2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Semantic-Aware Obstacle Tracking and Avoidance for Ceiling-Mounted Healthcare Robots

Marco Masannek[1,2], Rolf Schmidt[1,2], Andreas Deinlein, Thorsten Gecks, Stefan May[1] and Andreas Nüchter[2]

***Abstract*— Automation of healthcare workflows and devices demands safe and trustworthy robotic behavior, particularly in environments shared with patients and medical staff. For ceiling-mounted imaging robots, the key challenge lies in perceiving and monitoring the 3D workspace to plan safe, collision-free motions around people and equipment. Beyond simple obstacle avoidance, semantic understanding is essential to distinguish between object types — such as patients, walking aids, or medical tools — and to adapt motion behavior accordingly. We address this challenge with a semantic-aware obstacle tracking and avoidance pipeline that extends prior 2D semantic navigation concepts into full 3D space. The approach combines 2D semantic segmentation with depth projection to estimate object positions and dimensions in real time from RGB-D data. These detections are fused in a tracking module to build a continuous, semantic world model from which class-dependent safety margins are derived. The resulting information enables adaptive motion planning that increases distance from high-risk objects (e.g., persons) or reduces velocity when close interaction is required. Experiments on a real ceiling-mounted robot in laboratory scenarios demonstrate the system's ability to enhance safety, predictability, and contextual awareness during automated healthcare procedures.**

I. INTRODUCTION

Ceiling-mounted mechatronic systems for medical imaging [1] are being increasingly used in modern hospitals due to their electronic positioning abilities. These systems typically consist of gantry-style portal robots, combining two orthogonal ceiling rails for planar motion with a vertical lift that adjusts the height of the mounted imaging unit, such as an x-ray tube or CT detector. This configuration enables precise six-degree positioning of heavy components around the patient and allows remote operation through motorized drives, thereby relieving staff from physically demanding manual adjustments. However, current safety concepts still rely on a human-in-the-loop to monitor the workspace and avoid collisions with obstacles, which can lead to accidents when attention is diverted (e.g., during patient care). Moreover, quickly reaching an optimal scanning position requires experienced staff members, who are not always available. These factors have led to a growing demand for workflow automation in such systems [2], e.g., automatic positioning of imaging tools or intuitive gesture-control modes, thereby transforming them into collaborative robots (CoBots) capable of operating safely in clinical environments.

[1]Nuremberg Institute of Technology, Kesslerplatz 12, 90489 Nuremberg, Germany, {marco.masannek, rolf.schmidt, stefan.may}@th-nuernberg.de
[2]Julius-Maximilians-University Würzburg, Am Hubland, 97074 Würzburg, Germany, {andreas.nuechter}@uni-wuerzburg.de

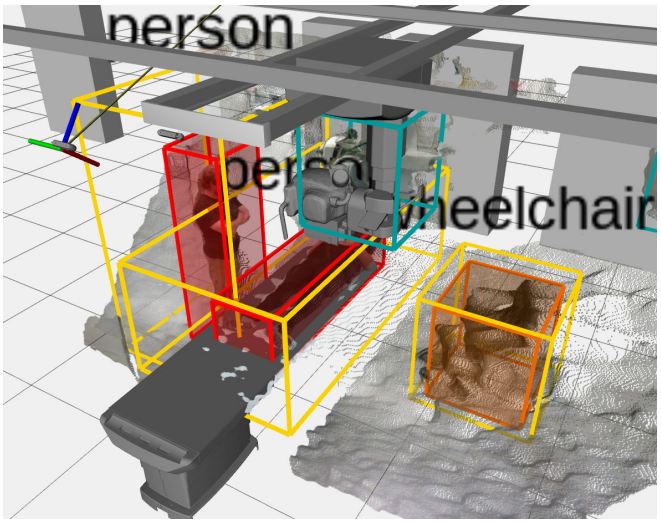

Fig. 1. Real-time environment model of a ceiling-mounted healthcare robot during a mock leg-scanning procedure. The gantry system enables free positioning of the x-ray tube (blue) within the 3D workspace above the patient. An RGB-D camera captures a colored point cloud of the scene, including the floor (white waves) and surrounding obstacles. Detected persons are classified as high-risk objects (red), while less critical items such as the wheelchair are shown in orange. Type-specific safety zones (yellow) visualize semantic safety constraints that enforce increased caution around sensitive objects.

CoBots are robotic systems designed to operate safely alongside humans, cooperating on physical and non-physical tasks such as assisted manufacturing [3]. While most industrial CoBots rely on internal or external sensors for collision avoidance, they typically work in controlled environments accessible only to trained personnel.

In contrast, medical environments involve untrained staff and vulnerable patients in close proximity to the robot, which demands not only heightened perception and safety awareness but also trustworthy system behavior to maintain user confidence and reduce stress during procedures. Reliable identification and tracking of people and equipment are therefore essential for safe and predictable motion. RGB-D cameras are well suited for this purpose, as they provide high-quality 3D data for indoor applications, enabling object detection and path planning when combined with semantic processing.

Building a consistent real-time world model from such data, however, remains challenging, since the medical domain has received limited attention in computer vision research and often requires substantial adaptation of existing methods.

In our prior work, we developed a safety-constraint navigation system for a mobile robotic nurse in an OR [4], combining semantic segmentation, position estimation, and semantic obstacle paddings for 2D navigation around safety-critical static objects.

