Traveling Drinksman — A Mobile Service Robot for People in Care-Homes

Michele Persiani[a], Çağatay Odabaşı[b], Florenz Graf[b], Mohit Kalra[b], Thomas Hellström[a], and Birgit Graf[b]
[a]Department of Computer Science, Umeå University, Umeå, Sweden
[b]Fraunhofer IPA, Stuttgart, Germany

Abstract

This paper describes ongoing work on the development of a service robot for serving drinks to people sitting at tables, for example in the recreation room of a care-house. The robot, denoted the *Traveling Drinksman*, should be able to detect the occupied tables, navigate safely according to defined policies, and interact with the humans sitting to serve them a drink. We present initial results addressing all of these problems with different sub-modules, including numerical results for the human detection module.

1 Introduction

With the globally growing elderly population foreseen for the coming decades, the need for infrastructure dedicated to elderly care has also increased [18]. As intelligent service robots become increasingly available, there will be many possibilities to delegate low-level, repetitive tasks to the robots, thereby giving human workers more time for interpersonal care.

Preliminaries user studies showed that robots can have a positive impact on elder population in care-house environments. In [19], the authors show how the introduction of a social robot strengthened the relationships between the guests and overall increased well-being. While also negative responses to socially assistive robots have been measured by e.g. the study in [12] (such as psychological factors emerging from the need of using assistive machines), the same study shows that, in general, there is a positive correlation between age and robot acceptance by the elders.

From a work perspective, it is still unclear the impact that service robotics will have on human labor [3], and most likely a balance between full automation and human-robot collaboration contains the sweet spot for service robotics. When seeing services as robot manufactured products, there is a fairly low threshold above which it becomes impossible for untrained human workers to resolve problems that might occur with the robot or with the provided services. Hence, fully automated service robots should keep a low level of complexity to avoid the constant need of expert personnel for the maintenance of the robots and their procedures.

Along these lines, this paper describes ongoing work on the development of a service robot for serving drinks, to be deployed for example in the recreation room of a care-house. However, the proposed solution can also in the future be extended towards more social aspects, such as entertainment or chit-chatting. The proposed robot - the *Traveling Drinksman* - is expected to take care of the drinking needs of several people in a room. The robot's task is to continuously detect and serve people at all occupied tables.

Figure 1 The robot is approaching a human to ask if he wants to drink something.

We extend previously developed work [4] by implementing two major components: a human tracker, that allows to track in real-time the person positions in the room, and a planning system, that is used to plan how to serve the detected persons most efficiently.

2 Related Work

There is a fairly large body of research dedicated to service robotics, with one of the factors driving the field being the need for enhancing the infrastructure for the elderly care [18, 8]. In this regard, robotics offers a wide range of possibilities to support the working staff in e.g. care houses. Few examples of developed robots are nursing robots [7], companion robots [19], or robots for assistance in handicapped mobility scenarios [5, 17].

Due to their inherent complexity, robot architectures for service robotics should be able to be extended and to host newly added components systematically and comprehensively such that the provided services become easy to man-age. The authors in [8] propose that languages such as the Unified Modeling Language are suitable to support such a requirement in architecture modularity, also for its suitability for system engineering methods.

Since our developed architecture provides a single service, we don't provide such a high-level description of the system and focus more on the implementation details. We use the Robot Operating System (ROS) [13], which is modular by construction and allows a seamless integration of components through their defined ROS API. ROS is a well-known robot development framework and is utilized by many projects. Other robot architecture frameworks are utilizable, such as ones developed in industry e.g. [7].

Two common problems in robotics that we also faced in this work are detecting persons and planning sequences of actions. Detecting humans can be necessary when the robot is working in shared spaces, and several solutions have been explored in past years using different combinations of sensors. For example using video [20], lidars [16], or sonars. Classification methods received an overall massive improvement in recent years due to the dawn of deep learning methods. Given a properly supervised dataset, these methods can provide reliable detections, also in real-time due to the parallelizable computations of neural networks. With this respect, we utilize a YoloV3 [15] classifier that provides bounding boxes for the trained image regions.

Planning is the task of finding the optimal sequence of actions that from initial conditions to achieve a desired goal condition. Many robotic applications using planners have been developed in past decades such as based on STRIPS [11], Hierarchical Task Networks, etc. For this work, we selected the Planning Domain Description Language (PDDL) [10], which is a standard language to specify planning domains for what is usually referred to as classical planning.

3 Method

We utilize an existing service robot prepared for the task of serving drinks [1]. The drinks are stored in plastic glasses in the robot's shelf-type hardware. The robot has an omnidirectional mobile base and uses a map to autonomously navigate collision-free through the environment. For making the robot approach and serve a given target position, we applied a previously developed approach module [4]. When triggered, this module makes the robot to approach a predefined area close to the person. By using an integrated tablet computer, users can request any of the available drinks stored inside the robot. In the proposed scenarios this procedure repeats until all present persons have been served.

