parts guided by human actions. In *Atti della 4ª Conferenza Italiana di Robotica e Macchine Intelligenti*, 2022.

[113] Wang Yuan, Zhijun Li, and Chun-Yi Su. Multisensor-based navigation and control of a mobile service robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(4):2624–2634, 2019.

[114] Kai-Wei Chiang, Guang-Je Tsai, HW Chang, Cyril Joly, and N Ei-Sheimy. Seamless navigation and mapping using an ins/gnss/grid-based slam semi-tightly coupled integration scheme. *Information Fusion*, 50:181–196, 2019.

[115] Lin Li, Fan Yang, Haihong Zhu, Dalin Li, You Li, and Lei Tang. An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells. *Remote Sensing*, 9(5):433, 2017.

[116] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023.

[117] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[118] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1(1):1–5, 2017.

[119] Fan Zheng and Yun-Hui Liu. Visual-odometric localization and mapping for ground vehicles using se (2)-xyz constraints. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3556–3562. IEEE, 2019.

[120] Xin Wei, Jixin Lv, Jie Sun, Erbao Dong, and Shiliang Pu. Gclo: Ground constrained lidar odometry with low-drifts for gps-denied indoor environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2229–2235. IEEE, 2022.

[121] Bo Li, Lionel Heng, Kevin Koser, and Marc Pollefeys. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1301–1307. IEEE, 2013.

[122] Weiming Zhi, Haozhan Tang, Tianyi Zhang, and Matthew Johnson-Roberson. Unifying scene representation and hand-eye calibration with 3d foundation models. *arXiv preprint arXiv:2404.11683*, 2024.

[123] Nicolas Andreff, Radu Horaud, and Bernard Espiau. Robot hand-eye calibration using structure-from-motion. *The International Journal of Robotics Research*, 20(3):228–248, 2001.

[124] Jan Heller, Michal Havlena, Akihiro Sugimoto, and Tomas Pajdla. Structure-from-motion based hand-eye calibration using l_∞ minimization. In *CVPR 2011*, pages 3497–3503. IEEE, 2011.

[125] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MASt3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *International Conference on 3D Vision 2025*, 2025.

[126] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *Proceedings of the IEEE international conference on computer vision*, pages 1401–1408, 2013.

[127] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[128] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[129] Toufik Benmessabih, Rim Slama, Vincent Havard, and David Baudry. Online human motion analysis in industrial context: A review. *Engineering Applications of Artificial Intelligence*, 131:107850, 2024.

[130] Giulio Giacomuzzo, Matteo Terreran, Siddarth Jain, and Diego Romeres. Decaf: a discrete-event based collaborative human-robot framework for furniture assembly. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7085–7091. IEEE, 2024.

[131] Kunyu Peng, Alina Roitberg, Kailun Yang, Jiaming Zhang, and Rainer Stiefelhagen. Delving deep into one-shot skeleton-based action recognition with diverse occlusions. *IEEE Transactions on Multimedia*, 25:1489–1504, 2023.

[132] Matteo Terreran, Leonardo Barcellona, and Stefano Ghidoni. A general skeleton-based action and gesture recognition framework for human–robot collaboration. *Robotics and Autonomous Systems*, 170:104523, 2023.

[133] Matteo Terreran, Margherita Lazzaretto, and Stefano Ghidoni. Skeleton-based action and gesture recognition for human-robot collaboration. In *International Conference on Intelligent Autonomous Systems*, pages 29–45. Springer, 2022.

[134] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13041–13051, 2023.

[135] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13232–13242, 2022.

[136] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *3DV*, pages 42–52. IEEE, 2021.

[137] Tianshu Zhang, Buzhen Huang, and Yangang Wang. Object-occluded human shape and pose estimation from a single color image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7376–7385, 2020.

[138] Michele Boldo, Mirco De Marchi, Enrico Martini, Stefano Aldegheri, Davide Quaglia, Franco Fummi, and Nicola Bombieri. Real-time multi-camera 3d human pose estimation at the edge for industrial applications. *Expert Syst. Appl.*, 252:124089, 2024.

[139] Mattia Guidolin, Luca Tagliapietra, Emanuele Menegatti, and Monica Reggiani. Hiros: Open-source multi-camera sensor fusion for real-time people tracking. *Comput. Vis. Image Und.*, 232:103694, 2023.

[140] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339, 2013.

[141] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformer for multi-view human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 1036–1037, 2020.

