Ref. Ares(2026)5995764 - 12/06/2026





SOPRANO

Socially-Acceptable and Trustworthy Human-Robot Teaming for Agile Industries

Project Number 101120990

D2.3 Industrial Context Sensing II

Version 1.0
27 May 2026
Final

Public Distribution

FORTH and NTUA

Project Partners: **AEGIS, ATB, CEA, Centro Ricerche Fiat, Circular Economy Foundation, E-TERRY, F6S, FORTH, Harokopio University of Athens, IFADO, Jade University, KUKA Assembly & Test, Netcompany-Intrasoft, NTUA, PROFACTOR, Technology Transfer Systems, The Open Group, University of Bremen, University of York**

Every effort has been made to ensure that all statements and information contained herein are accurate, however the SOPRANO Project Partners accept no liability for any error or omission in the same.

© 2026 Copyright in this document remains vested in the SOPRANO Project Partners.

PROJECT PARTNER CONTACT INFORMATION

AEGIS Nikos Kostalas Humboldtstrasse 25 38106 Braunschweig, Germany E-mail: n.kostalas@aegisresearch.eu	**ATB** Sebastian Scholze Wiener Strasse 1 28359 Bremen, Germany E-mail: scholze@atb-bremen.de
CEA Nikolaos Matragkas NanoInnov Bat. 862 PC 174 91191 Gif-sur-Yvette Cedex, France E-mail: nikolaos.matragkas@cea.fr	**Centro Ricerche Fiat** Felice Tauro Strada Torino 50 10043 ORBASSANO, Italy E-mail: felice.tauro@crf.it
Circular Economy Foundation Ioannis Basdekis Rue Breydel 34-36-40 1040 Brussels, Belgium E-mail: ioannis.basdekis@circular.foundation	**E-TERRY** Fabian Rösler Neuwerkstrasse 50 99084 Erfurt, Germany E-mail: fabian.roesler@e-terry.de
F6S Gergana Dimitrova 77 Lower Camden Street Dublin D02 XE80, Ireland E-mail: gergana@f6s.com	**FORTH** Manolis Lourakis N Plastira Str 100 70013 Heraklion, Greece E-mail: lourakis@ics.forth.gr
Harokopio University of Athens Iraklis Varlamis Eleftheriou Venizelou 70 176 71 Athens, Greece E-mail: varlamis@hua.gr	**IFADO** Felix Dreger Ardeystrasse 67 44139 Dortmund, Germany E-mail: dreger@ifado.de
Jade University George Athanassiou Friedrich Paffrath Strasse 101 26389 Wilhelmshaven, Germany E-mail: georgios.athanassiou@jade-hs.de	**KUKA Assembly & Test** Georgios Karantinakis Uhthoffstrasse 1 28757 Bremen, Germany E-mail: georgios.karantinakis@kuka.com
Netcompany-Intrasoft John Soldatos 2B Rue Nicolas Bové 1253 Luxembourg, Luxembourg E-mail: john.soldatos@netcompany-intrasoft.com	**NTUA** Maria Pateraki Heroon Polytechniou 9, Zographou Campus 157 80 Athens, Greece E-mail: mpateraki@mail.ntua.gr
PROFACTOR Sharath Akkaladevi Im Stadtgut D1 4407 Steyr-Gleink, Austria E-mail: sharath.akkaladevi@profactor.at	**Technology Transfer Systems** Antonio Avai Via Francesco d'Ovidio, 3 20131 Milan, Italy E-mail: avai@ttsnetwork.com
The Open Group Scott Hansen Rond Point Schuman 6, 7th Floor 1040 Brussels, Belgium E-mail: s.hansen@opengroup.org	**University of Bremen** Nico Hochgeschwender Bibliothekstrasse 1 28359 Bremen, Germany E-mail: ni_ho@uni-bremen.de
University of York Simos Gerasimou Deramore Lane York YO10 5GH, United Kingdom E-mail: simos.gerasimou@york.ac.uk	

Document Control

Version	Status	Date
0.1	Initial draft with outline	10 February 2026
0.2	Contributions by partners	9 April 2026
0.3	Review comments by editor	20 April 2026
0.4	Draft version for QA review	24 April 2026
0.5	QA review comments	8 May 2026
0.6	QA review comments addressed	14 May 2026
1.0	Harmonization, final version for EC submission	27 May 2026

Table of Contents

1 Introduction **1**

1.1 Purpose of the deliverable . . . 1

1.2 Relation to other deliverables . . . 1

1.3 Structure of the deliverable . . . 2

2 Visual Spatial Localization and Mapping [VLM] **3**

2.1 Technology overview and module specifications . . . 3

2.2 Methodology and implementation . . . 4

2.2.1 Existing approaches (SOTA) . . . 4

2.2.2 The approach in SOPRANO . . . 6

2.3 Results . . . 14

2.4 Discussion . . . 18

3 Object Perception [OBP] **19**

3.1 Technology overview and module specifications . . . 19

3.2 Methodology and implementation . . . 20

3.2.1 Existing approaches (SOTA) . . . 20

3.2.2 The approach in SOPRANO . . . 23

3.3 Results . . . 29

3.3.1 CRF use case . . . 29

3.3.2 E-TERRY use case . . . 33

3.4 Discussion and future work . . . 36

4 Non-Visual Localization [NVL] **37**

4.1 Technology overview and module specifications . . . 37

4.2 Methodology and implementation . . . 37

4.2.1 Existing approaches (SOTA) . . . 37

4.2.2 The approach in SOPRANO . . . 40

4.3 NTUA's robotic platform for NVL . . . 48

4.4 Results . . . 49

4.4.1 Testing using the simulator . . . 49

4.4.2 Testing of the LiDAR-based SLAM . . . 50

4.4.3 Testing BLE-based localization . . . 51

4.5 Discussion . . . 53

5 Human Monitoring [HMO] 55
5.1 Technology overview and module specifications . . . 55
5.2 Methodology and implementation . . . 55
5.2.1 Existing approaches (SOTA) . . . 55
5.2.2 The approach in SOPRANO . . . 59
5.3 Experimental results . . . 66
5.3.1 Human Detection . . . 67
5.3.2 Human body pose estimation . . . 67
5.3.3 Ergonomic posture evaluation . . . 69
5.3.4 Gesture recognition . . . 70
5.3.5 Monitoring of action / task progress . . . 71
5.4 HMO runtime performance assessment . . . 73
5.4.1 High-end workstation runtime performance (CRF) . . . 73
5.4.2 Embedded platform runtime performance (E-TERRY and KUKA) . . . 73
5.5 Discussion and upcoming work . . . 73

6 Conclusions 75

A Updates to Components Since D2.1 76
A.1 Visual Spatial Localization and Mapping (T2.1) . . . 76
A.2 Object Perception (T2.2) . . . 76
A.3 Non-Visual Localization (T2.3) . . . 77
A.4 Human Monitoring (T2.4) . . . 77

List of Figures

1 KUKA's operating environment . . . 10
2 Plane detection and annotation GUI . . . 11
3 Pose visualization in KUKA's environment . . . 12
4 Perspective-n-Line (PnL) example . . . 13
5 CRF's operating environment . . . 15
6 Feature tracking and sparse reconstruction in CRF's environment . . . 15
7 Robot trajectory on CRF's floor map . . . 16
8 VLM evaluation for the CRF use case . . . 16
9 VLM evaluation for the KUKA use case . . . 17
10 Feature-based vSLAM in KUKA's environment . . . 17
11 VLM in E-TERRY's environment . . . 18
12 Use case object detection examples . . . 20
13 Overview of the IndustryShapes dataset . . . 25
14 Annotated CRF screw detection dataset . . . 26
15 Sample images from the E-TERRY environment . . . 26
16 Qualitative results from theIndustryShapes dataset . . . 28
17 Overview of the multi-stage screw detection process . . . 30
18 Overview of the screw detection pipeline . . . 31
19 Visualization of screw geometry and CNOS-based segmentation . . . 31
20 Methods comparison across scenes of the IndustryShapes test set . . . 34
21 Confidence of pose estimates with the COBRA method . . . 35
22 Example results from the E-TERRY use case . . . 36
23 LiDAR mounting points on KUKA's robot . . . 40
24 Example of raw and filtered LiDAR scans . . . 42
25 The 3D robotic simulator for LiDAR-based SLAM . . . 43
26 LiDAR-based SLAM results with and without extrinsic calibration . . . 44
27 The ultrasonic radar sensor array . . . 45
28 Tool presence detector prototype . . . 47
29 The schematic diagram of the propulsion system electronics . . . 49
30 The laboratory testbed of the propulsion system . . . 49
31 RViz2 output running the simulator . . . 50
32 Comparison of VLM and NVL trajectories . . . 51
33 LiDAR-based SLAM in the KUKA environment . . . 52
34 BLE experimental setup . . . 53
35 The HMO component workflow . . . 60
36 Layout of CRF's manufacturing environment . . . 61

37 Ergonomic risk-inducing dynamic postures selected from EAWS 64
38 Pose estimation (2D) for multiple E-TERRY workers 65
39 Human detection for multiple E-TERRY workers 68
40 3D pose estimation for a CRF worker at close range 69
41 3D pose estimation for a distant CRF worker 69
42 Body postures classification and ergonomic evaluation 69
43 Gesture recognition evaluation 71
44 Confusion matrix for gesture recognition evaluation 71
45 Confusion matrices for assembly action classification in CRF 72

List of Tables

1 Camera types, features and applications in different use cases 24
2 OBP variations for different hardware and software setups 29
3 Screw detection performance on CRF dataset 32
4 Comparison of the EPOS and FoundationPose methods 32
5 Average recall per object for EPOS and FoundationPose 33
6 Localization success rates for EPOS and FoundationPose 33
7 Pretrained YOLOv5 human detection performance for E-TERRY 35
8 Fine-tuned YOLOv5 human detection performance for E-TERRY 35
10 Video datasets for human activity monitoring 59
11 HMO component variants 60
12 Features of HMO component across use cases 61
13 Classification results for EAWS-based ergonomic work postures 70
14 MocapNet2 3D pose estimation results 70
15 Performance of models for action/sub-goal classification 72
16 Execution times of the main pipeline components for CRF 73
17 Execution times of the main pipeline components for KUKA and E-TERRY 74

Executive Summary

This deliverable outlines the development and results of perception mechanisms that equip robots with advanced sensing capabilities, enhancing their operation in industrial environments targeted by SOPRANO. It summarizes the work performed in Tasks 2.1–2.4 of SOPRANO up to the submission date (M30).

The deliverable is structured around four sections, each focusing on a key perception technology employed within SOPRANO. While certain interdependencies exist across the document, each section is presented as largely self-contained, enabling independent review without loss of context.

Section 2 focuses on the visual spatial localization and mapping (VLM) component which is tasked with determining where a robot is in its operating environment. VLM uses structure-from-motion techniques for accurate camera pose estimation in a dual-system RGB-D framework. In environments with moderate-to-rich texture, localization is supported by a feature-based visual SLAM (simultaneous localization and mapping) pipeline. In texture-poor environments, a geometry-based approach based solely on depth measurements has been developed. Together, they ensure reliable operation across diverse conditions. Experimental results using data from the use cases are included, demonstrating the capabilities of the developed systems.

Section 3 focuses on the object perception (OBP) component, designed to enable real-time detection and 6D pose estimation of objects in industrial settings, also supporting robotic manipulation tasks. By employing deep learning and machine vision, OBP overcomes challenges posed by weak textures, occlusions, and reflective surfaces. The approach includes training on synthetic and annotated data to ensure robustness across diverse scenarios. Experimental results highlight its effectiveness in key use cases like vehicle assembly, demonstrating reliable object detection and pose estimation capabilities.

Section 4 explores non-visual localization (NVL) methods of the respective component, such as those utilizing LiDAR (scans), Bluetooth tags, or mm-wave radar signals. These techniques enhance robustness by compensating for visual system limitations through sensor fusion strategies. NVL methods can function as standalone systems, providing efficient and reliable localization, or tailored to complement visual sensing in challenging environments. The section presents the main methodologies, implementation choices, and testing activities carried out for the relevant SOPRANO use cases.

Section 5 focuses on the development and implementation of the human monitoring (HMO) component that facilitates safe human-robot collaboration. Toward this end, it estimates 2D/3D human location, fine-grained 2D/3D human body poses, monitors human activity, analyzes gestures, and assesses ergonomic postures using visual information as input. This component plays a crucial role in improving safety and efficiency in manufacturing and disassembly tasks, with applications that involve monitoring the workers' physical interactions with robotic systems across the SOPRANO use cases.

Reported developments in spatial localization, object detection, and human monitoring lay the groundwork for future research and implementation, with an emphasis on achieving higher Technology Readiness Levels (TRLs) through rigorous field testing and validation. The deliverable serves as a framework for subsequent advancements in robotic performance and adaptability within complex industrial environments.

1 Introduction

1.1 Purpose of the deliverable

The purpose of this deliverable is to present the advancements in perception and localization technologies within the SOPRANO project, emphasizing their integration and applicability in industrial environments. It aims to provide a comprehensive overview of the methodologies, technological components, and experimental results achieved in enhancing robotic perception and localization capabilities.

The deliverable outlines the progress made in four core perception components, namely Visual Spatial Localization and Mapping (VLM), Object Perception (OBP), Non-Visual Localization (NVL), and Human Monitoring (HMO). The VLM component focuses on improving visual-based localization and mapping in dynamic environments, while the OBP component addresses challenges in object and 2D human detection as well as 6D pose estimation. The NVL component complements these efforts by leveraging non-visual sensing to enhance localization in conditions where vision-based approaches are limited. The HMO component ensures safe and efficient collaboration by enabling robust 3D human detection and pose estimation, ergonomic risk assessment, as well as activity and gesture recognition.

The SOPRANO framework is demonstrated through three use cases which encompass building renovation (KUKA), vehicle door disassembly (CRF) and symbiotic farming (E-TERRY). All four SOPRANO components presented in this deliverable were designed and developed in accordance with the functional requirements defined by SOPRANO's specific use cases. To ensure operational relevance, the components were rigorously validated using real field data captured directly from the use case environments, establishing their reliability under real-world conditions. Representative results for each component are provided in this document, offering empirical evidence of their stability and cross-domain adaptability. Additional validation is reflected in the successful adoption of the components described herein by eight out of the ten Open Call projects funded through SOPRANO via the FSTP (Financial Support to Third Parties) mechanism.

1.2 Relation to other deliverables

Deliverable D2.3 is closely linked to the following SOPRANO deliverables:

- D1.1 - Project requirements, which describes the detailed requirements for the R&D activities to be carried out within the project.
- D2.1 - Industrial Context Sensing I, which is a preliminary, earlier version of the present deliverable.
- D2.2 - Customizable Task Execution I, which is related to the mechanisms to enhance task execution flexibility considering the outputs from the modules of this deliverable.
- D3.1 - Technologies for Safety Analysis and Testing of MH-MR Systems, which describes the various components that are contributing to simulate safety scenarios at design time and provide runtime uncertainty assessment.
- D4.1 - Modular, Reconfigurable and Flexible Tools I provides the tools and services for developing applications for MH-MR collaboration in dynamic contexts.
- D5.1 - Architecture Definition outlines the multiple views of the system architecture, the technical specifications and serves as the key lever for the research & development initiatives within the project.
- D5.2 - Integrated Technology I that reports on the first release of the Integrated Technologies.
- D5.3 - Use Cases Evaluation I, reporting on the first evaluation of SOPRANO's components by use case partners.
- D2.4 - Customizable Task Execution II, which will be an updated version of D2.2.
- D5.5 - Integrated Technology II, which will be an updated version of D5.2.
- D5.6 - Use Cases Evaluation II, which will be an updated version of D5.3.

1.3 Structure of the deliverable

The deliverable is structured around four main sections, each providing a detailed overview of the methodologies, technologies, and outcomes associated with corresponding components of the SOPRANO project, excluding the introduction and conclusion sections.

Although certain interdependencies do exist throughout the text, each section has been structured to remain largely self-contained, allowing for independent reading without loss of context.

Section 2 focuses on the Visual Spatial Localization and Mapping (VLM) component, which identifies the exact location of a robot in its operating environment. By reliably determining its own position, the robot gains the situational awareness necessary to autonomously navigate complex paths and execute tasks.

Section 3 concerns the Object Perception (OBP) component which aims to provide real-time detection and 6D pose estimation of objects in a robots' operating environment exploiting visual information from cameras installed on the robot as well in stationary locations in the environment.

Section 4 details the Non-Visual Localization (NVL) component that offers an alternative to visual localization in environments where visual texture is unreliable or unavailable, ensuring robust and accurate localization through sensor-based techniques.

Section 5 addresses the Human Monitoring (HMO) component, which observes workers with the aim of enabling efficient and smooth Human-Robot Interaction (HRI) and collaboration (HRC) across all use case scenarios. Thus, it is concerned with real-time monitoring of 2D/3D human poses, identification of physical ergonomic risks, recognition of worker activities and task progress during task cycles as well as gesture recognition.

Section 6 concludes by summarizing key achievements across these components, reflecting on actual results and integration within the SOPRANO project.

At the end of the document, appendix A summarizes the main advances for each component with respect to its early prototype.

2 Visual Spatial Localization and Mapping [VLM]

2.1 Technology overview and module specifications

An autonomous robot that navigates a partially known dynamic environment must be aware of its position with respect to its surroundings [239]. This awareness is facilitated by on-board sensors that are used to construct and maintain a representation of the environment (i.e., a map), while simultaneously keeping track of the robot's position and orientation within that representation. In robotics parlance, this is the computational problem of simultaneous localization and mapping, abbreviated as SLAM. SLAM is a challenging task whose difficulty stems from the fact that it represents a cause and effect dilemma: In order for the robot to localize itself, a map is needed and for constructing a map, a localization estimate is necessary. Of particular interest are visual SLAM (vSLAM) algorithms, which rely on camera-based environment observations. Such algorithms have gained significant traction in recent years, driven by the practical advantages of camera sensors, including their affordability, low power requirements, and compact physical footprint.

In SOPRANO, the visual spatial localization and mapping (VLM) module is assigned the task of determining the location of a robot in its operating environment. This is achieved by continuously estimating the robot's pose (i.e., its position and orientation) in the environment along with a sparse map pertaining to the environment's visible geometry. This map is primarily intended to support localization and is characterized by being sparse, of moderate accuracy, and lacking any semantic information (cf. Fig. 6 right). Therefore, despite being sufficient for path planning and obstacle avoidance, such a map is not suitable for tasks such as precise robotic manipulation, 3D modeling, or surveying. The VLM module estimates a general six-degree-of-freedom (6DoF) pose, which defines the spatial configuration of a robot in 3D space. However, in cases where the robot operates on a planar ground, this general pose is for simplicity converted to a 2D spatial location (x, y) plus a yaw rotation θ around the robot's vertical axis. The VLM outputs are consumed by other SOPRANO components such as Human Monitoring (HMO), MH-MR Task Allocation (CTA), Context Extraction Module (CEM), Robotic Capabilities Implementation (RCI) and Runtime Safety Assessment (RSA). In addition to these components, the VLM has also been employed by some of the SOPRANO Open Call beneficiaries to support their own demonstrators [50].

Early visual SLAM approaches relied on a single (i.e., monocular) intensity camera and were based on low-level features. However, to extract accurate, up to scale depth information, they usually required multiple images with large parallax, increasing complexity and computational cost. To overcome these issues, we decided to base VLM on 3D (depth) cameras [5]. More specifically, indoor localization employs the Intel RealSense D455 RGB-D camera [75], whereas outdoor localization relies on the Stereolabs ZED X camera [201]. Both the RealSense D455 and the Stereolabs ZED X are stereo vision cameras that generate RGB images and depth maps using paired sensors and onboard IMUs, supporting applications such as 3D perception and motion tracking. They also have global shutters that avoid motion distortions for fast camera movements. However, their operating principles differ in a way that influences their typical deployment environments. Specifically, the D45x series cameras use an active stereo method with a built-in infrared projector, which allows reliable depth estimation in low-texture and controlled lighting conditions, making them well suited to indoor use. However, their performance degrades in strong sunlight, as ambient infrared radiation interferes with the projected pattern, reducing depth accuracy and effective range. In contrast, the ZED family of cameras rely on passive stereo vision and GPU-based processing without artificial illumination, making them more effective in well-lit, large-scale outdoor environments with sufficient texture. Additionally, the relatively large baseline of ZED stereo cameras benefits their depth estimation performance at extended ranges.

Before describing the specific localization methods developed in SOPRANO, we briefly examine existing pertinent research in order to establish the necessary background.

2.2 Methodology and implementation

2.2.1 Existing approaches (SOTA)

Spatial mapping and localization are essential to support robot navigation in the three use case environments. Addressing this challenge requires SLAM, one of the most extensively studied topics in robotics research for more than three decades. A broad range of relevant algorithms have been developed, some of which have enabled localization systems to perform well in certain domains. Relevant solutions come in different variants, depending on the algorithmic approach, the type of sensors (e.g., sonar, laser, time-of-flight, monocular or stereo cameras and RGB-D) and the data representation used, as well as the particularities of an application. In the following, we provide a brief general overview with more emphasis on visual SLAM. More details can be found in the extended surveys in [21, 73, 17, 192, 8].

In the earliest approaches, SLAM was formalized in a probabilistic fashion that aimed to estimate the model parameters (map and robot state) that maximize the probability of obtaining the actual measurements. A SLAM system consists of two main components, namely the front and back ends. The front-end transforms sensor data so that they can be more suitable for estimation, while the back-end performs inference on the data provided by the front-end [21]. In the case of visual SLAM, the front-end extracts the pixel locations of discriminant points in the environment and associates them with specific landmarks, e.g. 3D points; this task is known as data association. Estimating the pose of a moving camera and the 3D geometry of a scene has been studied by the computer vision community under the name of *structure from motion* (SfM) [258]. Despite their similarities, SfM and vSLAM approached the problem from different perspectives. SfM emphasized accuracy of reconstruction at the cost of batch operation whereas vSLAM sought incremental camera motion estimation with online performance. From the early 2000s onward, the research on SfM and vSLAM has converged and the generic character of the localisation and reconstruction problems has become more apparent [36].

A possible formulation of SLAM is the full one, where the entire robot trajectory and the map are estimated given all the control inputs and all sensor measurements. Although this approach can lead to highly accurate results, it suffers from the drawback that the problem grows unboundedly with the number of considered variables. Thus, it cannot be solved in real time, which in turn limits its practical applicability. Another approach is that of online SLAM, where the problem is solved incrementally by updating the estimated model using only recently acquired sensor information. Estimation techniques can be classified as either filter or optimization based. Filter-based methods originate from Bayesian filtering and sequentially fuse image measurements by updating probability distributions over the map and pose parameters. They operate in two steps: First, a prediction of the model is made, and then the current sensor measurements are used to correct the previously predicted state. A Kalman Filter (KF) relies on the premise that the pose model is linear. The KF couples a prediction stage that uses the previous values to predict the current state, followed by an update phase that combines the predicted state with current sensor data. As the linearity assumption does not hold in practice, derivatives of the KF are used to handle nonlinear systems. The extended KF (EKF) performs linearization around the current estimate via first-order Taylor expansion. The EKF update time depends quadratically on the size of the state vector, hence EKF cannot support the growing map size of large-scale SLAM. To deal with this, new empty sub-maps (linked via a higher-level map) are introduced when the map size becomes large [140]. Still, the EKF approach has high computational demands. The unscented KF (UKF) performs better than the EKF for highly non-linear systems and does not require the calculation of the Jacobians. However, it is also computationally expensive. An alternative approach to KF filtering is provided by particle filters (PF), which use particles (i.e., samples) to represent the posterior distribution of a stochastic process given noisy or partial observations. PFs sample the state with a set of particles and perform displacement predictions and updates for each particle. During the update, particles are weighted according to their likelihood and then the most likely ones are retained while the rest are eliminated and replaced by new particles, which are generated near the high-weight ones. PFs avoid making any linearity or Gaussianity assumptions and can generate samples from a required distribution, without any presumptions about the state-space model or the state distributions. FastSLAM is the best known algorithm based on particle filtering [136].

Optimization-based methods also consist of two parts. The first finds a correspondence between new observations and the map, thus providing constraints. The model is then updated to accommodate the constraints provided by the new observations. Optimization can be carried out with Bundle Adjustment (BA) [115] and its variants or graph SLAM [93, 42]. Bundle adjustment simultaneously refines the sensor pose and the geometry of the scene to optimize a criterion involving the reprojection error of all points in all images. BA performs batch optimization on a large number of variables and yields accurate results at the expense of high computational cost. To limit this cost, BA is performed on selected images such as those in a sliding window or on spatially distributed key-frames [92]. A comparison of filter-based and optimization techniques for monocular SLAM concluded that having a large number of feature keypoints is more beneficial to the accuracy of SLAM compared to having a large number of frames [202]. Furthermore, since the computational cost of BA grows less with the number of keypoints compared to EKF approaches, the former outperforms the latter. Graph SLAM employs a graphical representation in which nodes represent poses and connecting edges correspond to spatial constraints between them. Constraints are linearized to obtain a sparse matrix corresponding to a sparse graph map. A reduction process removes the redundant map variables and optimization strives to find a configuration of nodes that minimize the error induced by the constraints [96].

Another taxonomy divides SLAM algorithms into direct and indirect [43]. Indirect methods preprocess raw sensor measurements and generate an intermediate representation (e.g. feature correspondences [220]) that is next interpreted in a model to estimate geometry and camera motion [36, 92, 142]. They minimize the reprojection error over batches of features and can thus yield accurate results, at relatively high computational cost. Direct methods bypass the pre-processing step and use the sensor values directly in a probabilistic model [44, 43]. This makes them highly resilient in indoor environments, where the surfaces may be smooth and have weaker texture. Still, they are susceptible to changes in lighting and induce higher computational costs. Yet another classification discriminates between sparse and dense methods. The former employ and reconstruct a selected set of independent image points (e.g. corner keypoints) [92, 142], whereas dense methods attempt to use and reconstruct all pixels in a 2D image [44]. Intermediate, semi-dense approaches refrain from reconstructing the complete surface but still aim at using and reconstructing a largely connected subset.

In recent years, inertial measurement unit (IMU) devices have become widely available. These transmit rotational speed and acceleration measurements at high rates which can enhance the performance and robustness of SLAM algorithms. Apart from requiring calibration, one of the main issues with IMUs is that they suffer from noise and bias, which accumulate over time and corrupt the measurement quality. To limit these errors, the common approach is to use information from a vision sensor to initialize and track the IMU parameters [22, 166, 173]. These visual-inertial methods are categorized according to the type of fusion they pursue as (i) loosely or (ii) tightly coupled. The first category builds on the assumption that an (up to scale) camera trajectory can be estimated accurately from the vision sensor, which is then used to initialize the IMU parameters. Tightly coupled methods derive closed-form solutions for equations that jointly optimize common IMU and vision factors (e.g., [125, 81]).

The performance of SLAM systems is affected by different factors, including sensor noise, rapid sensor movement, unfavorable lighting, low-textured environments, reflective materials, environment changes, and computational limitations. These manifest themselves as small errors in the pose estimate, known as drift, which accumulate over time and adversely affect localization. These factors are most prominent in outdoor environments, and can significantly affect the robustness of localization algorithms [145]. To mitigate their effect, methods such as geometric features from range-based sensors [48], visual-inertial fusion [192] or appearance-based loop closure [145] have been proposed.

Loop closure is a particularly effective approach that aims to detect whether an observed scene has been visited earlier [219]. Doing so allows previously unconnected parts of a map to be connected into a loop and the pose estimates and map to be updated accordingly in order to eliminate drift. Thus, incorporating informative scene representations into a SLAM system, is a crucial prerequisite for closing loops. Over the years, various representations have been proposed, including meshes, voxels, and truncated signed distance functions. More recently, SfM methods have demonstrated improved accuracy by recovering 3D structure and camera motion offline using incremental [183] or global [154] optimization techniques. SLAM systems have uti-

lized these representations by leveraging SfM data to design more effective loop closure mechanisms. These approaches often involve converting SfM data into volumetric or photometric representations using methods such as NeRF [133, 12] or 3D Gaussian splatting [85, 227]. For example, RD-SLAM leverages 3D Gaussian splatting for dense RGB-D SLAM, combined with Generalized Iterative Closest Point (G-ICP) for efficient pose estimation [85]. LoopSplat addresses global consistency challenges by registering 3D Gaussian splats using pose graph optimization, enhancing loop closure accuracy [84]. Loopy-SLAM focuses on dynamic submap generation and uses neural point clouds for local scene representation, employing pose graph optimization to ensure global consistency during loop closure [176].

On the deep learning front, DeepFactors [35] encodes scene geometry using a learned latent representation of depth, which is optimized during frame alignment. DROID-SLAM [209] extends this idea with a recurrent architecture that iteratively refines camera pose and dense depth, enabling robust performance even with noisy measurements. ManhattanSLAM [245] exploits orthogonal planar constraints to achieve reliable tracking and mapping in structured, man-made environments. GO-SLAM integrates full bundle adjustment (BA) and loop closure, addressing issues such as drift and map distortion in large-scale environments [7]. GS-SLAM improves the efficiency of dense SLAM by dynamically adjusting the number of Gaussian splats based on the complexity of a scene, achieving high localization accuracy and dense reconstruction [85]. SplaTAM extends these ideas by utilizing Gaussian splatting to optimize volumetric scene representations, allowing real-time tracking and rendering at high frame rates [84]. Nevertheless, a shortcoming of these methods is that they often generate redundant information to facilitate loop closure. Additionally, they rely on realistic local scene rendering, which presents significant challenges in real-time SLAM operations and struggles with scalability due to inherent issues with map representations.

Complementing purely photometric methods, another line of work leverages RGB-D sensors, which employ active sensing technologies such as structured light or time-of-flight to augment conventional intensity data with dense, registered depth measurements. By directly capturing geometry, these sensors provide an explicit representation of the scene that is largely independent of ambient illumination and surface appearance, partially mitigating the limitations caused by low visual texture. As a result, depth-based approaches have become widely adopted in dense 3D reconstruction and visual navigation. Building on this paradigm, Iterative Closest Point (ICP) [29] enables direct alignment of geometric structures in measurement space. KinectFusion [144] was the first system to achieve robust real-time tracking using only depth data. Subsequently, Dense Visual Odometry (DVO) [86] jointly minimized photometric and geometric errors across all pixels, using depth cues to constrain motion even in regions lacking distinctive visual features. BAD SLAM [186] introduced an efficient direct bundle adjustment formulation capable of execution in real-time on GPUs.

2.2.2 The approach in SOPRANO

The use cases targeted by SOPRANO involve environments with dissimilar characteristics that present considerable challenges for visual localization such as paucity of features and less than optimal imaging conditions. Outdoor operation has to cope with variable lighting and adverse weather conditions, including low-light, shadows, overexposure, and sun reflections or glare. On the other hand, some of the indoor spaces are characterized by feature-poor, planar surfaces with non-distinct visual features and scarce texture, which offer limited information for algorithms to extract. Furthermore, the presence of humans in close proximity to operating robotic systems imposes stringent performance constraints. These requirements include minimal delay and reduced error margins to facilitate effective and safe human-robot collaboration.

The localization component developed in SOPRANO leverages state-of-the-art structure-from-motion algorithms to achieve accurate and robust camera pose estimation. It is designed as two independent systems, each following an approach tailored to the unique environmental conditions of the use case environments and leveraging different data modalities from RGB-D sensors. This dual-system design ensures reliable localization operation across diverse real-world environments, ranging from feature-rich spaces to texture-deficient scenes.

The first approach, presented in Sec. 2.2.2.1, targets environments with moderate to rich texture. In such settings, sufficient natural features can be reliably detected and matched across photometric images, enabling precise pose estimation through a mature feature-based pipeline. This direction builds upon well-established principles in visual localization, taking advantage of the discriminative power of image features to achieve high accuracy and robustness. The second approach, discussed in Sec. 2.2.2.2, addresses feature-poor environments with limited texture, where traditional feature-based methods tend to degrade due to the lack of sufficient distinctive visual cues. In this case, geometric cues derived from the depth component of the RGB-D data, enable stable pose estimation even in challenging conditions where appearance-based matching alone is inadequate. Together, these two development strands provide a comprehensive localization solution capable of adapting to diverse environmental characteristics.

Regardless of the particular approach pursued for pose estimation, the VLM component executes on-board the use case robots in order to reduce latency and communication overhead. VLM is implemented in C++ to ensure high performance and efficient use of resources. To ensure seamless integration and scalability, the component leverages ROS 2 as its core communication framework. Regarding supported hardware architectures, VLM supports both x86-64 workstations and ARM64 Nvidia Jetsons, to accommodate real-time desktop processing and edge computing. To maintain a deterministic control loop for the autonomous robotic platform, it was decided to retain the VLM component on the CPU. This ensures a more consistent execution frequency and avoids the non-deterministic scheduling latencies often associated with shared GPU resources in multi-process systems. In addition, executing on the CPU preserves GPU capacity for essential parallel tasks, such as deep-learning-based perception (for example, the HMO; see Sec. 5), which are fundamentally dependent on GPU hardware acceleration. However, despite the absence of GPU acceleration, VLM's implementation achieves high performance by leveraging multi-threading and SIMD vectorization. To enforce strict dependency isolation and ensure consistent version alignment of third-party libraries, the VLM is containerized using Docker images. This approach guarantees seamless transition and environment parity between development and deployment, while decoupling localization logic from host-specific dependency conflicts and enabling reproducible execution across hardware platforms and architectures. For streamlined distribution and installation, the container images are stored on the GitHub Container Registry, a managed service for hosting and versioning containers.

The VLM component expects RGB-D image streams as input, using standard `sensor_msgs/Image` ROS messages for image data and `sensor_msgs/CameraInfo` for camera metadata. To conserve memory/network bandwidth, VLM supports RGB images that have been losslessly compressed as PNGs. The images can either originate from a live depth camera such as the RealSense D45x (through the RealSense ROS wrapper) or prerecorded datasets stored in image files and later republished using FORTH's open source player[1]. VLM also offers the option of directly accessing a RealSense depth camera using the RealSense SDK[2]. This direct approach eliminates the computational overhead of message serialization and enables zero-copy memory access and lower latency. In addition, it provides lower-level hardware control. However, it has the disadvantage that the image streams cannot be easily shared among multiple applications. Multi-camera configurations are supported by discriminating between physical devices via their unique serial numbers. The VLM component publishes the pose it estimates for each RGB-D image frame on user-defined topic using standard `PoseWithCovarianceStamped` ROS messages. ROS 2 serves as the backbone for time-critical operations, providing the low-latency, real-time communication necessary for local navigation via Nav2. Concurrently, pose messages from VLM are relayed to Kafka, which acts as the centralized message bus for the broader system, facilitating data exchange with external SOPRANO components. VLM is highly configurable, with key parameters defined in YAML and ROS launch files and support for command-line overrides. This design ensures operational flexibility and rapid prototyping by allowing real-time tuning without recompilation.

The following subsections discuss the localization methods developed in SOPRANO. The description is intentionally maintained at a high level, putting greater emphasis on the outcomes rather than the methodology, in

[1]FORTH RealSense ROS player: https://github.com/ehourdakis/RealsenseRecord/tree/main/ROS2/realsense_record_publisher

[2]Intel RealSense SDK: https://github.com/realsenseai/librealsense

order to preserve clarity, brevity and overall readability. Detailed technical specifics are available in the cited publications.

2.2.2.1 Feature-based vSLAM with absolute localization

Feature-based vSLAM approaches rely on matching distinctive features in the environment and using random sample consensus (RANSAC) [47] variants to filter out outliers. They offer several important advantages, such as robustness to environmental changes, computational efficiency, and scalability for large-scale environments. Using distinctive features, such methods ensure accurate and reliable localization and mapping, even in challenging conditions. Based on these considerations, FORTH has developed a cutting-edge feature-based vSLAM pipeline similar to ORB-SLAM2 [143] and PTAM [91]. The pipeline operates autonomously on each incoming frame, leveraging ORB [174] or BEBLID [208] features to ensure reliable feature detection and matching, and continuously delivers a 6DoF pose stream. Examples of the vSLAM system operating in industrial environments are illustrated in Figs. 6 and 10, whereas Fig. 11 depicts an outdoor environment.

The vSLAM system was improved through specialized upgrades of contemporary structure from motion algorithms combined with system-wide architectural restructuring, large-scale code refactoring, and numerous performance enhancements. Beyond custom algorithmic and feature improvements, significant effort was also invested in porting the vSLAM system from ROS 1 to ROS 2. This migration amounted to a substantial software engineering undertaking that was essential to adapt the software to the ROS 2 middleware architecture and ensure future-proof compatibility and long-term support. In the course of improving our pipeline, we extended the open-source OpenCV library by contributing our optimized Perspective-n-Point (PnP) solver implementation[3], which was successfully integrated into the official main repository via peer-reviewed pull requests[4]. We also made available the code[5] of our pose regression method [214] and made improvements to nanoflann[6].

The processing rate achieved by feature-based vSLAM depends on the capacity and computational load of the hosting hardware, and to a certain extent the content of the supplied images. Hence, the exact number of frames processed per second varies; however, it typically falls within several frames per second (fps), reaching up to 25-30 fps on capable desktop hardware. In the latter case, the primary speed limiting factor is the bandwidth achieved by ROS image streams rather than the per frame latency of vSLAM.

A practical issue pertaining to most feature-based vSLAM systems is that the 3D information they recover is expressed in an arbitrary coordinate system, typically determined from the starting segment of the camera trajectory. In other words, the employed reference frame is attached to the initial robot pose and subsequent position tracking relates to this arbitrary origin. Although a very common practice in vSLAM systems, this convention is nonetheless unsuitable for practical navigation that requires awareness of proximity to objects and/or regions that are integral parts of the workspace, situated in specific locations with respect to a physically meaningful reference. Hence, aligning a vSLAM-based estimate with a fixed coordinate system defined in the actual world is an important prerequisite for robot localization and planning. To achieve this, the similarity transformation that relates the two should be determined. This transformation consists of a rigid transformation (i.e., isometry) composed with isotropic scaling. Attaining such an absolute localization is clearly closely related to the ability to identify previously visited areas for loop closing [21, 219].

A key strength of the developed vSLAM system is that it integrates absolute localization capabilities in order to achieve the alignment of the SLAM coordinate system with a world reference frame. In this manner, the integrated system enables a robot to localize itself despite not having any prior knowledge of where exactly it is initially placed in its environment. The system is tailored to applications that involve repeated navigation in the same environment, expecting it to change moderately over time. The approach capitalizes on the observation that vSLAM exhibits sufficient accuracy when estimating small relative displacements between

[3]SQPnP solver: https://github.com/terzakig/sqpnp
[4]SQPnP in OpenCV: PR #1 and PR #2
[5]Rational regression for pose: https://github.com/terzakig/RRP
[6]Nanoflann KD-Tree library improvements PR.

proximate observations over short periods (in which cases vSLAM reduces to visual odometry [149]). However, as small errors accumulate over time, they can cause the camera's absolute position to drift from its true location. Hence, by relying on the vSLAM pipeline for real-time, incremental relative localization, and combining it with a subsystem that periodically provides absolute pose, continuous and accurate absolute localization can be ensured. The periodic absolute localization relies on representations constructed offline from casually collected imagery of the target environment. These representations capture the environment's structure and appearance in a way that accommodates fast look-ups and robust matching for images from a running vSLAM session. In addition to achieving absolute localization, this process also bounds the vSLAM localization error, promoting accuracy and reliability. For added flexibility, the vSLAM system also includes a mode that accepts absolute localization from an external source via a dedicated ROS topic. This allows the system to integrate localization data derived from PnP [213] or PnL [215] solutions. It is emphasized that the aforementioned features are possible without requiring a pre-deployed infrastructure such as fiducial markers or beacons, adding to the system's overall practicality. The availability of absolute poses facilitates the provision of an AR-like visualization for vSLAM: Given an estimated pose, the outline of a physical object can be drawn on the input image; its alignment with the actual object provides direct visual feedback regarding the accuracy of the estimated pose. This approach is adopted, for example, by visualizing the conveyor belt in Fig. 6.