We extend the functionalities offered by the approach module by new methods to increase autonomy, robustness and time efficiency. A robust human detection module (Section 3.1) and a global path planning optimization (Section 3.2) have been emerged as useful during previous tests in elderly care homes.

To realize the described robot behavior, we use the open-source software framework Robot Operating System (ROS) for development and testing. Figure 2 presents the three main parts of existing and new ROS modules, as well as their communication:

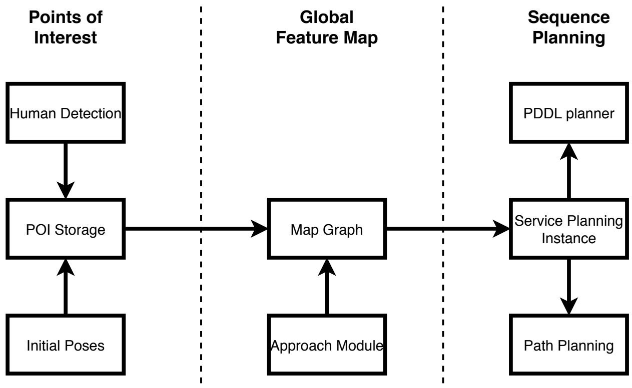

Figure 2 ROS software architecture of the high-level planning components of the system. Each block corresponds to a developed ROS node. Edges indicate how messages flow in between the nodes.

The first part provides the points of interest (POIs). In our case, these correspond to the person positions, the robot position, the glass refilling position, and the robot home position. Initially, these poses are set manually. The human detection simultaneously updates the person positions for approach and the next serving.

Human Detection: This ROS-node subscribes to the color and depth image of the robot for head detection. In the case of detection, the node provides the position of the head from the perspective of the camera. Using the transformation of the robot and from the localization, the node transforms the position into the global map frame.

Initial Positions: It provides the available sitting positions, the refilling, and home pose.

POI Storage: It publishes all the gathered detected positions. These include the person's global positions extracted from their head positions, the robot position, the sitting positions, the robot's home position, and the refilling position.

The second part of the architecture combines the global grid map of the robot with the POIs and the robot poses to approach these. Simultaneous Localization and Mapping provides the global grid map. Since the robot has to serve the drink close to the person, the approach node uses the costmap to provide the best not-occupied robot pose around the person pose. During the robot approaches a person, the approach module updates the best robot pose based on the current sensor information and human pose. This is required because of occluded and dynamic obstacles.

Approach Module: This module takes the pose of the person and returns the best robot pose for Robot Human Interaction as a ROS-Service.

Map Graph: Combines the robot map data with the approaching positions, the refilling and the home position.

The third part calculates the optimal overall sequence for serving the present people. First, the **Map Graph** node interconnects all POIs and store them as edges on a graph. Then, an external path planning service executes an A* search algorithm on the costmap to provide an estimated travel length. Afterward, we associate the travel length as cost and assign them to the edges. Finally, the PDDL-planner performs the overall planning, constraint by starting at the current robot position and ending at the refilling position. The PDDL-planner returns the optimal sequence of poses, that executes the state machine of the robot to serve all present person by a drink.

Path Planning: This node provides a path planning service using the global costmap map, which is used to compute the path length between two given sets of coordinates. The service request includes the positions and the service response includes the travel length.

Service Planning Instance: It is the node computing the service planning instance, that is updated in real-time using the global map. It allows us to plan for optimal paths over the feature map. For every plan request, a PDDL instance is generated using the feature map. The obtained plan is then transformed into sequences of approaches, refill, or go idle for the approach module. An edge associates every pair of nodes with associated weight obtained by calling **Path planning**.

PDDL Planner: This node provides a planner for PDDL problems. It is connected to the Map Graph to receive the graph and the travel lengths in between. After calculation, it returns the optimal sequence to the state machine.

More details on human detection and planning are given in the following sections.

3.1 Human Detection

In addition to traditional robot vision challenges such as obstacle detection, in our scenario, people are visible from all orientations. They may also be partly occluded, for example by a chair or other furniture. This causes problems for state-of-the-art approaches aiming at detecting the whole body. The other constraint is the need for 3D information. The robot needs to detect the precise position of the person in 3D space to approach him optimally. Therefore we decided to detect the head of people with an active stereo RGB-D camera that provides color and depth images.

Creating and hence finding a realistic RGB-D head dataset in uncontrolled environments is quite hard. Since the robot operates in uncontrolled environments like nursing homes, a deep network trained with a controlled dataset would not generalize to our problem well. That is why we start with the RGB-D human dataset [15] by Spinello et al., which contains thousands of full-body annotated RGB-D images from the people passing through a university hall. In the dataset, the people are mostly walking, standing, and their heads are visible from different angles to the camera. We extend it by annotating the head bounding boxes on both RGB and Depth domains.