In this paper, we integrate an update of our pipeline into a ceiling-mounted x-ray gantry system for healthcare applications, aiming to transfer the concept to automated workflows in 3D. To our knowledge, this is the first implementation of a semantic-aware 3D navigation framework for a ceiling-mounted healthcare robot, extending prior 2D semantic-constraint navigation concepts to a full volumetric planning and tracking system. Our key contributions are

- Real-time object detection and 3D pose estimation in the system workspace
- Modular tracking architecture for state estimation and semantic processing
- Custom message data type, which couples detection results and point cloud segments for better data association. Open-source implementations of our messages and visualization plugins are publicly available [5]
- Semantic-aware 3D path planning and collision avoidance for a ceiling-mounted healthcare robot

The remainder of this paper is structured as follows: Section II provides a detailed review of related work, focusing on object detection, state estimation and semantic path planning in CoBot applications. Section III then formulates a problem statement for the detection, tracking and motion planning modules. Our used methodology follows in IV, giving an architectural overview of our system design and details on used models and algorithms. We present our experiments in Section V, followed by results and discussion in Section VI, before concluding our work with an outlook for future research in Section VII.

II. RELATED WORK

In recent years, the main focus for robots in healthcare environments has been on telepresence and assistive applications [6][7], with patient aid and rehabilitation being a highly popular research field [8]. While these systems often target mobile robot applications (e.g. smart wheelchairs), only few works aim to improve the workflow of modern gantry robot systems in hospitals through automation. Some works exist that aim to optimize trajectories of robotic x-ray imaging [9], but their focus lies on improving the quality of resulting images rather than planning paths around obstacles. To address this issue, [10] aimed to provide a detection and avoidance pipeline for a ceiling-mounted system, but their work lacked real-time capabilities and did not include motion tests in realistic environments. Due to the sparse availability of testing systems for such medical devices, other works involving similar kinematic constraints often target applications in industrial automation, such as assembly or welding tasks [11][12]. They demonstrate smooth path planning for predefined scenes but often fail to include real-time detection methods for dynamic objects such as humans in real environments.

A. Camera-based object detection

Recent advances in convolutional neural networks have significantly improved real-time object detection and segmentation in robotics. Earlier approaches like R-CNN and its successors offered bounding box proposals with high accuracy but limited real-time applicability [13][14]. This limitation was addressed by YOLO (You only look once) and its successors [15][16], which reframed object detection as a regression problem, enabling real-time inference. The latest iterations, such as YOLOv11 [17][18] achieve state-of-the-art results in both speed and accuracy and are increasingly used for robotic perception tasks to detect e.g. humans [19].

To support pixel-level accuracy in custom models, instance segmentation networks such as Mask R-CNN [20] and frameworks like Segment Anything [21] have been employed to simplify the process of manual dataset creation for highly specific target domains, e.g. medical environments.

While 6-DOF pose estimation networks such as *foundation pose* [22] have shown promising results on household objects, they often only support predictions for a single object per frame, which can potentially lead to some obstacles being ignored. However, acquiring suitable training data for custom objects remains difficult, leading many approaches to combine 2D segmentation with 3D point-cloud data for pose estimation.

B. Collision avoidance for CoBots

Only few works aim to integrate real-time obstacle avoidance in CoBot applications based on recent sensor readings. While the authors of [23] attached proximity sensors directly to the robot, the authors of [24] used point cloud data from an external RGB-D camera to build and update a distance field of the workspace of a robotic arm. Although both approaches demonstrated collision-free motion planning and adaptation in the presence of humans, they did not include type-specific avoidance strategies and only applied their methods on robotic manipulators.

In the field of motion planning for mobile robots, the term *social navigation* has been primarily used for applications where agents navigate in the presence of humans [25]. Path planners often use proxemics [26] to model the comfort zone of a person and adapt their plan to provide more personal space when possible [28]. Other works extend the concept of tailored behavior for specific object types by combining semantic segmentation and classification on an underwater robot to exclude e.g. fish during collision checks [29].

In our prior work [4], we presented a semantic-constraint navigation pipeline for a mobile robotic nurse that detected high-risk obstacles in an operating room and enforced enlarged safety distances around them. While this approach successfully demonstrated dynamic avoidance of semantically labeled objects, it's hard constraints often blocked narrow yet traversable passages and were limited to 2D collision checks without considering object height.

In contrast, the proposed method introduces soft, semantic safety constraints that enable both efficiency and context-sensitive avoidance in 3D.

III. Problem Statement

Let $\mathcal{R} \subset \mathbb{R}^3$ denote the robot's workspace, observed by an RGB-D sensor providing a stream of synchronized color and depth frames:

$$I_t^{\text{rgb}}, I_t^{\text{depth}} \quad \text{at time } t \in \mathbb{R}_{\geq 0}$$

Our goal is to compute a safe, continuous trajectory $\xi : [0, T] \to \mathcal{R}$ for a ceiling-mounted robot, from a start configuration x_0 to a goal x_g, such that the trajectory avoids obstacles with class-dependent safety constraints.