[142] Haoyu Ma, Liangjian Chen, Deying Kong, Zhe Wang, Xingwei Liu, Hao Tang, Xiangyi Yan, Yusheng Xie, Shih-Yao Lin, and Xiaohui Xie. Transfusion: Cross-view fusion with transformer for 3d human pose estimation. *arXiv preprint arXiv:2110.09554*, 2021.

[143] Olivier Moliner, Sangxia Huang, and Kalle Åström. Geometry-biased transformer for robust multi-view 3d human pose reconstruction. In *FG*, pages 1–8. IEEE, 2024.

[144] Xiaoyue Wan, Zhuo Chen, and Xu Zhao. View consistency aware holistic triangulation for 3d human pose estimation. *Comput. Vis. Image Und.*, 236:103830, 2023.

[145] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13147–13156, 2022.

[146] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *Int. Conf. Comput. Vis.*, pages 11127–11137, 2021.

[147] Yi Zhang, Pengliang Ji, Angtian Wang, Jieru Mei, Adam Kortylewski, and Alan Yuille. 3d-aware neural body fitting for occlusion robust 3d human pose estimation. In *Int. Conf. Comput. Vis.*, pages 9399–9410, 2023.

[148] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 459–468, 2018.

[149] Yu Zhan, Fenghai Li, Renliang Weng, and Wongun Choi. Ray3d: ray-based 3d human pose estimation for monocular absolute 3d localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13116–13125, 2022.

[150] István Sárándi, Timm Linder, Kai Oliver Arras, and Bastian Leibe. Metrabs: metric-scale truncation-robust heatmaps for absolute 3d human pose estimation. *IEEE Trans. Biom. Behav. Identity Sci.*, 3(1):16–30, 2020.

[151] István Sárándi, Alexander Hermans, and Bastian Leibe. Learning 3d human pose estimation from dozens of datasets using a geometry-aware autoencoder to bridge between skeleton formats. In *WACV*, pages 2956–2966, 2023.

[152] István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. How robust is 3d human pose estimation to occlusion? *arXiv preprint arXiv:1808.09316*, 2018.

[153] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T Tan. Occlusion-aware networks for 3d human pose estimation in video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 723–732, 2019.

[154] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[155] Stefano Ghidoni, Matteo Terreran, Daniele Evangelista, Christian Eitzinger, Sebastian Zambal, Enrico Villagrossi, Nicola Pedrocchi, Nicola Castaman, Marcin Malecha, et al. A smart workcell for human-robot cooperative assembly of carbon fiber parts. In *Atti della III Conferenza Italiana di Robotica e Macchine Intelligenti*, 2021.

[156] Martin Schepers, Matteo Giuberti, Giovanni Bellusci, et al. Xsens mvn: Consistent tracking of human motion using inertial sensing. *Xsens Technol*, 1(8):1–8, 2018.

[157] Matteo Munaro, Filippo Basso, and Emanuele Menegatti. Openptrack: Open source multi-camera calibration and people tracking for rgb-d camera networks. *RAS*, 75:525–538, 2016.

[158] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *3DV*, pages 120–130. IEEE, 2018.

[159] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10975–10985, 2019.

[160] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3334–3342, 2015.

[161] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John P Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Brit. Mach. Vis. Conf.*, volume 2, pages 1–13. London, UK, 2017.

[162] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[163] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human action recognition from various data modalities: A review. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3200–3225, 2022.

[164] Qianqian Xiong, Jianjing Zhang, Peng Wang, Dongdong Liu, and Robert X Gao. Transferable two-stream convolutional neural network for human action recognition. *Journal of Manufacturing Systems*, 56:605–614, 2020.

[165] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[166] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[167] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[168] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1963–1978, 2019.

[169] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Pyskl: Towards good practices for skeleton action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 7351–7354, 2022.

[170] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021.

[171] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(11):4125–4141, 2021.

[172] Jun Liu, Amir Shahroudy, Gang Wang, Ling-Yu Duan, and Alex C Kot. Skeleton-based online action prediction using scale selection network. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1453–1467, 2019.

[173] Mejdi Dallel, Vincent Havard, Yohan Dupuis, and David Baudry. A sliding window based approach with majority voting for online human action recognition using spatial temporal graph convolutional neural networks. In *Proceedings of the 2022 7th International Conference on Machine Learning Technologies*, pages 155–163, 2022.