Another distinguishing feature of the developed pipeline is its support for map serialization and loading. Specifically, the vSLAM system can save its current map, i.e., the set of map-points, descriptors and key-frames collected in a session by serializing and storing it in a binary file. Then it can read this binary file and generate a distinct map, separate from the map of the current session. The loaded map remains inert, alongside the active map and once the latter has gained sufficient overlap with the loaded one, the loop closure module will detect a match, which will subsequently trigger a map merge. After the merge, the current map has acquired all map-points and key-frames from the persisted map of the previous session and can subsequently use them to localize and/or close a loop in parts of the workspace which have not yet been inspected during the current session. This ability of the vSLAM system to store and subsequently load maps from previous sessions (i.e., past runs throughout the workspace), in order to merge them with the current map allows it to recover from unexpected loss-of-localization events. Prior maps ensure the continuity of operation and eliminate the need to manually intervene in a robot's mission in order to safely lead it into "known territory" to force a relocalization.

In practical robotic applications, robots often need to operate in a discrete "start-and-stop" manner rather than moving continuously. This mode of operation typically involves a robot navigating to a specific location, stopping to execute a particular task, and then resuming its navigation. A practical challenge during these stationary periods is the potential for anticipated camera occlusion. For instance, a robot arm extending to manipulate an object within its workspace can obstruct the navigation camera's field of view, compromising the input needed for localization. To account for such temporary obstructions, the vSLAM system offers a mechanism to freeze its camera input. This capability effectively "disconnects" the camera feed on request, allowing the robot to perform its tasks while temporarily shutting off the camera. Once occlusion is no longer a concern, the system can be instructed to re-activate its camera input.

2.2.2.2 Absolute localization in texture-poor environments

Pose estimation from line–plane correspondences Despite the advantages of feature-based vSLAM, its efficacy is inherently tied to the quality of detected and successfully matched point features. In challenging environments with homogeneous texture, repetitive patterns, or poor illumination, the reliability of feature extraction and matching can degrade considerably. For example, SOPRANO's automated construction use case (KUKA) involves environments that primarily consist of empty walls with very sparse texture. Such environments give rise to few features and, as a result, considerably restrict the applicability of feature-based visual localization approaches.

To address this, we devised a novel, plane-based method [212] that combines the depth map captured by an RGB-D camera in a structured environment with knowledge of the scene planes to compute the pose of that

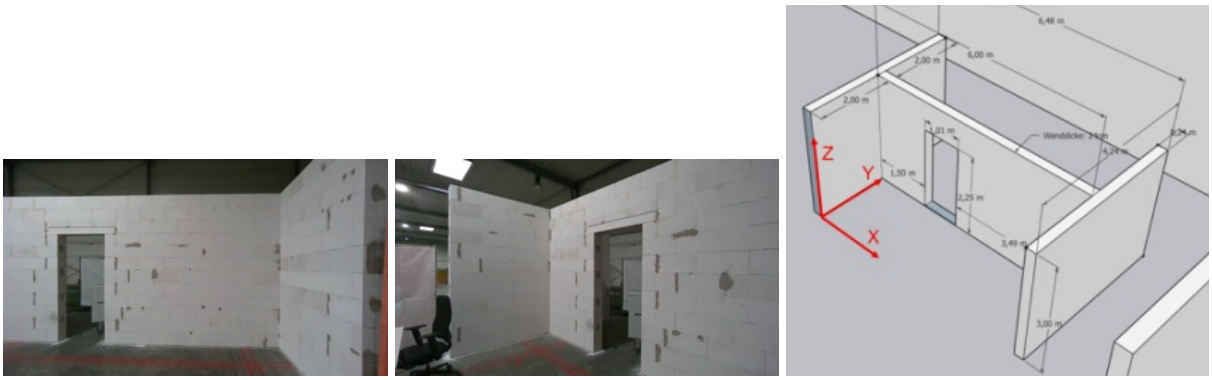


Figure 1: *Left, middle*: Representative views of KUKA's operating environment, a room of floor dimensions 6×2 m characterized by low-texture planar surfaces; *Right*: 3D model for the room, used to derive the parameters for the planes corresponding to the four walls and the floor; red arrows denote the employed coordinate triad.

camera with respect to an external, world coordinate system. This approach provides absolute localization by design, at the expense of minimal user input. Our method uses plane priors to form line–plane correspondences [134], i.e. associations between lines observed in an image and 3D scene planes, such that the back-projection of each line through the camera center lies on a specific 3D plane. These correspondences give rise to geometric constraints involving the pose that are enforced via a quadratic program. This program is solved by adapting a methodology we developed for the PnP problem [213]. The method employs automatic plane extraction that uses the depth channel of the RGB-D data to segment planar surfaces in the observed scene (see Fig. 2). Additionally, by synthesizing line segments on the basis of geometric support rather than the directional uniformity of photometric gradients, it overcomes the difficulties traditional line detectors face in low-texture regions and is able to estimate the pose of an RGB-D camera in an environment with textureless planes. The supplied line–plane correspondences are processed in a robust regression framework to detect outliers and determine the camera pose consistent with their majority. The method requires manual input for delineating world to image plane correspondences, which is achieved with the aid of a GUI that asks the user to color-code certain planar image regions to indicate their correspondence to specific world planes. An example of the method operating in the renovation use case environment (KUKA) is illustrated in Figures 2 and 3. Due to its reliance on human-in-the-loop input, the method is primarily suited for bootstrapping non-absolute localization systems, such as the SOPRANO Non-Visual Localization component (NVL; see Sec. 4) whose LiDAR-based approach is better suited to structured indoor settings such as offices or buildings. At this point, it is pointed out that this report uses VLM as an umbrella term for the component irrespective of algorithmic details; references to the individual vSLAM or plane-based approaches are inferred from the context.

A practical challenge arising in the estimation of camera pose from planar primitives regards the ambiguity of the process. Specifically, when the number of visible planes is minimal (i.e. 3), the underlying mathematical formulation can find multiple valid solutions for the camera's translation and rotation that satisfy the projection of observed planes into the image frame. As standard plane equations do not inherently distinguish between the "inside" and "outside" half-spaces of a boundary, the estimation may thus result in a "rotated" pose, equivalent to turning the camera upside down. In this case, the camera position is physically mirrored across the vertical axis of symmetry, resulting in a pose that is mathematically consistent with the visual observations but located in a physically unreachable region of the coordinate space. To resolve this ambiguity, a robust validation and corrective stage was introduced. Specifically, following the initial pose estimation, the resulting camera center is subjected to a physical consensus test using a majority voting scheme based on the original world normals. By calculating the signed distance of the camera center relative to the physical map boundaries, the process determines whether the estimated position resides within the valid navigable workspace. If a mirrored state is detected, the pose is corrected by a 180° rotation around the vertical axis. This dual-stage approach ensures that the final estimate maintains both mathematical precision and geometric consistency within the world frame. Pose mirroring is much less likely to occur if the number of matched visible planes is 4 or greater.

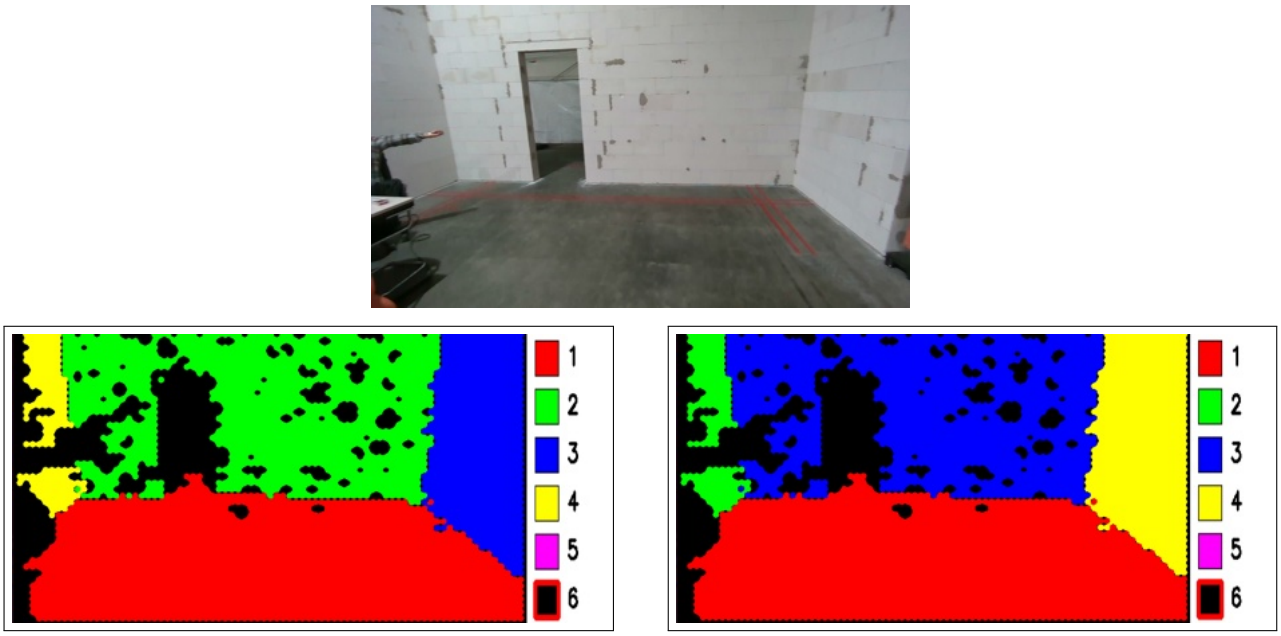


Figure 2: *Top*: An intensity image from an RGB-D pair. *Bottom*: Views of the plane detection GUI; on the left are the planes detected automatically from the depth image, with color labels assigned arbitrarily by the plane detection algorithm; black areas correspond to unclassified or missing depth pixels. On the right, planes are correctly annotated by the user, with the left wall in green, the middle wall in blue, the right wall in yellow, and the floor in red. The planes are annotated using a flood-fill mechanism, with color labels selected from the palette displayed on the right side of the GUI. User annotation establishes semantic correspondence between image and scene planes; for example, by assigning blue to the middle vertical plane (green in the plane detection on the left), the user explicitly identifies it as the middle wall.

It should also be noted that although Intel advertises a nominal depth accuracy of less than 2% at 4 meters for the D45x series, this figure is highly optimistic and assumes ideal conditions. In practice, we have observed much larger depth errors at that distance range, especially on low-texture or poorly lit surfaces where the stereo matching algorithm struggles. This empirical evidence agrees with evaluation studies like [74, 191] whose findings converge to the conclusion that D455's performance deteriorates significantly under common real-world conditions. Such conditions, for example, are when surfaces are angled relative to the camera, when the scene lacks texture, or when the camera is positioned beyond 3 meters away from the target surface. A primary reason for this behavior is that the infrared projector's signal decays at a longer range. Consequently, raw depth maps captured by D45x cameras should not be blindly trusted for high-precision applications without employing distance-based filtering or error-weighting mechanisms. VLM implements such mechanisms, for example the refinement of scene depths via optimizing multi-image observations (i.e., bundle adjustment [115]) or the robust plane fitting incorporated in line–plane pose estimation.

Pose estimation via Perspective-n-Line The Perspective-n-Line (PnL) problem seeks to determine the pose (i.e., position and orientation) of a camera from the image projections of a set of known 3D lines. Human-made environments typically abound with linear structures that give rise to straight line segments when imaged in 2D. Line segments can be accurately localized in images at a moderate computational cost, offering robust structural cues by delineating a scene's main elements. They are also moderately robust to noise, occlusions and illumination or viewpoint changes. Furthermore, they are often present on even poorly textured objects, for which techniques based on local patch detectors and descriptors, are not applicable. Hence, in certain structured environments with weak texture, straight line segments are the preferred type of visual features for a variety of visual analysis tasks. Within SOPRANO, we developed SQPnL, a novel PnL solver which can accommodate an arbitrary number of lines, has linear time complexity and consistently accurate performance [215]. Since the order of the general polynomial arising from PnL is 8, the solution to the PnL problem is not uniquely determined when only three line correspondences are available (i.e., $n = 3$) [238]. For $n \geq 4$,

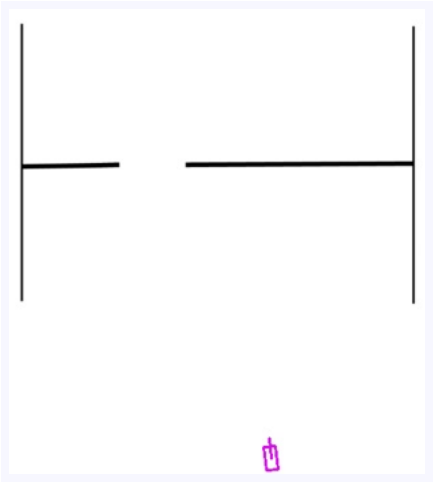


Figure 3: *Top*: Visualization of the pose estimated for the image in Fig. 2. The pose is shown superimposed on the room's floor map using a magenta rectangle with a line segment to indicate the orientation (i.e., camera viewing direction). The black lines correspond to the layout of the room's walls (only the I-shaped "inner" walls are drawn).

the solution usually becomes unique, provided that the available lines are not in a degenerate configuration, i.e. they are not coplanar or parallel. A C++ implementation of the SQPnL solver has been made publicly available[7]. Pose estimates computed by SQPnL are refined using the Levenberg-Marquardt non-linear least squares optimization algorithm to minimize the reprojection error between actual image line segments and their directions, as predicted with the estimated pose.

Similarly to the line–plane pose estimation, pose estimation from line correspondences is inherently subject to solution ambiguities arising from the symmetric structure of the problem. Unlike point-based methods, lines lack intrinsic cheirality: a 3D line and its mirror image through the camera center project to the same 2D line, yielding multiple geometrically consistent pose hypotheses that satisfy both the reprojection constraint and the positive-depth (cheirality) condition. This ambiguity is particularly acute when the available line correspondences are limited in number or exhibit insufficient geometric diversity — for instance, when lines are nearly coplanar or share similar orientations in 3D space — conditions that are common in structured indoor environments. In such cases, standard algebraic disambiguation criteria, including minimization of the reprojection error, do not identify the correct solution, as the competing hypotheses produce virtually identical residuals. To resolve this ambiguity, we adopted a depth-based disambiguation strategy that takes advantage of the metric depth measurements provided by the RGB-D sensor. Specifically, for each pair of 2D line segments, we compute their image-plane intersection and retrieve the corresponding depth measurement from the sensor's depth map using median filtering over a 3×3 pixel neighborhood to mitigate the effect of missing or erroneous readings. The predicted depth at each intersection is then obtained by transforming the corresponding 3D line intersection point — computed as the midpoint of the closest approach between the two world-frame lines — into the camera frame under each candidate pose. The pose hypothesis whose predicted depths are most consistent with the measured values, quantified as the average absolute depth discrepancy over all valid intersection points, is selected as the correct solution. This approach requires no additional correspondences, no manual intervention, and no assumptions about scene structure beyond the availability of metric depth, making it well suited for RGB-D based pose estimation pipelines.

To provide an additional source of pose estimates and thus increase redundancy, the SQPnL-based estimator has been integrated into the line–plane pose estimation pipeline discussed in Sec. 2.2.2.2. SQPnL leverages the pipeline's plane detection capability to automatically identify adjacent planes that give rise to structural 3D lines. These line primitives are extracted analytically by calculating the intersections of adjacent planes and then projected onto the image plane to populate a graphical user interface (GUI), allowing a human user to manually refine and edit the line features (see Fig. 4). This semi-automatic approach ensures that the resulting

[7] SQPnL perspective-n-line solver: `https://github.com/terzakig/sqpnl`

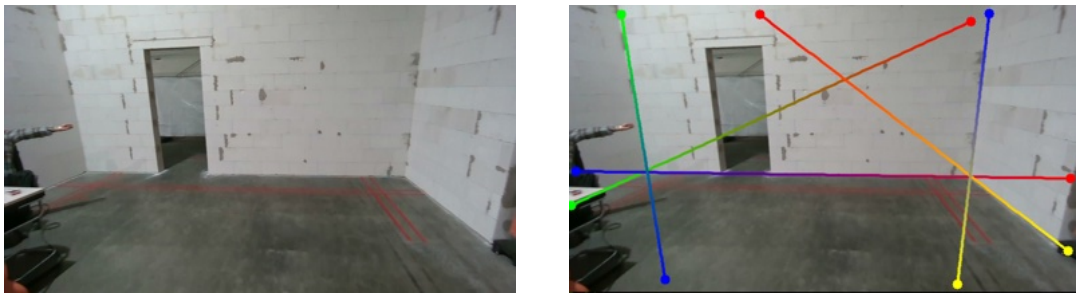


Figure 4: *Left*: The intensity image of Fig. 2 (top), repeated for convenience. *Right*: Line segments, defined by plane intersections, projected onto the image and manually refined to align with wall intersections. Their exact endpoints are irrelevant; only the supporting infinite lines matter. Line segment colors denote the planes that define each segment; for instance, the horizontal segment transitioning from blue to red corresponds to the intersection of the blue and red planes (i.e., the middle wall and the floor) in Fig. 2 (bottom right). The user can adjust the segment's position and orientation through point-and-click mouse interaction.

lines maintain high topological accuracy through both automated geometric derivation and expert-informed validation. The user also has the option of choosing between the estimates obtained with the line–plane and the PnL–based pose estimation.

Vertical Vanishing Point as an Orientation Cue The vertical vanishing point (VVP) provides a key geometric constraint on the orientation of a camera in 3D space. The VVP links a camera's internal coordinate system to the global gravity vector through the projection of parallel vertical lines. In structured or "Manhattan World" environments, features such as wall intersections, door frames, and columns converge at this point, which is determined solely by the camera's orientation and remains invariant to its translation [24]. By identifying the VVP, one can analytically solve for the roll and pitch of the camera, effectively reducing the rotational estimation from a three-degree-of-freedom search to a single-degree-of-freedom problem centered on the heading or yaw [95, 71]. In the particular SOPRANO use case where the camera is rigidly mounted to a mobile robot traversing a planar surface, its extrinsic orientation relative to the ground plane remains invariant over time. Specifically, if the ground is perfectly flat and the robot's suspension is neglected, the camera's pitch (tilt relative to the horizon) and roll (rotation about the optical axis) angles are fixed by the mechanical mounting. This physical constraint means that the VVP remains constant within the image frame, regardless of the robot's translation or heading. This, in turn, implies that the VVP can be estimated from an image having suitable vertical lines and then reused, regardless of whether suitable lines are present or not.

Efforts are currently underway to integrate VVP constraints into the PnL-based pose estimation framework to improve the accuracy of orientation estimation. Inclusion of the VVP also regularizes the PnL estimation, maintaining estimation stability even when the number of detected lines falls below those minimally required for a well-constrained pose.

2.2.2.3 Datasets

To evaluate the performance of vSLAM algorithms, established tools [58, 18] and datasets are available, including KITTI [53], TUM RGB-D [203], TUM-VI [187], EuRoC [19], ETH3D [186] and ICL-NUM [60]. A fairly recent review of relevant RGB-D datasets can be found in [250]. While publicly available datasets are invaluable for debugging and benchmarking VLM, they often fall short of capturing the full complexity, variability, and subtle nuances of particular use case environments. To address this, we have utilized our open-source RealsenseRecord[8] utility to collect several RGB-D image sequences from the actual operational environments at CRF and KUKA. Regarding the E-TERRY use case, since it does not use a RealSense camera, standard ROS2 bags were recorded for it. The RealsenseRecord utility application facilitates the acquisition of synchronized

[8]RealsenseRecord utility: `https://github.com/ehourdakis/RealsenseRecord/`.

RGB, depth and inertial measurements with RealSense cameras. It features a flexible playback mechanism that can stream captured data using either real-time system wall clocks or simulated capture timestamps to ensure temporal consistency during offline analysis. Within the course of SOPRANO's implementation, the RealsenseRecord utility has been systematically extended with new features to support emerging technical needs. The captured data have been used for testing and evaluating the VLM algorithms, allowing us to gain deeper insights into potential system limitations and identify areas for improvement. Some of these acquired sequences also include moving artifacts to evaluate the VLM's resilience to independent motion.

2.2.2.4 Accuracy evaluation metrics

The output of feature-based vSLAM consists of the estimated camera trajectory along with a map corresponding to a sparse representation of the environment structure. In principle, it is possible to evaluate the quality of the map. However, since accurate ground truth maps are difficult to obtain, map evaluation is not pursued in practice and only the quality of an estimated camera trajectory is evaluated.

To evaluate the localization error associated with a set of estimated trajectories, the average absolute trajectory error (ATE) [203] across these trajectories will be employed. Before computing the ATE, two trajectories must first aligned spatially and then their absolute pose differences are evaluated. ATE represents poses with homogeneous 4×4 transformation matrices each of which amounts to a rigid transformation. To compute it for a sequence of estimated poses $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$ and a sequence of ground truth poses $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_n$, the two sequences must first be aligned spatially as they are typically specified in different coordinate systems. This is achieved by computing the similarity transformation $\boldsymbol{S}$ corresponding to the least squares estimate that maps the estimated trajectory $\boldsymbol{P}_{1:n}$ onto the ground truth trajectory $\boldsymbol{Q}_{1:n}$. The absolute pose error at time i can then be computed as $\boldsymbol{E}_i = \boldsymbol{Q}_i^{-1} \boldsymbol{S} \boldsymbol{P}_i$. The translational error corresponding to $\boldsymbol{E}_i$ is the norm of its translation component, whereas the rotational error is given by the angle of the residual rotation that is equal to $\arccos\left((\mathrm{trace}(\boldsymbol{R}_i) - 1)/2\right)$, with $\boldsymbol{R}_i$ being $\boldsymbol{E}_i$'s rotation component, i.e. its top-left 3×3 submatrix. Finally, ATE is defined as the root mean square (RMS) of the translational components of the error matrices $\boldsymbol{E}_i$. A rotational error is similarly defined from the average of the rotational errors, i.e. the residual rotation angles. It is pointed out that the computation of ATE just outlined assumes that ground truth data regarding the trajectory of the employed camera are available. Obtaining such data is challenging, if not infeasible, in certain cases. Towards this end, the option adopted in SOPRANO is to rely on a batch structure from motion reconstruction engine such as COLMAP [183] in order to recover approximate ground truth (i.e., pseudo-ground truth).

The evaluation of line–plane pose estimation makes use of ideas similar to those employed to evaluate vSLAM. However, the lack of scene texture renders the batch process of extracting approximate ground truth employed in the vSLAM case inapplicable. To remedy this, the following semi-automatic procedure was adopted. First, by using the graphical interface developed for PnL, a few characteristic line segments whose pre-images in the world are easily identifiable can be delineated in an input image. Examples of such line segments are the intersections of walls or the outlines of wall openings such as doors. To reduce manual intervention, the GUI automatically augments this set with line segments corresponding to intersections of neighboring planes detected in the image. Image line segments can be interactively adjusted in the GUI, to ensure precise alignment with their physical counterparts. Once this is verified, a pose estimate is computed from these segments using our SQPnL solver [215]. The rotation error for a true camera rotation $\mathbf{R}_\mathrm{g}$ and an estimate $\mathbf{R}_\mathrm{e}$ is simply the angle of rotation about a unit vector that transfers $\mathbf{R}_\mathrm{e}$ to $\mathbf{R}_\mathrm{g}$, given by $\arccos\left((\mathrm{trace}(\mathbf{R}_\mathrm{g}\mathbf{R}_\mathrm{e}^\mathrm{T}) - 1)/2\right)$. The translation error for a true translation $\mathbf{t}_\mathrm{g}$ and an estimate $\mathbf{t}_\mathrm{e}$ is simply the Euclidean norm of their vector difference $|\mathbf{t}_\mathrm{g} - \mathbf{t}_\mathrm{e}|$. It is noted that since the PnL-based procedure described above for collecting ground truth pose data requires certain manual input for each frame, it is not suitable for use on large image datasets.

2.3 Results

The CRF manufacturing use case concerns a vehicle door disassembly scenario in which human workers perform repetitive tasks on door dollies standing on a conveyor belt. A wheeled mobile robot drives around the



Figure 5: View of the operating environment at CRF. The black rectangle with the yellow sawtooth border is a conveyor belt that is shown here carrying three assembly workstations *(left)*. A CAD drawing in which the horizontally hatched area is the conveyor belt whereas the I-shaped herringbone area corresponds to the light blue region in the picture *(right)*.

shop floor, carrying tools and parts as needed. A general view of the operating environment is shown in Fig. 5. The environment provides sufficient texture, permitting the application of feature-based SfM techniques.

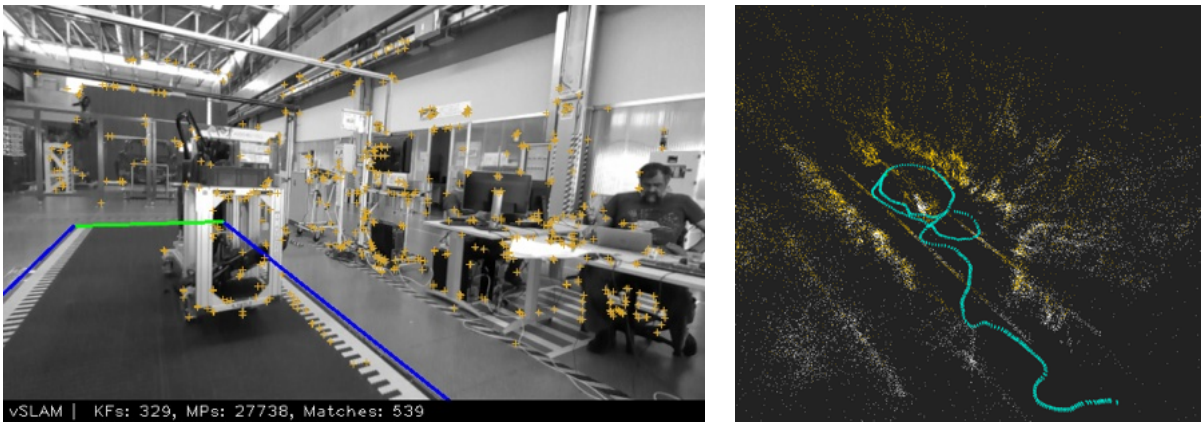


Figure 6: Sample tracking and reconstruction views from feature-based vSLAM in CRF. On the left, successfully tracked ORB features are shown as orange crosses on an input image. The estimated absolute pose is visualized by projecting the conveyor belt's outline on the image (blue and green lines). A sparse reconstruction of the workspace and corresponding key-frame viewpoints drawn in cyan are shown on the right. The current position of the camera corresponds to the topmost point of the upper loop of the 8-like segment of the trajectory near the middle of the right figure.

An example of the feature-based vSLAM (see Sec. 2.2.2.1) operating in CRF's environment is illustrated in Figs. 6 and 7. These show the estimated trajectory of camera poses along with the recovered sparse map of the environment. Additionally, the absolute pose estimated for a particular image is used to draw on that image the outline of the conveyor belt, providing direct visual feedback for the accuracy of the estimated pose.

Leveraging the environmental texture density, we extracted approximate ground truth from ten sequences captured with a handheld RealSense D455 camera. These sequences are characterized by rapid orientation fluctuations and variations in the imaged environment stemming from rearranged objects, background clutter, moving humans, illumination changes, etc, and were acquired in two batches which were two months apart chronologically. Each sequence is several thousand frames long and their pixel resolution is 640×360 at 30 fps.

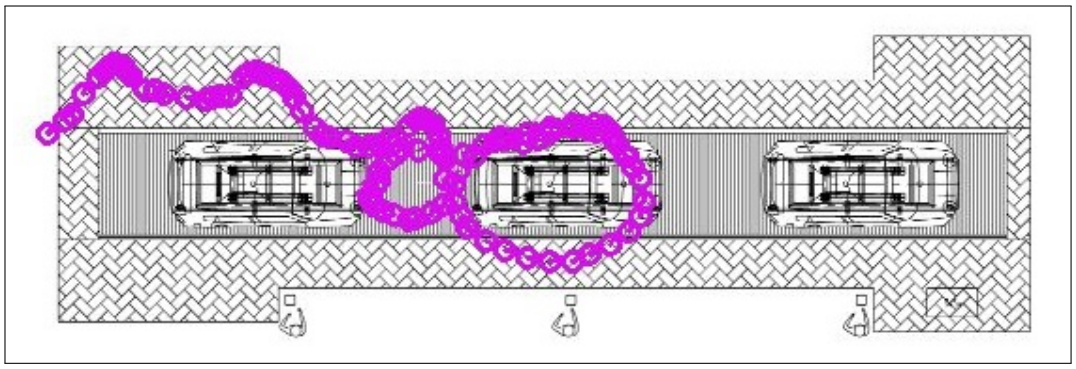


Figure 7: Robot 2D trajectory for the example in Figure 6. The trajectory is shown superimposed on CRF's floor map using magenta circles to denote positions, each with a marked radial direction indicating orientation (i.e., heading).

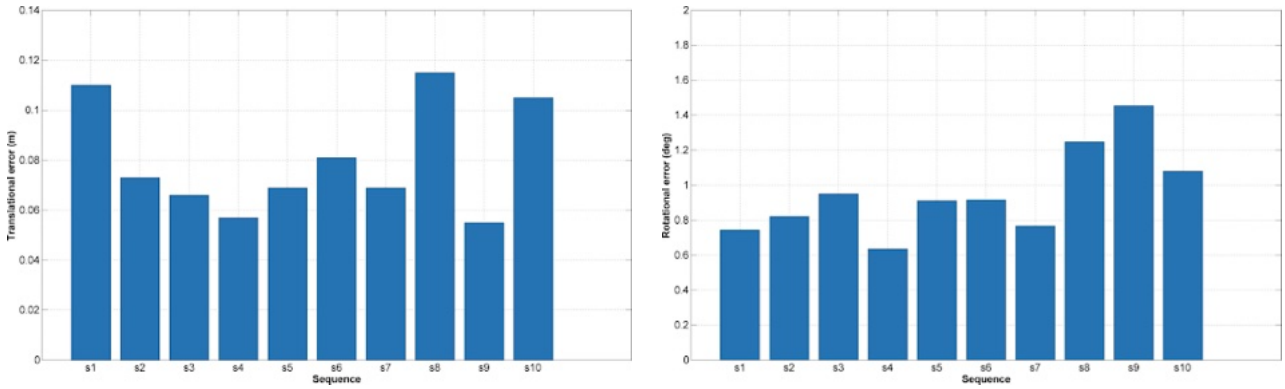


Figure 8: CRF: RMS translational and rotational errors (ATE) for feature-based VLM over ten test sequences. The average errors are respectively 0.08 m and $0.95°$.

Pseudo-ground truth poses for each sequence were obtained by temporally sub-sampling them by a factor of 5 and subsequently reconstructing them with the COLMAP structure from motion pipeline [183]. The temporal sub-sampling aimed to keep the number of images presented to COLMAP and the associated computational overhead, relatively moderate. Following this, the poses of the 4 intermediate, left-out images were computed with spherical linear interpolation (SLERP) from the COLMAP poses of their two adjacent image frames. This procedure yielded ground truth for over 75K images from CRF. Figure 8 illustrates the RMS translational and rotational errors computed for the ten sequences from the poses estimated by VLM. The translational and rotational errors achieved by VLM are respectively below 0.11 m and $1.6°$ for all sequences. Averaged over the 10 sequences, the **translational and rotational errors** are, respectively, $\mathbf{0.08}$ m and $\mathbf{0.95}°$. These errors are well below the desired thresholds of 0.2 m and $10°$, hence the relevant KPI-O1-1 has been achieved.

The renovation use case at KUKA involves a dedicated mock-up room constructed specifically for the execution of tasks such as drilling, cladding, and sanding. This space provides a controlled environment for these activities, ensuring safety and minimizing disruption to other areas. Sample views of the room along with a CAD model are provided in Fig. 1. Visual localization in this environment does not rely on the use of texture but involves a single depth image from which the camera's 6D pose is estimated as discussed in Sec. 2.2.2.2. The estimated pose is visualized by projecting it on the floor plane. Sample results are shown in Figs. 2 and 3.

To evaluate the localization accuracy of line–plane pose estimation for the KUKA use case, the rotation and translation errors pertaining to certain images were separately measured as detailed in Section 2.2.2.4. To collect ground truth pose for an image so that it can be used to measure localization performance, the PnL-based interactive procedure described in Sec. 2.2.2.2 was used. Figure 9 illustrates the average translational and rotational errors for the pose estimates of ten images obtained with VLM and the PnL-based procedure. These images were acquired with the robot's onboard camera, after moving the robot to various locations in the

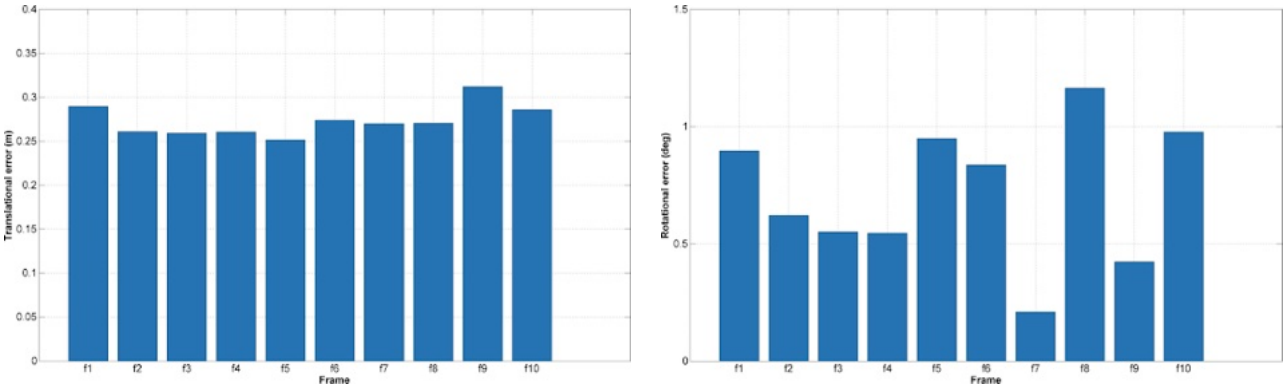


Figure 9: KUKA: Translational and rotational errors for plane-based VLM over ten test frames. The average errors are respectively 0.27 m and 0.72°.

KUKA operating environment. The translational and rotational errors achieved by VLM are respectively below 0.35 m and 1.2° for all frames. The average translational error achieved by VLM is 0.27 m and moderately exceeds the 0.2 m threshold of KPI-O1-1, while the rotational error is 0.72°, well below the desired 10°. Compared to those measured for CRF, the aforementioned translational errors are notably higher, which is a direct consequence of texture paucity in KUKA's environment. Still, these results are quite remarkable considering that they rely exclusively on noisy depth measurements whose accuracy quadratically degrades with distance from the camera [74].

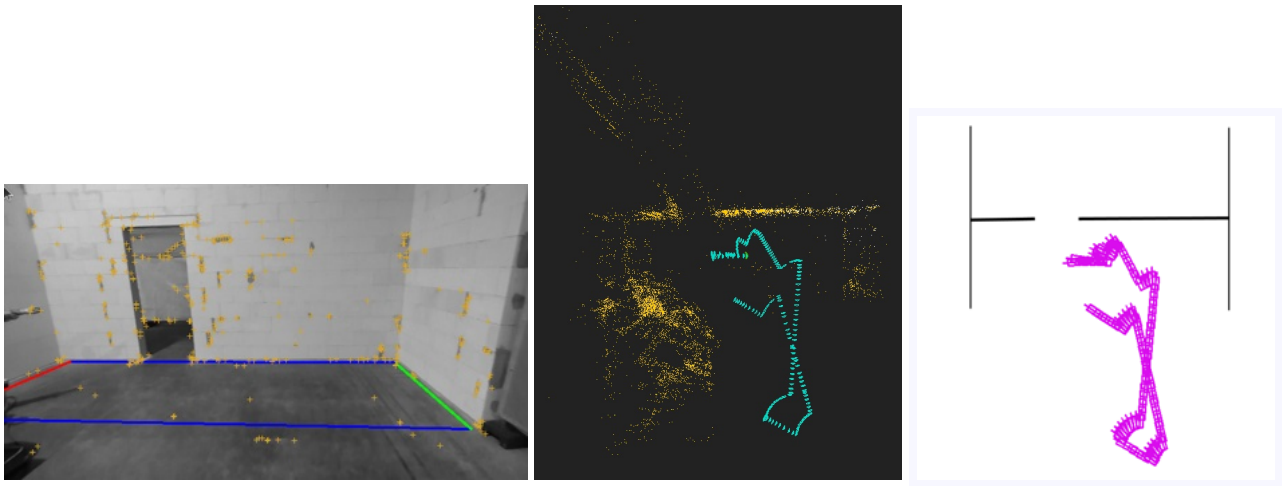


Figure 10: Sample results from the feature-based vSLAM applied in KUKA's environment. On the left, successfully tracked ORB feature locations are shown as orange crosses on an input image. The estimated absolute pose is visualized by projecting the floor's outline on the image (red, blue and green lines). A top view of the sparse 3D reconstruction of the workspace and corresponding key-frame viewpoints drawn in cyan are shown in the middle; the point cluster to the left of the camera trajectory originates from the desk next to the left wall of the room. On the right, magenta rectangles depict the robot's 2D trajectory in the room's floor map.

Although feature-based vSLAM remains operational in KUKA's low-texture environment (cf. Fig. 10), its robustness is diminished. This is because the scarcity of distinctive texture adversely affects the reliability of feature detection and matching. Consequently, in this environment, the vSLAM system becomes more sensitive to noise and fast motion, which can lead to tracking loss and intermittent failures that compromise both the continuity and accuracy of the localization. Conversely, NVL (Sec. 4) suits these environments better.

Regarding the symbiotic farming use case (E-TERRY), the corresponding outdoor environments abound in high-contrast natural features. Hence, they provide large sets of stable anchors for robust tracking, which positively impacts the reliability of feature-based vSLAM as a result. Additionally, the low velocity of the

platform ensures high temporal overlap and eliminates motion blur, further facilitating more robust feature tracking. The poses estimated by vSLAM in this case are relative, and therefore georeferenced (i.e., aligned to a real-world coordinate system) using onboard real-time kinematic global navigation satellite system (RTK-GNSS) measurements. An example of localization in an environment in which E-TERRY operates is in Fig. 11.