We define $\mathcal{O}_t = \{o_1^t, \ldots, o_N^t\}$ as a set of semantic obstacles detected at time t, where each obstacle $o_i^t = (P_i^t, \hat{p}_i^t, b_i^t, c_i, z_i^t)$ consists of:

- $P_i^t \subset \mathbb{R}^6$: the segmented RGB-XYZ point cloud (e.g., a set of 3D points with color)
- $\hat{p}_i^t \in \mathbb{R}^3$: the estimated 3D position of the object (e.g., centroid of P_i^t)
- $b_i^t \in \mathbb{R}^6$: the 3D bounding box (e.g., $(x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max})$)
- $c_i \in \mathcal{C}$: the semantic class label (e.g., `person`, `walker`)
- $z_i^t \in \mathbb{R}^6$: the semantic safety zone derived from c_i

Tracking is modeled as a data association problem, linking detections o_i^t over time to maintain persistent obstacle identities $\hat{o}_j = \{o_j^{t_0}, o_j^{t_1}, \ldots\}$ using nearest-neighbor matching or semantic filters. Detections with no recent association for $\Delta t > \Delta t_{\text{timeout}}$ are discarded.

To ensure robust classification across frames, we maintain a per-object confidence history for all tracked instances $\hat{o}_j$. At each time step, the current detection contributes a soft classification score vector over all known classes. These scores are accumulated over time using an exponential moving average, allowing transient misclassifications to be smoothed out. The final class label c_i is selected as the most probable class across this history.

Once a stable class label is determined, it is used to query a semantic database which defines type-specific physical dimensions and safety parameters for each object category. This information is used to compute the semantic safety zone z_i^t for each object o_i^t, which extends the physical bounding box b_i^t by a class-dependent padding term b_c:

$$z_i^t = b_i^t + b_c \tag{1}$$

Within this zone, a proximity-based velocity dampening function is applied to positions near obstacles, thereby applying soft safety constraints on the path planning.

Let $\mathcal{B}_i^t = B(p_i^t, r_i)$ be the ball of radius r_i around obstacle i at time t. The set of forbidden regions at time t is:

$$\mathcal{F}_t = \bigcup_{i=1}^{N} \mathcal{B}_i^t \tag{2}$$

The motion planning objective is to compute:

$$\xi^* = \arg\min_{\xi \in \Xi} \int_0^T \|\dot{\xi}(t)\| \, dt \quad \text{s.t.} \quad \xi(t) \notin \mathcal{F}_t \quad \forall t \in [0, T] \tag{3}$$

where Ξ is the set of dynamically feasible trajectories subject to actuator constraints.

Our proposed system solves this problem online, in a rolling-horizon manner, by:

- Detecting o_i^t from RGB-D frames via semantic segmentation and pose estimation
- Tracking $\mathcal{O}_t$ to maintain consistent obstacle representations
- Inferring risk zones via a semantic safety constraint model
- Planning ξ^* using a graph-based search in a dynamically updated costmap

IV. Methodology

A. Robot Description

We implement our obstacle tracking and avoidance pipeline on a ceiling-mounted technology demonstrator (TecDem) in a laboratory environment (see Fig. 2). The gantry-based robot operates within a workspace of approximately $5\,\text{m} \times 3.5\,\text{m} \times 1.8\,\text{m}$, combining orthogonal ceiling rails for planar motion with a vertical telescopic lift that adjusts the height of the x-ray tube mounted on its end effector. An Intel RealSense D455 camera is attached to the supporting ceiling frame, providing a full view of the workspace and streaming RGB-D data to the detection module in real time. System control and feedback are handled by custom software interfacing with the robot's kinematic controllers for position and velocity regulation. All components run on a workstation equipped with an Intel Xeon Silver 4116 CPU and an NVIDIA Quadro P2000 GPU. The system is running Ubuntu 22.04 and the Robot Operating System 2 (ROS2) Humble [30]. An overview of the software architecture is given in the following section.

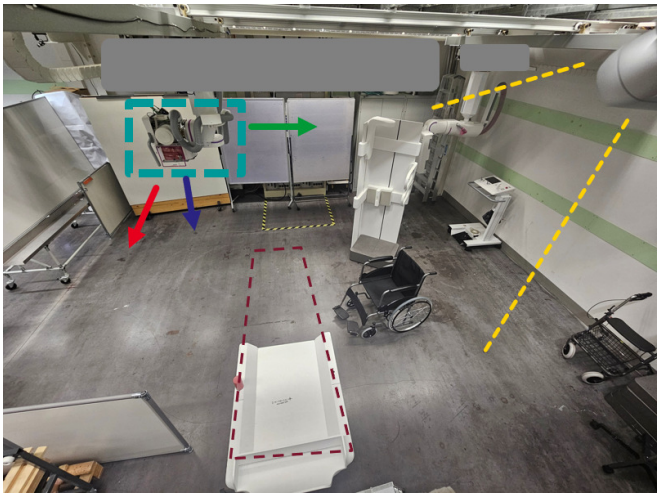

Fig. 2. TecDem setup in a laboratory: The petrol colored box marks the movable x-ray tube along the xyz-axis (rgb-arrows), while the yellow dashes symbolize the field of view of the RGB-D camera (grey housing). The red dashed trapezoid marks the position of the optional patient table when it is mounted. Example obstacles such as the wheelchair, walker and L-stand are placed randomly in this scene to visualize the dimensions.