[174] Joungbin An, Hyolim Kang, Su Ho Han, Ming-Hsuan Yang, and Seon Joo Kim. Miniroad: Minimal rnn framework for online action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10341–10350, 2023.

[175] Seunggeun Chi, Hyung-gun Chi, Qixing Huang, and Karthik Ramani. Infogcn++: Learning representation by predicting the future for online skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[176] Christos Gkournelos, Christos Konstantinou, Panagiotis Angelakis, Eleni Tzavara, and Sotiris Makris. Praxis: A framework for ai-driven human action recognition in assembly. *Journal of Intelligent Manufacturing*, 35(8):3697–3711, 2024.

[177] Marco Carraro, Matteo Munaro, Jeff Burke, and Emanuele Menegatti. Real-time marker-less multi-person 3d pose estimation in rgb-depth camera networks. In *Intelligent Autonomous Systems 15: Proceedings of the 15th International Conference IAS-15*, pages 534–545. Springer, 2019.

[178] Kevin P Murphy. Machine learning: A probabilistic perspective (adaptive computation and machine learning series). *The MIT Press: London, UK*, 2018.

[179] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[180] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[181] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pages 894–906. PMLR, 2022.

[182] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[183] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

[184] Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. Mocoda: Model-based counterfactual data augmentation. *Advances in Neural Information Processing Systems*, 35:18143–18156, 2022.

[185] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[186] Christian Gumbsch, Noor Sajid, Georg Martius, and Martin V Butz. Learning hierarchical world models with adaptive temporal abstractions from discrete latent dynamics. In *The Twelfth International Conference on Learning Representations*, 2023.

[187] Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems*, 33:3976–3990, 2020.

[188] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023.

[189] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

[190] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[191] Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.

[192] Thomas Kipf, Elise Van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

[193] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

[194] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[195] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[196] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[197] Fan Jia, Weixin Mao, Yingfei Liu, Yucheng Zhao, Yuqing Wen, Chi Zhang, Xiangyu Zhang, and Tiancai Wang. Adriver-i: A general world model for autonomous driving. *arXiv preprint arXiv:2311.13549*, 2023.

[198] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[199] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

[200] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[201] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.

[202] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[203] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016.

[204] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations*, 2019.

[205] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024.

[206] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: a physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–1, 2024.

[207] Lukas Meyer, Floris Erich, Yusuke Yoshiyasu, Marc Stamminger, Noriaki Ando, and Yukiyasu Domae. Pegasus: Physically enhanced gaussian splatting simulation system for 6dof object pose dataset generation. *arXiv preprint arXiv:2401.02281*, 2024.

[208] Haozhe Lou, Yurong Liu, Yike Pan, Yiran Geng, Jianteng Chen, Wenlong Ma, Chenglong Li, Lin Wang, Hengzhen Feng, Lu Shi, et al. Robo-gs: A physics consistent spatial-temporal model for robotic arm with hybrid representation. *arXiv preprint arXiv:2408.14873*, 2024.

[209] Sipu Ruan, Weixiao Liu, Xiaoli Wang, Xin Meng, and Gregory S Chirikjian. Primp: Probabilistically-informed motion primitives for efficient affordance learning from demonstration. *IEEE Transactions on Robotics*, 2024.

[210] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

[211] Siwei Chen, Yiqing Xu, Cunjun Yu, Linfeng Li, and David Hsu. Differentiable particles for general-purpose deformable object manipulation. *arXiv preprint arXiv:2405.01044*, 2024.

[212] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. PAC-neRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *The Eleventh International Conference on Learning Representations*, 2023.

[213] William F Whitney, Tatiana Lopez-Guevara, Tobias Pfaff, Yulia Rubanova, Thomas Kipf, Kim Stachenfeld, and Kelsey R Allen. Learning 3d particle-based simulators from RGB-d videos. In *The Twelfth International Conference on Learning Representations*, 2024.

[214] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[215] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *ICCV*, 2023.

[216] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.

[217] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

[218] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, pages 284–301. PMLR, 2023.

[219] Yen-Chen Lin, Pete Florence, Andy Zeng, Jonathan T Barron, Yilun Du, Wei-Chiu Ma, Anthony Simeonov, Alberto Rodriguez Garcia, and Phillip Isola. Mira: Mental imagery for robotic affordances. In *Conference on Robot Learning*, pages 1916–1927. PMLR, 2023.

[220] Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *European Conference on Computer Vision*, pages 349–366. Springer, 2025.