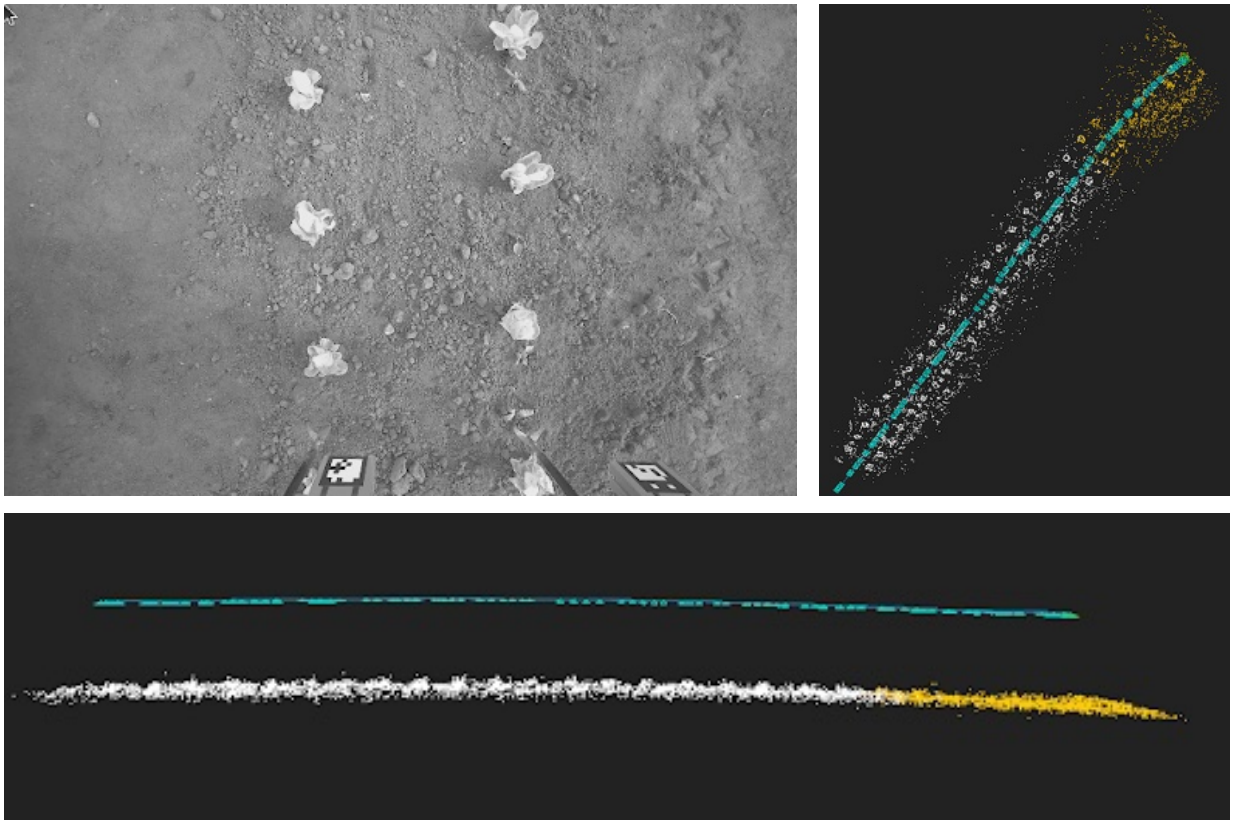


Figure 11: E-TERRY's view of an agricultural field where the camera is pointing down towards the soil and plants that appear brighter *(top left)*. Top and side views of the camera trajectory estimated with vSLAM (in cyan) and the recovered 3D map (white/yellow) *(top right, & bottom).*

2.4 Discussion

The efficacy of point feature extraction and matching depends on the characteristics of a particular environment and can be compromised in scenes lacking sufficient visual texture. To mitigate this limitation and provide a robust solution for high-fidelity pose estimation across diverse operational environments, the visual localization framework developed in SOPRANO proceeded in two directions. Each direction leverages a distinct data modality offered by RGB-D sensors, depending on whether the target environment has sufficient texture or not. Specifically, the framework utilizes a feature-based vSLAM system for photometric (i.e., RGB) matches in richly textured areas, while transitioning to a geometric, depth plane-based system for poorly-textured scenes. The integration of Perspective-n-Line (PnL) estimation provides an additional fallback in certain situations.

Collectively, the developments described above allow the localization component to refrain from a singular, critical reliance on visual texture, ensuring that it remains capable of determining precise spatial coordinates even in challenging, feature-degenerate environments. This dual-system approach establishes a reliable foundation for both continuous tracking and absolute pose bootstrapping within complex industrial and architectural contexts. Visual localization in SOPRANO has reached TRL 6, having been successfully demonstrated in the project's real-world environments, thereby confirming its feasibility and robustness. While all use cases have validated the core functionality of VLM, TRL 7 has already been achieved in CRF, as the component has been successfully validated against operational and performance requirements in the target environment.

3 Object Perception [OBP]

3.1 Technology overview and module specifications

The Object Perception (OBP) component aims to provide real-time detection and 6D pose estimation of objects in a robots' operating environment exploiting visual information from cameras deployed on the robot as well from stationary cameras in the environment. Object detection and 6D pose estimation are two cornerstone tasks in computer vision. Detection involves identifying objects in a scene and localizing them within an image by determining their precise position and size, typically represented as bounding boxes or segmentation masks. In contrast, 6D pose estimation extends this by determining an object's position (3D translation) and orientation (3D rotation) relative to the 3D coordinate frame of the camera. These capabilities are essential for robotic systems interacting with objects, enabling tasks such as grasping, manipulation, and assembly. OBP tackles challenges in detecting and localizing static and moving objects, including cases of objects with complex surface properties, varying occlusions, and diverse shapes and sizes commonly encountered in the industrial context of the SOPRANO use cases. In addition, it addresses the broader challenges of uncontrolled imaging conditions and variations in indoor and outdoor industrial settings.

To tailor its approach to specific use case requirements, OBP categorizes problems into two primary classes: a) object detection (and 3D position estimation), identifying and classifying objects in the 2D image plane and estimating their 3D position in the real world, and b) 6D object pose estimation, providing precise position and orientation of object(s) to support robotic manipulation, such as grasping and unscrewing. The SOPRANO use cases prioritize different aspects of the OBP capabilities. Specifically, the E-TERRY scenarios focus on the detection of specific object classes in the scene (e.g. boxes) and estimation of their distance to the robot to support tasks related to safety and recognizing humans in the vicinity of the robot to initiate further interaction and collaboration (see HMO sec. 5). CRF scenarios, in addition to the visual detection of specific object classes in the assembly line (e.g. car door, humans), also necessitate the estimation of the full 6D pose to support tasks requiring precise physical interaction between robots and objects (e.g. tools, tool holders, screws).

The OBP component processes visual data from on-board and off-board cameras, primarily exploiting RGB images for object detection and 6D object pose. In addition, depth information from RGB-D sensors (passive or active) is exploited to estimate the object positions in 3D space. The component employs machine and deep learning methods, requiring annotated data to train the neural networks employed. These data are necessary for accurate model training, enabling reliable object detection and pose estimation in real-world scenarios. These typically include images, with corresponding annotations such as object labels, bounding boxes, and 6D pose information. A summary of the captured and annotated data for each use case is shown in Fig. 12. The OBP component can operate in two modes depending on the application requirements: (i) Continuous mode, where object detection and pose estimation are performed at a fixed rate (e.g., per frame or at a predefined frequency), enabling real-time monitoring of dynamic environments. This mode is primarily used in scenarios such as human detection and safety monitoring (as in E-TERRY, CRF workstation monitoring), where timely updates are critical. (ii) On-demand mode, where perception is triggered explicitly by other system components (e.g., task planner or robot controller) when a specific action is required, such as grasping or unscrewing. This mode reduces computational load and allows the use of more computationally intensive methods (e.g., high-precision 6D pose estimation) when needed.

Additionally, OBP provides an unbiased confidence score for pose quality, ensuring reliability before executing robotic actions (e.g., grasping). Furthermore, to enhance scalability and efficiency, OBP automates training data generation for 6D pose estimation, and explores new methods for novel object 6D pose estimation exempting reliance on training data, thus supporting continuous improvements in pose estimation algorithms (see sec. 3.2.2.4). The outputs of OBP, including 2D segmentation masks, 6D poses, and confidence scores, are utilized by other SOPRANO components such as Human Monitoring, MH-MR Task Allocation, Robotic Capabilities Implementation, and Safety Tools.

In the sections that follow, we discuss related methods to the problem of 2D object detection and 6D pose estimation before providing further details on the approach followed in SOPRANO and related preliminary results. We note that the analysis of the state-of-the-art was done during the first phase of the project.



(a) (b) (c)

Figure 12: Use case environment examples: (a) tool 6D pose estimation and (b) screw detection in CRF, (c) person detection in E-TERRY.

3.2 Methodology and implementation

3.2.1 Existing approaches (SOTA)

3.2.1.1 2D object detection

Object detection is a fundamental problem in computer vision that involves identifying and locating objects within an image or video. State-of-the-art methods for 2D object detection have been extensively reviewed in several works, e.g. [257, 194, 6, 195, 252, 194], which categorize methods based on different criteria, such as the underlying computational approaches (e.g., convolutional neural networks versus transformers), the type of supervision (fully, semi-, or self-supervised learning), and the specific target applications [114, 152]. Traditionally, 2D object detection was dominated by convolutional-based architectures, such as YOLO [170], RetinaNet [109], and R-CNNs [55], which offer impressive computational efficiency and real-time performance but struggle with accuracy on smaller or overlapping objects due to information loss in the feature extraction process. However, more recent works have increasingly shifted towards leveraging transformer-based approaches, such as Co-DETR [256] and DINO [152], which incorporate attention mechanisms to better handle contextual information and long-range dependencies, ultimately improving detection accuracy in cluttered environments.

The latest advances also explore the integration of self-supervised and semi-supervised learning techniques, which have helped address the problem of limited labeled datasets. For example, Siamese DETR [30] and Unbiased Teacher [114] utilize large amounts of unlabeled or sparsely labeled data to improve model robustness, particularly in scenarios where labeled data is either scarce or expensive to obtain. These methods allow models to learn generalized features without depending solely on large, labeled datasets, improving their applicability in new environments or for novel object classes.

In addition to classic object detection, instance segmentation remains a significant branch of 2D object detection. Region-based convolutional neural networks (R-CNNs) and their variations [55] have long been used for this purpose. These methods work by generating region proposals based on assumptions of color, texture, and size before performing the detection task on the regions of interest. More recent works incorporate transformers [105] to further enhance segmentation, addressing weaknesses related to occlusion and handling objects in intricate settings with overlapping boundaries. Additionally, the advent of foundation models, such as Segment Anything Model (SAM) [90], has further revolutionized segmentation tasks by providing versatile and adaptable frameworks. These models are pre-trained on massive datasets and designed to generalize across a wide range of segmentation tasks without the need for extensive fine-tuning, making them highly effective in handling complex scenarios and diverse object categories.

Recent works also focus on advancing *semantic segmentation* techniques, with architectures like U-Net [172] employed to classify each pixel in an image as belonging to a specific class (e.g., background, ground, wall).

Despite its success, U-Net can produce artefacts like unclosed boundaries or disconnected segments. To address these shortcomings, novel methods such as CNOS [147] have been proposed, leveraging CAD models to improve segmentation accuracy for previously unseen objects. This approach improves the ability of models to generalize to novel object categories, reducing the need for exhaustive manual labeling and enhancing the applicability of 2D detection in dynamic and variable environments.

Overall, while the latest 2D object detection techniques continue to show improvements in accuracy and robustness, they still face challenges in environments involving complex object interactions, clutter, and occlusions. Innovations such as transformer-based architectures, the integration of synthetic data, and self-supervised learning are pushing the boundaries of current capabilities, but there remain opportunities to enhance detection and segmentation performance for transparent, reflective, and weakly textured objects, especially in real-world industrial applications.

3.2.1.2 6D object pose estimation

6D pose estimation aims to estimate the transformation from the object coordinate system to the camera coordinate system. This transformation is defined as a translation and a rotation in 3D.

State-of-the-art methods for 6D pose estimation are extensively reviewed in several works, e.g. [33, 87, 64, 175, 169, 234, 255], that usually categorize object localization according to the prior information, the number of views, the type of computer vision methods being used and whether they exploit 2D or 3D information derived by different sensors [67, 199, 64], single or multiple views [97], classic or deep learning methods [67, 64] and correspondence, template or voting based approaches [41]. However, recent works primarily exploit deep learning methods, dominating the field in estimating the 6D pose of objects from a single image [70].

Most instance-based methods, for which the 3D model of the object is available, manage to achieve very high scores in recall accuracy on benchmark datasets [70]. However, several challenges persist especially in industrial objects: symmetries may cause ambiguous pose predictions; reflective surfaces distort key features due to inconsistent lighting; weak textures exhibit lack of distinct visual features for reliable correspondences. These challenges often necessitate advanced techniques, such as symmetry-aware deep learning models, specialized imaging methods, or augmented training data, to improve robustness in real-world applications. In addition, most deep learning methods require a large amount of training data, as well as obtaining both training and testing data from the same object space. These assumptions, though, limit the applicability and robustness of object detection and localization, especially in real-time robotic applications in industrial environments, where similar objects with the aforementioned characteristics are commonly met.

The latest works try to tackle those issues using different approaches. To enhance model accuracy while relying on a smaller amount of training data, data augmentation techniques can be employed. In most cases, simple methods such as geometric and intensity transformations (e.g., rotation, translation, jittering, blurring) [67, 63] are commonly utilized. However, more advanced approaches, including offline texture generation and photorealistic rendering, may also be applied, albeit at the cost of increased processing time and human effort. Building upon this idea, recent works leverage synthetic data to overcome the time-consuming and non-scalable annotation of real data, while also addressing challenges such as the lack of realism. For instance, Chen et al. [26] introduced TexPose, a pipeline aiming to bridge the gap between synthetic pre-training and real data usage. The pipeline begins by pre-training a pose estimator on a synthetic dataset to produce coarse initial poses for objects in a real dataset and further use these poses by a texture learner to extract realistic object textures from real images.

Regarding the object special characteristics, mainly objects with occlusions, cluttered background, weak texture, symmetries, transparencies, and reflective surfaces are still open issues. In this direction, EPOS [67] managed to handle global and partial object symmetries by defining object-surface fragments and using a single RGB image. DOPE++ [79] used RGBD images and applied a random mask local processing method and a multiscale fusion pose estimation module to cope with occlusions and scale differences for weakly textured objects. Pose estimation of transparent objects has also been examined [20, 78] however it remains one of the

most challenging tasks, mainly because of the refraction, reflection, and absorption of light and the fact that the transparent objects appear differently in different backgrounds and illumination conditions.

Another challenging task involves estimating the 6D pose of unseen objects, as it requires the generalization of methods to objects not encountered during training. This capability is critical for advancing robotic applications, particularly in tasks such as object grasping and manipulation in dynamic, real-world environments [57, 98]. Whether in industrial warehouses, domestic or outdoor settings, enabling robots to handle novel objects without costly retraining or extensive data collection ensures scalability and versatility [45, 148]. Methods in this domain are typically divided into three main categories: **model-based**, which rely on precise 3D models of the objects for alignment and pose estimation; **model-free**, which avoid explicit dependence on object models, relying on video or images capturing the object to infer its pose; and **category-based** approaches, which aim to generalize across object categories by leveraging shared semantic or geometric properties.

Model-based methods dominate the field when precise 3D models of objects are available. These approaches often achieve high accuracy by aligning observed features with those of rendered images created from the object model using techniques such as correspondence matching [148, 153] or iterative refinement [98, 137]. However, their reliance on detailed 3D models makes them less scalable and adaptable, particularly when encountering new objects or variations for which no 3D models are available. This dependency limits their effectiveness in dynamic, real-world environments, where creating or obtaining accurate 3D models for every object is both challenging and time-consuming.

Model-free methods address the challenges of pose estimation by bypassing the need for explicit 3D models and instead relying on more readily available data to infer an object's pose. These approaches typically require manually labeled reference views of the target object to establish correspondences or reconstruct representations necessary for pose estimation [63, 233, 207]. Dense RGBD template matching frameworks [63] and methods leveraging graph attention networks [207] have been proposed to exploit semantic and geometric relationships between query images and reference views. Recent approaches aim to bridge the gap between model-based and model-free paradigms. In particular, FoundationPose [229] introduces a unified framework that supports both setups by leveraging neural implicit representations, enabling pose estimation either from CAD models or from a small set of reference images without object-specific retraining. Weakly supervised methods like NeRF-Pose [102] employ Neural Radiance Fields (NeRFs) to reconstruct implicit 3D object models from multi-view weak labels such as segmentation masks and relative camera poses. These NeRF-based approaches integrate object reconstruction with pose regression pipelines, achieving accurate 6DoF pose estimation even under weak supervision. Although these techniques demonstrate strong generalization capabilities for unseen objects, challenges such as reliance on multiple reference views and limitations in handling low-textured objects persist [62, 206]. Continued research aims to address these limitations by reducing the dependency on extensive reference data and enhancing robustness across diverse scenarios.

Category-based methods generalize pose estimation to objects belonging to a known category rather than specific instances. Techniques such as normalized object coordinate spaces (NOCS) [225] enable alignment of unseen objects with canonical category representations, facilitating pose estimation for novel objects within a known category. These methods leverage shared semantic and geometric properties across objects in the same category, allowing for greater flexibility when precise object models are unavailable. However, they may still face challenges in generalizing to objects outside predefined categories or handling fine-grained variations within a category [123]. Recent improvements incorporate deep learning-based feature extraction and geometric consistency checks, enhancing performance across diverse category-level tasks.

3.2.1.3 Datasets

A variety of benchmark datasets are widely used for object detection and localization. Those datasets depict objects of varying sizes and characteristics, either isolated or within complex scenes with multiple object instances. LM [66], LM-O [16], HB [83], RU-APC [171], IC-MI [210] and YCB-V [236] contain objects of everyday life, depicted in complex scenes. T-LESS [68] depicts industrial objects with symmetries, occlusions,

and weak texture, while ITODD [40] and IC-BIN [39] depict multiple instances of identical objects, simulating the bin-picking problem. ITODD [40] represents one of the most hard-to-solve cases of identical overlapping, reflective metallic objects under imperfect illumination conditions. Additionally, the BOP (Benchmark for 6D Object Pose Estimation) [70] provides a standardized framework for evaluating 6D pose estimation methods across multiple of these datasets[9] and in various tasks. For 2D object detection tasks, the COCO (Common Objects in Context) dataset[10] [110] is widely used, offering a large collection of annotated images spanning various object categories in natural scenes, which provides a benchmark for evaluating state-of-the-art detection algorithms. Most of the developed methods in the literature are initially tested on such benchmark datasets and some of them have also created and introduced new synthetic photorealistic datasets [63, 25, 79, 20] in order to serve the research needs of the respective survey.

3.2.1.4 Evaluation methodologies

The accuracy and precision of the 6D object pose estimation are evaluated through the comparison between the predicted and the known ground truth pose of the depicted object. There is a well-established strategy in evaluating the 6D object pose estimation in the literature [69] that involves several metrics. The most widely used metrics are: (i) Average Distance for distinguishable (ADD or ADD(-S)), symmetric distinguishable (ADD-S), and indistinguishable (ADI) objects, which calculates the average distance between the vertices in the 3D space. (ii) Visible Surface Discrepancy (VSD), which defines the object location using distance maps between the input and rendered image, while being invariant to symmetries and encountering only the visible object parts. (iii) Maximum Symmetry-Aware Surface Distance (MSSD) that computes the maximum distance and is independent of the object geometry. (iv) Maximum Symmetry-Aware Projection Distance (MSPD) that is similar to MSSD but for the 2D space, leaving out the Z-axis misalignment.

For object detection, evaluation methodologies focus primarily on intersection over union (IoU), which measures the overlap between the predicted and ground truth bounding boxes, and precision-recall curves, which assess performance over varying confidence thresholds. Metrics such as average precision (AP) and mean average precision (mAP) are standard for quantifying the overall detection performance.

These metrics reflect a different perspective of object pose estimation and are usually used as a combination to extract the final estimation accuracy [69]. The most common metrics for object detection and localization accuracy evaluation is the average of the average recalls for the different aforementioned metrics.

3.2.2 The approach in SOPRANO

The OBP component is designed to address the varied requirements of real-time object detection and 6D pose estimation in diverse industrial settings. The use cases target robot collaboration tasks in vehicle door assembly (CRF), and smart agriculture (E-TERRY). OBP integrates different sensor modalities from on-board and off-board stationary cameras for covering several tasks. More specifically, OBP employs: a) RGB images for 2D object detection and 6D pose estimation and b) depth information from RGB-D sensors for estimation of 3D positions. Different instances of the OBP can be deployed on workstations or ARM devices (i.e. NVIDIA Jetson) to provide tailored solutions for the SOPRANO use cases. Specifically:

- *CRF Use case*: Tools, screw, and car door detection in vehicle door assembly (Fig. 12 (a-b)). Synthetic data generation is exploited to automate training for the pose estimation task, ensuring the system can adapt to the dynamic nature of the assembly line.
- *E-Terry Use case*: Focuses on recognizing specific object classes (e.g. boxes) and humans in agricultural environments (Fig. 12 (c)) to address challenges such as cluttered scenes, occlusions, and non-standard imaging conditions (e.g., vegetation and uneven lighting), overall ensuring safe operation amidst moving machinery and natural elements.

[9] https://bop.felk.cvut.cz/datasets/
[10] https://cocodataset.org/

To effectively address the above, the OBP component consists of two independent subcomponents, the first one handling 6D pose estimation (OBP-6D) and the second handling 2D object detection and 3D position estimation (OBP-3D).

In the following, we provide a detailed description of each of the identified approaches, focusing on their methodologies, technical implementations, and how they contribute to solving the complex automation challenges faced in the respective industrial use cases.

3.2.2.1 Cameras & hardware

Different types of cameras for image acquisition are considered depending on the specific task requirements, operation in indoor or outdoor environments, and whether the camera is installed on-board or off-board the robotic platform. In outdoor environments, low-cost passive stereo cameras are preferred as active sensors often exhibit degraded performance due to interference with sunlight. Additionally, a wide field of view is advantageous for covering a large portion of the surroundings with a single sensor. In the E-TERRY use case, Stereolabs ZED X [201] cameras are chosen as they are already available and integrated with the E-TERRY robot.

In indoor environments, both active and passive stereo cameras can be utilized, with the choice primarily depending on the required depth range and the specific task they are intended to support. For the CRF use case, both types of cameras are used. RealSense cameras based on active stereo technology are deployed onboard the two robots of the CRF use case. The RealSense D455 [75], mounted on the arm of the mobile robotic platform, supports 6D pose estimation and grasping of tools and components. The camera is physically connected to a Jetson Xavier AGX board, which features 32GB memory and a 512-core GPU. In addition, the RealSense D405, chosen for its higher accuracy at close ranges (less than 0.5 meters), was utilized for the UR10 robot to support the task of screw detection and unscrewing operation. Furthermore, for monitoring the workstation areas (humans and car doors) a pair of StereoLabs ZED 2 stereo sensors is used. Each sensor model features two 4M pixel RGB sensors of wide-angle view and is also capable of capturing synchronised stereo RGB images sequences of 1080p at a frame rate of 30 fps and depth images of the same resolution, up to 100 fps for a depth range of 0.3 to 15 meters. Each sensor is mounted on a metallic rod at a height of 1.85 meters and is oriented to point towards the working area of the production line and the conveyor belt. These cameras are physically connected to the same workstation/desktop PC via a USB3.0 interface and acquire time-synchronized visual data from both sides of the workstation during assembly activities. Each PC is equipped with an Intel® CoreT M i9 Processor with 16 cores, 32 GB of RAM memory, and NVidia® RTX A4000 GPU with 16 GB of RAM. A summary of camera types, placement, and applications in the different use cases is provided in Table 1.

Camera Type	Sensor Type	Placement	Use Case	Task
Intel RealSense D455	Active	On-board	CRF	6D pose estimation of tools and components
Intel RealSense D405	Active	On-board	CRF	Screw detection
Stereolabs ZED 2	Passive	Off-board	CRF	Car door and human detection
Stereolabs ZED X	Passive	On-board	E-TERRY	Human and object detection

Table 1: Summary of camera types, features, and applications in different use cases.

3.2.2.2 Objects, 3D models and SOPRANO datasets

To support the development and evaluation of the OBP component, dedicated datasets were constructed across the SOPRANO use cases. These datasets include ground truth annotations such as 2D bounding boxes and 6D object poses, enabling both model training and quantitative evaluation.

CRF use case datasets. For the CRF use case, two complementary datasets were developed to address the different perception tasks:

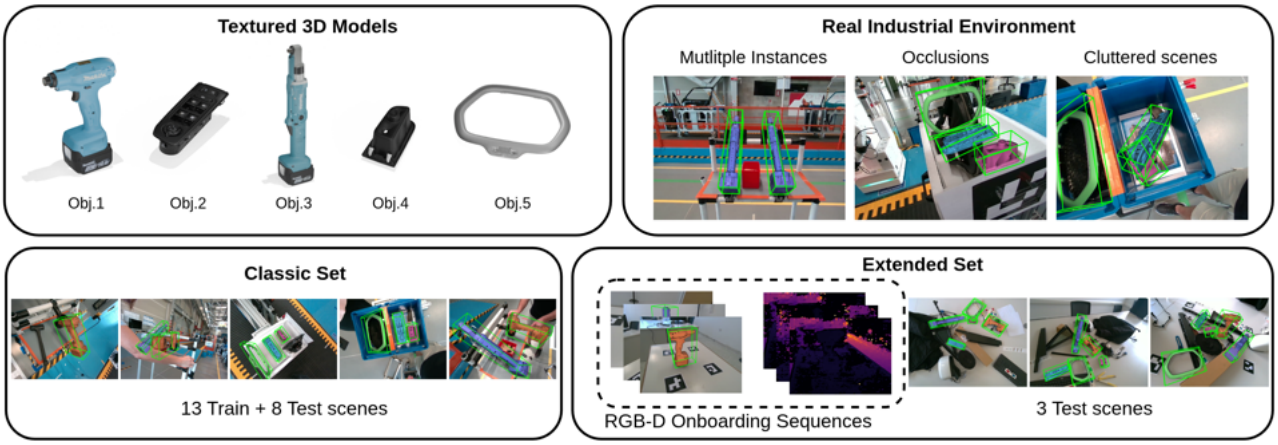


Figure 13: Overview of the IndustryShapes dataset for the CRF use case, featuring industrial tools and components with challenging properties (e.g., reflective surfaces, weak texture, symmetries), organized into a Classic set (instance-level) and an Extended set (novel-object evaluation with RGB-D onboarding sequences).

- *IndustryShapes dataset (6D pose estimation)*: IndustryShapes is an RGB-D dataset specifically designed for 6D object pose estimation of industrial tools and components involved in grasping tasks. It includes five representative objects (e.g., electric screwdrivers and car components) exhibiting challenging characteristics such as weak or absent texture, reflective surfaces, symmetries, and thin structures.
  The dataset is organized into two subsets:
  - *Classic set*: Captured primarily in the CRF industrial environment, it supports instance-level pose estimation. It includes over 4.6k RGB-D images and approximately 6k annotated poses, with training and test splits defined across different scenes to ensure robust evaluation.
  - *Extended set*: Introduced to support novel-object pose estimation methods. It includes RGB-D static onboarding sequences (two per object) and challenging multi-object test scenes under unconstrained conditions. This enables evaluation of both model-based and model-free approaches.

  The dataset follows the BOP [69] benchmark format and includes high-quality 3D CAD models for all objects, facilitating standardized evaluation using metrics such as ADD, VSD, MSSD, MSPD, and AR. It combines real and synthetic RGB-D data and reflects realistic industrial conditions encountered in robotic assembly environments. A more detailed analysis of the IndustryShapes dataset can be found in [180] which is accepted for presention at the 2026 IEEE International Conference on Robotics & Automation (ICRA 2026).
- *Screw detection dataset (2D detection)*: A dedicated dataset of screws was constructed to support the unscrewing task. It consists of RGB-D images captured in the CRF environment under realistic operating conditions, including variations in lighting, viewpoint, and object placement. All screw instances were manually annotated with bounding boxes and unique instance IDs.
  This dataset is specifically designed to evaluate 2D detection performance under the requirements of the robotic unscrewing process, where accurate localization is critical for tool alignment.

E-TERRY use case dataset. In the E-TERRY use case, a dataset of 12 videos was captured across diverse agricultural scenarios to represent dynamic and unstructured outdoor environments. These scenarios include harvesting activities, unloading and transportation of boxes, human presence and interaction, and autonomous robot operation within crop rows. The captured data reflect realistic field conditions, including cluttered backgrounds, vegetation-induced occlusions, varying illumination, and continuous motion of both humans and E-TERRY robot.

The dataset captures a wide range of situations, such as the movement and placement of boxes, human interaction with objects, and robot navigation in narrow agricultural field rows. These characteristics make the



Figure 14: Screw detection dataset collected in CRF, consisting of RGB-D images with manually annotated screw instances under realistic operating conditions.

dataset particularly suitable for evaluating object and human detection methods under challenging real-world conditions.

Building upon the initial data collection, individual frames extracted from the recorded videos were manually annotated to support supervised training of deep learning models. Specifically, annotations were created for object classes relevant to the use case, including *boxes* and *humans*. Although human monitoring is primarily addressed by the HMO component (see Sec. 5), human detection is performed within OBP as part of a unified perception pipeline. This avoids the need for separate object- and human-specific detectors, ensures consistent spatial representations across components, and enables direct integration of human detections into downstream modules such as safety monitoring and task allocation. The annotations consist of 2D bounding boxes and were formatted according to standard object detection annotation schemes (e.g., YOLO/COCO formats), enabling compatibility with widely used detection frameworks such as YOLO-based models.

This annotated dataset is used to train and evaluate the OBP-3D subcomponent for robust object and human detection in agricultural environments. The combination of diverse scenarios, realistic conditions, and curated annotations enhances the generalization capability of the trained models, supporting reliable perception in outdoor robotic operations.

Images illustrating the annotated boxes and detected humans from the E-TERRY use case are shown in Fig. 15.



Figure 15: Sample images from the E-TERRY environment, featuring human and carrying boxes, captured using Stereolabs ZED X.

3.2.2.3 2D object detection

The developed 2D object detection framework is designed to address a wide range of scenarios across the SOPRANO use cases, supporting both generic object categories and use case-specific targets under diverse environmental conditions.

For commonly encountered object classes, such as humans, the YOLOv5 [80] model is employed with pretrained weights derived from large-scale datasets such as COCO [110]. This enables strong baseline performance without additional training, particularly in scenarios where the target classes are well represented in existing datasets.

For use case-specific object classes, such as boxes in agricultural environments (E-TERRY) and screws in industrial assembly (CRF), the framework supports supervised fine-tuning using annotated datasets collected within the SOPRANO project. In the E-TERRY use case, frames extracted from recorded videos were manually annotated with bounding boxes for humans and boxes, enabling the adaptation of the detector to challenging outdoor conditions, including occlusions, clutter, and variable illumination. In the CRF use case, a dedicated screw detection dataset was constructed and annotated to support high-precision localization required for the unscrewing task.

In scenarios requiring instance-level detection, synthetic data generation is incorporated into the training pipeline. Specifically, photorealistic images generated using Blender [15] are used to augment training datasets, improving robustness and enabling the detection of small or visually ambiguous objects, such as screws, under varying viewpoints, lighting conditions, and occlusions.

Furthermore, for previously unseen objects with known 3D models, training-free approaches such as CNOS [147] are employed. These methods leverage CAD models and foundation models to perform object localization without requiring additional data collection or retraining, supporting rapid deployment in dynamic environments.

This combination of pretrained models, use case-specific fine-tuning, synthetic data augmentation, and training-free methods improves the robustness, scalability, and adaptability of the 2D detection framework, enabling reliable performance in diverse scenarios.

3.2.2.4 6D pose estimation

The developed 6D pose estimation framework combines state-of-the-art techniques and a flexible training pipeline to address the challenges of industrial object localization and manipulation. Both instance-based and generalizable approaches were investigated, enabling operation in scenarios involving known as well as previously unseen objects.

EPOS and FoundationPose were investigated as complementary approaches addressing different operational requirements. FoundationPose was selected for integration due to its scalability and ability to operate without object-specific retraining, making it suitable for dynamic environments where new objects may frequently be introduced. In contrast, EPOS remains the preferred solution in scenarios where object-specific training data are available and high task-level localization reliability is required.

Initially, representative instance-level methods from the BOP Challenge, such as EPOS [67], were evaluated and adapted to the industrial requirements of the use cases. These methods rely on object-specific training and provide strong performance when sufficient annotated data are available.

To address the limitations of instance-based approaches in dynamic environments, recent methods for 6D pose estimation of novel objects were also investigated. In particular, model-based and model-free techniques were assessed, focusing on their ability to generalize without requiring object-specific retraining. Based on this evaluation [222], **FoundationPose** was selected for integration into the OBP framework, due to its robustness, scalability, and ability to operate in both model-based and model-free configurations.

The framework leverages a combination of real and synthetic data, as described in the datasets presented above (e.g., IndustryShapes), ensuring both realism and sufficient variability for robust operation. Real-world annotated data ground the system in operational conditions, while synthetic data complement training by increasing viewpoint and appearance diversity, particularly for objects with limited observations.

To further ensure reliability and precision, a novel confidence score calculation method is incorporated [177], motivated by the need for a consistent and unbiased assessment of 6D pose estimation quality. The scoring pipeline relies on an efficient shape representation method capable of supporting fast spatial queries. This need led to the development of two novel methods for shape representation. The first approach [178] leverages Gaussian Process priors to generate a lightweight, explicit representation of the object. The second, a significantly faster and more robust method named SHARC [179], represents complex shapes through a collection of distance fields encoded via Spherical Harmonics (SH), which are anchored at optimally selected reference points inside the object's interior. Ultimately, utilizing either of these compact representations enables the fast and accurate 2D-3D correspondence queries necessary to calculate the confidence score for each estimated pose.

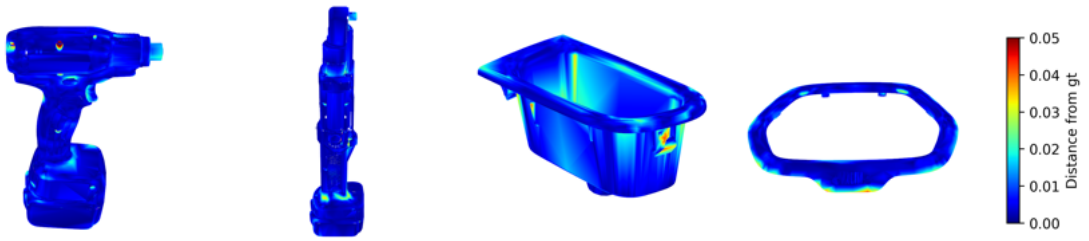


Figure 16: Qualitative results of our GP-based method [178] for the models of the IndustryShapes dataset. The heatmap quantifies the distance from the ground truth point-cloud to the reconstructed; superimposed into the ground truth point cloud.

3.2.2.5 ROS integration and deployment

The overall communication of the OBP module with other modules is mainly achieved using ROS2. The required installations were packaged into Docker images to ensure portability and consistency across systems.

Object Perception Module:

For the CRF use case, both OBP subcomponents are ported to ROS2 to take full advantage of the improved reliability, security and communication protocols of ROS2 and enable the component's smooth integration with the SOPRANO ecosystem. The subcomponents are implemented as ROS2 actions to facilitate robust and pre-emptive communication, ensuring smooth execution of tasks like object detection, pose estimation, and alignment refinement. For the ETERRY use case, OBP-3D is implemented independently of ROS2, using Kafka broker-based communication instead.

To address the diverse use case scenarios and extend the component's applicability to workflows outside of SOPRANO, variations of the subcomponents have been implemented that support different hardware and software setups. These variants are summarized in Table 2. Note that *ROS2* refers to Humble in all cases; currently, no other ROS2 distributions are supported. For the specific camera models, the reader may refer to Table 1.

Docker Container Deployment: To enhance portability and sustainability, the entire object perception system, along with its dependencies and ROS2 middleware, was containerized using Docker. Custom Dockerfiles were created for Ubuntu-based containers, incorporating Compute Unified Device Architecture (CUDA), ROS2 (i.e. ROS2 Humble), OpenCV, and other required libraries. This ensured consistent installation and configuration of all dependencies.

	Architecture		Jetpack		Camera Type		Middleware	
	x86	ARM	5.x	6.x	RealSense	ZED	ROS2	Kafka
OBP-6D	✓	✓	✓	✓	✓	✗	✓	✓
OBP-3D-CRF	✓	✗	✗	✗	✓	✗	✓	✓
OBP-3D-ETERRY	✗	✓	✗	✓	✗	✓	✗	✓

Table 2: OBP support matrix for hardware and software setups.

This modular and containerized approach using ROS2 and Docker ensures the robustness, scalability, and ease of integration of the object perception component, facilitating its deployment across various hardware platforms and robotic systems.

Adaptation for resource-constrained devices: The variation of EPOS used in the OBP-6D component has been adapted to support DNN inference with the ONNX [11] and TensorRT [12] engines, enabling its efficient execution on resource-constrained hardware. Furthermore, specialized Docker containers were tailored for the NVIDIA Jetson AGX Xavier and Orin platforms deployed on the use cases, optimizing the deployment for onboard processing.

Integration with AIO: Distributed variants of OBP-6D and OBP-3D are under development for integration with the AI Optimizer component and deployment on the CRF and E-TERRY use cases, respectively. The OBP-6D pipeline is comprised of two stages: (1) DNN inference using DeepLabv3+ [27] and (2) post-processing with Graph-Cut RANSAC [11] to extract the 6D pose. The respective distributed pipeline further splits Stage 1 into two stages to perform distributed inference using the partitioning of the DeepLab model produced by the AIO Model Splitter. The 3 resulting stages are wrapped in ROS2 services, orchestrated by a ROS2 action server in order to preserve the communication convention of the monolithic pipeline.

In a similar manner to OBP-6D, DNN inference and postprocessing are separate stages for the distributed version of OBP-3D. More specifically, stage 1 & 2 comprise the distributed inference segment, executing one submodel each. The submodels are the result of the Model Splitter applied to YOLOv5. In addition, stage 1 preprocesses the input RGB image received from the appropriate ZED camera ROS topic by applying letterboxing before running inference. The raw tensor output is published to a topic and consumed by stage 3 which implements YOLOv5 postprocessing for filtering and upscaling bounding boxes. Stage 4 estimates the 3D pose of the detected objects of interest (humans, boxes) using point cloud and depth data and publishes the 3D poses along with the refined detections from stage 3. All stages are implemented as ROS2 nodes for continuous detection using a publisher-subscriber pattern.

3.3 Results

The following sections present results of the OBP considering the two use cases: CRF (sec. 3.3.1) and E-TERRY (sec. 3.3.2) and exploitation of the OBP framework for specific tasks, including object detection, human detection, and 6D pose estimation.