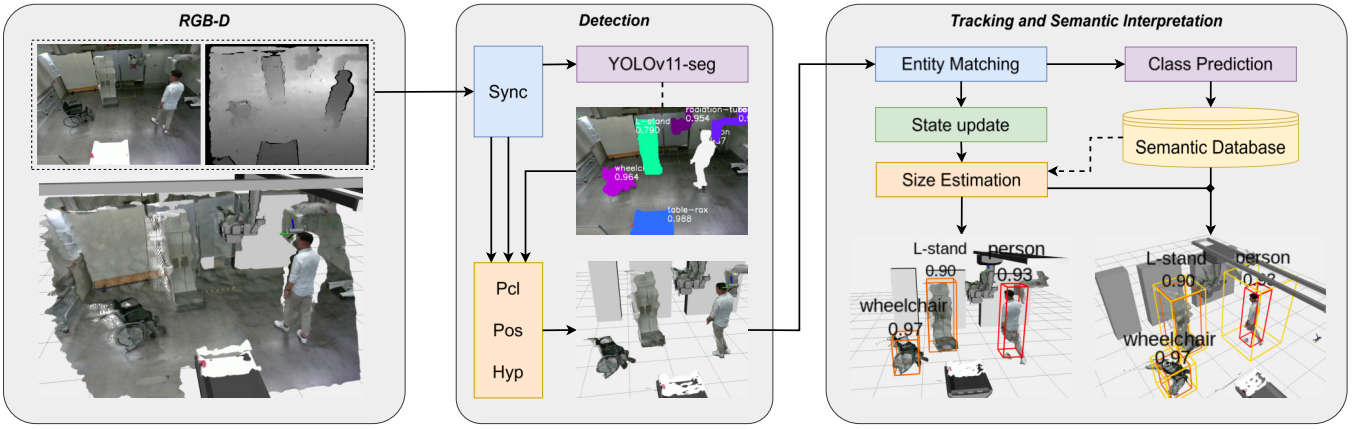

Fig. 3. High-level architecture of the proposed obstacle detection and tracking pipeline. Synchronized RGB-D input (left) is processed through a semantic segmentation network and 3D point-cloud projection (center) to generate class-aware detections. The resulting detections are forwarded to a centralized tracking module that maintains a consistent world model and enriches it with geometric and safety-related information from a semantic database (right). This semantic-aware world model is continuously provided to the motion-planning module, which adapts paths and velocities online according to class-dependent safety constraints.

B. System Architecture

Our system follows a modular ROS2-based architecture designed for reliable and transparent operation in safety-critical medical environments. Its distributed structure allows perception, tracking, and planning processes to run in parallel while exchanging information in real time to maintain a continuously updated world model. This design supports flexibility and scalability, enabling future integration of additional perception modules such as multi-camera setups or hospital information systems. Together, the modules form a semantic-aware framework that allows the robot to interpret its surroundings and adapt motion behavior to nearby people and equipment (see Fig. 3).

C. Object Detection

Our object detection utilizes a single state-of-the-art neural network for semantic segmentation, namely a YOLOv11-Seg [17][18] model, which has been fine-tuned to include domain-relevant classes such as *person, wheelchair, walker, L-stand, patient-table, x-ray-tube, and x-ray-detector*. The model processes RGB images and outputs pixel-wise segmentation masks, which are then used to identify and locate objects in the scene. To obtain the 3D position and dimensions of each detected object, we project the segmented pixels into 3D space using the corresponding depth image and the pinhole camera model. Specifically, for each pixel (u, v) in the segmentation mask, the 3D coordinates (x, y, z) are computed as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = d(u,v) \cdot K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{4}$$

where $d(u, v)$ is the depth value at pixel (u, v) and K is the camera intrinsic matrix. The resulting set of 3D points forms a point cloud segment P_i^t for each object.

We then compute an estimate for the object position $\hat{p}_i^t$ from the centroid, which is obtained by calculating the center of mass $\mathbf{c}$ of the segment as

$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^{N} p_i^t \tag{5}$$

where p_i^t are the 3D points belonging to the object and N is the total number of points.

The detected object set $\mathcal{O}_t$, containing class IDs c_i, confidence, centroid positions $\hat{p}_i^t$, and point cloud segment P_i^t, is packaged into a custom data type message (CDT) and forwarded to the tracking module. This ensures that geometric and semantic information remain coupled for every detection, enabling straightforward data association and entity-level merging during tracking. We therefore extend the *vision_msgs* package to include *PointCloud2* fields for each *Detection3d*. Our fork is available on GitHub [5] and implements the message types and RVIZ2 plugins for visualization (see Fig. 3).

D. Object Tracking

The object-tracking module maintains a list of active entities in the robot's environment, continuously updating their states based on new detections and managing their semantic attributes. During entity matching, the geometric properties of all detections o_i^t are transformed into the world frame and compared with the existing object pool $\hat{o}_j$ based on spatial proximity. If a new detection result (DR) lies within a predefined spatial threshold of an existing object, it is added to its detection pool. If no match is found, a new object entity is created instead.

In each state-update cycle, the module iterates over all tracked objects and merges new detection results (DRs) with their existing states. This involves updating the position estimate $\hat{p}_j^t$, refreshing the point-cloud segment P_j^t from the latest DR, and computing an axis-aligned bounding box b_j^t based on the updated point cloud. Additionally, the class hypotheses of all DRs in the pool are aggregated to compute a mean class label c_j for each object.

Using this class label, type-specific information is retrieved from the semantic database, including the collision relevance R_c, typical bounding-box dimensions (w_b, h_b), and the semantic safety zone size z^t, which constrains the maximum permitted velocity around nearby objects.

To mitigate the effect of noisy depth measurements on object edges, the estimated physical dimensions are compared to the semantic entries, and bounding-box sizes are corrected if the deviation exceeds a predefined threshold. For objects labeled *person*, a principal component analysis (PCA) is performed on the point-cloud segment to extract the orientation of the main axis, enabling rough posture estimation (e.g., *standing* vs. *lying*).