[221] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[222] Nicholas Corrado and Josiah P. Hanna. Understanding when dynamics-invariant data augmentations benefit model-free reinforcement learning updates. In *The Twelfth International Conference on Learning Representations*, 2024.

[223] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

[224] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[225] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[226] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. In *NeurIPS*, 2020.

[227] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *Int. Conf. Learn. Represent.*, 2023.

[228] Ziyi Wu, Yulia Rubanova, Rishabh Kabra, Drew A Hudson, Igor Gilitschenski, Yusuf Aytar, Sjoerd van Steenkiste, Kelsey R Allen, and Thomas Kipf. Neural assets: 3d-aware multi-object scene synthesis with image diffusion models. *arXiv preprint arXiv:2406.09292*, 2024.

[229] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Self-supervised visual reinforcement learning with object-centric representations. In *International Conference on Learning Representations*, 2021.

[230] Andrii Zadaianchuk, Georg Martius, and Fanny Yang. Self-supervised reinforcement learning with independently controllable subgoals. In *Conference on Robot Learning*, pages 384–394. PMLR, 2022.

[231] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[232] Haochen Wang, Cilin Yan, Shuai Wang, Xiaolong Jiang, Xu Tang, Yao Hu, Weidi Xie, and Efstratios Gavves. Towards open-vocabulary video instance segmentation. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 4057–4066, 2023.

[233] Aniket Didolkar, Andrii Zadaianchuk, Anirudh Goyal, Mike Mozer, Yoshua Bengio, Georg Martius, and Maximilian Seitzer. Zero-shot object-centric representation learning. *arXiv preprint arXiv:2408.09162*, 2024.

[234] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in Neural Information Processing Systems*, 36, 2024.

[235] Lorenzo Bianchi, Fabio Carrara, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi. The devil is in the fine-grained details: Evaluating open-vocabulary object detectors for fine-grained understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22520–22529, 2024.

[236] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[237] Marcel Torne Villasevil, Arhan Jain, Vidyaaranya Macha, Jiayi Yuan, Lars Lien Ankile, Anthony Simeonov, Pulkit Agrawal, and Abhishek Gupta. Scaling robot-learning by crowdsourcing simulation environments. In *RSS 2024 Workshop: Data Generation for Robotics*, 2024.

[238] Marius Memmel, Andrew Wagenmaker, Chuning Zhu, Dieter Fox, and Abhishek Gupta. Asid: Active exploration for system identification in robotic manipulation. In *The Twelfth International Conference on Learning Representations*, 2024.

[239] Daniel M Wolpert, R.Chris Miall, and Mitsuo Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, 1998.

[240] Raman Mehra. Optimal input signals for parameter estimation in dynamic systems–survey and new results. *IEEE transactions on automatic control*, 19(6):753–768, 1974.

[241] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[242] Ziyan Gao, Armagan Elibol, and Nak Young Chong. Estimating the center of mass of an unknown object for nonprehensile manipulation. In *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1755–1760. IEEE, 2022.

[243] Peide Huang, Xilun Zhang, Ziang Cao, Shiqi Liu, Mengdi Xu, Wenhao Ding, Jonathan Francis, Bingqing Chen, and Ding Zhao. What went wrong? closing the sim-to-real gap via differentiable causal discovery. In *Conference on Robot Learning*, pages 734–760. PMLR, 2023.

[244] Allen Z Ren, Hongkai Dai, Benjamin Burchfiel, and Anirudha Majumdar. Adaptsim: Task-driven simulation adaptation for sim-to-real transfer. In *Conference on Robot Learning*, pages 3434–3452. PMLR, 2023.

[245] Núria Armengol Urpí, Marco Bagatella, Marin Vlastelica, and Georg Martius. Causal action influence aware counterfactual data augmentation. *arXiv preprint arXiv:2405.18917*, 2024.

[246] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[247] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

[248] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024.

[249] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[250] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.

[251] Joseph E. Banta, L. R. Wong, Christophe Dumont, and Mongi A. Abidi. A Next-Best-View System for Autonomous 3D Object Reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics*, 30(5):589–598, 2000.

[252] Frederic Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. Information Based Adaptive Robotic Exploration. In *International Conference on Intelligent Robots and Systems*, pages 540–545, 2002.

[253] Wen Jiang, Boshu Lei, and Kostas Daniilidis. FisherRF: Active View Selection and Mapping with Radiance Fields Using Fisher Information. In *European Conference on Computer Vision*, pages 422–440, 2024.