3.3.1 CRF use case

3.3.1.1 Car door and human detection

In the CRF use case, car door and human detection are essential prerequisites to enable robotic tasks. Car door detection provides a useful prior for the HMO component (see sec. 5) by a) selecting a single active worker operating in the observed workstation area in case multiple workers/humans are visible by the two cameras, b)

[11] https://onnxruntime.ai/
[12] https://developer.nvidia.com/tensorrt

estimating the human pose more reliably and in cases where the worker is partially occluded by the car door and c) as an additional feature set to the classifier of the human actions monitoring approach. In addition, the car door can be integrated to the updated environment map as a dynamic moving object to aid robot navigation as well as robot tasks involving grasping of objects located on the moving cart and locating components, such as screws, within the door frame (stage 1 in screw detection depicted in Fig. 17). The focus was on accurately segmenting car door regions while handling overlaps with human areas. For the car door 2D segmentation, the Mask R-CNN deep learning model [61] was utilized, implemented within the Detectron2 framework [235]. Since the car door represents a specific object class not included in pretrained models, it was necessary to annotate the car door area in sample images extracted from assembly line videos. These annotated images (in total 257), along with their corresponding 2D segmentation masks, were then used to fine-tune the Mask R-CNN R50 FPN model for the newly introduced car door class while maintaining segmentation capabilities for humans (class person). Once the 2D segmentation mask of the car door is detected, depth information and camera extrinsic parameters are exploited to estimate the car door's pose. By transforming the mask's extremities into the camera and global coordinate systems, the door's orientation and centroid are deduced with high accuracy.

We used 257 frames from [157] for annotation for the *car door* class and initially trained the model with the car door class and without the 2D segmentation masks for humans. Overall, the results with the first set exhibited sufficient accuracy, with 85.07% average precision for the bounding boxes. We have further fine-tuned the model using annotation data containing the 2D mask of the car door and those of the humans. The results did not exhibit significant differences in the way the model dealt with the prediction of the two classes, as both the car doors and humans were correctly detected with a confidence score above 95%, for both classes. The average precision with the COCOEvaluator is 88.20% on bounding boxes and 79.63% on segmentation masks. Related results are reported in [155].

3.3.1.2 Screw detection



Figure 17: Overview of the multi-stage screw detection process with corresponding input images. (Stage 1) Door detection is performed using an off-board Stereolabs ZED 2 camera for long-range detection. (Stages 2-3) Screw detection utilizes an onboard Intel RealSense D405 camera mounted on a UR robot, covering close-range and very close-range detection, respectively.

For the CRF use case, screw detection is an essential task aimed at enabling precise localization for robotic manipulation with the UR10 robot. The screw detection approach constitutes the second and third stages of the overall door-screw detection pipeline, as depicted in Fig. 17.

The screw detection pipeline comprises three main stages: synthetic data generation, 2D detection, and 3D refinement (Fig. 18).

To train the detection model, **synthetic datasets are generated** using Blender [15], leveraging detailed 3D screw models and HDRI (High Dynamic Range Imaging) maps to simulate realistic lighting conditions. Varia-

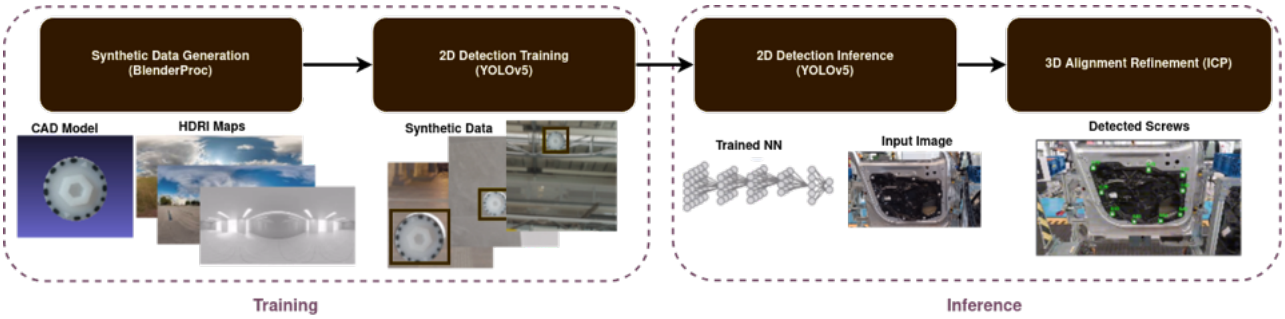


Figure 18: Overview of the screw detection pipeline for the CRF use case, illustrating synthetic data generation, 2D detection and 3D alignment refinement.

tions in orientation, occlusions, and illumination are introduced to ensure robustness under real-world industrial conditions.

The **2D detection stage** is performed using a YOLOv5-based model [80], which is trained on the synthetic dataset and provides real-time detection performance. The model is capable of detecting screws under challenging conditions, including clutter, reflections, and partial occlusions.

Following detection, **a refinement step** is applied using geometric priors. Given the known screw layout on the car door, detected screw instances are first associated with their expected positions, enabling an initial estimation of the door pose. This estimate is then iteratively refined by alternating between correspondence assignment and pose re-estimation, improving alignment consistency between observed detections and the underlying geometric model. Initialization is achieved either through image-based door detection or CAD-based matching using CNOS [147], enabling robustness in dynamic configurations.



(a) (b)

Figure 19: (a) Known screw geometry and positions used for refinement. (b) Example of CNOS-based segmentation for door localization.

The evaluation of the screw detection component was conducted using a dedicated dataset collected in the CRF environment, consisting of RGB-D images with manually annotated screw instances. Performance was assessed using standard detection metrics, including mAP@0.5 (mean Average Precision at an Intersection-over-Union (IoU) threshold of 0.5), mAP@0.5:0.95 (mean Average Precision averaged over IoU thresholds from 0.5 to 0.95 in steps of 0.05), precision, and recall, as well as a task-specific metric (mAP@CRF) reflecting the localization accuracy required for the unscrewing operation.

The evaluation follows a two-stage setup corresponding to the operational pipeline. In **Stage 2**, multiple screws are simultaneously visible and detection is evaluated at a moderate IoU threshold (65%), allowing coarse localization prior to refinement. In **Stage 3**, only a single screw remains visible before the unscrewing operation, requiring high-precision localization; therefore, a stricter IoU threshold (90%) is applied.

As shown in Table 3, the detection model achieves consistently high performance across both stages. Stage 2 demonstrates strong performance under multi-object conditions, while Stage 3 achieves higher precision under stricter localization requirements. These results confirm that the detection framework reliably meets the accuracy requirements for robotic unscrewing tasks.

Table 3: Screw detection performance on the CRF dataset across operational stages. The mAP@CRF metric reflects task-specific localization accuracy at the required IoU threshold.

Stage	mAP@0.5	mAP@0.5:0.95	Precision	Recall	mAP@CRF
Stage 2 (IoU 65%)	99.4	61.3	99.7	97.1	96.6
Stage 3 (IoU 90%)	99.7	89.1	99.7	100.0	92.4

Overall, the screw detection pipeline combines synthetic data generation, deep learning-based detection, and geometric refinement to deliver a reliable and scalable solution for industrial applications.

3.3.1.3 6D pose estimation of tools and components

We compared the performance of two state-of-the-art 6D pose estimation frameworks, namely EPOS [67] and FoundationPose [229]. EPOS follows a dense correspondence-based approach, predicting 2D–3D correspondences between image pixels and object surface fragments, which are subsequently used for pose recovery through a PnP-based optimization scheme. In contrast, FoundationPose is a recent model-based method that supports novel-object pose estimation, enabling pose inference directly from CAD models without requiring object-specific retraining. This property makes it particularly suitable for industrial environments where new objects are frequently introduced.

EPOS was trained on object-specific data, combining real and laboratory-captured samples of the Makita DFT object (Obj 1), while FoundationPose was evaluated in a model-based setting without additional training, relying solely on the CAD model at inference time.

The evaluation was conducted on unseen real images using the BOP evaluation protocol, which includes complementary metrics such as ADD, MSPD, MSSD, VSD, and Average Recall (AR). These metrics evaluate different aspects of pose estimation quality, including geometric alignment accuracy, robustness to symmetries, and consistency of visible surface projections. As shown in Table 4, both EPOS and FoundationPose achieve competitive overall performance, with EPOS providing slightly stronger localization accuracy, while FoundationPose demonstrates robust generalization capabilities despite not requiring object-specific training.

Method	ADD	MSPD	MSSD	VSD	AR
EPOS	0.64	0.54	0.56	0.41	0.51
FoundationPose	0.59	0.55	0.56	0.42	0.51

Table 4: Quantitative comparison of EPOS and FoundationPose using BOP evaluation metrics.

A broader evaluation on the IndustryShapes dataset, using BOP metrics (i.e. ADD, MSPD, MSSD, VSD and AR), is presented in Table 4. For this evaluation, EPOS was trained on all objects of the IndustryShapes dataset. The results show comparable overall performance between the two methods, while their behavior differs depending on object characteristics and training data availability. A more detailed per-object analysis is provided in Table 5. The five evaluated objects correspond to the industrial tools and components of the IndustryShapes dataset shown in Fig. 13, selected to capture challenging properties such as symmetries, reflective surfaces, weak texture, and thin structures. The results highlight that EPOS performs better on well-represented objects, whereas FoundationPose is more robust on objects with limited training data. The particularly low AR observed for Object 5 is mainly attributed to its challenging geometry and appearance characteristics, including reflective surfaces and limited distinctive visual features, which negatively affect stable correspondence estimation. In addition, Object 5 is not part of the current robotic grasping validation scenario, and improving performance on such challenging objects is part of ongoing dataset extension and future work.

For task-driven evaluation in the CRF grasping scenario, localization success rates across different ADD thresholds are reported in Table 6. These results confirm that EPOS consistently satisfies the required KPI

Method	Obj 1	Obj 2	Obj 3	Obj 4	Obj 5
EPOS	0.66	0.26	0.59	0.25	0.00
FoundationPose	0.68	0.34	0.52	0.40	0.06

Table 5: Average Recall (AR) per object for EPOS and FoundationPose.

threshold, while FoundationPose provides competitive performance at lower thresholds but does not reach the same reliability at the target tolerance.

Indicative qualitative results of pose estimation for both frameworks are shown in Fig. 20.

Threshold (mm)	Object 1		Object 3	
	EPOS	FoundationPose	EPOS	FoundationPose
2	0.0	0.4	4.7	0.3
4	2.0	11.8	16.7	6.65
6	25.3	40.7	35.4	24.8
8	45.3	60.0	50.6	41.6
10	62.0	72.3	63.0	56.5
12	75.3	79.4	75.3	65.0
14	80.0	83.0	81.7	70.1
16	84.7	86.1	84.9	74.1
18	87.3	87.3	90.0	76.4
20	88.7	88.1	90.8	78.3
22	90.7	88.5	92.0	79.0
24	91.3	88.9	93.6	80.3
26	92.7	88.9	96.0	80.8
28	93.3	88.9	96.4	81.3
30	94.7	88.9	96.9	81.9

Table 6: Localization success rates (based on ADD) across thresholds for EPOS and FoundationPose on Objects 1 and 3.

Finally, building upon the described confidence score calculation framework, experimental validation was conducted to assess its effectiveness. The validation focused on evaluating the method's performance across various scenarios, measuring its correlation with state-of-the-art metrics such as ADD. This comparison highlights the reliability of the confidence score in reflecting the true quality of 6D pose estimates. For instance, the consistency of the score across different objects and environments demonstrates its robustness and practical applicability. An example comparison with ADD-based ground truth evaluation is illustrated in Fig. 21.

3.3.2 E-TERRY use case

3.3.2.1 Human and object detection

In the E-TERRY use case, the focus was on detecting both humans and agricultural objects (e.g., boxes) in dynamic field environments. These scenarios are characterized by cluttered backgrounds, strong illumination variations, and frequent occlusions due to vegetation and ongoing activities.

A YOLOv5 model pre-trained on COCO [110] was initially employed without additional retraining. While the model demonstrated strong generalization capabilities, particularly for human detection, it exhibited systematic misclassifications caused by domain shift. As shown in the preliminary results (Fig. 22(a)), crouching humans were occasionally misclassified as visually similar COCO categories (e.g., animals such as sheep), highlighting the limitations of directly applying pretrained models in agricultural environments.

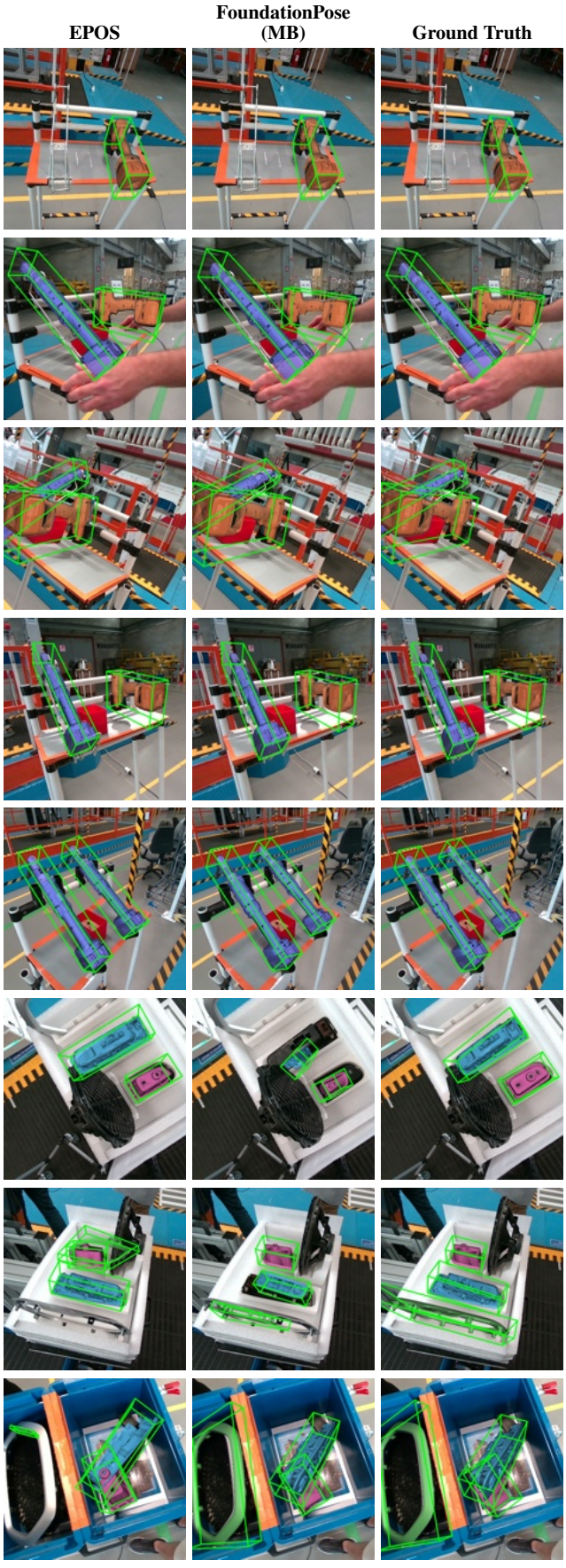


Figure 20: Comparison of the methods across selected scenes of the IndustryShapes test set.

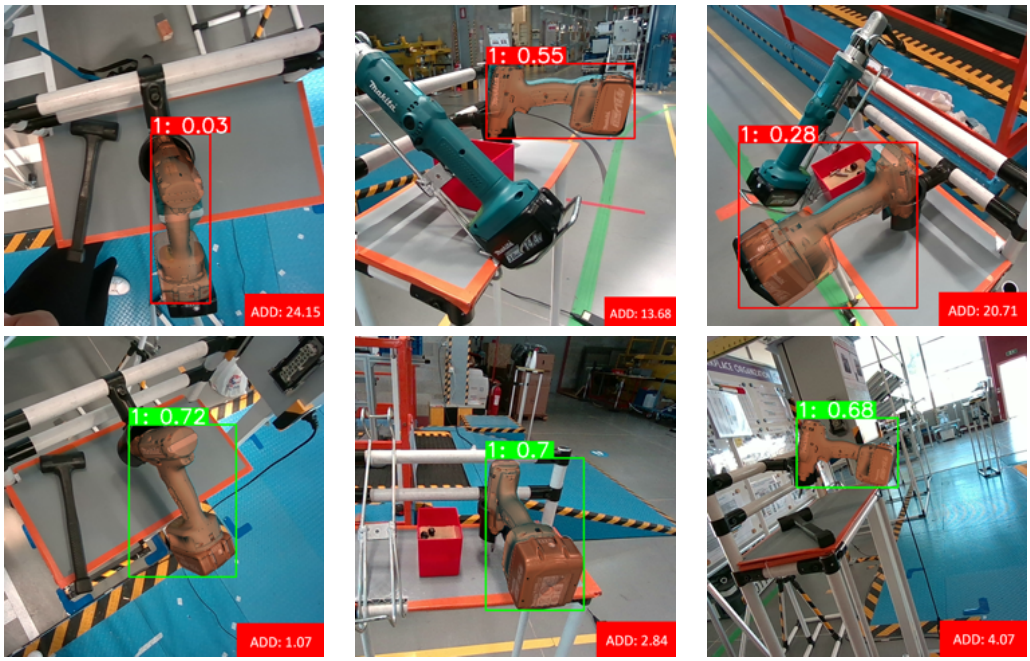


Figure 21: Accepted (green) and rejected (red) pose estimates based on their confidence metrics. The corresponding ADD metric can be seen in the lower right corner of each image.

To address these limitations, a dedicated annotation and retraining process was performed. Incorrect labels were corrected (e.g., reassignment of misclassified humans), and additional annotations were introduced for both human and box instances. The model was subsequently fine-tuned on the updated dataset, enabling reliable detection of both classes. The fine-tuning process was performed on approximately 1200 annotated frames over 100 epochs using standard YOLOv5 training settings, including data augmentation techniques such as scaling and flipping. Despite the relatively modest dataset size, this process proved sufficient to significantly improve domain adaptation performance.

Table 7 summarizes the performance of the pretrained model for human detection, while Table 8 reports the results after fine-tuning for both humans and boxes. The evaluation was conducted using standard detection metrics, including mean Average Precision (mAP), precision, and recall.

Class	**mAP@0.5**	**Precision**	**Recall**
Human	0.837	0.891	0.705

Table 7: Detection performance of the pretrained YOLOv5 model (COCO) on the E-TERRY dataset for human detection.

Class	**mAP@0.5**	**Precision**	**Recall**
Human	0.957	0.960	0.912
Box	0.670	0.623	0.811

Table 8: Detection performance after fine-tuning on the E-TERRY dataset for both human and box detection.

The results indicate that fine-tuning significantly improves detection performance, reducing misclassifications and enabling robust detection of both humans and agricultural objects. Qualitative results are illustrated in Fig. 22(b), demonstrating reliable detection under diverse and challenging field conditions.

In addition to the above evaluation, task-level performance was assessed in the integrated system. The object detection and localization component achieved an average success rate of 79%, which increased to over 96%



(a) (b)

Figure 22: Example results from the E-TERRY use case. (a) Initial misclassification of humans due to agricultural elements resembling COCO categories. (b) Improved detection results after manual label correction and retraining.

when objects (boxes) were fully visible to the camera. This highlights the sensitivity of the system to occlusions and viewing angles, while confirming its reliability under standard operating conditions, indicating that the system performs reliably when boxes are sufficiently visible, while occlusions and unfavorable viewpoints remain the main limitations. For human detection and localization, a success rate of 94.6% was achieved, corresponding to excellent performance. The perception pipeline demonstrated robust behavior across varying lighting conditions and human poses, with occasional failures primarily attributed to temporary occlusions or extreme viewpoints. Overall, these results confirm that the OBP component provides accurate and timely information to support safety monitoring and navigation in agricultural environments.

3.4 Discussion and future work

The Object Perception (OBP) component has reached a maturity level consistent with TRL 6 for the evaluated CRF and E-TERRY scenarios, demonstrating strong performance in both 2D object detection and 6D pose estimation under realistic operational conditions. Ongoing work is targeting TRL 7 through full system integration and validation in operational environments. In particular, the integration of EPOS for instance-level pose estimation and FoundationPose for novel-object scenarios, together with YOLO-based detection pipelines, has enabled robust perception capabilities in both industrial (CRF) and agricultural (E-TERRY) environments.

In the CRF use case, the component achieved reliable 6D localization performance, satisfying the defined KPI requirements for robotic grasping tasks. In parallel, the screw detection pipeline demonstrated high accuracy under real operating conditions, leveraging synthetic data generation and geometric refinement. In the E-TERRY use case, the detection framework showed strong generalization capabilities, while also highlighting the importance of domain adaptation through targeted retraining for agricultural environments.

Key priorities for future work include improving robustness under challenging conditions such as occlusions, reflective surfaces, and dynamic scenes, as well as extending support to a wider range of object categories and scales. Particular emphasis will be placed on enhancing generalization capabilities through the integration of novel-object pose estimation approaches and extension of datasets, such as IndustryShapes. Additionally, further improvements in detection performance will be pursued through domain-specific data annotation and model fine-tuning, especially for scenarios with limited or highly variable training data.

Through these developments, the OBP component is expected to achieve increased adaptability, scalability, and reliability, supporting complex perception-driven robotic tasks in both industrial and agricultural environments.

4 Non-Visual Localization [NVL]

4.1 Technology overview and module specifications

As already pointed out in Sec. 2.1, autonomous robot navigation requires either an existing map of the operating environment or the ability to construct and update such a map online, while continuously estimating the robot's position and orientation within it This localization can be achieved either via visual data collected with camera sensors (cf. Sec. 2), or using non-visual sensory modalities and positioning technologies, such as LiDAR, mm-wave radar, and Bluetooth-based systems (among others).

Non-visual localization (NVL) plays a critical role by offering an alternative to visual localization in environments where visual data is unreliable or unavailable, such as in low-light conditions or areas with sparse texture. It can operate independently to provide robust positional data or be integrated with visual localization, combining the strengths of both approaches to enhance accuracy, robustness, and overall system performance. For example, visual localization can offer rich contextual information, while non-visual sensors provide precise depth and motion tracking, creating a complementary system that performs well in a variety of conditions [146]. Despite lacking detailed environmental context and semantic understanding, NVL can offer reliability in challenging environments where visual systems may be unsuitable, e.g. during rapid motion, in darkness, or when visual landmarks are absent. It also tends to be less affected by environmental factors like glare or shadows.

4.2 Methodology and implementation

4.2.1 Existing approaches (SOTA)

Simultaneous localization and mapping (SLAM) with 2D LiDAR sensors can offer highly efficient and reliable solutions for a variety of robotics applications (e.g. [163]). In recent years, modern, extensible frameworks are developed accommodating real-time performance, robust localization, and mapping capabilities [243]. One such example comprises graph-based approaches, where the robot's pose and sensor measurements are represented as nodes and edges of a graph. Pose graph optimization and loop closure detection are also recent optimization techniques, that are widely used and can ensure consistent mapping over extended operations [119]. Real-time mapping scenarios are now sophisticated enough to enable robots to adapt seamlessly to dynamic and evolving environments. [116] demonstrates the efficiency, accuracy, and effectiveness of the pose graph SLAM paradigm in the management of the mapping problem during long-term robot operation, utilizing inter- and intra-session loop closure detection.

2D LiDAR-based SLAM allows incremental mapping, serialization, and map merging, features that are crucial for applications requiring long-term operation and multi-session mapping; service robots navigating complex indoor environments are a prime example of applications that are benefited from advancements of the field. In these latter cases, maintaining high accuracy while minimizing computational overhead is highly important. 2D LiDAR based SLAM can provide such capabilities and further limit the required processing power and resources. Integration of efficient algorithms for scan matching, often leveraging techniques like Iterative Closest Point (e.g. [31]) or Normal Distributions Transform (e.g. [14]), allows for rapid alignment of LiDAR scans, which is essential for real-time performance.

Modern SLAM systems can additionally provide great flexibility in terms of handling diverse operational modes. They can adapt to varying requirements depending on application scope and efficiently handle real-time mapping, navigation, and detailed map refinement. Moreover, lifelong mapping capabilities allow for continuous improvement and updates to the map as the robot revisits familiar areas, ensuring that the representation of the environment remains consistent and up-to-date.

Recently, techniques that incorporate LiDAR intensity information have been introduced, that allow for improved scan matching and reliable mapping, while maintaining computational efficiency [224]. These techniques could especially assist seamless adaptation to large-scale or dynamic environments, aiding in accurate mapping with little computational overhead. It is worth noting that graph-based SLAM approaches can use pose-graph optimization to identify and correct drift over extended missions, effectively assisting in precise loop closure detection. Revisiting familiar areas can thus lead to incremental refinements even under conditions with limited computational resources.

When dealing with highly dynamic, evolving environments, active SLAM techniques can be used [163]. These techniques extend traditional mapping approaches through the integration of trajectory planning and localization. Both localization accuracy and mapping robustness can be enhanced while at the same time map coverage can be optimized leading to uncertainty minimization. Recently, [101] proposed a semantic SLAM approach based on LiDAR point clouds (SD_SLAM), dedicated to dynamic scenes, effectively differentiating between dynamic and semi-static landmarks. SD_SLAM can improve localization and mapping performance in dynamic scenes, while simultaneously constructing a static semantic map with multiple semantic classes for enhanced environment understanding. Generation of Pose Graph-based maps from Occupancy Grid Maps (OGM), when building information models (BIM) are available, has also been shown to improve localization and pose estimation in the latter case [223].

Targeting computational efficiency and scalability of LiDAR-based SLAM systems, [111] introduces a Fisher Information-based keyframe selection strategy. The key concept in this approach is to retain only essential keyframes, and thus minimize redundancy without compromising map quality. In turn, this allows for faster pose graph optimization, while the reduced dataset size leads to reduced constraints on computation resources.

In addition to SLAM-based approaches, probabilistic localization methods operating on pre-existing maps are widely used in robotics. A prominent example is Adaptive Monte Carlo Localization (AMCL), which employs particle filters to estimate the robot pose within a known occupancy grid map. In this framework, the robot state is represented by a set of weighted particles, each corresponding to a pose hypothesis. These particles are propagated using a motion model (e.g., odometry) and updated through a measurement model that evaluates the consistency between sensor observations (e.g., LiDAR scans) and the reference map. Resampling techniques are then applied to concentrate computational resources on the most likely hypotheses. AMCL has been shown to provide robust and efficient localization in indoor environments, even in the presence of sensor noise and partial map ambiguities [51, 216]. Its adaptive variant further improves efficiency by dynamically adjusting the number of particles based on the uncertainty of the pose estimate, making it particularly suitable for real-time robotic applications.

The performance of both SLAM and map-based localization methods is strongly influenced by the accuracy of the underlying motion estimation, commonly referred to as odometry. Odometry provides an estimate of the robot's incremental motion and is typically derived from wheel encoders, inertial measurement units (IMUs), or directly from exteroceptive sensors such as LiDAR. While encoder- and IMU-based approaches can offer high-frequency motion estimates, they are often affected by wheel slip, calibration errors, and drift over time. Alternatively, LiDAR-based odometry methods estimate motion by aligning consecutive scans, providing improved robustness in environments where geometric features are sufficiently informative [248, 76]. However, their accuracy remains dependent on environmental structure and sensor quality, making odometry a critical factor in the overall performance of localization and mapping systems.

The current state of 2D LiDAR SLAM reflects a mature and highly effective approach to robotics mapping, leveraging modularity, extensibility, and precision to meet the demands of contemporary applications. Ultrasonic distance sensors have evolved to achieve millimeter-level accuracy for indoor positioning applications, with recent implementations demonstrating localization accuracies of ±3 mm for frontal measurements and ±6 mm for lateral positioning in mobile robot applications [141].

Ultrasonic ranging accuracy is influenced by several factors including the divergence angle of the ultrasonic beam, the incident angle relative to target surfaces, and environmental conditions such as temperature and humidity, with angle compensation techniques improving accuracy from 12.24 mm to 3.4 mm within 1-meter

range at 60° angle positions [167]. Advanced ultrasonic indoor positioning systems implementing time-of-flight (ToF) and time-difference-of-arrival (TDoA) methodologies have achieved sub-centimeter accuracy with resolution in the millimeter range, particularly when deployed in controlled environments with optimized beacon layouts [129]. Modern trilateration-based ultrasonic localization systems report standard deviation accuracies of up to 15 mm, combining low-cost wireless sensor networks with sophisticated signal processing algorithms to maintain reliable positioning in complex indoor spaces [205].

Contemporary mobile robot localization systems employ multi-sensor fusion strategies that integrate ultrasonic sensors with complementary technologies such as LiDAR, cameras, inertial measurement units (IMUs), and ultra-wideband (UWB) systems, with fusion architectures classified into cooperative, complementary, and competitive approaches to overcome individual sensor limitations [221]. Recent implementations combining IMU, UWB, and ultrasonic measurements through factor graph optimization have demonstrated decimeter-level accuracy in complex environments, with experimental results showing 12.3 cm RMSE in dynamic motion conditions and 38% improvement over conventional Kalman filter approaches [247]. For three-dimensional positioning applications, ultrasonic local positioning systems (ULPS) deployed on multiple planes with extended Kalman filter (EKF) and adaptive Kalman filter (AKF) fusion methods achieve centimeter-to-decimeter range accuracy across large coverage volumes, effectively merging position estimates from different beacon configurations [124].

Integration of ultrasonic ranging with wheel odometry and attitude heading reference systems (AHRS) through Kalman filter-based fusion significantly outperforms single-sensor approaches, reducing noise and enhancing consistency while correcting odometry drift in differential-drive mobile platforms [218]. Extended Kalman Filter (EKF) implementations serve as the predominant framework for fusing heterogeneous sensor data in mobile robot pose estimation, utilizing discrete models with first-order approximation for nonlinear systems to enable complementary compensation for each sensor's limitations, with the resulting fused system performance exceeding individual sensor capabilities [4].

Advanced EKF-based sensor fusion methods combining visual, inertial, and magnetic sensors achieve orientation root mean square errors of 1°-1.5° and position RMSEs of 3.5-10 mm in experimental conditions, demonstrating the effectiveness of tightly-coupled approaches that process all measurements within a unified statistical filtering framework [108]. Multi-sensor fusion algorithms integrating ultrasonic or UWB positioning with LiDAR and IMU data through EKF methods reduce tracking errors by up to 25.7% compared to single-sensor approaches, successfully mitigating poor localization accuracy in non-line-of-sight (NLOS) environments and trajectory drift in feature-sparse conditions [253]. The fusion of ultrasonic sensors with laser-based range finders creates comprehensive eight-sensor arrays capable of precise three-dimensional coordinate determination and angular orientation estimation, enabling centimeter-level positioning accuracy essential for autonomous navigation in confined industrial spaces.

Mesh connections between Bluetooth devices with minimal power consumption (i.e., less than 10 mW) are now possible thanks to Bluetooth low energy (BLE) [89]. When BLE beacons receive Bluetooth signals, they can use the signal strength (RSSI) and beacon position to identify the person's location. "In-phase and Quadrature" (IQ) sampling has been added to the Bluetooth protocol in recent years, enabling angle of arrival and angle of departure computations via Bluetooth [204]. Bluetooth low energy is a low-cost wireless transmission technology that is optimized for short-range (i.e., theoretical range <50 m) communication [231].

The primary advantage of BLE is that its extremely low power consumption (usually 0.367 mW) enables battery-operated devices to operate for years without needing to be recharged [89]. Furthermore, many mobile devices have BLE hardware installed, enabling safe and effective communication [89], [150], [244]. Interference and erratic BLE signal detection by the beacons used to detect them are among the disadvantages. Utilizing BLE is restricted to the 2.4 GHz band, which is heavily used, and may conflict with Wi-Fi signals.

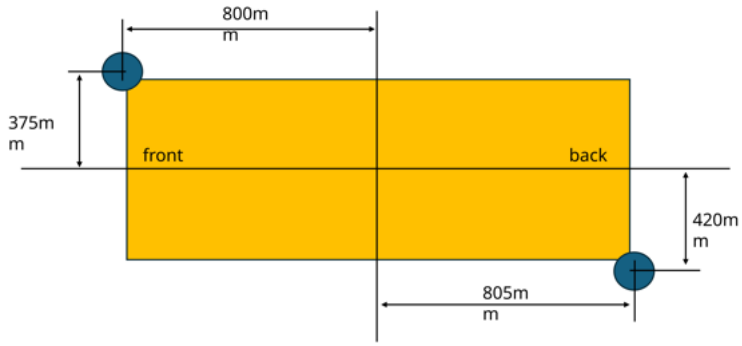


Figure 23: LiDAR mounting points on KUKA's robot.

4.2.2 The approach in SOPRANO

SOPRANO involves three distinct use cases operating in diverse environments, each posing different challenges: (a) KUKA is targeting building renovation operating in construction environments, (b) CRF involves vehicle door assembly, and (c) E-TERRY operates outdoors on agricultural tasks, being the only case where outdoor non-visual localization can be reliably employed; in this latter case though, a localization stack based on GPS and IMU sensors is already employed by E-TERRY. Thus, we will focus on the KUKA and CRF use cases in the following.

The KUKA Use Case

For the KUKA use case, the current NVL payload includes two 2D planar LiDAR sensors (Hokuyo UST/URG family) mounted as depicted in Figure 23, which use time-of-flight laser ranging to acquire distance measurements over a planar field of view, producing 2D range scans of the environment. The payload also includes a series of two ultrasound-mm radar sensors mounted on the robot's side. The NVL architecture is designed to follow a two-stage approach. First, LiDAR-based (see Sec. 4.2.2.1) and radar-based pose estimates (see Sec. 4.2.2.2) are computed independently. Subsequently, these estimates are fused using a Kalman filtering approach, taking into account their associated uncertainty (i.e. covariance), in order to derive a final, optimal NVL pose estimate. Ultrasonic sensors can complement LiDAR sensing, since they provide comparable distance-measurement accuracy while offering a different sensing footprint, which can improve obstacle-detection performance in the regions where they are installed.

The LiDAR-based stack supports two distinct modes of operation. In the first mode, a *pure SLAM* configuration is employed, in which no prior map is available and the environment is constructed online. In the second mode, a *pure localization* configuration is used, where an *a priori* occupancy grid map is provided and serves as a global reference frame for pose estimation.

The CRF Use Case

For the CRF case, two sub tasks were addressed: a tool presence detection system and the localization of a dolly. Although tool presence detection does not constitute localization in the strict sense, it falls within the broader scope of non-visual sensing and was therefore developed within the NVL stack. The sensor provides information on the tool presence, therefore the tool is expected to be found in the specific location of the tool stand before grasping is initiated. The approach that we followed evolves a low-cost, wireless IoT solution that automatically detects (using IR sensors) and reports tool presence at workstations in real-time, enabling centralized monitoring and data collection for industrial or laboratory environments (see Sec.4.2.2.5).

For the localization of the dolly, Bluetooth Low Energy (BLE) tags are exploited that utilize the phase difference of radio signals across multi-antenna arrays to calculate precise spatial coordinates (see Sec. 4.2.2.6). Two

approaches are implemented: the first is the Angle of Arrival (AoA) approach, where a mobile tag transmits a signal to a fixed receiver that uses its antenna array to determine the incoming signal's direction. The second is the Angle of Departure (AoD) approach which involves a fixed transmitter sending signals from multiple antennas, allowing a mobile device to calculate its own position based on the specific phase shifts it detects. These approaches based on BLE tags are currently being evaluated in a lab environment to assess their metric accuracy before deployment.

4.2.2.1 LiDAR-based SLAM (KUKA UC)

Taking into account the planar motion of the robot, the payload configuration and the indoor KUKA workspace, a graph-based 2D LiDAR SLAM approach (see 4.2.1) was selected as the main NVL solution. Unlike the feature-based VLM pipeline (see Sec. 2), which estimates full 6-DoF pose, this LiDAR-based method operates in $SE(2)$, estimating only (x, y, θ). This aligns with the KUKA use case, where the robot moves in largely planar indoor environments with wall-like structures that support reliable 2D scan matching and mapping. In contrast, the CRF environment has fewer clear structural boundaries and contains more thin and cluttered elements, making it less suitable for robust 2D LiDAR-based localization as the primary localization approach.

In this setting, the selected 2D LiDAR-based approach offers a number of practical advantages, including:

- scalability and efficiency,
- handling of drift and errors,
- robust loop closure detection,
- high-quality 2D mapping,
- customization, modularity, extensibility and ease of integration with ROS2.

At the core of this approach are four main elements: scan matching, which estimates relative motion between successive observations; a pose graph, in which robot poses are represented as nodes and sensor-based spatial constraints as edges; loop-closure detection, which identifies revisited areas and introduces additional constraints; and graph optimization, which reduces accumulated error and improves global map consistency.

To implement this pipeline, the `slam_toolbox`[13] package was selected as the main software backend (see e.g. [118]). It provides a mature ROS-based framework for real-time and offline 2D SLAM, supporting scan matching, loop closure, pose-graph optimization, and persistent map handling. Its modularity and ease of integration with ROS2 make it particularly suitable for the present use case, while also allowing targeted configuration and further development within SOPRANO.

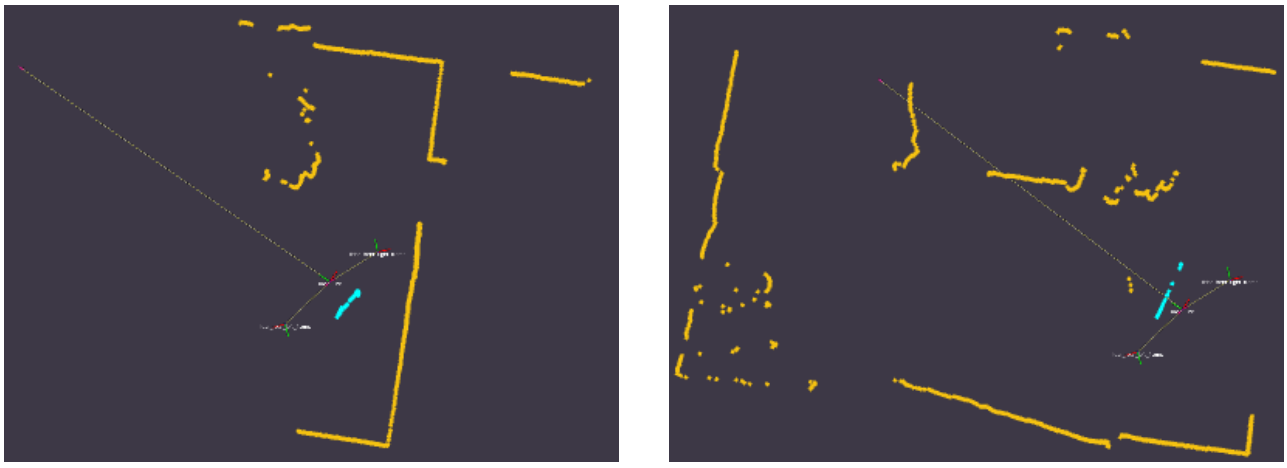
To support operation in environments where an a priori occupancy grid map is available and navigation goals are expressed in a global reference frame, the system employs a probabilistic localization module based on Adaptive Monte Carlo Localization (AMCL). AMCL estimates the robot's pose by maintaining a set of weighted particles representing hypotheses of the robot's state, which are updated using LiDAR observations and a motion model derived from odometry. This approach enables robust localization in the presence of sensor noise and environmental uncertainty, ensuring that navigation remains consistent with the predefined map.