The object pool is then filtered to remove objects that have not been detected for a certain time period $\Delta t > \Delta t_{\text{timeout}}$ or that overlap with tracked duplicates caused by motion. Finally, collision-relevant entities are identified by excluding non-obstacles (e.g. robot components), and semantic safety zones z_j^t are added around high-risk objects (see Fig. 3). The resulting set of semantic obstacles is then forwarded to the motion-planning module as CDT messages, where class-specific attributes are incorporated during path search.

E. Semantic Database

Tab. I lists the parameters (R_c, w_b, h_b, z^t) defined in our semantic database, representing the characteristic properties of potential obstacles for a ceiling-mounted imaging robot.

TABLE I

SEMANTIC DATABASE ENTRIES PER OBJECT TYPE

Parameter	X-ray Tube	Patient Table	Person	L-stand	Wheelchair	Walker
R_c	no	no	yes	yes	yes	yes
w_b [m]	0.7	0.6	0.5	0.8	0.8	0.8
h_b [m]	0.8	1.0	1.8	2.0	0.9	1.2
z^t [m]	-	-	1.4	0.3	0.3	0.3

F. Motion Planning

The motion planning module must fulfill several non-functional requirements, most notably efficiency, robustness, and user trust. To meet these demands, we employ a coarse workspace grid that limits the number of feasible routes while enabling a global A* search to be performed in real time—within each control cycle (typical planner runtime $\approx 5\,\text{ms}$). The resulting path is subsequently smoothed using second-degree splines to ensure feasible and natural motion.

Although the coarse resolution can lead to planning failure in highly cluttered environments, such scenarios are considered rare in the target domain. In these cases, a stationary robot is generally perceived as safer and more acceptable than risky or unpredictable behavior.

The cost function used in the A* search explicitly models time as the primary optimization criterion by incorporating the estimated traversal time for each grid edge. Importantly, speeds are reduced when the robot moves towards a human to improve perceived safety. This time-based cost implicitly encourages the robot to maintain greater distances from people, leading to more socially aware trajectories. However, if the system is forced to move close to people (e.g. for procedures), the areas remain traversable with lowered speed.

V. EXPERIMENTS

We are using the TecDem setup for our experiments. In order to verify our system, we conducted three trial runs each for three selected scenarios (see Fig. 4), recording world belief states and robot trajectories during autonomous motion. While all experiments focus on static scenes, the full semantic tracking and planning pipeline runs continuously in real time. Object positions are detected and updated frame-by-frame rather than predefined, simulating the real perception conditions of a dynamic environment. By applying semantic safety buffers around detected objects, we directly control the occupancy of the workspace and adapt the path planning scene accordingly.

As implementing external baseline methods would require a complete reintegration of perception and control components, we deliberately selected three scenarios in which the effects of our approach are clearly observable by experimental design. Scene 1 & 2 are intended to demonstrate the differences between semantic vs. non-semantic path planning for similar workspace configurations, while the impact of semantic velocity constraints is evaluated in Scene 3, where the robot is forced to travel close to persons.

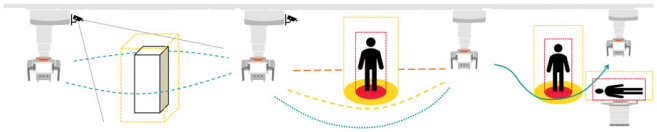

Fig. 4. Three example scenarios for system validation: (left) Scene 1 contains a static object obstructing the direct path between start and goal position. (center) Scene 2: A person obstructing the path, requiring decision-making between faster trajectories keeping more distance and closer, slower paths. (right) Scene 3 describes a typical workflow of an x-ray imaging procedure, with the nurse setting up the patient on the table. The system must wait for the patient to lie down and avoid the nurse and patient during autonomous motion.

A. Scene 1: Misplaced L-Stand

The first scene consists of an L-stand at an unknown position in the workspace when commanding autonomous motion, which may cause accidents in daily clinical routines. The goal is to reliably detect and localize the object and plan a path around it. Since the L-stand is not a high-risk object on its own, the standard proxemic rules apply with the main goal of planning a feasible path around the object.

B. Scene 2: Standing Human

Our second scene demonstrates the system's ability to handle a standing human in the workspace. The goal is to autonomously navigate around the person safely, which means deciding either to keep a larger distance or to reduce the travel speed when close to the human.

C. Scene 3: Demo Procedure

In the third scenario, we aim to replicate automation of an exemplary scanning workflow of a leg-fracture. We therefore act as a nurse positioning a patient on the table while the system is in a parking position (see also Fig. 1). Once the patient is correctly positioned and rests calmly on the table, the system is commanded to move safely into an approximated scan position above the patient. This scenario is particularly challenging, as the patient's posture on the table must be reliably estimated while avoiding the standing nurse at the same time. Since the system is forced to travel close to the persons to reach the final scan position, our semantic velocity constraints shall apply when approaching the humans, effectively lowering the travel speed.

VI. RESULTS & DISCUSSION

A. Runtime Performance

In our tests, our proposed pipeline achieved an average frame rate of ≈ 17 fps, with an overall processing delay of ≈ 280 ms between raw sensor data and tracked obstacles.

Although this latency may appear significant, it remains practical for real-time applications for several reasons. First, once an object has been detected, the tracking and fusion module can continue to provide updated object positions at a high rate, even in the absence of new detections, ensuring timely outputs for downstream tasks. Second, considering the computational demands of semantic processing, the achieved runtime is competitive with state-of-the-art systems. Finally, there is potential for further improvements, e.g., by making use of ROS2 components to reduce data copies between nodes, thereby eliminating transmission delays and increasing the effective frame rate.