[254] Maria Isabel Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43(46):3736–3741, 2004.

[255] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. Neuraldiff: Segmenting 3d objects that move in egocentric videos. In *2021 International Conference on 3D Vision (3DV)*, pages 910–919. IEEE, 2021.

[256] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 9302–9311. IEEE, 2025.

[257] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, and Others. Habitat 3.0: A Co-Habitat for Humans, Avatars, and Robots. In *International Conference on Learning Representations*, 2023.

[258] Christian Dornhege and Alexander Kleiner. A frontier-void-based approach for autonomous exploration in 3d. *Advanced Robotics*, 27(6):459–468, 2013.

[259] Richard Pito. A Solution to the Next Best View Problem for Automated Surface Acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1016–1030, 2002.

[260] Richard Pito. A Sensor-Based Solution to the "Next Best View" Problem. In *International Conference on Pattern Recognition*, pages 941–945, 1996.

[261] Chao Cao, Hongbiao Zhu, Howie Choset, and Ji Zhang. TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments. *Robotics: Science and Systems*, 5:2, 2021.

[262] Howie Choset, Sean Walker, Kunnayut Eiamsa-Ard, and Joel Burdick. Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph. *International Journal of Robotics Research*, 19(2):126–148, 2000.

[263] Antoine Guédon, Tom Monnier, Pascal Monasse, and Vincent Lepetit. Macarons: Mapping and coverage anticipation with rgb online self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 940–951, 2023.

[264] Shiyao Li, Antoine Guédon, Clémentin Boittiaux, Shizhe Chen, and Vincent Lepetit. NextBestPath: Efficient 3D Mapping of Unseen Environments. In *International Conference on Learning Representations*, 2025.

[265] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[266] Ziyue Feng, Huangying Zhan, Zheng Chen, Qingan Yan, Xiangyu Xu, Changjiang Cai, Bing Li, Qilun Zhu, and Yi Xu. Naruto: Neural Active Reconstruction from Uncertain Target Observations. In *Conference on Computer Vision and Pattern Recognition*, pages 21572–21583, 2024.

[267] Liren Jin, Xingguang Zhong, Yue Pan, Jens Behley, Cyrill Stachniss, and Marija Popović. Activegs: Active Scene Reconstruction Using Gaussian Splatting. *IEEE Robotics and Automation Letters*, 2025.

[268] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-Time 3D Model Acquisition. In *ACM SIGGRAPH*, May 2002.

[269] Thibaut Weise, Bastian Leibe, and Luc Van Gool. Accurate and Robust Registration for In-Hand Modeling. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[270] Pengyuan Wang, Fabian Manhardt, Luca Minciullo, Lorenzo Garattoni, Sven Meie, Nassir Navab, and Benjamin Busam. DemoGrasp: Few-Shot Learning for Robotic Grasping with Human Demonstration. In *International Conference on Intelligent Robots and Systems*, pages 5733–5740, 2021.

[271] Dimitrios Tzionas and Juergen Gall. 3D Object Reconstruction from Hand-Object Interactions. In *International Conference on Computer Vision*, 2015.

[272] Shreyas Hampali, Tomas Hodan, Luan Tran, Lingni Ma, Cem Keskin, and Vincent Lepetit. In-Hand 3D Object Scanning from an RGB Sequence. In *Conference on Computer Vision and Pattern Recognition*, 2023.

[273] Yufeng Jin, Vignesh Prasad, Snehal Jauhri, Mathias Franzius, and Georgia Chalvatzaki. 6DOPE-GS: Online 6D Object Pose Estimation Using Gaussian Splatting. In *International Conference on Computer Vision*, 2025.

[274] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024.

[275] Hyungtae Lim, Daebeom Kim, Gunhee Shin, Jingnan Shi, Ignacio Vizzo, Hyun Myung, Jaesik Park, and Luca Carlone. Kiss-Matcher: Fast and Robust Point Cloud Registration Revisited. In *International Conference on Robotics and Automation*, pages 11104–11111, 2025.

[276] Michael Korn, Martin Holzkothen, and Josef Pauli. Color Supported generalized-ICP. In *International Conference on Computer Vision*, pages 592–599, 2014.

[277] Chunyu Ju, Qinghua Luo, and Xiaozhen Yan. Path planning using an improved a-star algorithm. In *2020 11th international conference on prognostics and system health management (PHM-2020 Jinan)*, pages 23–26. IEEE, 2020.

[278] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[279] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018.