Algorithmic approach

The 2D LiDAR-based SLAM process (see Sec. 4.2.2.1) aims to construct a map of an indoor environment while simultaneously estimating the robot's pose within it. The method adopts a graph-based SLAM approach, in which nodes represent robot poses at different times and edges encode spatial constraints derived from sensor observations. The overall workflow comprises LiDAR acquisition, scan preprocessing, scan matching, pose-graph update, loop-closure detection, graph optimization, and map generation.

LiDAR data are acquired in real time from the onboard sensors and provided to the NVL stack through ROS2 topics. Before further processing, a filtering stage is applied to suppress measurements that are not useful

[13] https://github.com/SteveMacenski/slam_toolbox



(a) (b)

Figure 24: Raw (cyan) and filtered (yellow) scans from the two LiDAR sensors overlaid while operating in localization mode; the raw scans contain a self-return artifact from the robot perimeter/frame, which is removed by filtering. (a) Front LiDAR scan and (b) rear LiDAR scan.

for environment reconstruction, in particular self-observations caused by the robot body or sensor mounting structures (see Figure 24). This preprocessing step improves the robustness of downstream modules, including LiDAR-based odometry, SLAM, and map-based localization.

Pose estimation is then obtained by scan matching, where incoming scans are aligned with previous observations or with the current map in order to estimate relative motion. These estimates are used to incrementally update the pose graph. When previously visited areas are detected, loop-closure constraints are added, and the graph is optimized to reduce accumulated drift. In the present implementation, graph optimization is performed with the Ceres Solver [3]. The optimized trajectory is finally used to generate or update a 2D occupancy-grid map representing free, occupied, and unknown space.

In addition to the SLAM workflow described above, the localization stack also supports operation in the presence of an *a priori* occupancy-grid map. In this case, map construction is no longer required; instead, the objective is to estimate the robot pose within an already defined reference frame, so that navigation targets expressed in map coordinates can be reached consistently. To this end, the system incorporates a probabilistic localization module based on AMCL.

AMCL estimates the robot pose by maintaining a set of weighted particles representing pose hypotheses in the known map. These hypotheses are propagated using the motion estimate provided by odometry and are updated by comparing incoming LiDAR observations against the reference occupancy grid. In this way, the robot can localize itself within the predefined environment while remaining consistent with the global map frame. This functionality is essential when the robot must operate relative to externally supplied maps and navigation goals defined in that map.

Simulation framework for testing LiDAR-based localization

The algorithmic approach outlined in Sec. 4.2.2.1, was implemented using the Gazebo[14] simulator. Gazebo is a widely used (e.g. [1] and [106]) open-source robotics simulator, providing a realistic environment for testing and developing algorithms including sensor simulation. It integrates seamlessly with ROS offering a feature-full virtual setting.

The motivation for designing and developing the simulator described in this section, is based on a twofold reasoning: on the one hand, real-time data acquisition from the on-board KUKA LIDAR sensor proved to

[14] https://gazebosim.org/home

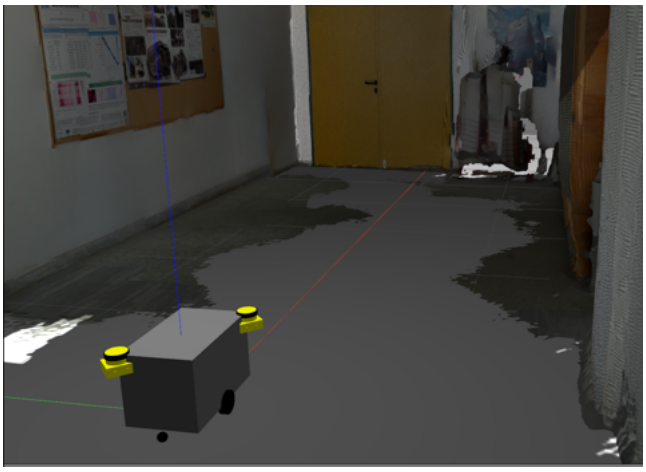


Figure 25: The 3D robotic simulator developed to perform LiDAR-based SLAM.

be a technically demanding task at the early development steps. Several technical issues had to be resolved beforehand, a fact that could delay the development of the NVL component. Adding to that, remote access to the sensor could lead to potential security issues. On the other hand, deploying the NVL on a simulator enables extensive testing and validation, is not limited by the current payload and computing resources, facilitates a trial-and-error workflow and finally can lead to the production of a ready-to-deploy software stack.

Thus, a software framework was designed to enable the operation, simulation, and development of a mobile robot tailored to industrial environments, resembling the KUKA use case in SOPRANO. The robot is equipped with a Hokuyo UST/URG 2D LiDAR scanner (matching the actual KUKA payload) for SLAM, and a Realsense D435 camera for depth and visual sensing. Although the NVL component itself does not rely on visual sensing, the camera is included in the simulation payload to support broader development, testing, and validation activities if required. The system is built upon the ROS2 Humble distribution, operating within an Ubuntu 22.04 environment, and provides comprehensive capabilities for navigation, mapping, and teleoperation, as well as simulation in a virtual industrial environment.

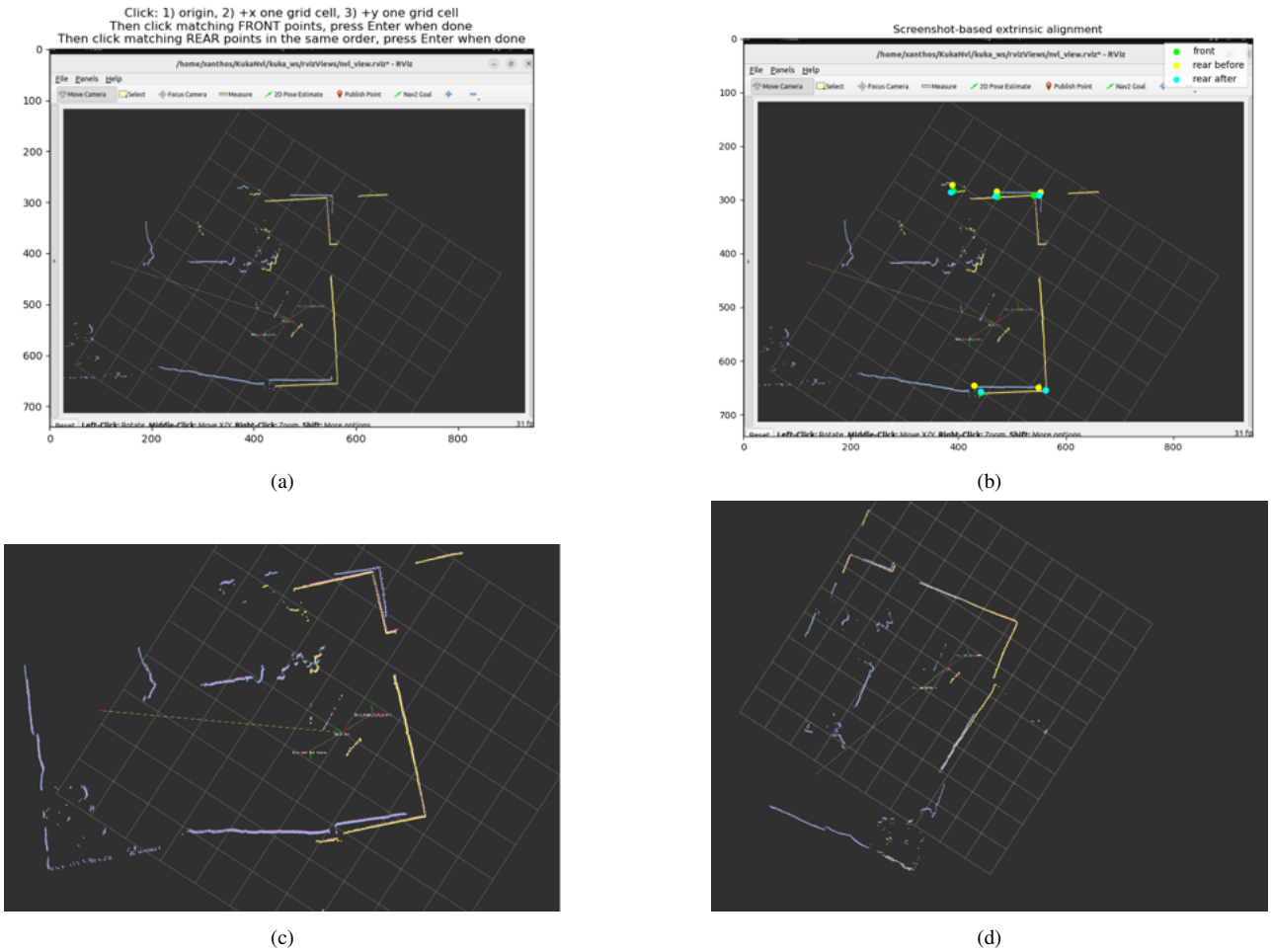
The software facilitates straightforward installation and building and is published in a remote repository. ROS2 packages and dependencies are needed, including drivers for the LiDAR and Realsense D435 camera, as well as navigation and SLAM toolkits such as nav2[15] [120] and `slam_toolbox`. The system includes a fully functional Gazebo simulation environment, allowing for virtual testing and evaluation of the (virtual) robot's capabilities. Using predefined launch files, the robot can be deployed in simulated industrial scenarios. The simulation integrates with ROS2 tools, enabling tasks such as navigation and mapping in a controlled virtual world. The resulting simulation environment is shown in Figure 25.

Extrinsic calibration of the LiDAR sensors

Accurate estimation of the relative pose (extrinsics) between the LiDAR sensors is a critical requirement for multi-sensor fusion. In the NVL payload, two 2D LiDARs are mounted at different locations on the robot and their scans are merged into a common reference frame. Even small errors in the relative translation or orientation between the sensors can lead to significant misalignment in the merged scan, which in turn degrades scan matching, SLAM accuracy, and localization performance.

In practice, manual measurements of the sensor mounting points are often insufficient to achieve the required precision. Errors on the order of a few centimeters or a few degrees in yaw can result in visible inconsistencies between the scans, especially in structured indoor environments. For this reason, an additional data-driven extrinsic calibration procedure was developed.

[15] https://docs.nav2.org/



(a) (b)

(c) (d)

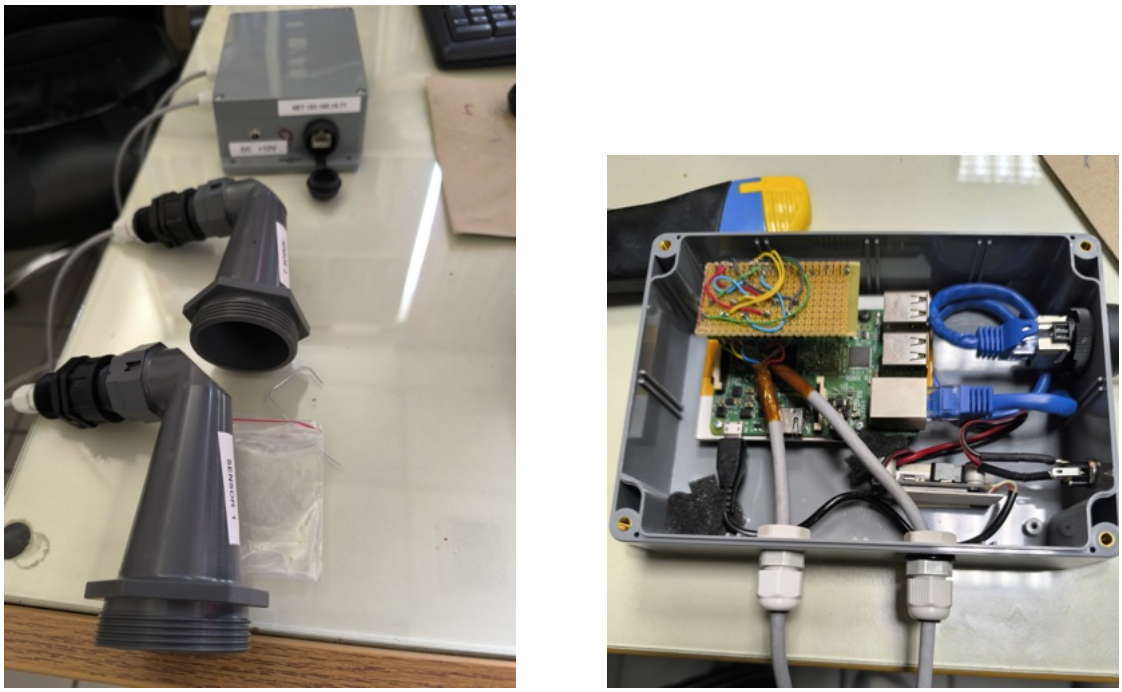
Figure 26: (a) and (b) Extrinsic calibration procedure applied to a snapshot from a LiDAR-based SLAM experiment. (c) SLAM result without using the extrinsic calibration procedure. (d) SLAM result with extrinsic calibration employed.

The calibration process is implemented in a dedicated software module, which estimates the rigid-body transformation between the two LiDAR frames based on corresponding geometric features observed in both scans. The method relies on manually selecting pairs of corresponding points (e.g., wall intersections or planar features) from the two LiDAR observations. The optimal 2D rigid transformation is then computed by a least-squares Procrustes analysis procedure that yields the relative translation and yaw rotation between the two sensor frames [184]. The resulting transformation is incorporated into the robot description (URDF/Xacro) as a static transform.

This calibration procedure allows for significantly improved alignment between the two LiDAR scans, leading to more consistent scan merging and improved performance of downstream components such as SLAM and localization. Qualitative experimental results indicate that refining the extrinsic parameters reduces visible map inconsistencies and improves overall trajectory accuracy. Representative calibration outputs and their effect on LiDAR-based SLAM are shown in Figure 26.

4.2.2.2 Ultrasound radars

The implemented solution comprises a dual MaxBotix MB7851 ultrasonic radar sensor array (see Figure 27) mounted on the KUKA robot, providing millimeter-resolution distance measurements for precise Y-axis room positioning. The two sensors, designated as Y1 room and Y2 room, measure the perpendicular distances from the robot to opposite room boundaries, enabling accurate determination of the robot's lateral position within the operational space. The ultrasonic measurements complement six SICK laser distance sensors (X1



(a) (b)

Figure 27: The dual MaxBotix MB7851 ultrasonic radar sensor array.

front, X2 front, X1 back, X2 back, Y1 wall, Y2 wall) that are interfaced through the Siemens PLC, creating a comprehensive eight-sensor fusion system capable of full three-dimensional robot localization with centimeter-level accuracy. The sensor data acquisition is implemented through a Python application utilizing the `pigpio` library for software serial communication, extracting distance values from the ASCII-formatted sensor output and transmitting measurements to the PLC via OPC UA protocol.

To ensure maximum system reliability in industrial environments, the implementation incorporates multiple layers of fault tolerance and recovery mechanisms. A hardware watchdog service (bcm2835_wdt) continuously monitors critical system parameters including network connectivity via periodic ping tests to the OPC UA server, Ethernet interface status, CPU load thresholds (max-load-1=16 for the 4-core Pi 3B+), and temperature limits (80°C threshold with normal operating temperature at 51°C), automatically triggering system reboot upon detection of kernel freezes, network failures, or thermal events to maintain continuous operation without manual intervention.

The sensor fusion algorithm operates on a Jetson platform running ROS, receiving distance measurements from eight independent sensors through two distinct data acquisition pathways: two MaxBotix MB7851 ultrasonic sensors (Y1 room, Y2 room) streaming real-time measurements via OPC UA from the Raspberry Pi at 1 Hz, and six SICK laser sensors (X1 front, X2 front, X1 back, X2 back, Y1 wall, Y2 wall) polled from the Siemens PLC datablock via Modbus TCP or OPC UA at 10 Hz. The system establishes a fixed room coordinate frame with known wall positions and geometric constraints, where the ultrasonic sensors measure perpendicular distances from the robot center to the Y-axis room boundaries, while the laser sensors provide distances to the X-axis front/back walls and redundant Y-axis wall measurements, along with inclination data from onboard IMU sensors. Each sensor measurement is preprocessed to account for sensor-specific characteristics including the ultrasonic beam divergence angle, laser sensor mounting positions relative to the robot's geometric center, and environmental factors such as temperature compensation for speed of sound variations (affecting ultrasonic measurements) and reflectivity variations (affecting laser readings). The measurement model transforms raw distance readings into a predicted robot pose by solving the geometric inverse problem: given the known room dimensions and wall positions, the eight distance measurements constrain the robot's position to a spatial region that is refined through probabilistic fusion, with measurement uncertainties quantified based on

sensor datasheets (1mm resolution for MB7851, sub-millimeter for SICK lasers) and empirically characterized noise covariance matrices capturing temporal and spatial measurement variations. The core fusion algorithm implements an Extended Kalman Filter (EKF) estimating the robot state vector $x = [p_x, p_y, p_z, \theta, \phi, \psi]^\mathsf{T}$ representing position (p_x, p_y, p_z) and orientation (θ, ϕ, ψ) denoting roll, pitch and yaw respectively, with associated covariance matrix P encoding estimation uncertainty.

The EKF operates in a prediction-update cycle at 10 Hz: the prediction step propagates the state estimate using a kinematic motion model incorporating wheel odometry from the robot's drive system and angular velocity from the IMU, while the update step assimilates the eight sensor measurements through a nonlinear measurement function h(x) that predicts expected sensor readings given the current pose estimate, comparing predictions against actual measurements to compute Kalman gains and refine the state estimate. The measurement function h(x) accounts for sensor geometry, transforming the robot pose into expected distances by computing line-of-sight vectors from sensor mounting positions to wall surfaces, with Jacobian matrices computed analytically for the linearization required by EKF. Sensor outlier rejection is implemented through Mahalanobis distance gating (χ^2 test with 95% confidence threshold) to exclude anomalous readings caused by specular reflections, multipath propagation, or temporary occlusions, while adaptive measurement noise covariance estimation adjusts sensor weights based on real-time signal quality indicators from the laser sensors and signal-to-noise ratios from the ultrasonic receivers. The fused pose estimate is published as a geometry_msgs/PoseWithCovarianceStamped ROS message at 10 Hz on the /robot_pose topic, where the pose field encodes the 6-DOF position and orientation in the room reference frame, the covariance field provides the 6×6 uncertainty matrix for downstream navigation algorithms.

4.2.2.3 Odometry estimation (RF2O)

Accurate motion estimation is a fundamental requirement for both SLAM and localization. Since no wheel encoders or other external odometry sources are available in the KUKA setup, the NVL stack relies on a LiDAR-based odometry method, namely Range Flow-based Odometry (RF2O) [76], to estimate the robot motion directly from consecutive laser scans.

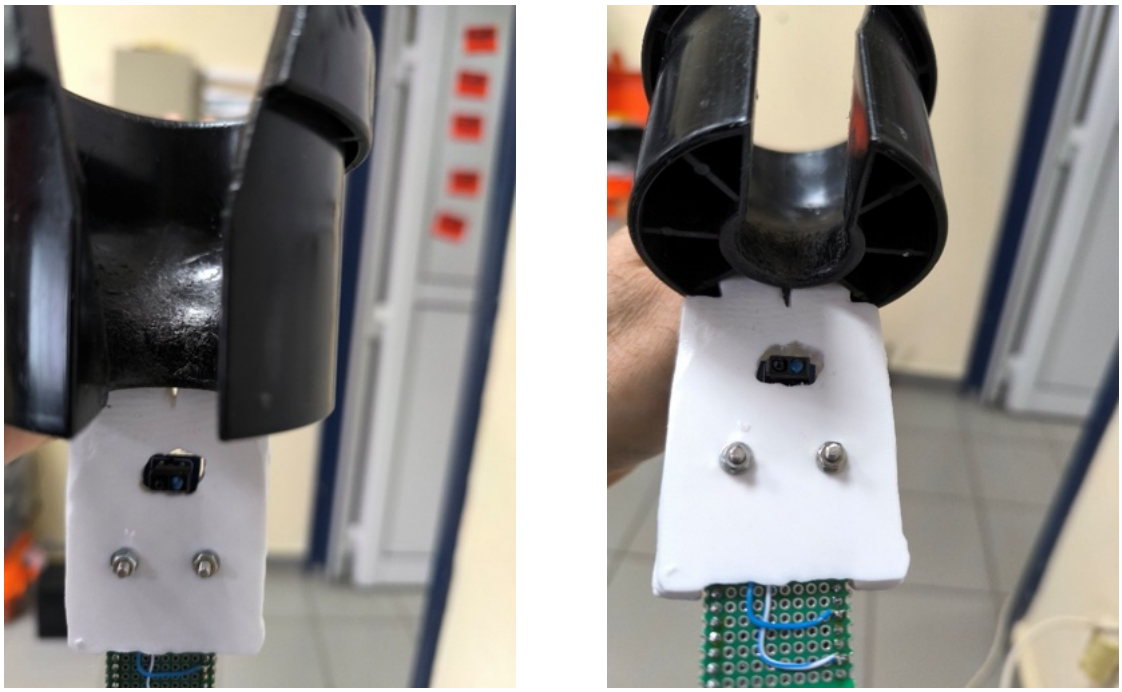
RF2O computes the relative motion between successive scans by exploiting the temporal variation of range measurements and the spatial structure of the observed environment. In the implemented system, it publishes the transform between the `odom` and `base_link` frames, thereby providing the motion estimate required by higher-level components. In particular, this transform serves as the initial estimate for scan matching in `slam_toolbox` and as the prediction model for AMCL.

The choice of RF2O is motivated by the lack of reliable proprioceptive sensing and by the need for a lightweight LiDAR-only solution compatible with the available payload. At the same time, LiDAR-based odometry remains sensitive to environmental structure, dynamic objects, and calibration inaccuracies, which may introduce drift and bias in the estimated trajectory. Despite these limitations, RF2O constitutes a practical and computationally efficient motion estimation module for the present NVL stack.

4.2.2.4 ROS2 integration and deployment

The NVL stack is deployed in ROS2 on the KUKA robot and provides the software infrastructure required for LiDAR-based mapping and localization. It is implemented mainly in Python, with lower-level components in C++, and is organized around ROS2 launch scripts and configuration files defining the robot model, sensor interfaces, static transforms, odometry, and the higher-level SLAM/localization modules.

In SLAM mode, where no *a priori* map is available, the stack uses `slam_toolbox` for online asynchronous mapping. LiDAR observations and odometry are combined to perform scan matching, loop closure, and incremental occupancy-grid map generation, yielding a map that is directly compatible with Nav2. When an external occupancy-grid map is available, the stack instead operates in localization mode: map construction is disabled, the reference map is published by `nav2_map_server`, and AMCL estimates the robot pose in



(a) (b)

Figure 28: Prototype of the tool presence detector attached to a toolholder.

the global map frame using LiDAR observations and odometry. In this mode, an initial pose must be provided either manually through RViz or programmatically by another node, such as the visual localization module. The resulting software stack was validated both in simulation and on the KUKA platform, as summarized in Sec. 4.4.

4.2.2.5 Tool presence detection (CRF UC)

For the CRF use case, initially a tool presence detector was developed, shown in Figure 28. This system implements a real-time tool presence detection solution using an ESP8266 microcontroller and an infrared (IR) sensor. The primary objective is to monitor whether a tool is present at a designated workstation and transmit this status information to a central Kafka message broker for further processing, logging, or integration with other industrial systems. The IR sensor provides analog readings that vary based on the presence or absence of an object, and the ESP8266 processes these readings using a software-configurable threshold to determine the tool's status. When the sensor value exceeds the threshold, the system reports "NO tool" (GripStatus=0), and when the value falls below the threshold, it reports "tool present" (GripStatus=1). The system only transmits messages when the status changes, reducing network traffic and ensuring efficient communication.

The system includes a comprehensive calibration mode that enables users to determine the optimal threshold value for their specific sensor setup and environment. During calibration, the system collects 40 averaged samples for each state (with and without tool), providing statistics including average, minimum, maximum, and range values. The process is interactive, requiring the user to press ENTER in the Serial Monitor after positioning the sensor for each stage, ensuring accurate measurements. Upon completion, the calibration routine calculates and displays a suggested threshold value based on the midpoint between the two states, along with warnings if the separation between states is insufficient for reliable detection.

4.2.2.6 Bluetooth tags (CRF UC)

For the CRF localization use case, the Bluetooth technology (BLE) was selected because it offers low cost per node while providing acceptable accuracy for distances typically encountered in CRF facilities. Although

alternative technologies may outperform BLE in specific aspects, each has drawbacks that make it less suitable overall: Ultra-Wideband (UWB) has higher cost and is better suited to shorter distances, ZigBee requires specialized equipment, Wi-Fi is prone to interference with existing 2.4 GHz and 5 GHz devices and involves complex fingerprinting, while RFID offers lower accuracy and longer response times. Therefore, the proposed solution will follow a Bluetooth-based approach, using a configuration with at least four anchor points where in each point an antenna array (8 antennas) will be used. Each monitored dolly will be equipped with a Bluetooth 5.1 tag. In the case of AoA (see Sec. 4.2.2), the mobile asset is equipped with a tag that transmits a Bluetooth direction finding signal, which includes a constant tone extension packet (CTE). In this scenario, measurements made by the antenna arrays are used to determine the angle of the incoming signals using a network-based engine. The signals transmitted by the mobile tag reach the individual antennas that comprise the anchor's multi-antenna array with a slight phase shift relative to the rest. Assuming that the signal propagates a planar wave, the slight phase differences observed at each antenna can be used to calculate its angle of arrival.

In the case of AoD, the mobile tag receives Bluetooth direction finding signals transmitted by one or several antenna arrays. In this case, the mobile client uses measurements of the incoming signal to compute the signal's direction of departure from the antenna array. Using AoD, the direction finding signals transmitted by each element of the anchor's antenna array reach the client with a slight phase difference relative to the rest. With information on the geometry of the antenna arrays, the client can calculate the angle of departure of the signal from the antenna array using the measured phase differences.

4.3 NTUA's robotic platform for NVL

Building on the simulation environment developed (see Sec. 4.2.2.1) and utilizing the FOOSBOT[16] platform, NTUA went on to create a DIY robot, equipped with a similar to KUKA NVL payload. This initiative is expected to enable further research activity feeding off SOPRANO technology.

FossBot - The propulsion system

The FossBot has three points of contact with the ground: two via the rear wheels and one via a ball-shaped wheel attached at the front. Its movement is primarily based on the rear wheels' ability to perform differential steering, while the ball-shaped wheel provides essential stability. Therefore, in this section, we present the propulsion system from both a software and hardware perspective.

The hardware

The propulsion hardware consists of a Raspberry Pi Pico, two DC motors driving the rear wheels, and a motor controller interfacing the control signals with the power stage. As shown in Figure 29, four Pico GPIO lines are connected to the controller inputs to command the left and right motors for differential-drive motion. A common ground is shared between the Pico and the motor controller to ensure a consistent electrical reference. The motor controller is powered by the robot's main power source, while the Pico is powered and communicates with the rest of the system through USB.

The software

To ensure compatibility with the rest of the system, micro-ROS was adopted and installed on the Raspberry Pi Pico. Since vehicle motion is addressed in this section, the ROS 2 geometry_msgs/msg/Twist topic is used to properly structure the corresponding commands. When a new message arrives, it is unpacked to extract the linear.x and angular.z values. These values are then used to compute the appropriate speed for each motor according to $left_speed = linear.x - angular.z$ and $right_speed = linear.x + angular.z$, where the linear

[16]FOSSBOT is an open technology robotic system for education, see https://fossbot.gr/landing-page

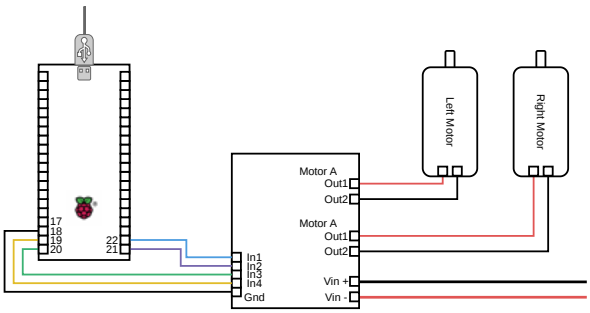


Figure 29: The schematic diagram of the propulsion system electronics

component corresponds to linear velocity and the angular component corresponds to angular velocity. Finally, the set_motor_speed function translates the calculated speeds into appropriate PWM signal values.

The testbed

In Figure 30, the propulsion system is presented as tested on the laboratory testbed. In this setup, a laboratory power supply (PSU) was used to power the system, while a desktop PC acted as a micro-ROS agent to simulate communication with the rest of the robot. During the initial testing phase, several ROS 2 messages were generated by the desktop PC and published to the Twist topic, while the system's response was evaluated by a human in the loop. Further testing using quantitative metrics is planned for future work.

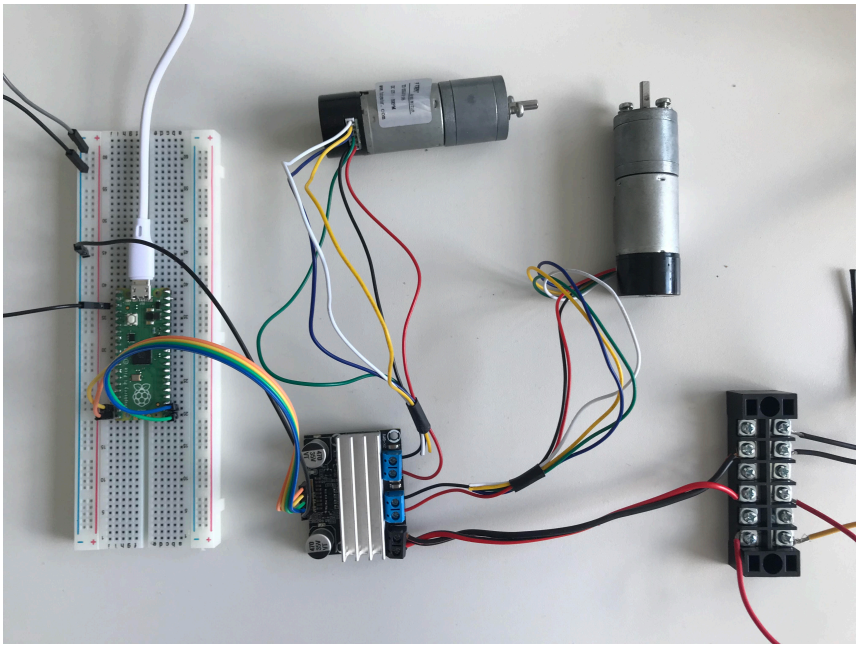


Figure 30: The laboratory testbed of the propulsion system

4.4 Results

4.4.1 Testing using the simulator

The simulator described in Sec 4.2.2.1 and Sec. 4.2.2.4 was used extensively to evaluate the LiDAR-based SLAM pipeline under controlled conditions. The testing workflow consisted of launching the stack in asyn-

chronous online mode, ensuring that LiDAR data were published on the appropriate ROS2 topic, and monitoring the resulting map and robot trajectory in RViz2 while the robot moved through the environment and revisited previously mapped areas. An example RViz2 output from the simulation workflow is presented in Figure 31.

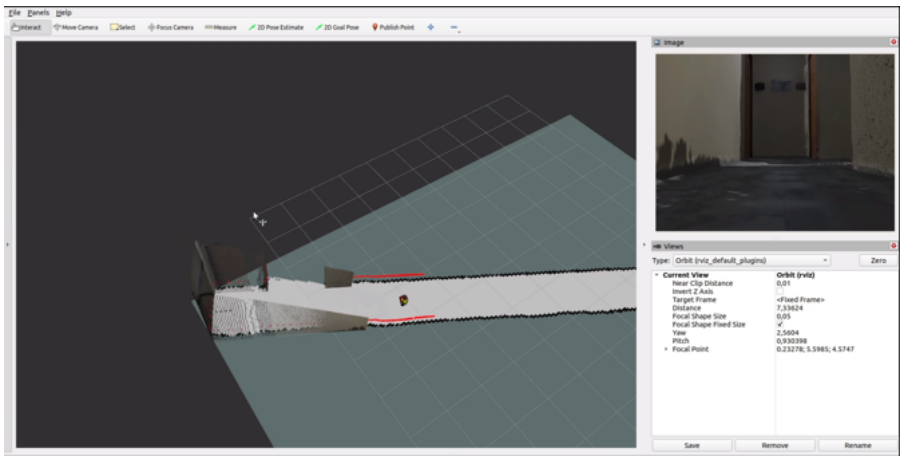


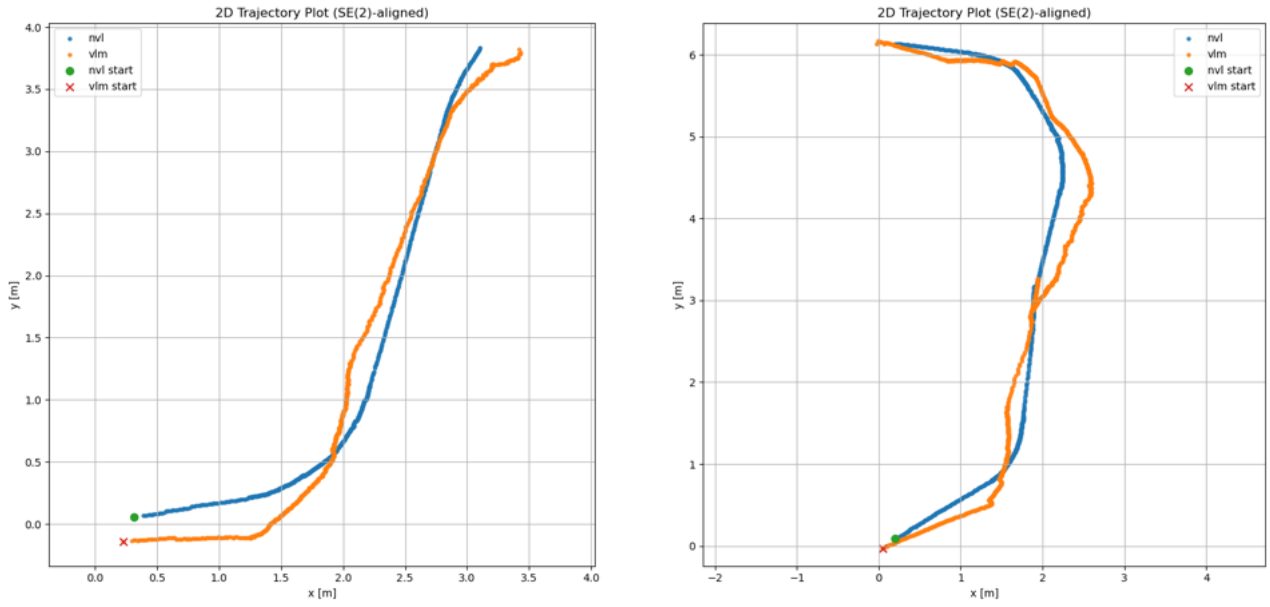
Figure 31: RViz2 output from running the simulator.

As a first benchmark, a building with a known blueprint was selected as the test environment. Since the environment remained static over time, it was suitable for assessing the robustness and accuracy of the simulator. SLAM was used to reconstruct the environment from LiDAR scans, and the resulting occupancy map was compared against the building blueprint. The generated map was found to be consistent with the ground truth, with key structural features such as walls, doors, and major room layouts accurately reproduced.

In addition, pose estimation accuracy was assessed using predefined trajectories within the simulated environment. The estimated robot path observed in RViz2 was compared against the expected motion and found to be in good agreement. Finally, computational performance was also examined, including responsiveness, mapping speed, and loop-closure behavior, allowing further tuning and refinement of the overall stack.

4.4.2 Testing of the LiDAR-based SLAM

To evaluate the localization accuracy of the LiDAR-based NVL component, a performance assessment was conducted using a rigorous evaluation protocol encompassing data acquisition, independent pose estimation, precise temporal association, and finally localization error calculation. Specifically, RGB-D and LiDAR timestamped datasets were recorded on-board as the robot was driving in KUKA's test environment along trajectories comprising both straight segments and curved turns. Distinct datasets were acquired, each comprising a few thousand individual samples, and collectively ensuring a comprehensive sampling of environmental features and operational conditions for localization accuracy testing. Following data collection, two localization pipelines were applied to the raw sensor data: a batch Structure-from-Motion (SfM) pipeline using the RGB-D data for camera pose estimation and the NVL pipeline using the LiDAR data for localization. The aforementioned SfM pipeline involves a COLMAP engine [183] modified by FORTH, whose comprehensive offline processing ensures that it yields highly accurate poses. The poses estimated from both localization methods were temporally synchronized using the timestamps and NVL's performance was compared against the poses obtained from the SfM pipeline which served as the ground truth (see Figure 32). NVL's accuracy was evaluated using the standard Absolute Trajectory Error (ATE) metric, which computes the Root Mean Square Error (RMSE) between the estimated and ground-truth trajectories after spatial alignment. The ATE results for all sequences we examined reveling an RMSE below 20 cm. Localization errors when using an *a priori* reference map (i.e. in *localization mode*, see Sec. 4.2.2.4) and incorporating the radar-based results are expected to decrease further, although this remains to be verified experimentally (see Sec. 4.5).



(a) (b)

Figure 32: Comparison of two VLM and NVL trajectories over common time intervals after temporal synchronization and 2D rigid alignment.

Representative snapshots of the LiDAR-based SLAM system deployed in the KUKA environment are shown in Figure 33.