B. Qualitative Results

Our recorded object estimates and robot paths in Fig. 5 (see Appendix) illustrate the system's performance for each scenario, displaying waypoints and velocity traces along the trajectories. In Scene 1, the robot initially navigates onto the sparse grid before choosing a path around the L-stand. However, after reaching the first waypoint, the robot travels safely around the object, maintaining cruise speed during the rest of the trajectory. Scene 2 demonstrates the effect of our semantic velocity constraints, as the robot naturally chooses a slightly longer path around the person to maintain a safe distance, allowing it to travel at higher speeds along the trajectory. Compared to Scene 1, the distance and time to reach the goal are only marginally increased, demonstrating our path planner's ability to adapt to the environment while maintaining efficiency.

In Scene 3, our pipeline is able to reliably detect both the lying patient and the standing nurse, allowing the system to safely navigate around the nurse and above the patient. Since the robot must travel close to persons to reach the goal position, our path planner now chooses a more direct path towards the goal while heavily reducing the velocity to ensure safety.

C. Quantitative Results

Our quantitative results are displayed in Tab. II and show the averaged path metrics for each scenario. As proposed by [31], we report the path length $\mathbf{l_p}$, the path time $\mathbf{t_p}$, the time cruise speed $\mathbf{t_{cs}}$ and the closest distance $\mathbf{d_{o,c}}$ to each object class c for all detected objects $\hat{o}$. They confirm that the system is able to reliably detect and avoid obstacles in the workspace. while maintaining a safe behavior due to our semantic velocity constraints.

TABLE II
QUANTITATIVE RESULTS FOR THE THREE VALIDATION SCENARIOS (AVERAGE)

	Scene 1 (L-Stand)	**Scene 2** (Person)	**Scene 3** (Workflow)
$\mathbf{l_p}$ [m]	5.63	6.0	5.8
$\mathbf{t_p}$ [s]	18.95	19.90	30.95
$\mathbf{t_{cs}}$ [s]	10.2	13.4	7.1
$\mathbf{d_{o,l}}$ [m]	0.3	-	-
$\mathbf{d_{o,p}}$ [m]	-	1.38	0.43
$\mathbf{d_{o,w}}$ [m]	-	-	0.72

D. Discussion

The results highlight the effectiveness and practicality of the proposed obstacle tracking and avoidance pipeline for ceiling-mounted robots in clinical settings. Our runtime analysis demonstrates that the system achieves near real-time performance, with a total latency that remains within acceptable bounds for responsive navigation.

Qualitative evaluations across diverse scenarios confirm that the system can robustly avoid detected obstacles, including challenging cases such as prone patients and standing staff during mockup workflows. The integration of semantic information into the planning process enables easy trajectory and velocity adaptation based on the type of encountered object, resulting in behavior that is both safe and efficient.

The quantitative results further support these findings, showing consistent path metrics and safe minimum distances across all test cases. The system's ability to maintain cruise speed where appropriate, while dynamically adjusting to environmental constraints, demonstrates the benefit of semantic-aware planning.

Despite these strengths, several limitations should be acknowledged. The current evaluation focuses on static scenes to ensure controlled validation. However, with the framework being designed for dynamic tracking of objects, future work should include experiments with moving obstacles. Additionally, object size estimation relies on class-based heuristics, which may not capture the full variability encountered in practice. Improvements in perception accuracy and more sophisticated semantic reasoning could therefore enhance system robustness.

VII. CONCLUSIONS & OUTLOOK

In this paper, we presented our improved semantic-aware obstacle detection and avoidance pipeline for healthcare robots and successfully integrated it on a ceiling-mounted x-ray system in a laboratory environment. As demonstrated by

our real-world experiments during mockup workflows, our system is capable of reliably detecting and avoiding objects in the 3D workspace in real time. The planned and executed trajectories of our semantic-aware motion planner further highlighted the benefits of our detection pipeline, allowing the robot to plan safer paths around high-risk objects such as humans. Posture estimation for persons further enables safe 3D path planning above laying patients, demonstrating the potential for real workflow application.

In the future we plan to deploy the system in a clinical prototype setup to evaluate safety and user trust in a real-world hospital setting. Furthermore, we plan to deploy the pipeline in a system with higher-precision sensors and accurate calibration to enable ground truth measurements and evaluation of detection accuracy. This would allow us to further improve the system's performance and reliability in complex environments. Finally, the integration of pose estimation modules could enhance posture and limb detection accuracy, thereby enabling more precise identification of human poses and activities.

Ultimately, ensuring predictable and trustworthy robotic behavior is essential for the acceptance of autonomous systems in clinical practice.

ACKNOWLEDGMENTS

This work was supported by Siemens Healthineers AG. The authors would like to thank Vladimír Jendrol' and Barbora Kubalcová for their contributions.

REFERENCES

[1] Siemens Healthineers AG, (2025 June 2), Multitom Rax, [Online]. Available: https://www.siemens-healthineers.com/de-ch/robotic-x-ray/twin-robotic-x-ray/multitom-rax

[2] A. Thacharodi et al., 'Revolutionizing healthcare and medicine: The impact of modern technologies for a healthier future—A comprehensive review', Health Care Sci, vol. 3, no. 5, pp. 329–349, Oct. 2024, doi: 10.1002/hcs2.115.

[3] Z. Saleem, F. Gustafsson, E. Furey, M. McAfee, and S. Huq, 'A review of external sensors for human detection in a human robot collaborative environment', J Intell Manuf, vol. 36, no. 4, pp. 2255–2279, Apr. 2025, doi: 10.1007/s10845-024-02341-2.

[4] M. Masannek, R. Schmidt, V. Jendrol, C. Coïc, L. Bernhard, C. Guo, D. Wilhelm, S. May, and A. Nüchter, "Safety-oriented Semantic-Constraint Navigation in Clinical Environments", in: Proc. of the 19th International Conference IAS-19, 2025, Springer, to appear.

[5] M. Masannek, (2025 June 25), "Public Fork for extended ROS2:Vision Msgs". [Online]. Available: https://github.com/marcomasa/vision_msgs

[6] D. Silvera-Tawil, "Robotics in Healthcare: A Survey," SN Computer Science, vol. 5, no. 1, p. 189, Jan. 2024, doi: 10.1007/s42979-023-02551-0.

[7] J. Holland et al., "Service Robots in the Healthcare Sector," Robotics, vol. 10, no. 1, Art. no. 1, Mar. 2021, doi: 10.3390/robotics10010047.

[8] A. A. Morgan, J. Abdi, M. A. Q. Syed, G. E. Kohen, P. Barlow, and M. P. Vizcaychipi, "Robots in Healthcare: a Scoping Review", Current Robotics Reports, vol. 3, no. 4, pp. 271–280, Dec. 2022, doi: 10.1007/s43154-022-00095-4.

[9] G. Herl, J. Hiller, M. Thies, J.-N. Zaech, M. Unberath, and A. Maier, 'Task-Specific Trajectory Optimisation for Twin-Robotic X-Ray Tomography', IEEE Transactions on Computational Imaging, vol. 7, pp. 894–907, 2021, doi: 10.1109/TCI.2021.3102824.

[10] M. Mahmeen, R. D. D. Sanchez, M. Friebe, M. Pech, and S. Haider, 'Collision Avoidance Route Planning for Autonomous Medical Devices Using Multiple Depth Cameras', IEEE Access, vol. 10, pp. 29903–29915, 2022, doi: 10.1109/ACCESS.2022.3159239.

[11] X. Zhou, X. Wang, Z. Xie, J. Gao, F. Li, and X. Gu, 'A Collision-free path planning approach based on rule guided lazy-PRM with repulsion field for gantry welding robots', Robotics and Autonomous Systems, vol. 174, p. 104633, Apr. 2024, doi: 10.1016/j.robot.2024.104633.

[12] M. N. Vu, A. Lobe, F. Beck, T. Weingartshofer, C. Hartl-Nesic, and A. Kugi, 'Fast trajectory planning and control of a lab-scale 3D gantry crane for a moving target in an environment with obstacles', Control Engineering Practice, vol. 126, p. 105255, Sep. 2022, doi: 10.1016/j.conengprac.2022.105255.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Oct. 22, 2014, arXiv: arXiv:1311.2524. doi: 10.48550/arXiv.1311.2524.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jan. 06, 2016, arXiv: arXiv:1506.01497. doi: 10.48550/arXiv.1506.01497.

[15] J. Redmon, S. Divvala, R. Girshick, und A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", 9. Mai 2016, arXiv: arXiv:1506.02640. doi: 10.48550/arXiv.1506.02640.

[16] M. Hussain, "YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision," Jul. 03, 2024, arXiv: arXiv:2407.02988. doi: 10.48550/arXiv.2407.02988.

[17] R. Khanam und M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements", 23. Oktober 2024, arXiv: arXiv:2410.17725. doi: 10.48550/arXiv.2410.17725.

[18] G. Jocher and J. Qiu, Ultralytics YOLO11. 2024.

[19] T. Linder, K. Y. Pfeiffer, N. Vaskevicius, R. Schirmer, and K. O. Arras, "Accurate detection and 3D localization of humans using a novel YOLO-based RGB-D fusion approach and synthetic training data," in 2020 IEEE International Conference on Robotics and Automation (ICRA), May 2020, pp. 1000–1006. doi: 10.1109/ICRA40945.2020.9196899.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," Jan. 24, 2018, arXiv: arXiv:1703.06870. doi: 10.48550/arXiv.1703.06870.

[21] A. Kirillov et al., "Segment Anything," Apr. 05, 2023, arXiv: arXiv:2304.02643. doi: 10.48550/arXiv.2304.02643.

[22] B. Wen, W. Yang, J. Kautz and S. Birchfield. "FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects", 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 17868-17879. 10.1109/CVPR52733.2024.01692.

[23] S. J. Moon, J. Kim, H. Yim, Y. Kim, and H. R. Choi, 'Real-Time Obstacle Avoidance Using Dual-Type Proximity Sensor for Safe Human-Robot Interaction', IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 8021–8028, Oct. 2021, doi: 10.1109/LRA.2021.3102318.

[24] U. Ali, L. Wu, A. Mueller, F. Sukkar, T. Kaupp, and T. Vidal-Calleja, 'Interactive Distance Field Mapping and Planning to Enable Human-Robot Collaboration', Oct. 23, 2024, arXiv: arXiv:2403.09988. doi: 10.48550/arXiv.2403.09988.

[25] C. Mavrogiannis et al., "Core Challenges of Social Robot Navigation: A Survey," J. Hum.-Robot Interact., vol. 12, no. 3, p. 36:1-36:39, Apr. 2023, doi: 10.1145/3583741.