4.4.3 Testing BLE-based localization

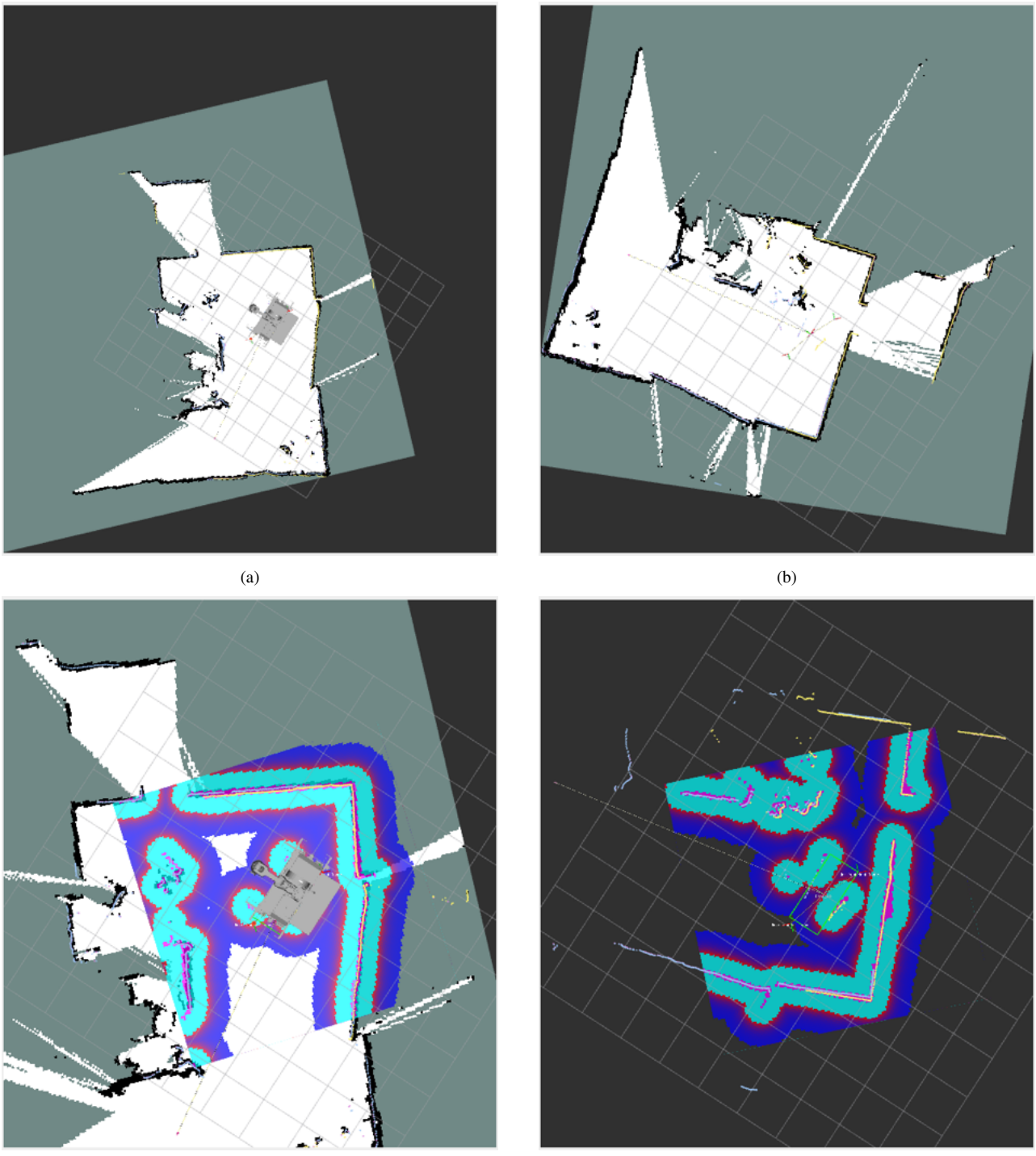
The experimental setup begins with the physical deployment of four u-blox ANT-B10 anchor nodes at the corners of the 12m×8m facility at a fixed height of 3.8m. Each anchor must be precisely oriented; a misalignment of even a few degrees in azimuth or tilt can result in significant spatial deviation. Once the anchors are secured, a local coordinate system is established, typically with the origin $(0, 0, 0)$ at one corner of the floor. Each anchor's location and orientation are then registered within the u-locateEmbed positioning engine to define the spatial boundaries. To ensure that the environment is representative of an industrial assembly line, robotic arms and metallic structures are kept in their operational positions to account for multipath interference and signal shadowing during the measurement phase.

Data collection involves moving the four BLE beacons through 50 pre-defined ground truth (GT) positions across the room, specifically targeting "challenging" zones near the robotic arms and low-visibility areas. At each GT point, the system captures the Constant Tone Extension (CTE) from the beacons, which the anchors process into Phase Difference (PD) values to calculate the Angle of Arrival. These raw angles are then processed by the positioning engine to generate an estimated coordinate $(\hat{x}, \hat{y}, \hat{z})$. The final evaluation is conducted by calculating the Root Mean Square Error (RMSE) using the formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}$$

where d_i represents the Euclidean distance between the ground truth and the system estimate. This process is repeated for both the horizontal plane and the vertical component to provide a comprehensive accuracy profile.

The environment is a 12m×8m factory floor with a 4m ceiling. The four u-blox ANT-B10 nodes are mounted in the corners of the ceiling to maximize Line-of-Sight (LoS) coverage across the robotic assembly area (see Figure 34).



(a) (b)

(c) (d)

Figure 33: LiDAR-based SLAM in operation, deployed on the KUKA robot and environment; front LiDAR scan in yellow, rear scan in light blue. Images (a) and (c) depict the URDF/Xacro robot model; (c) and (d) include LiDAR-enabled proximity-based obstacle detection (“bumper” layer).

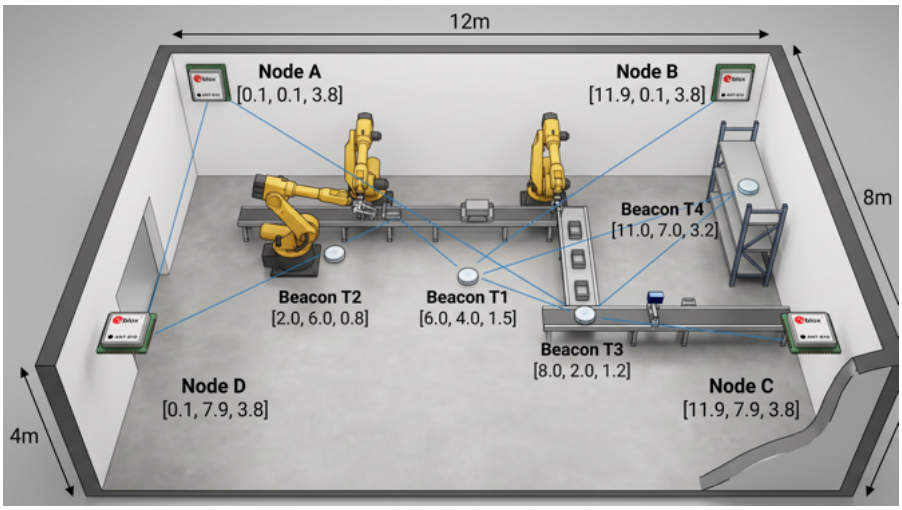


Figure 34: BLE-based localization experimental setup.

Node (Anchor) Coordinates:

1. Node A: $(0.1, 0.1, 3.8)$
2. Node B: $(11.9, 0.1, 3.8)$
3. Node C: $(11.9, 7.9, 3.8)$
4. Node D: $(0.1, 7.9, 3.8)$

Obstacles (Robotic Arms):

- Robot 1 Base: Located at $(2.0, 6.0, 0.0)$ (Near Node D).
- Robot 2 Base: Located at $(8.0, 2.0, 0.0)$ (Near Node B).
- Conveyor Belt: Running centrally from $X = 2$ to $X = 10$ at $Y = 4$.

Results of RMSE from 50 measurements are presented in Table 9. The results can justify the BLE superiority over widely used UWB (which costs 10 times more per node) due to specific hardware advantages of the u-blox EVB-ANT-1.

Advanced Angular Resolution : Unlike basic BLE, the 8-patch antenna array in the ANT-B10 provides significantly higher angular resolution than standard UWB anchors, which rely solely on timing.

Multilateration Smoothing : By using four nodes, the system benefits from over-determined positioning. The u-locateEmbed engine uses advanced Kalman filtering to "smooth" the jumps that metallic reflections (multipath) usually cause in assembly lines.

Superior Vertical Control : UWB vertical accuracy is notoriously poor when anchors are placed on the same horizontal plane (the ceiling). Because AoA measures the angle directly, it is often more stable at determining the height of a tag than a UWB system relying on TDOA (Time Difference of Arrival).

4.5 Discussion

The NVL component is largely in place and tested in laboratory conditions, simulated environments, and relevant real work environments for the KUKA use case. The results obtained so far indicate that the component

Metric	BLE AoA
Horizontal RMSE	0.29 m
Vertical RMSE	0.52 m
Max Horiz. Error	0.48 m
Max Vert. Error	0.88 m
Min Horiz. Error	0.03 m

Table 9: BLE-based localization experimental results.

operates as intended, producing consistent localization estimates and supporting higher-level functionalities built on top of it, most notably navigation.

Through testing and validation, it has become clear that the main limitation affecting the accuracy of the current LiDAR-based SLAM pipeline is the quality of motion estimation, i.e. the odometry. In the present system, odometry is derived directly from LiDAR scan matching (RF2O [76]), and therefore its accuracy depends strongly on environmental structure and sensor configuration. Compared to dedicated proprioceptive sensors such as wheel encoders or IMUs, LiDAR-based odometry can exhibit reduced consistency, particularly in environments with limited geometric features or dynamic elements.

Furthermore, the accuracy of motion estimation is affected by factors such as the surrounding environment and the precision of sensor mounting (extrinsic calibration). In the localization case, this limitation can be partially mitigated by the constraints imposed by an *a priori* reference map, which helps correct drift through global consistency. In addition, the fusion of LiDAR-based localization with radar-based localization is expected to improve robustness and accuracy.

Upcoming work

An important upcoming development concerns the integration of LiDAR-based and radar-based localization outputs into a common final 2D pose estimate for the NVL component. The overall design for this fusion stage is already available and its deployment is imminent. The current plan is to implement this fusion through a Kalman-filtering approach, either in the form of an Extended Kalman Filter (EKF) or an Unscented Kalman Filter (UKF), depending on the results obtained from real-world testing.

5 Human Monitoring [HMO]

5.1 Technology overview and module specifications

The Human Monitoring (HMO) component addresses challenging computer vision tasks within SOPRANO, enabling efficient and seamless Human-Robot Interaction (HRI) and Human-Robot Collaboration (HRC) across use case scenarios. The overarching goal of the HMO component is the on-line visual monitoring of worker motion and activities within the actual operational environments of the SOPRANO use cases.

To achieve this, the component tackles five primary categories of vision-based tasks:

a) 3D human detection and body orientation estimation within the 3D workspace or the shop floor.
b) 3D body pose estimation capturing the positions of anatomical body keypoints (e.g., elbows, knees, shoulders, and hips).
c) Spatiotemporal analysis of the estimated 3D body poses to detect ergonomically sub-optimal (awkward) postures during work activities, evaluating physical ergonomics throughout task cycles.
d) Spatiotemporal analysis of 3D body poses to recognize use-case-specific upper-body, arm, and hand gestures used by workers to issue relevant robot commands.
e) Recognition of performed work actions and monitoring of task cycle progress during execution.

Consequently, the HMO component comprises an ensemble of methods ranging from foundational human body pose detection and tracking to more advanced tasks, such as action recognition and work progress evaluation within the context of assigned tasks. These methods rely heavily on machine learning and deep learning approaches, necessitating annotated datasets to train the underlying neural network models. Such datasets are essential for achieving accurate and robust task performance; Depending on the task and use case, the video datasets include bounding boxes, image sequences annotated with bounding boxes, 2D masks, and detailed 3D human pose information, alongside semantic labels for actions, gestures, and ergonomically sub-optimal postures.

To tailor functionalities to SOPRANO's specific requirements, the HMO utilizes various visual sensors to continuously acquire RGB and depth (RGB-D) image sequences. Depending on the use case and environmental setting, this involves deploying single or multiple sensors—either mounted on mobile robots or placed at stationary locations across the shop floor.

The outputs generated by the HMO component are subsequently utilized by other SOPRANO modules, including MH-MR Task Allocation (CTA), the Context Extraction Module (CEM), Object Perception (OBP), Runtime Safety Assessment tools (RSA), and the Human-Digital Twin (HDT).

The following sections discuss existing developments related to these vision-based problems and targeted tasks.

5.2 Methodology and implementation

5.2.1 Existing approaches (SOTA)

A brief overview of existing state-of-art approaches related to the vision tasks of interest for human monitoring is reported in the following.

5.2.1.1 Human pose estimation

Human pose estimation (HPE) in 2D or 3D, in images or videos is a challenging yet fundamental vision task due to depth ambiguity, self-occlusion, and the fact that it represents an inverse, highly ill-posed problem. In addition, the rather time-consuming process of collecting high-quality and fine-grained annotations (i.e.,

labeled/ground truth data) for the human body leads to limited access to training data. In particular, 3D annotation requires dedicated motion capture systems, which are not easy to obtain and install to capture human motion and activities in various contexts and environmental conditions.

OpenPose [23] is the first open-source real-time approach for multi-person 2D pose detection in images. The core feature of this method refers to an explicit non-parametric representation of the associations among the estimated body keypoints, called Part Affinity Fields (PAFs), which encodes both the position and orientation of the candidate human limbs. A Convolutional Neural Network model that features two branches is used to learn how to estimate the observed 2D body poses; the first branch predicts a set of 2D confidence maps that indicate the locations of body parts, while the second branch defines a set of 2D vector field of PAFs. In essence, the output of this approach comprises a set of 2D image coordinates (x, y) that correspond to the locations of $K = 25$ skeletal body joints in the image plane according to the BODY25 pose output format. FORTH's MocapNet2 [165] method is an efficient end-to-end approach that relies on ensembles of Deep Neural Networks to regress the 3D coordinates and angles of 15 human body joints per frame. MocapNet2 uses the input skeleton-based 2D body poses for regression with respect to the 3D body pose and an Inverse Kinematics mechanism for fine-tuning the estimated 3D human poses with respect to the anthropometrics of the observed human body (i.e., lengths, scale of body limbs/parts). Moreover, MocapNet2 splits the hierarchy of the skeletal body model into upper and lower body and estimates the 3D configuration of the joints for each body part independently, also estimating the global orientation of the human body in 3D space. This important design feature allows us to tackle cases of extreme occlusions of the human body. For example, even when the whole lower body is not visible by the camera, either due to self-occlusions or other objects in the scene, the method can still produce accurate pose estimation results for the visible upper body part. The output of the MocapNet2 approach is a Biovision Hierarchy (BVH) character animation file format [132] representing the estimated human motion throughout the input video.

Graformer, a recently proposed method for HPE [251], relies on 2D pose estimates to learn the representation of the underlying graph-structure data and to infer 3D human poses using a novel Transformer architecture based on Graph-based Convolutional Networks. The MHFormer method [104] also relies on a Transformer-based network to learn spatiotemporal representations of multiple plausible 3D pose hypotheses. This work extends the previously proposed PoseFormer method [254] that employs a purely transformer-based approach for 3D human pose estimation in videos without convolutional architectures. The MHFormer achieves state-of-the-art results on the two challenging datasets Human3.6M [131] and MPI-INF-3DHP [131]. Another approach aspires to exploit the deterministic 3D-to-2D mapping to solve the ambiguous inverse 2D-to-3D problem for monocular HPE using a probabilistic framework based on normalizing flows [228]. Finally, the recently proposed method in [197] tackles the 2D HPE for multiple persons in images using an end-to-end architecture based on Transformers and a novel joint decoder that is utilized to refine the estimated poses by exploring the kinematic relations between body joints.

5.2.1.2 Gesture recognition

Gesture recognition [59] is a rapidly advancing field in Computer Vision, driven by its broad applications in human-computer interaction, virtual reality, and assistive technologies. The introduction of robust sensing technologies and advanced algorithms has enabled the recognition of both static and dynamic gestures across various modalities, including RGB, depth, and skeleton data. The primary challenge lies in capturing and interpreting complex spatio-temporal patterns inherent in human gestures, which vary widely in speed, trajectory, and context.

Deep learning has revolutionized gesture recognition with models like CNNs, RNNs, and hybrid architectures. These models automate feature extraction and classification, overcoming the limitations of hand-crafted features. Molchanov et al. [135] applied 3D CNNs for capturing spatio-temporal dynamics in multi-modal data. Attention mechanisms and graph-based networks, such as those by Hou et al. [72], enhance model capabilities by focusing on critical spatio-temporal features, resulting in improved performance across various datasets. More specifically, skeleton data, often combined with depth information, provides precise joint-level details,

enabling models to capture intricate hand movements. This approach is advantageous due to its robustness against changes in lighting and background noise. Notable contributions include Chen et al. [28] and Lai et al. [99], who employed bidirectional RNNs to classify gestures. Maghoumi and LaViola [121] proposed an end-to-end deep network model that achieves state-of-the-art results on challenging datasets. Shi et al. [198] introduced a dynamic graph-based spatial-temporal attention (DG-STA) model, while Lui et al. [112] utilized a combination of 3D CNNs and HMMs for robust gesture recognition. Devineau et al. [37] focused on using CNNs to extract motion features from skeleton data, specifically targeting global hand translations and rotations. This approach allows for effective spatio-temporal feature extraction, as seen in Li et al. [107], who achieved high accuracy rates on the DHG 14/28 dataset [200].

5.2.1.3 Human action understanding

Understanding human actions in visual data is linked to advances in areas like human dynamics, domain adaptation, and semantic segmentation. Extensive research in this domain follows the latest deep learning methodologies [65, 139, 100, 103, 46]. These approaches often model coarse geometric and appearance features using spatial regions-of-interest (ROIs) [56] to classify actions in short video clips. More advanced methods represent the temporal and spatial structure of entities, such as 2D/3D skeletal models [23, 165], 2D hand/object masks [10], and 3D poses of hand(s)-object(s) [211, 52].

Spatiotemporal relationships in Human-Object Interactions (HOI) are modeled using attention mechanisms [117], Graph Neural Networks (GNN) [103], CNNs [88], RNNs, LSTMs [249, 122], and Transformer models [246, 54, 232]. However, many approaches treat HOI as non-composite, monolithic activities, limiting their ability to generalize across diverse actions. More recent research considers HOI as composite activities with complex spatiotemporal relationships [126], integrating high-level semantics [9], logic rules [237], and graph-based methods [159]. Novel representations encode human-object relationships over time using spatio-temporal graphs [77], with some focusing on visual reasoning to uncover causal relationships [10].

Finally, recent advances in activity understanding in videos focus on the challenge of modeling procedural activities—tasks that unfold as ordered sequences of interdependent actions [113, 38] Traditional action recognition methods often fail to capture this temporal and causal structure, limiting their ability to reason about progress, detect mistakes, or segment actions online. To overcome these issues, those recent research has focused on integrating task-graph representations and progress-aware temporal modeling into video analysis frameworks. More specifically, the work in [188] introduce Differentiable Task Graph Learning, a method that learns a soft, differentiable task graph from sequences of key-steps in egocentric procedural videos, optimizing edge weights via maximum likelihood and integrating the graph into neural architectures for improved mistake detection. This approach not only recovers procedural structure from data but also uses the learned graph to detect deviations online. In contrast, [196] propose Progress-Aware Online Action Segmentation (ProTAS), which adapts typical segmentation backbones to causal models and augments them with a module that predicts the progress of ongoing actions, combining that with a learned task graph to stabilize segmentation in real time. While the former emphasizes structure learning and mistake detection, the latter focuses on robust online action segmentation in streaming settings, especially for egocentric procedural tasks.

5.2.1.4 Vision-based ergonomic posture evaluation

Several screening tools for physical ergonomics assessment are commonly used in industry, based on different evaluation protocols, sets of postures, and other related human activities such as grasping and lifting. These ergonomic analysis tools include the Rapid Entire Body Assessment (REBA) [128], the European Assembly Worksheet [181] (EAWS), the OCRA checklist [151], the MURI risk analysis [230] and the Ovako Working Posture Analyzing System (OWAS) [82].

Based on these methods, various efficient vision-based approaches have been proposed to tackle the challenging problem of automated posture evaluation during work activities. For example, Parsa et al. [162] exploited spatial and temporal visual features from RGB-D frames using CNNs for predicting the ergonomic risk of

object manipulation actions according to the REBA method. Shafti et al. [193] automatically extract 3D skeletal body information using RGB-D frames to continuously analyze the user's posture and understand the safe range of arm motions during welding actions following the RULA posture monitoring method. Both methods in [190] and [241] address the task of vision-based posture recognition on construction sites using the OWAS approach. Mehrizi et al. [130] also proposed a deep learning approach for markerless 3D pose estimation optimized in the context of object lifting tasks by using RGB images from two different viewpoints. Recently, Parsa et al. [161, 160] proposed a novel approach based on Graph-based CNNs and LSTMs to recognize object manipulation actions (e.g., lifting, moving boxes, etc.) and predict the REBA scores in videos. Recently, a novel multi-stream deep network has been proposed [94] to compute the REBA score based on 3D skeletal data sequences extracted during work activities.

5.2.1.5 Toolkits

In the domain of vision-based pose estimation and gesture recognition [59], several state-of-the-art toolkits provide robust frameworks for analyzing human poses in 2D or 3D and interpreting gestures performed as dynamic postures in videos. These include OpenPose, MMPose, AlphaPose, and MediaPipe, each offering unique capabilities tailored to different applications and research requirements. These toolkits support sign language and gesture recognition tasks, addressing various challenges such as real-time performance, multi-person tracking, and context-specific customization. Their applicability spans diverse domains, including healthcare, robotics, education, and entertainment.

OpenPose[17], developed by the Carnegie Mellon Perceptual Computing Lab, is one of the first open-source frameworks for real-time multi-person 2D pose detection. It is capable of recognizing detailed human body movements, including body gestures, hand gestures, facial expressions, and even foot positions. This makes it a versatile tool for applications such as sign language recognition, fitness tracking, and human-computer interaction.

MMPose[18], part of the OpenMMLab ecosystem, is an advanced toolbox for pose estimation, emphasizing modularity and extensibility. While it does not focus solely on gesture recognition, it supports detailed human pose estimation across a wide range of datasets and configurations. Researchers often use MMPose for tasks such as full-body gesture recognition, sign language analysis, and ergonomic posture detection.

AlphaPose[19]provides an efficient and fast framework for human pose estimation, particularly in challenging scenarios involving crowded scenes or dynamic activities. It can detect body gestures with high precision and track multiple people over time, making it suitable for applications such as sports analytics, action recognition, and surveillance.

Moreover, the MediaPipe Toolkit [20] provides a comprehensive set of cross-platform tools for real-time machine learning applications. Its Gesture Recognizer task specifically focuses on hand gesture recognition, identifying gestures such as thumbs up, peace signs, or custom-defined gestures based on hand landmarks.

5.2.1.6 Datasets

We summarize available video datasets related to the visual understanding of human manufacturing activities in industrial environments, as shown in Table. 10. Only a few of them provide annotation data related to ergonomic analysis of body postures during work activities. The UW-IOM dataset [162] features limited object manipulation actions involving awkward poses and repetitions and frame-level annotations according to the REBA ergonomic risk index. The sustAGE User Postures & Actions Monitoring dataset [157] provides time-synced video recordings from multiple static RGB-D cameras and heart rate data of real line workers

[17]OpenPose GitHub Repository: https://github.com/CMU-Perceptual-Computing-Lab/openpose
[18]MMPose GitHub Repository: https://github.com/open-mmlab/mmpose
[19]AlphaPose GitHub Repository: https://github.com/MVIG-SJTU/AlphaPose
[20]MediaPipe Toolkit - Gesture Recognizer

acquired using wearable sensors recorded during manufacturing activities. Annotation data related to assembly actions and ergonomically sub-optimal postures based on the MURI postural grid for physical ergonomics assessment are included [21]. The AnDy multi-modal dataset [127] captures human motions in industry-relevant activities such as screwing and manipulating loads[22]. Annotation data regards whole-body kinematics of 13 participants recorded with optical motion capture, finger pressure force, videos, and postural events following the EAWS approach [181]. The IKEA ASM dataset [226] focuses on human action segmentation and fine-grained recognition providing multifaceted annotation data for a realistic scenario of chair assembly actions in videos captured from different viewpoints[23].

The IndustReal [185] [24] and the ENIGMA-51 [168] [25] datasets comprise egocentric videos of maintenance or assembly activities on a construction-toy assembly set and repair procedures on electrical boards, respectively. The Assembly101 dataset [189] [26]features exocentric and egocentric recordings of people assembling and disassembling 101 "take-apart" toy vehicles. The HA4M multi-modal dataset [32] demonstrates assembly activities for building an Epicyclic Gear Train [27]. The ATTACH dataset [2] demonstrates simulated assembly actions in the context of a human-robot collaborative scenario. Finally, the BRIO-TA [138] dataset simulates operations in factory assembly consisting of scenarios for normal and anomalous work processes. The recently proposed multi-modal dataset for car door assembly activities noted as CarDA [155], comprises a set of time-synchronized multi-camera RGB-D videos and motion capture data of the 3D body kinematics acquired using a motion capture system during car door assembly activities performed by real line workers in an experimental, real-world car manufacturing environment. The dataset is freely available online[28].

Dataset	Type of actions	Data	Viewpoint (cameras)	Ergonomics	Videos	Duration (h)
ATTACH [2]	assembly actions	Video RGB-D, IR, 3D poses	exocentric (3)	-	378	17
BRIO-TA [138]	assembly toy set	Video (RGB)	exocentric (1 top-view)	-	75	2.9
IKEA ASM [13]	assembly furniture	Video RGB-D, 3D poses	exocentric (3)	-	381	35.3
IndustReal [185]	maintenance, assembly toy set	Video RGB-D, Gaze	egocentric (1)	-	84	5.8
ENIGMA-51 [168]	repairing electrical board	Video, audio instructions	egocentric (1)	-	51	22
Assembly101 [189]	assembly toy vehicles	Video RGB-D, 3D hand poses	exo-egocentric (8-4)	-	362	167
HA4M [32]	assembly mechanical parts	Video	exocentric (1)	-	217	5.9
sustAGE [157]	car door assembly	Video RGB-D, heart rate	exocentric (2)	MURI	40	0.8
UW-IOM [162]	manipulating boxes-rods	Video RGB-D, 3D poses	exocentric (1)	REBA	20	0.63
AnDy [127]	screwing, manipulating loads	Video RGB-D, motion capture	exocentric (2)	EAWS	390	5
CarDA [155]	**car door assembly**	**Video RGB-D, motion capture**	exocentric (2)	**EAWS**	**25**	**1.48**

Table 10: Overview of video datasets focusing on human activity monitoring in industrial or industrial-like environments.

5.2.2 The approach in SOPRANO

SOPRANO encompasses three distinct use cases across diverse industrial domains and environments: building renovation (KUKA), vehicle door assembly (CRF), and smart agriculture (E-TERRY). The underlying scenarios involve complex monitoring activities. For example, the CRF use case requires multi-worker pose estimation, action recognition, and postural ergonomic evaluation during car door disassembly tasks. The E-TERRY use case involves 3D worker pose estimation, alongside action and gesture recognition (e.g., carrying, harvesting, or issuing "go" and "stop" commands) for agricultural tasks. Similar functionalities are deployed within the KUKA construction environments. Collectively, these use cases demonstrate the extensive applicability of advanced, markerless, vision-based human monitoring technologies across varied industrial settings.

[21] https://zenodo.org/records/7043787
[22] https://andydataset.loria.fr/
[23] https://github.com/IkeaASM/IKEA_ASM_Dataset
[24] https://github.com/TimSchoonbeek/IndustReal
[25] https://iplab.dmi.unict.it/ENIGMA-51/
[26] https://assembly-101.github.io/
[27] https://baltig.cnr.it/ISP/ha4m
[28] https://doi.org/10.5281/zenodo.13370887

HMO use case	Architecture support	Camera Type	Communication
ETERRY	ARM (Jetpack SDK 6.x)	StereoLabs ZED X	Kafka
KUKA	ARM (Jetpack SDK 6.x)	Intel RealSense	Kafka
CRF	x86-64 (MS Windows)	StereoLabs ZED 2	Kafka
Open Call	- x86-64 (Linux) - ARM (Jetpack SDKs 5.x, 6.x)	- Intel RealSense - StereoLabs ZED	–

Table 11: HMO component variants for supporting the SOPRANO use cases and the SOPRANO Open Call.

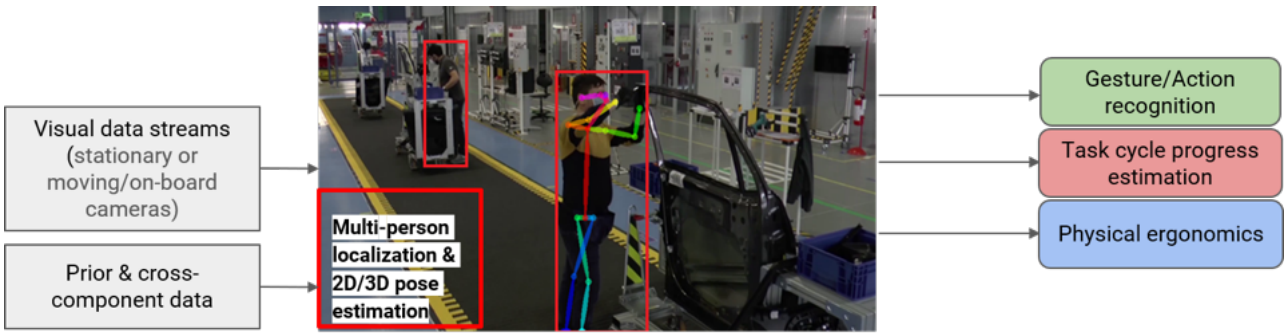


Figure 35: The Human Monitoring component workflow in SOPRANO illustrated with an image captured from the real CRF use case workplace.

A list of different camera types/models used and the vision-based competencies offered by the HMO component is reported in Table 12, while in Table 11 an overview of the HMO variants implemented for use in the SOPRANO use cases and the SOPRANO Open Call is reported.

To address the unique challenges posed by these use cases, the following key tasks have been identified:

- 3D Human Body Detection and Pose Estimation: Two distinct approaches are applied across the SOPRANO use cases to accurately monitor worker body configurations and analyze behavioral motion patterns.
- Action Recognition: Focused primarily on the CRF scenarios, this task aims to recognize worker actions during prolonged industrial activities and to estimate task progress throughout car door disassembly cycles.
- Vision-Based Analysis of Physical Ergonomics: Specifically aimed at the CRF use case, this task evaluates the postures of workers to identify potential physical ergonomic risks.
- Gesture Recognition: Implemented for the KUKA and E-TERRY use cases, this enables seamless human-robot interaction through upper-body, arm, and hand gestures.

A general workflow of the HMO approach that covers all different use cases in SOPRANO is illustrated in Fig. 35.

Furthermore, to fulfill the defined functional requirements across all three use cases, customized hardware specifications and setups are required. Providing efficient solutions to these challenging vision tasks inevitably leads to the design of different variants of the HMO pipeline. For instance, the CRF use case (indoor workspace) relies on a stationary RGB-D multi-camera system paired with a workstation PC for data acquisition and processing. In contrast, the E-TERRY (outdoor workspace) and KUKA (indoor workspace) use cases utilize a single on-robot RGB-D camera connected to an embedded (ARM-based) computing platform.

These tasks underline SOPRANO's aim of enhancing automation and human-robot collaboration through state-of-the-art vision-based monitoring methodologies tailored to the specific demands of each industrial domain. It is important to note that the HMO component shares the camera, hardware equipment, and its setup with the Object Perception (OBP) component, as described in 3.2.2. By using a shared camera and hardware infrastructure for both components, the SOPRANO platform streamlines the software/hardware footprint and maximizes

Use Case	Camera Type	Sensor Type	Processing	Task
CRF	2x Stereolabs ZED 2	Passive	Off-board	Human detection, Pose Estimation, Ergonomic posture evaluation, Action/Task progress classification
E-TERRY	Stereolabs ZED X/2i	Passive	On-board	Human detection, Pose Estimation, Gesture Recognition
KUKA	Intel RealSense D456	Active	On-board	Human detection, Pose Estimation, Gesture Recognition
Open Call	All	Active Passive	On-board Off-board	Human detection, Pose Estimation, Ergonomic posture evaluation

Table 12: Summary of camera types, features, and vision-based tasks deployed in all three use cases concerning the HMO component.

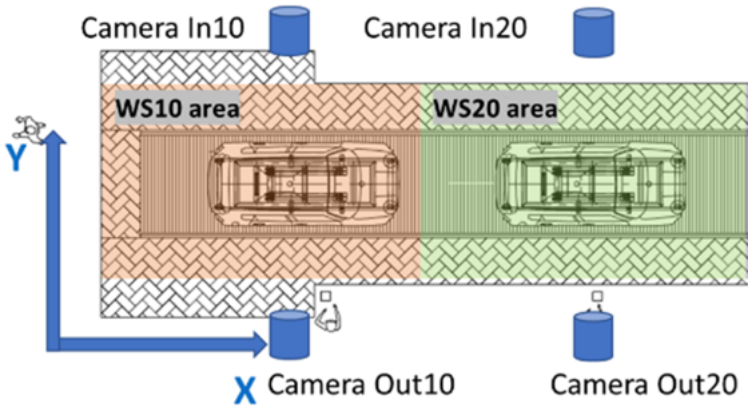


Figure 36: The outline of the manufacturing environment's shop floor for the CRF use case. Two workstation areas (shown as color-coded rectangles) are virtually defined on the conveyor belt area (see also Fig. 5). A pair of two stereo cameras (InXX, OutXX) (blue cylinders) are installed atop poles on both sides of each workstation.

the utility of limited hardware resources in industrial environments to ensure that visual human monitoring and object perception occur within an identical spatial and temporal frame of reference, reducing calibration errors and data synchronization issues/latencies that often hinder performance in multi-sensor systems.

The design for the placement of the multi-camera system in stationary locations on the CRF shop floor is illustrated in Fig. 36. The following sections detail each of the vision tasks and functionalities supported by the HMO component. Within each dedicated section, the specific methodologies and approaches proposed to address these tasks in each respective use cases are analyzed.

5.2.2.1 Human detection and pose estimation (all use cases)

Recent advances demonstrate that deep convolutional neural networks and transformer-based models deliver state-of-the-art results in human motion capture from images. In SOPRANO, we set out to explore innovative approaches to address the task challenges and ensure online runtime performance, occlusion-tolerant performance in unstructured and complex workplaces, in indoor or outdoor environments. These include deep learning techniques for high-resolution representation learning, which integrate multiscale feature fusion through sequential high-to-low resolution sub-networks. This allows for the detection of fine-grained body movements as well as larger-scale motion, adapting to the varying size of the observed human body.

CRF Use Case: To this end, two key technologies have been tested and integrated in the SOPRANO CRF use case for human detection and pose estimation, namely OpenPose [23] and FORTH's MocapNet [165]. OpenPose is a deep learning-based method that extracts 2D joint coordinates from images, identifying key

body parts such as the head, arms, and legs. The set of 2D image coordinates and a confidence value derived by OpenPose (ranging between $[0,\ 100\%]$) per detected body joint is obtained as a result. The 2D skeleton output per frame is filtered according to the following heuristic approach: The 2D skeleton data is considered valid, if more than 50% of body joints provide a confidence score higher than 50%, and is forwarded to estimate the 3D body pose of the detected human subject, otherwise it is rejected as non-valid. The same heuristic filtering approach is followed on the 3D body pose outcome per frame. However, 2D body pose estimation alone is not sufficient for accurate analysis of the fine human postures and motion during work activities in a three-dimensional space, especially when considering human-robot interactions in a manufacturing environment. To address this, MocapNet2 is used to uplift 2D coordinates into a 3D skeletal model, providing a more accurate representation of human postures and movements in the workspace being also robust in the presence of extent (i.e. up to half of the human body) and long-term occlusions of the human body. MocapNet utilizes a compact 2D pose representation, convolutional neural networks, inverse kinematics, and hierarchical decomposition of the body into upper and lower segments. This approach allows for accurate regression of the 3D pose and global orientation of the full or partial human body, making it a strong candidate for our component. This combination of 2D and 3D analysis enables a more robust system to monitor human workers in dynamic industrial environments.

KUKA and E-TERRY Use Cases: To tackle this problem in the KUKA and E-TERRY use cases, we deployed a deep learning approach, the Yolo-v5 [80] pre-trained model that relies on CNNs (DenseNet) to learn and infer the 2D bounding box of multiple humans in RGB images. The Yolo-v5 model has been fine-tuned using real video data acquired by E-TERRY and annotation data that demonstrate real workers during use case-specific work activities/scenarios in the actual outdoor environment.

To achieve accurate 3D human pose estimation, the proposed pipeline employs a sequential, multi-stage architecture. Initially, a YOLOv5 object detection model is utilized to identify human instances within the camera's field of view and extract their corresponding 2D bounding boxes. Operating directly on these localized regions, 2D pose estimation is then performed using NVIDIA's trt-pose framework[29], configured with either a ResNet-18 or DenseNet-121 backbone. This model extracts 17 standard COCO anatomical keypoints for each detected individual. Illustrations of the pose estimation predictions are shown in Figure 38. To ensure near real-time processing capabilities, particularly on edge devices, the TRT-pose model is highly accelerated using TensorRT optimization tailored for NVIDIA Jetson GPUs. Finally, the extracted 2D skeletal keypoints are uplifted into a 3D coordinate space by mapping them against the corresponding spatial depth information acquired from aligned RGB-D sensor data, yielding the final 3D skeletal posture.

5.2.2.2 Action monitoring

In SOPRANO, the focus is on analyzing videos to enable online recognition of three specific types of human behavior: (i) Worker's Actions Monitoring: A predefined set of elementary actions performed in sequence during complex work activities to facilitate human-robot collaboration and estimate work/task progress during a work shift. (ii) Gestures for Commands: Gestures executed using upper-body or both-hand configurations to issue predefined robot or system commands. (iii) Dynamic Body Postures: Also referred to as postural events, these involve a limited sequence of prolonged, ergonomically suboptimal body configurations lasting a few seconds. These are analyzed to evaluate ergonomic risk and physical body stress during work activities.

In order to build efficient solutions for each type of human behavior that is analyzed, separate HMO subcomponents have been designed and implemented. The first component is described in this section, while the others are described in the following sections.

Despite their varied contexts, these three categories share overlapping motion patterns and can be characterized as actions of short duration (under a minute) and of low to medium complexity (based on subsequent body configurations). Recognizing such diverse human behaviors within a unified framework is a promising approach, and our goal is to develop a versatile deep learning-based architecture capable of inferring these

[29] https://github.com/NVIDIA-AI-IOT/trt_pose

target classes. To enhance model performance and reduce the domain gap, we plan to collect image/video datasets from real-world pilot environments. These datasets will showcase target action and gesture classes, allowing for the fine-tuning of models to capture nuanced variations in human behavior.

Our research explores efficient solutions for monitoring human actions and task progress by exploring the performance of Long Short-Term Memory (LSTM) and Transformer-based models to encode spatiotemporal representations of human motion patterns. Rich motion data are acquired representing 2D and 3D body poses and joint-based trajectories and serve as input to the models. Additionally, we incorporate semantic information about interacting objects (if applicable) and the temporal structure of work activities to improve recognition performance. An analysis of the developed approach for recognition of actions and estimation of task progress is provided in [156] using videos that demonstrate car door assembly activities in the CRF use case workplace.

The HMO also focuses on monitoring human actions during work activities with emphasis on task progress estimation during car door disassembly task cycle executions for the CRF use case. To realize this HMO sub-component, we leverage sequential models that can capture the temporal dynamics of the human body poses aided by the car door position in the 3D shop floor, as described in the following.

We implemented and evaluated three deep sequence models: a single-layer Long Short-Term Memory (LSTM) network where the final hidden state summarizes the input sequence and feeds into a fully-connected softmax classifier, a Bidirectional LSTM (BiLSTM) that processes the sequence both forward and backward in time, concatenating hidden states from both directions, and a Transformer encoder that utilizes multi-head self-attention to model relationships between all input segments. All models ingest the same multi-modal time-series data and aim to classify the current *subgoal* of the assembly task. Each model processes a sequence of synchronized measurements describing the human–object interaction over time, including the 3D location of the car door, the 3D position of the human operator, and the operator's full skeletal joint angles/positions. These are expressed in a consistent world coordinate frame. Car door positions are computed as centroids of segmentation masks detected via a fine-tuned Mask R-CNN using annotation data for the car door shape/masks in videos captured during CRF work activities. 3D human skeletal data are transformed into a body-centric coordinate system with the base spine joint as the origin. Measurements are grouped into ten non-overlapping temporal segments, each yielding a feature vector input to the models. All models are trained with cross-entropy loss using the Adam optimizer and a 70-10-20 train-validation-test split.