[26] Hall, E.T. The Hidden Dimension: An Anthropologist Examines Man's Use of Space in Private and Public; Anchor Books; Doubleday &Company Inc.: New York, NY, USA, 1966.

[27] X.-T. Truong und T.-D. Ngo, "Dynamic Social Zone based Mobile Robot Navigation for Human Comfortable Safety in Social Environments", Int J of Soc Robotics, Bd. 8, Nr. 5, S. 663–684, Nov. 2016, doi: 10.1007/s12369-016-0352-0.

[28] J. Jang and M. Ghaffari, "Social Zone as a Barrier Function for Socially-Compliant Robot Navigation," Oct. 11, 2024, arXiv: arXiv:2405.15101. doi: 10.48550/arXiv.2405.15101.

[29] J. Hong, K. de Langis, C. Wyethv, C. Walaszek, and J. Sattar, "Semantically-Aware Strategies for Stereo-Visual Robotic Obstacle Avoidance," in 2021 IEEE International Conference on Robotics and Automation (ICRA), May 2021, pp. 2450–2456. doi: 10.1109/ICRA48506.2021.9561863.

[30] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, 'Robot Operating System 2: Design, Architecture, and Uses In The Wild', Sci. Robot., vol. 7, no. 66, May 2022, doi: 10.1126/scirobotics.abm6074.

[31] C. Rondoni et al., "Navigation benchmarking for autonomous mobile robots in hospital environment," Sci Rep, vol. 14, no. 1, p. 18334, Aug. 2024, doi: 10.1038/s41598-024-69040-z.

APPENDIX

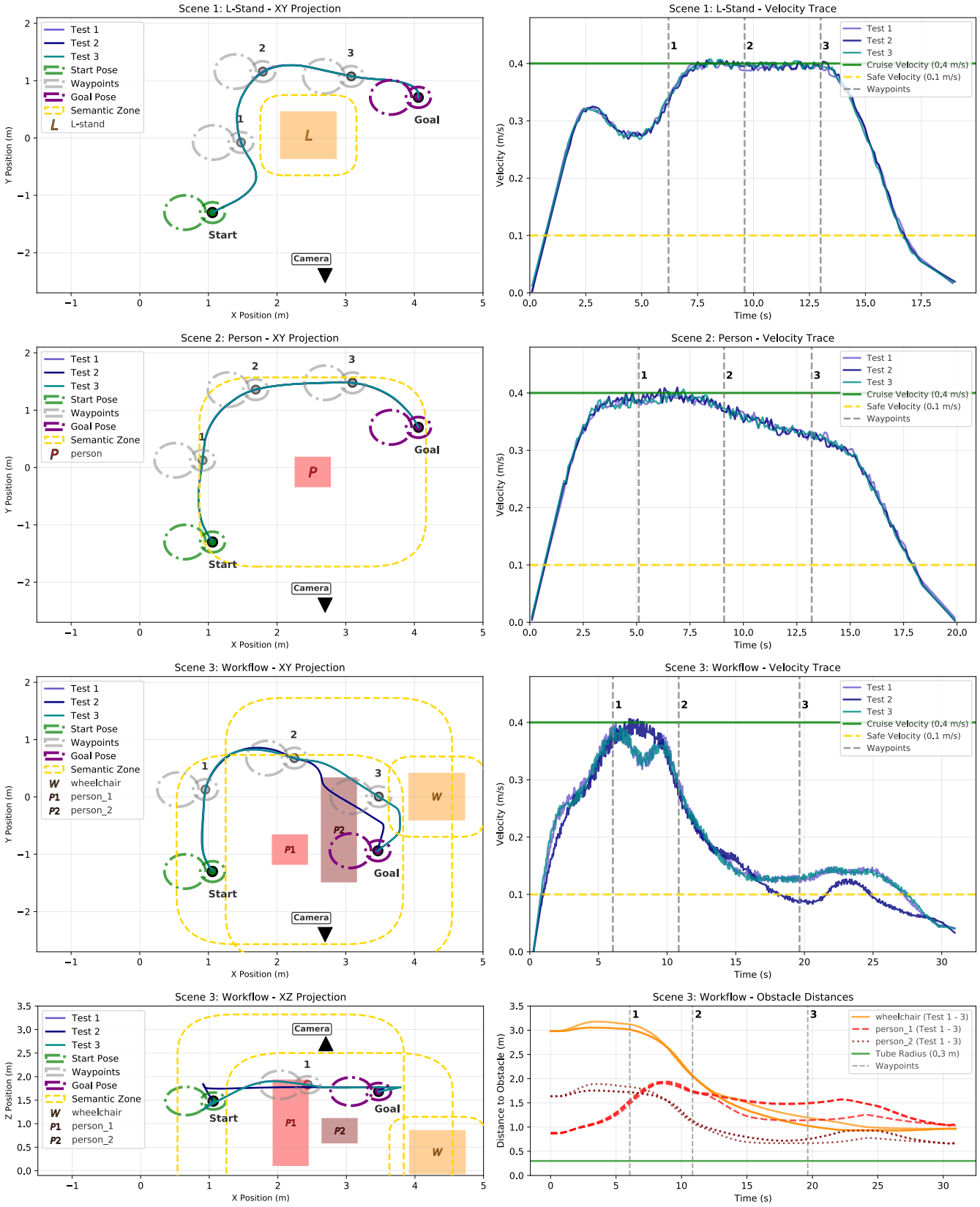

Fig. 5. Experiment results: (left) 2D-Projections of detected objects, safety zones and the robots path. (right) Velocity and obstacle distance traces.