5.2.2.3 Gesture recognition

To tackle the task of gesture recognition required for the E-TERRY and KUKA use cases, a separate HMO sub-component has been designed and implemented by introducing a comprehensive vocabulary of worker gestures for online recognition by robotic systems in the KUKA and E-TERRY use cases. The list of supported gestures has been selected after evaluation in collaboration with IFADO and the KUKA and E-TERRY use case partners with the goal to establish seamless and fluent human-robot interaction during each use-case-specific scenario. The supported gesture set is as follows.

- Raise arm
- Arm pointing
- Open palm
- Closed palm/fist
- Thumbs up
- Thumbs down
- Both arms extended horizontally to the side

Each gesture is considered valid only if it is consistently recognized within a time window of at least one second. Any single-hand gesture may be executed using either hand. For gestural human–robot interaction, the HMO automatically designates the worker closest to the camera and its optical axis as the active user.

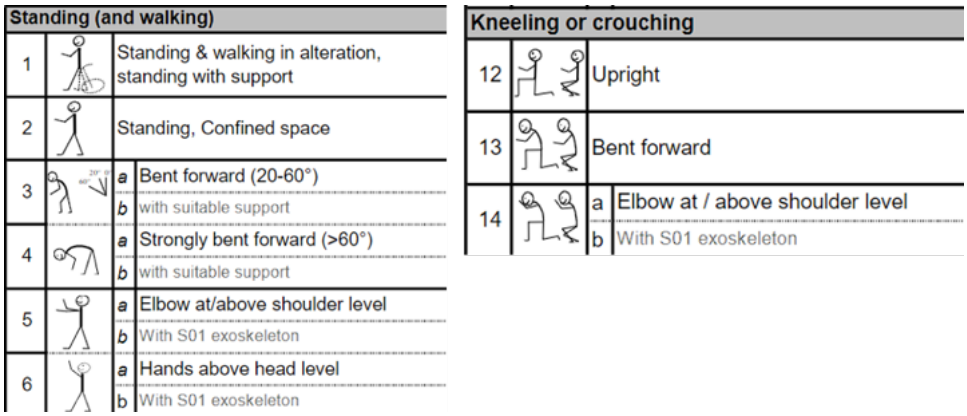


Figure 37: Ergonomic risk-inducing dynamic postures selected based on the EAWS tool.

We employ the established Google's Mediapipe framework [30] that relies on deep neural models to detect the 2D poses of human hands and recognize dynamic hand gestures in videos. The HMO sub-component for gesture recognition takes advantage of the RGB-D data acquired by the on-robot cameras for the two use cases to uplift the 2D hand poses to 3D and improve accuracy of the gesture recognition task. The sub-component is finally deployed on the ARM-based NVidia Jetson AGX platform embedded in the robot.

5.2.2.4 Recognition of ergonomic postures (CRF use case)

A primary objective of the framework developed in SOPRANO is to utilize the information derived from human poses and motion in 3D space to evaluate workers' postural performance in terms of physical ergonomics.

To achieve this goal, we formulate the vision-based task of efficient classification of ergonomically suboptimal postures—those considered awkward or physically harmful—during labor-intensive tasks. This is particularly relevant to the car door assembly process within the CRF use case, where repetitive and physically demanding activities are common. By identifying and evaluating these postures, the framework aims to provide indicators of ergonomic risks of physical stress and fatigue, which ultimately contribute to worker safety and well-being in industrial environments.

This task considers a small predefined vocabulary comprising different types of dynamic body postures deemed ergonomically suboptimal according to an ergonomic risk index/evaluation tool. In the literature, each suboptimal posture has a minimum duration of 4 seconds. In addition, a scoring scheme to evaluate the level of body stress induced by each postural type is considered as prior information/ground truth data. An ergonomic risk index is an assessment tool that involves an observational and graphical protocol or a checklist that associates the intensity, frequency, and duration of suboptimal working postures or other observations with the level of physical strain and the risk of musculoskeletal disorders (MSDs) as a single-valued score or set of scores. The European Assembly Worksheet (EAWS), the OCRA checklist, and MURI risk analysis are ergonomic assessment tools commonly used in the industry to design and optimize processes.

In SOPRANO, we focus on seven types of dynamic body postures based on the basic postures of the EAWS worksheet [127, 182, 181]. The EAWS tool is commonly employed in manufacturing to assess workers' physical workload and ergonomics. Each basic posture type is treated as a time-varying event, consisting of a sequence of body configurations with a minimum duration of 4 seconds. The EAWS tool uses a scoring/labeling scheme based on the occurrences and total duration of each posture type per minute throughout a work activity. Some of the basic postures refer to similar types of body configuration with higher levels of body strain. The selected EAWS-based posture types are listed next (using their original numbering/naming):

- 1. Standing & walking in alteration,

[30] https://github.com/google-ai-edge/mediapipe

- 2. Standing,
- 3. Bend forward,
- 4. Strongly bend forward,
- 5. Upright with elbows on or above the shoulders,
- 6. Upright with elbows above the head,
- "Trunk Rotation" and the "Lateral Trunk Bending" types from the Asymmetric postures.

Sketches of the selected postures are also included in Fig. 37. The set of dynamic ergonomic risk-inducing postures was selected based on the EAWS tool (Basic postures) for vision-based monitoring of physical ergonomics during car door assembly activities for the CRF use case (illustration of the postures originally appears in https://www.eaws.it/). Dynamic postures lasting at least 4 seconds are identified as valid, according to the EAWS instructions. Analysis of current research suggests that deep learning, specifically spatio-temporal GNNs, and transformer-based models, holds promise for tackling online recognition of dynamic body postures and predicting ergonomic risk scores in videos using sequences of 3D body poses as input. To achieve this goal, the approach in [155] is implemented and fine-tuned for the case of car door disassembly activities. The method formulates the evaluation of ergonomic work postures during manufacturing activities as a vision-based multi-class action detection and classification problem. To this end, given a video that demonstrates car door assembly activities, the sequence of 3D body poses is extracted for each frame. The 3D skeletal sequences are used to feed a spatiotemporal Graph Convolutional Network model (ST-GCN) [240], which has been trained to encode the input representations of the target working postures into a shared embedding/feature space. Finally, the pairwise temporal alignment cost between the embeddings of an unlabeled 3D skeletal sequence and those of a 3D skeletal sequence of a known class type is computed using the soft Dynamic Time Warping approach (softDTW) [34]. This pairwise cost is a similarity measure for classifying the unlabeled sequence among the target types of working posture. A sliding window scheme of 8 seconds, with an overlap of 4 seconds, is used to detect and classify the occurrences of any ergonomic posture classes.



Figure 38: Output of the 2D pose estimation for multiple workers in frames from the E-TERRY video dataset captured during agriculture (harvesting) activities for the E-TERRY use case.

5.2.2.5 Deployment and Communication

For the E-TERRY and KUKA use cases, the HMO component adopts a Docker-based containerized approach, facilitating seamless deployment across diverse hardware platforms and robotic systems.

Additionally, the HMO employs a Kafka broker-based communication framework to dispatch the computed outputs of all supported vision tasks to the broader SOPRANO system.

5.2.2.6 SOPRANO Open Call HMO implementation

In the context of the SOPRANO Open Call, an additional variant of the HMO perception component has been implemented [49] to support up to five beneficiaries (private companies) interested in deploying and advancing the component within their respective use cases. This new variant was developed because most OC beneficiaries/users needed both HMO sub-components to operate on embedded ARM-based platforms. Since the second sub-component was not originally designed for this environment, we re-implemented it for embedded systems and orchestrated a new vision-based pipeline.

To address these requirements, the HMO Open Call variant (HMO-OC) focuses on a targeted subset of functionalities:

- 2D/3D human detection and pose estimation
- EAWS-based ergonomic evaluation of postures

Furthermore, the HMO-OC accommodates diverse target hardware requirements by providing two distinct builds:

- An x86-64 platform-based version with support for various GPU families.
- An ARM64-based version optimized for NVIDIA Jetson AGX Xavier, Orin, and NX embedded platforms.

Both configurations support visual data acquisition using either Stereolabs ZED 1/2 or Intel RealSense camera sensors.

5.3 Experimental results

HMO is fully implemented and partially tested using data collected from the real operational environments of the CRF and the E-TERRY use cases so far. Three different datasets have been compiled and exploited to perform experimental evaluation quantitatively and qualitatively for the different HMO sub-components.

Firstly, the CarDA dataset is a multi-modal dataset for car door assembly activities and comprises a set of time-synchronized multi-camera RGB-D videos and motion capture data acquired during car door assembly activities performed by real line workers in the CRF car manufacturing environment [155, 156]. Rich annotation data concern temporal segmentation and semantics of sequential assembly actions for three different workflows of the car door production line and evaluation of physical ergonomics based on the EAWS method, as provided by manufacturing experts. The following sections provide a detailed description of the deployment environment and the data acquisition and annotation procedures followed to compile the CarDA dataset. The dataset is freely available online [31]. Thus, CarDA was used to assess the performance for 3D human pose estimation, ergonomic posture evaluation and monitoring of action/task progress in the CRF use case. Because dedicated disassembly annotations are not yet available at the same scale, CarDA is used as a proxy dataset for quantitative evaluation. This is justified by the similarity of body poses, motion patterns, and ergonomic postures between car-door assembly and disassembly activities, but final validation should use disassembly-specific data. The latter resembles car door assembly activities in terms of human poses, dynamic motion, and ergonomic postures required to achieve the goal of disassembly of car doors.

A video data set was constructed for visual human detection for the E-TERRY use case, noted hereby ETERRY-Detection dataset, consisting of 12 videos acquired in a challenging, real outdoor environment involving workers during harvesting activities. The data set provides semi-automatic annotation for 2D human body bounding

[31] https://zenodo.org/uploads/13370888

boxes and is utilized to assess the human detection performance of the HMO component, as shown in Fig. 39 and Fig. 38. The dataset includes up to 5.000 annotations of 2D human body bounding boxes with confidence values based on 1.162 images/frames.

Furthermore, a preliminary video dataset was developed for gesture recognition, noted HMO-GR, comprising five videos of hand and arm gesture classes relevant to the E-TERRY and KUKA use cases. Data were captured and annotated under controlled laboratory conditions to evaluate the performance of the HMO gesture recognition module.

5.3.1 Human Detection

The human detection performance of the HMO component was evaluated in two stages. This involved using the E-TERRY-HD video dataset, followed by field validation data collected by the E-TERRY partner during initial prototype testing.

During the first stage, the deep learning-based human detector (YOLOv5)—fine-tuned specifically for the estimation of the 2D human bounding box estimation—was assessed under controlled benchmark conditions. Ground-truth annotation data for human detection comprise 12 video sequences from the E-TERRY dataset, as illustrated in Fig. 38 and Fig. 39.

The human detector was quantitatively evaluated using established object detection performance indicators: Mean Average Precision (mAP), Precision and Recall. The detailed results of this evaluation are presented in Table 8 shown in Section 3.3.2. The obtained $\mathbf{mAP@0.5}$, **Precision and Recall** metric scores for 2D human detection are **95.7%, 96%, & 91.2%**, respectively. Together, these metrics provide complementary evidence on the detector's ability to accurately locate human instances while maintaining high detection completeness in all evaluated sequences.

In the second stage, and within the broader evaluation activities of the SOPRANO components, the E-TERRY partner independently assessed the performance of the deployed HMO component under realistic operational conditions, as reported in Deliverable D5.3. The outcome of this initial evaluation phase is summarized below. During this initial prototype evaluation phase, a representative sample of 100 video frames acquired during field operation was manually annotated by E-TERRY personnel, with bounding boxes corresponding to all visible workers present in the scene. The detections produced by the deployed perception pipeline were subsequently compared with the reference annotations on a frame-by-frame basis. Detection effectiveness was assessed using the **Recall** metric, defined as the proportion of correctly identified human instances compared to the total number of ground-truth annotations. Additionally, localization performance was evaluated using a strict **Intersection-over-Union (IoU)** threshold of $\mathbf{80\%}$, ensuring that only detections with sufficiently precise spatial overlap are considered valid. Based on this combined detection and localization criterion, the **Human Detection module achieved an overall success rate of** $\mathbf{95\%}$. This result provides evidence on the robustness and operational reliability of the perception stack under the realistic deployment conditions encountered during initial prototype evaluation.

Validation activities involving additional multi-worker scenes and use-case-specific scenarios, with increased inter-person occlusions, such as those that appear in the ETERRY-Detection video dataset are foreseen for the final prototype phase of the platform in order to further assess the scalability and resilience of the Human Detection module under realistic operational conditions.

5.3.2 Human body pose estimation

Human pose estimation relies on human body detection predictions (i.e., 2D human body bounding boxes and masks), as described above. Real-time human pose estimation is achieved by combining two state-of-the-art methods to predict the 2D and, subsequently, the 3D body configurations for each frame, leveraging known 2D and 3D skeletal body models. Given the predicted 2D human body masks per color image, the



Figure 39: Output of the 2D human detection for multiple workers in frames from the ETERRY-Detection video dataset captured during agriculture (harvesting) activities for the E-TERRY use case.

OpenPose deep learning-based framework [23] is initially applied to estimate the 2D body joint coordinates for up to 25 body joints per detected masked individual within the image plane. Once the 2D joint coordinates are obtained for each human, they are given as input to the MocapNet2 method [165] for 3D localization and pose estimation (15 body joint angles and positions) with respect to the camera coordinate system. For offline video sequences, MocapNet2 outputs results in the Biovision Hierarchy (BVH) format [132], widely used in character animation. This format facilitates direct comparisons between predicted 3D joint positions and ground-truth motion capture data.

To further enhance the accuracy of the 3D pose predictions, variations and optimizations of the aforementioned 2D/3D pose estimation pipeline have been explored during the SOPRANO project, by employing data acquired by multiple cameras as in the CRF use case or fine-tuning the deep-learning models using use-case-specific datasets, such as the CarDA [155].

We estimate the performance of the proposed approach based on accuracy and precision scores, as shown in Table 14. Overall, the proposed approach can **identify 3D poses with 76.7% accuracy and 68.27% mAP scores**. Table. 14 summarizes the results, where the average MPJPE [217] over all joints and all samples of the CarDA subset is below 140 mm and 68.8% of the 3D joints are considered successfully detected throughout all data sequences based on the PCK@150 metric [242], for the case of using a single stereo camera and triangulation to enhance 3D pose estimation. A noticeable performance gain of more than 60 mm and 17%, respectively, is achieved when using visual data from two different stereo cameras installed in the workplace, at the side and at opposite locations with respect to the working area to estimate more accurate and stable 3D human poses using also the Iterative Closest Point (ICP) algorithm for the 3D alignment of individual 3D poses estimated by each stereo camera.

Figures 40 and 41 illustrate the predicted 3D body pose of a single worker on the CRF work place. The images shown were acquired from the stationary camera installed on the side of the production line at the CRF workspace. The estimated 3D skeleton-based body pose is rendered (slightly translated to the right of the worker's body for better clarity) using blue and green colors for the limbs and joints of the human body, respectively.

Qualitative results for the estimation of 2D human pose for the E-TERRY use case are also shown in Fig. 38, where the poses of multiple humans are predicted during different work activities and body configurations in the real work environment. The 2D pose estimation relies on the fine-tuned Yolo-v5 model using the ETERRY-Detection dataset. No ground truth data for 2D or 3D human body poses is available for the E-TERRY videos.



Figure 40: Output of the 3D pose estimation for a single worker at a close-to-mid distance using MocapNet2 [165]. The reconstructed 3D skeletons—visualized with green markers for body joints and blue segments for limbs—are horizontally offset relative to the observed subject to enhance visual clarity.



Figure 41: Output of the 3D pose estimation for a single worker at a mid-to-far distance using MocapNet2 [165]. The 3D skeletons are again visualized next to the workers for clarity.

5.3.3 Ergonomic posture evaluation

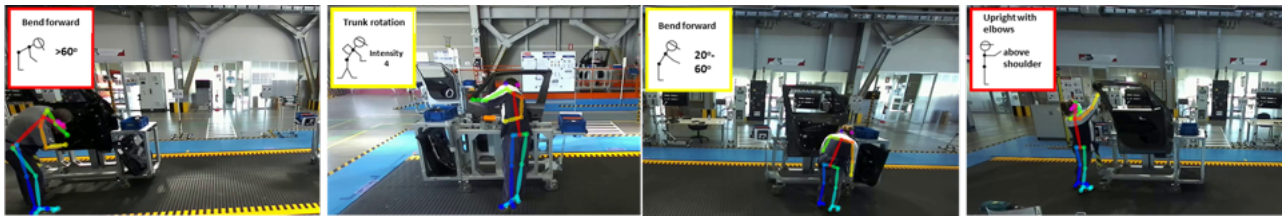


Figure 42: Visual classification and ergonomic evaluation of EAWS-based body postures during car door assembly activities.

We formulate the evaluation of ergonomic risk-inducing postures during work activities as a vision-based multi-class action recognition problem. To this end, given a video that demonstrates a sub-goal during a vehicle door task cycle for a specific workstation, we extract the sequence of 3D body poses using the MocapNet2 method for each frame. The set of basic EAWS-based postures of interest is illustrated in Fig. 37. To achieve

Classification of risk-inducing human postures (EAWS)	CarDA dataset	
	Accuracy	Precision
Single stereo camera (stereo triangulation, heuristic)	76.67	68.27
Multi-view (stereo triangulation, ICP, heuristic)	**84.36**	79.54

Table 13: Experimental results for the vision-based classification of EAWS-based ergonomic work postures (as described in Sec. 5.2.2.4) using the Car Door Assembly (CarDA) dataset [156]. The mean accuracy and mean average precision (mAP) are presented as a percentage %.

Estimation of 3D body poses	CarDA dataset	
	MJPE	PCK@150
Single stereo camera (stereo triangulation)	138.2	68.8
Multi-view (stereo triangulation, ICP)	**65.75**	86.58

Table 14: Experimental results for 3D pose estimation using the MocapNet2 method [164]. MPJPE (Mean Per Joint Position Error). PCK@150 (Percentage of Correct Key-points metric between 3D poses using the commonly used threshold of 150mm). The average scores for pairs of symmetric body joints are reported, e.g., hands.

this goal, we adapted and re-trained the previously proposed approach in [158] for detecting and classifying EAWS-based postures throughout a video. Qualitative experimental results are shown in Fig. 42. The estimated 3D human body pose and the type and risk level (i.e., severity) of ergonomic risk-inducing posture are annotated in each image. The approach proposed in [155] was utilized using the estimations of dynamic 3D poses extracted from video clips during simulated assembly activities in the CRF workplace.

Quantitative analysis is reported in Table 13. Overall, the proposed approach can **identify the correct class of postures performed by line workers with 84.36% accuracy and 79.54% mAP scores for the multi-view configuration**. Given the challenging conditions captured by the CarDA dataset in a real environment of simulated manufacturing activities, the results indicate a promising potential for the proposed approach in real-world applications.

5.3.4 Gesture recognition

Within the scope of the HMO evaluation activities, a dedicated gesture recognition video dataset (HMO-GR) was collected in a controlled indoor laboratory setting. The dataset serves as a validation benchmark for assessing the robustness and recognition accuracy of the developed gesture recognition pipeline. The dataset includes five video sequences of approximately 45 seconds average duration, featuring a single actor performing overall 50 gesture instances across the supported gesture vocabulary. Both single-arm/hand gestures and simultaneous two-arm/two-hand gestures are included, thereby covering both unilateral and bilateral gesture patterns relevant to the targeted scenarios for the KUKA and the E-TERRY use cases. Qualitative results for gesture recognition are shown in Figure 43, where gestures using one or both hands are performed by a single user in the range of 1.5 or 2.5 meters from the camera in an indoor lab setting. Quantitative results of the evaluation are reported as a confusion matrix in Figure 44, which indicates that the accuracy of the HMO sub-component exceeds 92%. A richer video dataset and annotations concerning gestures performed by real workers in the real indoor or outdoor use case specific workplaces will be needed to fine-tune the gesture recognition model and further advance its accuracy performance in view of the final prototype tests.

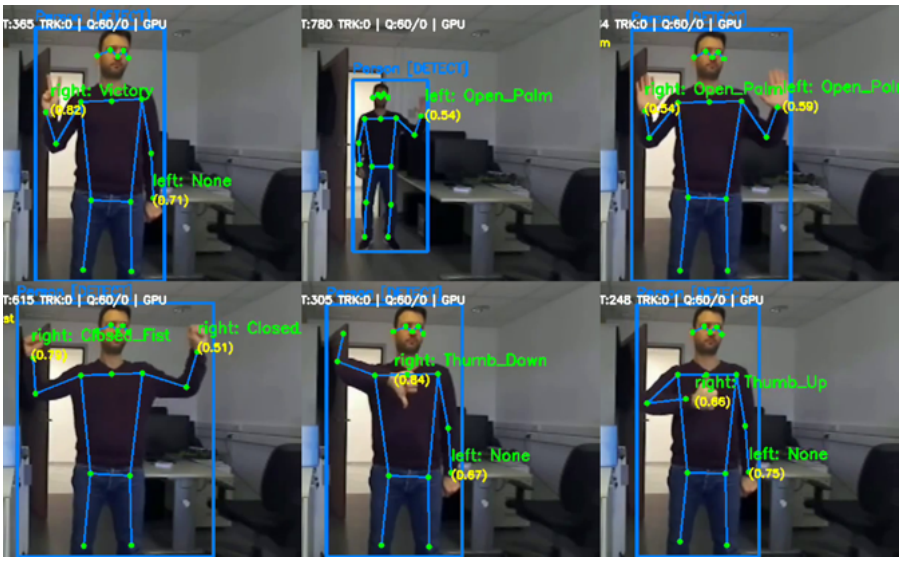


Figure 43: Visual recognition/classification of gestures using the HMO-GR video dataset captured in an indoor lab setting.

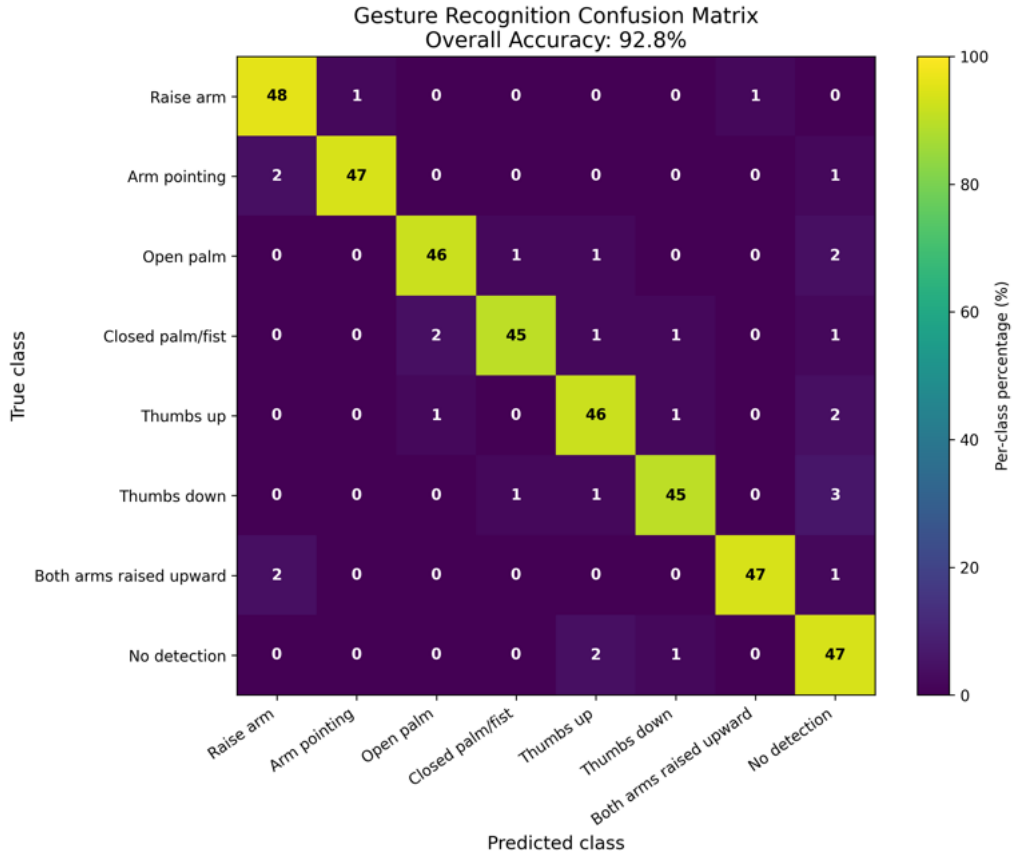


Figure 44: Visual classification of gestures in videos of the (HMO-GR dataset) captured in an indoor, lab setting.

5.3.5 Monitoring of action / task progress

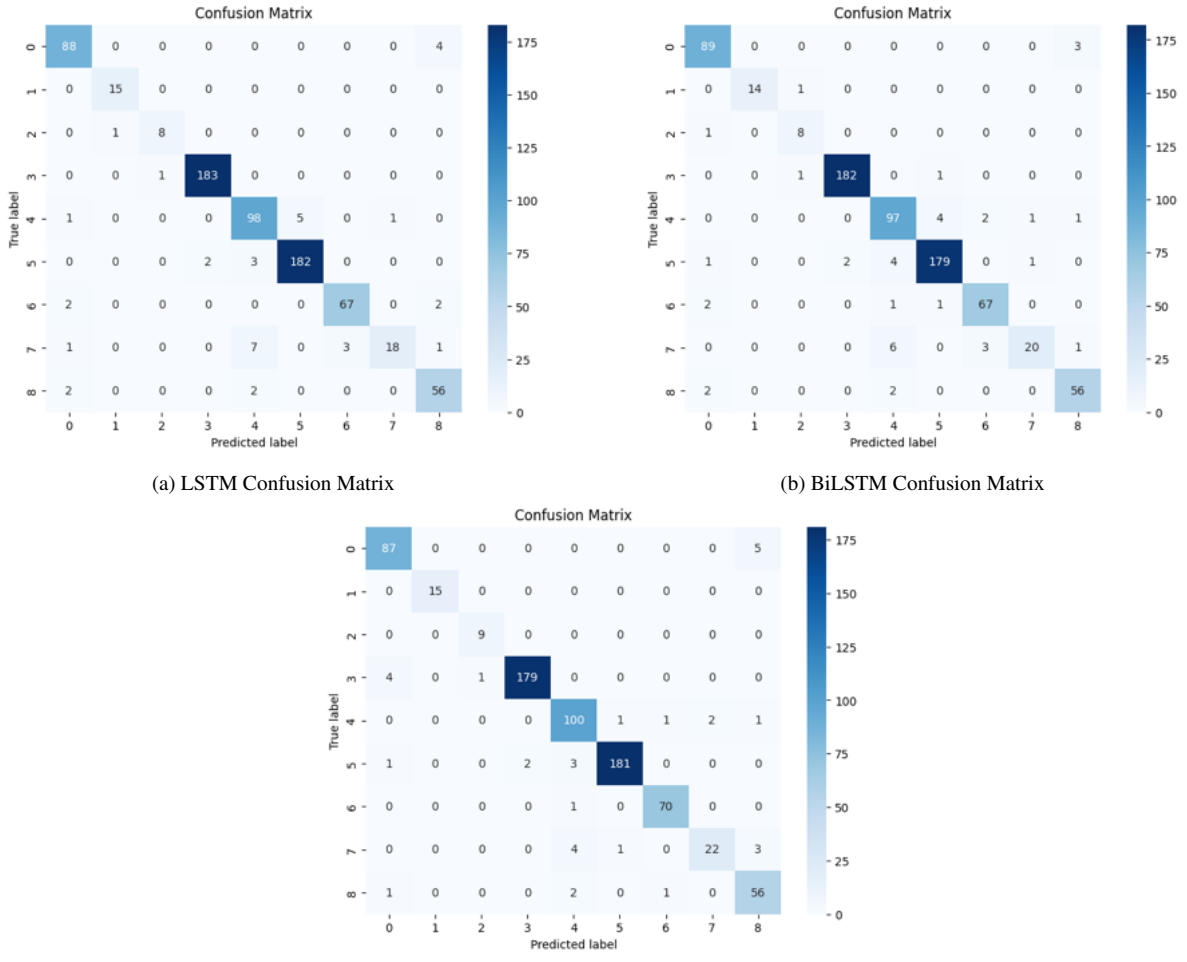
The performance of the action / subgoal classification of the models was evaluated using validation Recall, F1-score, and macro-average ROC AUC (Area Under the Receiver Operating Characteristic Curve). Table 15 summarizes the comparative results.

Table 15: Validation performance of sequential and graph-based models for action/sub-goal classification

Model	Recall (%)	F1-score (%)	Macro AUC
LSTM	91.4	91.8	0.996
Bidirectional LSTM	92.0	92.5	0.996
Transformer Encoder	94.3	93.5	0.969

The confusion matrix analysis in Fig. 45 revealed that:

- LSTM, BiLSTM and Transformer models exhibit strong diagonal dominance, indicating strong classification performance.
- Minor confusion persisted between visually or temporally similar subgoals, such as subgoal 7 and subgoal 4.



(a) LSTM Confusion Matrix

(b) BiLSTM Confusion Matrix

(c) Transformer Confusion Matrix

Figure 45: Confusion matrices for assembly action/sub-goal classification using different sequential models. The overall F1-score performance of the models is 91.8%, 92.5% and 93.5%, respectively, as reported in Table 15.

The Transformer encoder achieved the best Recall and F1-score, while LSTM and BiLSTM achieved slightly higher Macro AUC. Overall, all three models demonstrate strong discriminative performance, with minor confusion between visually or temporally similar sub-goals.

5.4 HMO runtime performance assessment

A quantitative runtime performance assessment was conducted for the Human Monitoring Component (HMO) across its primary deployment scenarios. The evaluation focuses on the CRF use case, which represents the most computationally demanding pipeline within the framework as well as the KUKA and ETERRY use cases that target an embedded platform for GPU accelerated edge computing.

Table 16: Execution times of the main pipeline components for the CRF use case (x64-based).

Component	**Avg Time per frame (ms)** / FPS (approx.)
RGB-D Video Acquisition (StereoLabs SDK)	33 ms / 30
Human Detection (Mask R-CNN, R50-FPN)	150 ms / 12
2D Pose Estimation (OpenPose, Model25) [23]	40 ms / 30
3D Pose Estimation (MocapNet2) [165]	10 ms / 150
Full pipeline (end-to-end)	**258 ms / 4**

5.4.1 High-end workstation runtime performance (CRF)

Table 16 details the processing latency per frame for each HMO sub-component on a high-end x64 workstation (Intel Core i9, 32 GB RAM, and NVIDIA RTX A4000 with 16 GB VRAM). This on-site configuration achieves an end-to-end processing rate of 4 fps, satisfying the "on-line" performance requirements for the CRF integrated framework. For the CRF use case, 4 fps is considered sufficient because the monitored assembly/disassembly actions evolve over several seconds and do not require frame-level intervention.

5.4.2 Embedded platform runtime performance (E-TERRY and KUKA)

The runtime performance for the E-TERRY and KUKA use cases, deployed on the NVIDIA Jetson Orin AGX embedded platform, is summarized in Table 17. Data acquisition relies on the StereoLabs SDK when using ZED RGB-D sensor (E-TERRY) or the Intel SDK for the Realsense RGB-D sensor (KUKA). Utilizing TensorRT-optimized models for human Pose Estimation based on the TRT-Pose library, the system achieves online execution rates of 9 fps with the ResNet18 backbone and 6.6 fps with the DenseNet121 backbone. The primary computational bottleneck remains the MediaPipe Gesture Recognition module. Due to current technical constraints in the implementation, this sub-component does not leverage GPU acceleration, limiting the overall throughput on the embedded hardware. Overall, the HMO pipeline for embedded reaches $6.6 to 9$ fps, supporting online gesture and pose monitoring under the evaluated conditions.

5.5 Discussion and upcoming work

Human monitoring in the SOPRANO project has been validated offline through the optimization of deep learning models and the assessment of sub-component inference performance using diverse datasets across all use cases. Those findings along with the outcomes of the initial prototype deployment and its subsequent evaluation phase confirm the following claims.

For the E-TERRY use case, the HMO pipeline—which includes 2D/3D human detection and tracking—has been demonstrated in relevant simulated environments. Quantitative experiments have validated the performance of these sub-components, achieving TRL 6. Gesture recognition sub-component was not part of the initial prototype.

Table 17: Execution times of the main pipeline components for the KUKA and E-TERRY use cases (ARM platform NVidia Jetson Orin AGX). Input RGB-D data acquisition is set to 15 fps and HD resolution. Visual data acquisition timing is included in that of the pose estimation task.

TRT-Pose model	**Task**	**Avg. latency (ms)**	**FPS**
StereoLabs or Intel SDK	RGB-D Video Acquisition	18	15.0
ResNet18 (default)	2D/3D Pose Estimation (GPU)	23.5	29.8
	Gesture Recognition (CPU)	70.2	12.8
	Full pipeline (end-to-end)	**111.7**	**9.0**
DenseNet121	2D/3D Pose Estimation (GPU)	55.4	14.6
	Gesture Recognition (CPU)	73.60	12.2
	Full pipeline (end-to-end)	**150.03**	**6.6**

Online runtime performance and high accuracy for the CRF HMO pipeline have been verified within its real-world operational environment. This pipeline integrates vision tasks for 2D human detection, 3D pose estimation, and the estimation of actions and task progress during industrial activities. Consequently, this use case specific HMO pipeline has also reached TRL 7, while the sub-component for estimating work actions and task progress reached TRL 6.

In the final stages of the SOPRANO project, the optimized HMO prototype will be deployed and demonstrated in real-world operational environments for all use cases. This final phase aims to bring all use-case-specific pipelines and all their respective sub-components to TRL 7.

Upcoming work for the HMO component will focus on improving seamless human-robot interaction and robust model training through the following key activities:

- Gesture vocabulary compilation and data acquisition: Capturing video data for the selected comprehensive vocabulary of worker gestures intended for online recognition by robotic systems in the KUKA and E-TERRY use cases.
- Final HMO dataset publication: Compilation of a new video dataset that captures these target gestures and other work activities performed by diverse users. The comprehensive data set will consolidate all image, video, and annotation data utilized to train HMO models in all challenging visual tasks and operational environments.

6 Conclusions

The deliverable has summarized the developments in SOPRANO's perception components. Sections 2 – 5 present the contributions of four components: Visual Spatial Localization and Mapping (VLM), Object Perception (OBP), Non-Visual Localization (NVL), and Human Monitoring (HMO). Each respective section addresses the specific challenges posed by complex and dynamic industrial scenarios, focusing on improving the adaptability, precision, and safety of robotic operations. Together, these advancements form the foundation for achieving seamless human-robot collaboration, robust navigation, and reliable object manipulation in challenging settings. A brief summary for each component is given below.

The Visual Spatial Localization and Mapping (VLM) component builds on state-of-the-art structure-from-motion techniques to provide accurate and robust camera pose estimation. It is implemented as a dual-system framework, with two complementary approaches tailored to different environmental conditions and leveraging RGB-D data modalities. This ensures reliable operation across a wide range of real-world scenarios, from richly textured environments to feature-sparse scenes. The first approach targets environments with sufficient visual structure, where natural image features can be reliably detected and matched across frames. In these conditions, a standard feature-based visual localization pipeline is used to achieve accurate pose estimation by exploiting robust feature correspondences. The second approach addresses feature-poor environments where appearance-based methods degrade due to a lack of distinctive visual cues. In this case, depth-derived geometric information from RGB-D sensors is used to constrain motion estimation and enable robust localization.

The Object Perception (OBP) component is dedicated to real-time object detection and 6D pose estimation, enabling robotic manipulation tasks in industrial and agricultural settings. By employing deep learning methods, the OBP system tackles challenges such as occlusions, and weak textures. Automated training data generation enhanced scalability and adaptability across use cases. The system's confidence scoring mechanism ensures reliable robotic actions, making it suitable for precise robotic operations like grasping. Extensive evaluation of the component across the SOPRANO use cases showcased its capabilities in reliable object detection and pose estimation and integration in diverse industrial and agricultural applications.

The NVL component has reached a stable implementation stage within SOPRANO, with its most mature branch corresponding to the KUKA use case, where a complete LiDAR-based localization stack has been developed, deployed and validated for both SLAM and map-based localization. This branch supports downstream functionalities (e.g. navigation) and forms the core operational capability of the current NVL solution. At the same time, additional non-visual sensing branches have been advancing in parallel, including an ultrasonic/radar-based localization subsystem for KUKA and dedicated CRF solutions for tool presence detection and BLE-based localization.

The Human Monitoring (HMO) component supports human-robot collaboration (HRC) by enabling real-time monitoring of 2D/3D human poses, along with the identification of physical ergonomic risks, recognition of worker activities and task progress during task cycle execution, as well as human gestures recognition toward fluent HRI. Advanced deep learning models are utilized for detailed recognition of the body and motion of multiple humans, allowing EAWS-based physical ergonomic risk assessment to improve workplace safety. The component also facilitates monitoring of human-robot interactions, supporting adaptive and context-aware safety measures. Different hardware platforms and visual sensors are supported across all SOPRANO use cases and multiple beneficiaries for the SOPRANO Open Call, demonstrating accurate performance in real unstructured work environments with diverse conditions and severe human body occlusions. By addressing both efficiency and safety in collaborative environments, the HMO component is pivotal in supporting the broader objectives of SOPRANO for industrial robotics.

Appendix A that follows summarizes the main advances achieved by each component relative to its early prototype.

A Updates to Components Since D2.1

This appendix summarizes the main advances for each of the perceptual components with respect to their early prototypes presented in D2.1 - Industrial Context Sensing I that was submitted in M15.

A.1 Visual Spatial Localization and Mapping (T2.1)

Notable enhancements introduced to the feature-based vSLAM variant of VLM since M15 include:

- Included the BAD (aka TEBLID) binary descriptor.
- Additional improvements to the parallel symmetric positive definite (SPD) solvers in bundle adjustment (BA).
- Accelerated frame-to-frame pose tracking with custom G-N & L-M implementations.
- Accelerated rigid transformation of point sets combining multi-threading and SIMD.
- Added support for directly capturing data from RealSense cameras via Intel's SDK.
- Added multiple RealSense device support (serial number identification) and manual exposure control.
- Implemented on-demand publishing of depth and 3D vSLAM maps as ROS 2 point clouds.
- Added support for externally provided vSLAM to world alignment.
- Added support for user-customizable visualization overlays.
- Migrated the middleware wrapper from ROS 1 to ROS 2.
- Revamped docker containers for both x86-64 and ARM64 architectures.
- Experimented with feature selection techniques.
- Bug fixes and codebase tidy-up.
- Several improvements concerning efficiency, data-management, and code clarity:
    - Implemented map serialization and loading
    - Improved pose prediction via rational regression
    - Improved handling of outliers for local BA
    - Introduced on-request premature termination for BA optimization
    - Accelerated non-linear least squares via parallel SPD solvers (Cholesky/LDLt) in BA
    - Improved keypoint feature tracking

Regarding the plane-based variant of VLM, recent upgrades focus on the following:

- Developed a disambiguation scheme for pose estimates via a consensus test based on majority voting
- Added automatic, sampling-based plane detection in depth images
- Developed a graphical interface for intuitive plane labeling
- Integrated line-based pose estimation using interactive line refinement and Perspective-n-Line (PnL)
- Added live (i.e., RealSense) and file-based image input support
- Developed a ROS 2 wrapper for interfacing and communication with external nodes
- Experimented with depth denoising
- Containerized code and ported to the ARM64 architecture
- Several refactoring updates and bug fixes

A.2 Object Perception (T2.2)

- The IndustryShapes dataset was compiled for 6D object pose estimation of industrial tools and components involved in grasping tasks, supporting both instance-based and novel-object methods.
- Developed, extended and annotated dedicated datasets for screw detection and outdoor agricultural object detection.

- In D2.1, the performance comparison of 6D pose estimation methods included representative instance-based methods. D2.3 was updated to consider novel-object 6D pose estimation techniques, focusing on model-based and model-free methods capable of generalizing to previously unseen industrial objects.
- Building on the novel confidence scoring method described in D2.1, two novel methods for shape representation were developed. The approaches leverage Gaussian Process priors and distance fields encoded via Spherical Harmonics to generate lightweight, explicit representations of complex objects, in order to support fast and reliable pose quality estimation.
- Finalized the CRF OBP pipeline for screw detection and quantitatively evaluated its performance using standard detection metrics on a custom RGB-D dataset with manually annotated screw instances collected in the CRF environment.
- To address the diverse use case scenarios and extend the component's applicability to workflows outside of SOPRANO, variations of the OBP-6D and OBP-3D subcomponents have been implemented that support different hardware and software setups.
- Distributed variants of OBP-6D and OBP-3D are under development for integration with the AI Optimizer component and deployment in the CRF and E-TERRY use cases, respectively.

A.3 Non-Visual Localization (T2.3)

- Added support for a dedicated map-based localization mode using AMCL, alongside the existing online LiDAR-based SLAM workflow. This enables operation against a prior occupancy-grid map in addition to map building.
- Introduced a LiDAR scan filtering stage to suppress self-returns from the robot body and mounting structures before downstream processing. This improves the robustness of LiDAR-based odometry, SLAM, and localization.
- Developed a dedicated extrinsic calibration procedure for the dual-LiDAR setup, estimating the relative 2D rigid transform between the two sensors from corresponding geometric features.
- Updated the simulation framework to better reflect the actual LiDAR payload, while retaining Gazebo as a controlled environment for development, validation, and tuning.
- Expanded the ROS2 deployment from a mainly SLAM-oriented stack to a fuller architecture covering sensor interfaces, static transforms, odometry, SLAM mode, and localization mode.
- Refined LiDAR-only odometry (RF2O) and ranked it as a critical performance factor, sensitive to environment structure, dynamic objects, and calibration inaccuracies.
- Added real-world quantitative evaluation against reference trajectories provided by SfM/VLM, reporting ATE RMSE below 20 cm for the tested sequences.

A.4 Human Monitoring (T2.4)

- Implemented a gesture recognition methodology and interface for the KUKA and E-TERRY UCs based on the MediaPipe library and additional customized gestural events.
- Implemented estimation and 3D body tracking of the closest worker for the KUKA and E-TERRY UCs.
- Implemented a methodology for the recognition of car door disassembly actions and estimation of work progress during task cycles.
- Revamped the Docker HMO containers for the final prototype versions of the KUKA, E-TERRY and Open Call variants.
- Improved vision-based recognition and evaluation of physical ergonomic postures based on the EAWS evaluation tool for the CRF use case.
- Revamped the HMO component for the Open Call adding new methodology for vision-based 3D human pose estimation and evaluation of physical ergonomics based on the EAWS approach to improve the ergonomic design of a workplace. Added support for i) multiple imaging sensors (StereoLabs ZED and

Intel Realsense camera families) for data acquisition and ii) GPU-powered computing based on edge ARM-based (NVidia Jetson AGX platform) and workstation x86-64 architectures.

- Acceleration of the HMO runtime performance for the CRF use case based on several improvements: implemented matrix inversion and caching, sped-up Detectron's 2 inference time for human mask detection, added multi-threading for separating visual data acquisition and vision-tasks computing, added tracking and skip-frame detection mechanisms for multiple human localization.
- Codebase tidy-up.

References

[1] Ilya Afanasyev, Artur Sagitov, and Evgeni Magid. ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment. In *Advanced Concepts for Intelligent Vision Systems*, pages 273–283, Cham, 2015. Springer International Publishing.

[2] Dustin Aganian, Benedict Stephan, Markus Eisenbach, Corinna Stretz, and Horst-Michael Gross. Attach dataset: Annotated two-handed assembly actions for human action understanding. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11367–11373, 2023.

[3] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver. `https://github.com/ceres-solver/ceres-solver`, 10 2023.

[4] Mary B. Alatise and Gerhard P. Hancke. Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter. *Sensors*, 17(10), 2017.

[5] Guillem Alenyà, Sergi Foix, and Carme Torras. Using ToF and RGBD cameras for 3D robot perception and manipulation in human environments. *Intelligent Service Robotics*, 7(4):211–220, 2014.

[6] Ershat Arkin, Nurbiya Yadikar, Xuebin Xu, Alimjan Aysa, and Kurban Ubul. A survey: object detection methods from cnn to transformer. *Multimedia Tools and Applications*, 82(14):21353–21383, 2023.

[7] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D Surface Reconstruction. In *CVPR*, 2022.

[8] Rana Azzam, Tarek Taha, Shoudong Huang, and Yahya Zweiri. Feature-based visual simultaneous localization and mapping: a survey. *SN Applied Sciences*, 2(2):224, 2020.

[9] Konstantinos Bacharidis and Antonis Argyros. Extracting action hierarchies from action labels and their use in deep action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 339–346. IEEE, 2021.

[10] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018.

[11] Daniel Barath and Jiri Matas. Graph-cut ransac, 2017.

[12] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021.

[13] 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 (WACV)*, pages 847–859, January 2021.

[14] Peter Biber and Wolfgang Straßer. The normal distributions transform: a new approach to laser scan matching. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 3:2743–2748 vol.3, 2003.

[15] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2023.

[16] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[17] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sebastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, September 2017.

[18] Mihai Bujanca et al. SLAMBench 3.0: Systematic automated reproducible evaluation of SLAM systems for robot vision challenges and scene understanding. In *International Conference on Robotics and Automation (ICRA)*, pages 6351–6358, 2019.

[19] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[20] Munkhtulga Byambaa, Gou Koutaki, and Lodoiravsal Choimaa. 6D pose estimation of transparent objects using synthetic data. In *International Workshop on Frontiers of Computer Vision*, pages 3–17. Springer, 2022.

[21] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[22] Carlos Campos, Richard Elvira, Juan J. Gómez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[23] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2019.

[24] Bruno Caprile and Vincent Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[25] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[26] Hanzhi Chen, Fabian Manhardt, Nassir Navab, and Benjamin Busam. Texpose: Neural texture learning for self-supervised 6d object pose estimation, 2023.

[27] 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.

[28] X. Chen, H. Guo, G. Wang, and L. Zhang. Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2881–2885. IEEE, 2017.

[29] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings International Conference on Robotics and Automation*, pages 2724–2729, 1991.

[30] Zeren Chen, Gengshi Huang, Wei Li, Jianing Teng, Kun Wang, Jing Shao, Chen Change Loy, and Lu Sheng. Siamese detr. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15722–15731, 2023.

[31] Won-Seok Choi, Yang-Shin Kim, Se-Young Oh, and Jeihun Lee. Fast iterative closest point framework for 3d lidar data in intelligent vehicle. *2012 IEEE Intelligent Vehicles Symposium*, pages 1029–1034, 2012.

[32] Grazia Cicirelli, Roberto Marani, Laura Romeo, Manuel García Domínguez, Jónathan Heras, Anna G Perri, and Tiziana D'Orazio. The ha4m dataset: Multi-modal monitoring of an assembly task for human action recognition in manufacturing. *Scientific Data*, 9(1):745, 2022.

[33] Frincy Clement, Kirtan Shah, and Dhara Pancholi. A review of methods for textureless object recognition. *arXiv preprint arXiv:1910.14255*, 2019.

[34] Marco Cuturi and Mathieu Blondel. Soft-dtw: A differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 894–903. JMLR.org, 2017.

[35] Jan Czarnowski et al. DeepFactors: Real-time probabilistic dense monocular SLAM. *IEEE Robotics and Automation Letters*, 5(2):2561–2568, 2020.

[36] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1403–1410, 2003.

[37] G. Devineau, F. Moutarde, W. Xi, and J. Yang. Deep learning for hand gesture recognition on skeletal data. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 106–113. IEEE, 2018.

[38] Gerard Donahue and Ehsan Elhamifar. Learning to predict activity progress by self-supervised video alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18667–18677, June 2024.

[39] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6D object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3583–3592, 2016.

[40] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing MVTec ITODD - a dataset for 3D object recognition in industry. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 2200–2208, 2017.

[41] Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review*, pages 1–58, 2020.

[42] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30(1):177–187, 2014.

[43] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018.

[44] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Computer Vision – ECCV 2014*, pages 834–849, Cham, 2014. Springer International Publishing.

[45] Zhiwen Fan, Panwang Pan, Peihao Wang, Yifan Jiang, Dejia Xu, Hanwen Jiang, and Zhangyang Wang. Pope: 6-dof promptable pose estimation of any object, in any scene, with one reference. *arXiv preprint arXiv:2305.15727*, 2023.

[46] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019.

[47] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[48] Gergely Fodor and Csaba Viktor. Range feature-based loop closure detection for visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10021–10028. IEEE, 2023.

[49] Foundation for Research & Technology – Hellas. Human Monitoring Component Technical Cookbook. SOPRANO GitHub repository, 2025. V1.1, 13 Oct. 2025.

[50] Foundation for Research & Technology – Hellas. Visual Localization and Mapping Component Technical Cookbook. SOPRANO GitHub repository, 2025. V0.3, 15 Sep. 2025.

[51] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 343–349, 01 1999.

[52] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2018.

[53] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[54] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.

[55] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[56] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8359–8367, 2018.

[57] Walter Goodwin, Sagar Vaze, Ioannis Havoutis, and Ingmar Posner. Zero-shot category-level object pose estimation. In *European Conference on Computer Vision*, pages 516–532. Springer, 2022.

[58] Michael Grupp. Python package for the evaluation of odometry and SLAM. `https://github.com/MichaelGrupp/evo`, 2017. Accessed: 17-12-2024.

[59] Lin Guo, Zongxing Lu, and Ligang Yao. Human-machine interaction sensing technology based on hand gesture recognition: A review. *IEEE Transactions on Human-Machine Systems*, 51(4):300–309, 2021.

[60] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for RGB-D visual odometry, 3d reconstruction and SLAM. In *2014 IEEE international conference on Robotics and automation (ICRA)*, pages 1524–1531. IEEE, 2014.

[61] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[62] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. Onepose++: Keypoint-free one-shot object pose estimation without cad models. *Advances in Neural Information Processing Systems*, 35:35103–35115, 2022.

[63] Yisheng He, Yao Wang, Haoqiang Fan, Jian Sun, and Qifeng Chen. FS6D: Few-shot 6d pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6814–6824, 2022.

[64] Zaixing He, Wuxi Feng, Xinyue Zhao, and Yongfeng Lv. 6d pose estimation of objects: Recent technologies and challenges. *Applied Sciences*, 11(1):228, 2021.

[65] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition. *Image Vision Comput.*, 60(C):4–21, 04 2017.

[66] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012.

[67] Tomáš Hodaň, Dániel Baráth, and Jiří Matas. EPOS: Estimating 6D pose of objects with symmetries. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[68] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017.

[69] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. In *European Conference on Computer Vision*, pages 577–594. Springer, 2020.

[70] Tomáš Hodaň, Martin Sundermeyer, Yann Labbé, Van Nguyen Nguyen, Gu Wang, Eric Brachmann, Bertram Drost, Vincent Lepetit, Carsten Rother, and Jiří Matas. BOP challenge 2023 on detection, segmentation and pose estimation of seen and unseen rigid objects. *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024.

[71] Nora Horanyi and Zoltan Kato. Generalized pose estimation from line correspondences with known vertical direction. In *International Conference on 3D Vision (3DV)*, pages 244–253, 2017.

[72] Y. Hou, Z. Wang, J. Cheng, J. Lai, and H. Lu. Constructing dynamic graphs for hand gesture recognition via spatial-temporal attention. *arXiv preprint arXiv:1907.08871*, 2019.

[73] Shoudong Huang and Gamini Dissanayake. A critique of current developments in simultaneous localization and mapping. *International Journal of Advanced Robotic Systems*, 13(5):1–13, 2016.

[74] P. Hübner, J. Hou, and D. Iwaszczuk. Evaluation of Intel RealSense D455 Camera Depth Estimation for Indoor SLAM Applications. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-1/W2-2023:1207–1214, 2023.

[75] Intel RealSense depth camera D455. https://www.realsenseai.com/products/real-sense-depth-camera-d455f/. Accessed: 20-03-2026.

[76] Mariano Jaimez, Javier Monroy, and Javier González-Jiménez. Planar odometry from a radial laser scanner: A range flow-based approach. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4476, 2016.

[77] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020.

[78] Xinbei Jiang, Zichen Zhu, Tianhan Gao, and Nan Guo. Ebfa-6d: End-to-end transparent object 6d pose estimation based on a boundary feature augmented mechanism. *Sensors*, 24(23), 2024.

[79] Mei Jin, Jiaqing Li, and Liguo Zhang. Dope++: 6D pose estimation algorithm for weakly textured objects based on deep neural networks. *PloS one*, 17(6):e0269175, 2022.

[80] Glenn Jocher and Ultralytics Team. YOLOv5 by Ultralytics. https://github.com/ultralytics/yolov5, 2020. Accessed: 2024-06-17.

[81] Jacques Kaiser, Agostino Martinelli, Flavio Fontana, and Davide Scaramuzza. Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation. *IEEE Robotics and Automation Letters*, 2(1):18–25, 2016.

[82] Osmo Karhu, Pekka Kansi, and Iikka Kuorinka. Correcting working postures in industry: A practical method for analysis. *Applied ergonomics*, 8(4):199–201, 1977.

[83] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[84] 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 *CVPR*, 2024.

[85] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023.

[86] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for RGB-D cameras. In *Intl. Conference on Robotics and Automation (ICRA)*, pages 3748–3754, 2013.

[87] Seong-heum Kim and Youngbae Hwang. A survey on deep learning based methods and datasets for monocular 3D object detection. *Electronics*, 10(4):517, 2021.

[88] Tae Soo Kim and Austin Reiter. Interpretable 3D human action analysis with temporal convolutional networks. In *IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 1623–1631. IEEE, 2017.

[89] Tan Kim Geok, Khaing Zar Aung, Moe Sandar Aung, Min Thu Soe, Azlan Abdaziz, Chia Pao Liew, Ferdous Hossain, Chih P. Tso, and Wong Hin Yong. Review of indoor positioning: Radio wave technology. *Applied Sciences*, 11(1), 2021.

[90] 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.

[91] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, pages 225–234. IEEE, 2007.

[92] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 83–86, 2009.

[93] Kurt Konolige and Motilal Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[94] Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. Towards real-time generalized ergonomic risk assessment for the prevention of musculoskeletal disorders. In *14th ACM International Conference on Pervasive Technologies Related to Assistive Environments Conference (PETRA)*, June-July 2021.

[95] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In *2010 Asian Conference on Computer Vision*, pages 216–229. Springer Berlin Heidelberg, 2011.

[96] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[97] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[98] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *Conference on Robot Learning (CoRL)*, pages 715–725, 2022.

[99] K. Lai and S.N. Yanushkevich. CNN+RNN depth and skeleton-based dynamic hand gesture recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3451–3456. IEEE, 2018.

[100] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[101] Feiya Li, Chunyun Fu, Dongye Sun, Jian Li, and Jianwen Wang. SD-SLAM: A semantic slam approach for dynamic scenes based on lidar point clouds, 2024.

[102] Fu Li, Shishir Reddy Vutukur, Hao Yu, Ivan Shugurov, Benjamin Busam, Shaowu Yang, and Slobodan Ilic. NeRF-Pose: A first-reconstruct-then-regress approach for weakly-supervised 6D object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2123–2133, 2023.

[103] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[104] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. MHFormer: Multi-hypothesis transformer for 3D human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13147–13156, 2022.

[105] Xiangtai Li, Henghui Ding, Haobo Yuan, Wenwei Zhang, Jiangmiao Pang, Guangliang Cheng, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[106] Y. Li, X. Wang, Q. Zhang, and H. Liu. A method for evaluating the slam algorithm based on the ros_gazebo simulation platform. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1234–1240. IEEE, 2023.

[107] Yangke Li, Dongyang Ma, Yuhang Yu, Guangshun Wei, and Yuanfeng Zhou. Compact joints encoding for skeleton-based dynamic hand gesture recognition. *Computers & Graphics*, 97:191–199, 2021.

[108] Gabriele Ligorio and Angelo Maria Sabatini. Extended kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: Comparative analysis and performance evaluation. *Sensors*, 13(2):1919–1941, 2013.

[109] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[110] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[111] Yuan Lin, Haiqing Dong, Wentao Ye, Xue Dong, and Shuogui Xu. InfoLa-SLAM: Efficient lidar-based lightweight simultaneous localization and mapping with information-based keyframe selection and landmarks assisted relocalization. *Remote Sensing*, 15(18), 2023.

[112] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan. Decoupled representation learning for skeleton-based gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5751–5760, 2020.

[113] Weizhe Liu, Bugra Tekin, Huseyin Coskun, Vibhav Vineet, Pascal Fua, and Marc Pollefeys. Learning to align sequential actions in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2181–2191, June 2022.

[114] Yen-Cheng Liu, Chih-Yao Ma, and Zsolt Kira. Unbiased teacher v2: Semi-supervised object detection for anchor-free and anchor-based detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9819–9828, 2022.

[115] Manolis I. A. Lourakis and Antonis A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1):1–30, 2009.

[116] María T. Lázaro, Roberto Capobianco, and Giorgio Grisetti. Efficient long-term mapping in dynamic environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 153–160, 2018.

[117] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2018.

[118] Steve Macenski. On use of the SLAM Toolbox: A fresh(er) look at mapping and localization for the dynamic world. In *Proceedings of ROSCon 2019*, 2019.

[119] Steve Macenski and Ivona Jambrecic. SLAM Toolbox: SLAM for the dynamic world. *Journal of Open Source Software*, 6(61):2783, 2021.

[120] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The marathon 2: A navigation system. *CoRR*, abs/2003.00368, 2020.

[121] Mehran Maghoumi and Joseph J. LaViola Jr. DeepGRU: Deep gesture recognition utility. In *International Symposium on Visual Computing, ISVC 2019*, volume 11844 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2019.

[122] Behrooz Mahasseni and Sinisa Todorovic. Regularizing long short term memory with 3D human-skeleton sequences for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3054–3062, 2016.

[123] Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minciullo, Xiangyang Ji, and Nassir Navab. CPS++: Improving class-level 6D pose and shape estimation from monocular images with self-supervised learning. *arXiv preprint arXiv:2003.05848*, 2020.

[124] Khaoula Mannay, Jesús Ureña, Álvaro Hernández, José M. Villadangos, Mohsen Machhout, and Taoufik Aguili. Evaluation of multi-sensor fusion methods for ultrasonic indoor positioning. *Applied Sciences*, 11(15), 2021.

[125] Agostino Martinelli. Closed-form solution of visual-inertial structure from motion. *International journal of computer vision*, 106(2):138–152, 2014.

[126] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1049–1059, 2020.

[127] Pauline Maurice, Adrien Malaisé, Clélie Amiot, Nicolas Paris, Guy-Junior Richard, Olivier Rochel, and Serena Ivaldi. Human movement and ergonomics: An industry-oriented dataset for collaborative robotics. *The International Journal of Robotics Research*, 38(14):1529–1537, 2019.

[128] Lynn McAtamney and Sue Hignett. Rapid entire body assessment. In *Handbook of Human Factors and Ergonomics Methods*, pages 97–108. CRC Press, 2004.

[129] C. Medina, José Segura, and Ángel Torre. Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy. *Sensors*, 13:3501–3526, 03 2013.

[130] Rahil Mehrizi, Xi Peng, Zhiqiang Tang, Xu Xu, Dimitris N. Metaxas, and K. Li. Toward marker-free 3D pose estimation in lifting: A deep multi-view solution. *2018 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2018)*, pages 485–491, 2018.

[131] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal V. Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation in the wild using improved CNN supervision. *2017 International Conference on 3D Vision (3DV)*, pages 506–516, 2017.

[132] Maddock Meredith, Steve Maddock, et al. Motion capture file formats explained. *Department of Computer Science, University of Sheffield*, 211:241–244, 2001.

[133] 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 *ECCV*, 2020.

[134] Faraz M. Mirzaei and Stergios I. Roumeliotis. Globally optimal pose estimation from line-plane correspondences. Technical Report 2010-005, MARS Lab, University of Minnesota, 2010.

[135] P. Molchanov, S. Gupta, K. Kim, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4207–4215, 2016.

[136] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Eighteenth National Conference on Artificial Intelligence*, pages 593–598, USA, 2002. American Association for Artificial Intelligence.

[137] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. Genflow: Generalizable recurrent flow for 6d pose refinement of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10039–10049, 2024.

[138] Kosuke Moriwaki, Gaku Nakano, and Tetsuo Inoshita. The brio-ta dataset: Understanding anomalous assembly process in manufacturing. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1991–1995, 2022.

[139] Md Golam Morshed, Tangina Sultana, Aftab Alam, and Young-Koo Lee. Human action recognition: A taxonomy-based survey, updates, and opportunities. *Sensors*, 23(4), 2023.

[140] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007.

[141] Wei-Yi Mu, Guang-Peng Zhang, Yu-Mei Huang, Xin-Gang Yang, Hong-Yan Liu, and Wen Yan. Omnidirectional scanning localization method of a mobile robot based on ultrasonic sensors. *Sensors*, 16(12):2189, December 2016.

[142] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, October 2015.

[143] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[144] Richard A Newcombe et al. KinectFusion: Real-time dense surface mapping and tracking. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011.

[145] Paul Newman, David Cole, and Kin Ho. Outdoor SLAM using visual appearance and laser ranging. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3401–3407. IEEE, 2006.

[146] Thien-Minh Nguyen, Muqing Cao, Shenghai Yuan, Yang Lyu, Thien Hoang Nguyen, and Lihua Xie. Viral-fusion: A visual-inertial-ranging-lidar sensor fusion approach. *IEEE Transactions on Robotics*, 38(2):958–977, 2022.

[147] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. CNOS: A strong baseline for CAD-based novel object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2134–2140, 2023.

[148] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. GigaPose: Fast and robust novel object pose estimation via one correspondence. In *CVPR23*, 2023.

[149] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I, 2004.

[150] Huthaifa Obeidat, Wafa Shuaieb, Omar Obeidat, and Raed Abd-Alhameed. A review of indoor localization techniques and wireless technologies. *Wireless Personal Communications*, 119:289–327, 2021.

[151] Enrico Occhipinti. OCRA: a concise index for the assessment of exposure to repetitive movements of the upper limbs. *Ergonomics*, 41(9):1290–1311, 1998.

[152] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[153] Evin Pınar Örnek, Yann Labb'e, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomás Hodan. Foundpose: Unseen object pose estimation with foundation features. In *ECCV24*, 2024.

[154] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, pages 58–77, 2024.

[155] Konstantinos Papoutsakis, Nikolaos Bakalos, Konstantinos Fragkoulis, Athena Zacharia, Georgia Kapetadimitri, and Maria Pateraki. A vision-based framework for human behavior understanding in industrial assembly lines. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops - T-CAP 2024 Towards a Complete Analysis of People: Fine-grained Understanding for Real-World Applications*, 2024.

[156] Konstantinos Papoutsakis, Nikolaos Bakalos, Athena Zacharia, Konstantinos Fragkoulis, Georgia Kapetadimitri, and Maria Pateraki. A vision-based framework and dataset for human behavior understanding in industrial assembly lines. *Computer Vision and Image Understanding*, 263:104592, 2026.

[157] Konstantinos Papoutsakis, George Papadopoulos, Michail Maniadakis, Thodoris Papadopoulos, Manolis Lourakis, Maria Pateraki, and Iraklis Varlamis. Detection of physical strain and fatigue in industrial environments using visual and non-visual low-cost sensors. *Technologies*, 10(2), 2022.

[158] Konstantinos Papoutsakis, Papadopoulos Thodoris, Michalis Maniadakis, Manolis Lourakis, Maria Pateraki, and Iraklis Varlamis. Detection of physical strain and fatigue in industrial environments using visual and non-visual sensors. In *The 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA 2021)*, pages 270–271, 04 2021.

[159] Konstantinos E Papoutsakis and Antonis A Argyros. Unsupervised and explainable assessment of video similarity. In *BMVC*, page 151, 2019.

[160] Behnoosh Parsa and Ashis G. Banerjee. A multi-task learning approach for human activity segmentation and ergonomics risk assessment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2352–2362, January 2021.

[161] Behnoosh Parsa, Athmanarayanan Lakshmi narayanan, and Behzad Dariush. Spatio-temporal pyramid graph convolutions for human action recognition and postural assessment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[162] Behnoosh Parsa, Ekta U. Samani, Rose Hendrix, Cameron Devine, Shashi M. Singh, Santosh Devasia, and Ashis G. Banerjee. Toward ergonomic risk prediction via segmentation of indoor object manipulation actions using spatiotemporal convolutional networks. *IEEE Robotics and Automation Letters*, 4(4):3153–3160, 2019.

[163] Julio A. Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A. Castellanos. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, 39(3):1686–1705, 2023.

[164] Ammar Qammaz and Antonis Argyros. A unified approach for occlusion tolerant 3d facial pose capture and gaze estimation using mocapnets. In *IEEE/CVF International Conference on Computer Vision Workshops (AMFG 2023 - ICCVW 2023)*, pages 3178–3188, Paris, France, October 2023. IEEE.

[165] Ammar Qammaz and Antonis A. Argyros. Occlusion-tolerant and personalized 3D human pose estimation in rgb images. In *IEEE International Conference on Pattern Recognition (ICPR 2020)*, January 2021.

[166] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[167] Zurong Qiu, Yaohuan Lu, and Zhen Qiu. Review of ultrasonic ranging methods and their current challenges. *Micromachines*, 13(4):520, March 2022. Review article.

[168] Francesco Ragusa, Rosario Leonardi, Michele Mazzamuto, Claudia Bonanno, Rosario Scavo, Antonino Furnari, and Giovanni Maria Farinella. Enigma-51: Towards a fine-grained understanding of human behavior in industrial scenarios. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4549–4559, 2024.

[169] Mohammad Muntasir Rahman, Yanhao Tan, Jian Xue, and Ke Lu. Recent advances in 3D object detection in the era of deep neural networks: A survey. *IEEE Transactions on Image Processing*, 29:2947–2962, 2019.

[170] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[171] Colin Rennie, Rahul Shome, Kostas E Bekris, and Alberto F De Souza. A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. *IEEE Robotics and Automation Letters*, 1(2):1179–1185, 2016.

[172] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[173] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.

[174] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision*, pages 2564–2571, 2011.

[175] Caner Sahin and Tae-Kyun Kim. Recovering 6d object pose: a review and multi-modal analysis. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[176] Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-SLAM: Dense Neural Point Cloud-Based SLAM. In *ICCV*, 2023.

[177] Panagiotis Sapoutzoglou, George Giapitzakis, George Terzakis, and Maria Pateraki. Cobra–confidence score based on shape regression analysis for method-independent quality assessment of object pose estimation from single images. *arXiv preprint arXiv:2404.16471*, 2024.

[178] Panagiotis Sapoutzoglou, George Terzakis, Georgios Floros, and Maria Pateraki. Shape representation using Gaussian Process mixture models. *arXiv:2604.00862*, 2026.

[179] Panagiotis Sapoutzoglou, George Terzakis, and Maria Pateraki. SHARC: Reference point driven spherical harmonic representation for complex shapes, 2026.

[180] Panagiotis Sapoutzoglou, Orestis Vaggelis, Athina Zacharia, Evangelos Sartinas, and Maria Pateraki. Industryshapes: An RGB-D benchmark dataset for 6D object pose estimation of industrial assembly components and tools, 2026.

[181] Karlheinz Schaub, Gabriele Caragnano, Bernd Britzke, and Ralph Bruder. The European assembly worksheet. *Theoretical Issues in Ergonomics Science*, 14(6):616–639, 2013.

[182] Karlheinz G Schaub, Jens Mühlstedt, Benjamin Illmann, Sebastian Bauer, Lars Fritzsche, Torsten Wagner, Angelika C Bullinger-Hoffmann, and Ralph Bruder. Ergonomic assessment of automotive assembly tasks with digital human modelling and the 'ergonomics assessment worksheet'(EAWS). *International Journal of Human Factors Modelling and Simulation*, 3(3-4):398–426, 2012.

[183] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[184] Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[185] Tim J Schoonbeek, Tim Houben, Hans Onvlee, Fons Van der Sommen, et al. Industreal: A dataset for procedure step recognition handling execution errors in egocentric videos in an industrial-like setting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4365–4374, 2024.

[186] Thomas Schops, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–144, 2019.

[187] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The tum vi benchmark for evaluating visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1680–1687. IEEE, 2018.

[188] Luigi Seminara, Giovanni Maria Farinella, and Antonino Furnari. Differentiable task graph learning: Procedural activity representation and online mistake detection from egocentric videos. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[189] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022.

[190] JoonOh Seo and SangHyun Lee. Automated postural ergonomic risk assessment using vision-based posture classification. *Automation in Construction*, 128:103725, 2021.

[191] Michaela Servi, Andrea Profili, Rocco Furferi, and Yary Volpe. Comparative evaluation of Intel RealSense D415, D435i, D455, and Microsoft Azure Kinect DK sensors for 3D vision applications. *IEEE Access*, 12:111311–111321, 2024.

[192] Myriam Servieres, Valerie Renaudin, Alexis Dupuis, and Nicolas Antigny. Visual and visual-inertial SLAM: State of the art, classification, and experimental benchmarking. *Journal of Sensors*, 2021:1–26, February 2021.

[193] A. Shafti, A. Ataka, B. Urbistondo Lazpita, A. Shiva, H.A. Wurdemann, and K. Althoefer. Real-time robot-assisted ergonomics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1975–1981, 2019.

[194] Shrishti Shah and Jitendra Tembhurne. Object detection using convolutional neural networks and transformer-based models: a review. *Journal of Electrical Systems and Information Technology*, 10(1):54, 2023.

[195] Tahira Shehzadi, Khurram Azeem Hashmi, Didier Stricker, and Muhammad Zeshan Afzal. 2D object detection with transformers: a review. *arXiv preprint arXiv:2306.04670*, 2023.

[196] Yuhan Shen, Niluthpol Chowdhury Mithun Das, and Larry S Davis. Protas: Procedural task-aware segmentation of egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[197] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11069–11078, 2022.

[198] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. *Asian Conference on Computer Vision (ACCV)*, 2020.

[199] Yifei Shi, Junwen Huang, Xin Xu, Yifan Zhang, and Kai Xu. StablePose: Learning 6D object poses from geometrically stable patches. *arXiv preprint arXiv:2102.09334*, 2021.

[200] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Skeleton-based dynamic hand gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1206–1214. IEEE Computer Society, 2016.

[201] ZED X Series: GMSL2 stereo cameras for perception. https://www.stereolabs.com/zed-x/. Accessed: 20-03-2026.

[202] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Visual SLAM: Why filter? *Image and Vision Computing*, 30(2):65–77, February 2012.

[203] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.

[204] Santosh Subedi, Goo-Rak Kwon, Seokjoo Shin, Suk-seung Hwang, and Jae-Young Pyun. Beacon based indoor positioning system using weighted centroid localization approach. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 1016–1019. IEEE, 2016.

[205] Marek Sukop, Maksym Grytsiv, Rudolf Jánoš, and Ján Semjon. Simple ultrasonic-based localization system for mobile robots. *Applied Sciences*, 14(9), 2024.

[206] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[207] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022.

[208] Iago Suárez, Ghesn Sfeir, José M. Buenaposada, and Luis Baumela. BEBLID: Boosted efficient binary local image descriptor. *Pattern Recognition Letters*, 133:366–372, 2020.

[209] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 16558–16569, 2021.

[210] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class Hough forests for 3D object detection and pose estimation. In *European Conference on Computer Vision*, pages 462–477. Springer, 2014.

[211] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2019.

[212] George Terzakis and Manolis Lourakis. Pose estimation from line–plane correspondences with application to camera localization in low-texture environments. Under submission.

[213] George Terzakis and Manolis Lourakis. A consistently fast and globally optimal solution to the perspective-n-point problem. In *European Conference on Computer Vision*, pages 478–494. Springer Nature Switzerland, 2020.

[214] George Terzakis and Manolis Lourakis. Efficient pose prediction with rational regression applied to vSLAM. In *International Conference on Robotics and Automation (ICRA)*, pages 11970–11976. IEEE, 2024.

[215] George Terzakis and Manolis Lourakis. Fast and consistently accurate perspective-n-line pose estimation. In *Intl. Conference on Pattern Recognition (ICPR)*, pages 97–112, 2024.

[216] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[217] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[218] Quoc-Khai Tran and Young-Jae Ryoo. Multi-sensor fusion framework for reliable localization and trajectory tracking of mobile robot by integrating uwb, odometry, and ahrs. *Biomimetics*, 10(7):478, July 2025.

[219] Konstantinos A. Tsintotas, Loukas Bampis, and Antonios Gasteratos. The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):19929–19953, 2022.

[220] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177—280, July 2008.

[221] Vygantas Ušinskis, Michał Nowicki, Andrius Dzedzickis, and Vytautas Bučinskas. Sensor-fusion based navigation for autonomous mobile robot. *Sensors*, 25(4):1248, February 2025. Review article.

[222] Orestis Vaggelis, Evangelos Sartinas, Panagiotis Sapoutzoglou, Georgios Floros, Georgios Terzakis, and Maria Pateraki. 6d object pose estimation for unseen instances: Literature review and comparative evaluation. Under Review.

[223] M.A. Vega Torres, A. Braun, and A. Borrmann. *OGM2PGBM: Robust BIM-based 2D-LiDAR localization for lifelong indoor navigation*, page 567–574. CRC Press, March 2023.

[224] Han Wang, Chen Wang, and Lihua Xie. Intensity-SLAM: Intensity assisted localization and mapping for large scale environment. *CoRR*, abs/2102.03798, 2021.

[225] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019.

[226] Yeping Wang, Gopika Ajaykumar, and Chien-Ming Huang. See what i see: Enabling user-centric robotic assistance using first-person demonstrations. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 639–648, 2020.

[227] Silvan Weder, Johannes L. Schonberger, Marc Pollefeys, and Martin R. Oswald. NeuralFusion: Online Depth Fusion in Latent Space. In *CVPR*, 2021.

[228] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic monocular 3D human pose estimation with normalizing flows. In *International Conference on Computer Vision (ICCV)*, 2021.

[229] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024.

[230] JP Womack. From lean tools to lean management. *Lean Enterprise Institute*, 21, 2006.

[231] Martin Woolley. Bluetooth direction finding. *A technical Overview*, 2019.

[232] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. MeMViT: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13577–13587, 2022.

[233] Jun Wu, Yue Wang, and Rong Xiong. Unseen object pose estimation via registration. In *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 974–979. IEEE, 2021.

[234] Junhui Wu, Dong Yin, Jie Chen, Yusheng Wu, Huiping Si, and Kaiyan Lin. A survey on monocular 3D object detection algorithms based on deep learning. In *Journal of Physics: Conference Series*, volume 1518, page 012049. IOP Publishing, 2020.

[235] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. Accessed: 2024-06-17.

[236] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[237] Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and Mohan S. Kankanhalli. Learning to detect human-object interactions with knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[238] Chi Xu, Lilian Zhang, Li Cheng, and Reinhard Koch. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(6):1209–1222, 2017.

[239] Meng Xu, Youchen Wang, Bin Xu, Jun Zhang, Jian Ren, Zhao Huang, Stefan Poslad, and Pengfei Xu. A critical analysis of image-based camera pose estimation techniques. *Neurocomputing*, 570:127125, 2024.

[240] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2018.

[241] Xuzhong Yan, Heng Li, Chen Wang, JoonOh Seo, Hong Zhang, and Hongwei Wang. Development of ergonomic posture recognition technique based on 2D ordinary camera for construction hazard prevention through view-invariant features in 2D skeleton motion. *Advanced Engineering Informatics*, 34:152–163, 2017.

[242] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392, 2011.

[243] Xiangdi Yue, Yihuan Zhang, Jiawei Chen, Junxin Chen, Xuanyi Zhou, and Miaolei He. Lidar-based slam for robotic mapping: state of the art and new frontiers. *Industrial Robot: the international journal of robotics research and application*, 51(2):196–205, 2024.

[244] Xiaoping Yun, Eric R Bachmann, Hyatt Moore, and James Calusdian. Self-contained position tracking of human movement using small inertial/magnetic sensor modules. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2526–2533. IEEE, 2007.

[245] Raza Yunus, Yanyan Li, and Federico Tombari. ManhattanSLAM: Robust planar tracking and mapping leveraging mixture of Manhattan frames. In *International Conference on Robotics and Automation (ICRA)*, pages 6687–6693, 2021.

[246] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Temporal query networks for fine-grained video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4486–4496, 2021.

[247] Fengyun Zhang, Jia Li, Xiaoqing Zhang, Shukai Duan, and Shuang-Hua Yang. Indoor fusion positioning based on "imu-ultrasonic-uwb" and factor graph optimization method, 2025.

[248] Ji Zhang and Sanjiv Singh. Loam : Lidar odometry and mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, pages 109–111, 01 2014.

[249] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017.

[250] Shishun Zhang, Longyu Zheng, and Wenbing Tao. Survey and evaluation of RGB-D SLAM. *IEEE Access*, 9:21367–21387, 2021.

[251] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. GraFormer: Graph-oriented transformer for 3D pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20438–20447, 2022.

[252] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[253] zhaoxia Zhou, yixuan Lu, and jingbo Xia. Research on the fusion positioning system of uwb/lidar based on the algorithm of spf and kf. *Physical Communication*, 67:102529, 2024.

[254] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3D human pose estimation with spatial and temporal transformers. *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[255] Yingzhao Zhu, Man Li, Wensheng Yao, and Chunhua Chen. A review of 6D object pose estimation. In *2022 IEEE 10th Joint Intl. Information Technology and Artificial Intelligence Conf. (ITAIC)*, volume 10, pages 1647–1655, 2022.

[256] Zhuofan Zong, Guanglu Song, and Yu Liu. DETRs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023.

[257] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023.

[258] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017.