We train two separate head detectors for RGB and Depth domains since both RGB and Depth domains have their advantages and disadvantages. The head detectors are object detection networks called YoloV3 [14]. RGB head detector is robust against sunlight but fails in the weak light. The depth detector works well in low light but is sensitive to sunlight due to the nature of the infrared-based depth camera.

The resulting RGB head detector is robust against sunlight but fails in the weak light. The depth detector works well in low light but is sensitive to sunlight due to the nature of the infrared-based depth camera. For these reasons, we fused the outputs of the two detectors to improve robustness. The fusion is done at the bounding box level, where the bounding boxes generated by the two modalities are fed into the SORT tracker [2], which then provides the fused detections.

Lastly, we transformed the head positions from the camera frame into the map frame to obtain all head detections in one frame for path planning. The head detection provides bounding boxes around the head block. To capture the distance of it, we calculate the median value inside the bounding box, since the assumption is that the head must be the most prominent and dominant object inside of the bounding box. Median filtering eliminates the background and provides us a simple, fast, and reasonably accurate estimation.

3.2 Planning

The robot has a map of the environment as well as the fixed table positions. The robot continuously updates the map with obstacles and persons by observing the environment with its available sensors [1]. As several people may be detected at the same time, a global planner determines the order in which the sitting positions should be served. Only sitting positions with at least a detected human are due for serving.

The defined problem corresponds to the Traveling Salesman Problem (TSP), which is the problem of visiting a set of points of interest while minimizing a cost metric (e.g. traveled path length), returning then to the starting point. In our case, there are additional constraints that the planner should consider, and for this reason, we used a general-purpose PDDL based planner, rather than a specific TSP algorithm. For example, only a limited number of glasses are available inside the robot, and plans should include also actions to refill the robot.

We selected Metric-FF [6] as a planner, which complies with PDDL 2.1. As the search strategy, we selected Dijkstra's algorithm. In our experiments the inclusion of a heuristics (i.e. turning the algorithm into A*) decreases the planning time but our tests showed that the planner also often returned sub-optimal plans. Our untested hypothesis is that the planner's available heuristics-based of deleting

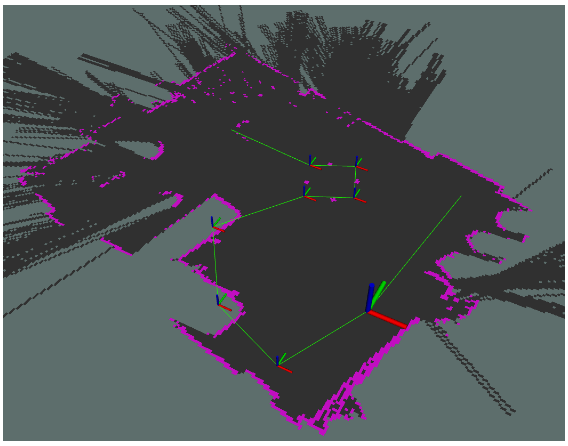

Figure 3 Obtained service plan for a scenario with 7 persons. From its initial position, the robot visits the refilling position, marked by the bigger axes, to then pass by every sitting position. Every plan ends with the robot reaching a predefined idle position. Service plans are then transformed into sequences of approaches for the approach module.

negative effects from the PDDL operators are sub-optimal for TSP planning instances (see Section 4).

A PDDL instance based on the map of the environment is continuously updated, and the planner is utilized to determine the path to follow in order to visit all currently detected occupied sitting positions, i.e. positions with at least a head detected in close proximity. While the goal of the generated plans is always the one of visiting all occupied sitting positions, the planning instance can be configured to minimize a chosen metric such as required time, traveled distance, a "First In First Served" policy, or a priority queue policy. The generated plans are dependent on the selected policy and correspond to an ordered list of sitting positions that is forwarded to the approaching module described in previous work [4]. The approaching module makes the robot approach the target position and start the serving procedure. After a position is approached, getting a drink or dismissing the robot through the tablet makes the robot continue to the next position in the plan.

4 Evaluation

We implemented planning between different tables in a real care house environment. Planning in a real scenario confirmed that our setting allows for the service robot to reach all of the tables in the most efficient way (by minimizing e.g. traveled length) from any position while avoiding obstacles. Obstacle avoidance is provided by the local planner. The robot plans also successfully consider refilling necessities by having the robot returning to its refilling station when the drink storage is empty.

A video of the robot executing a plan to visit the tables of a re-creation room at a care-house facility is at *https://youtu.be/hofICXrIvhE*. In the shown experiment, head tracking is disabled and the robot always visits all of the pre-programmed sitting positions.

We further compare the A* search strategy with Dijkstra

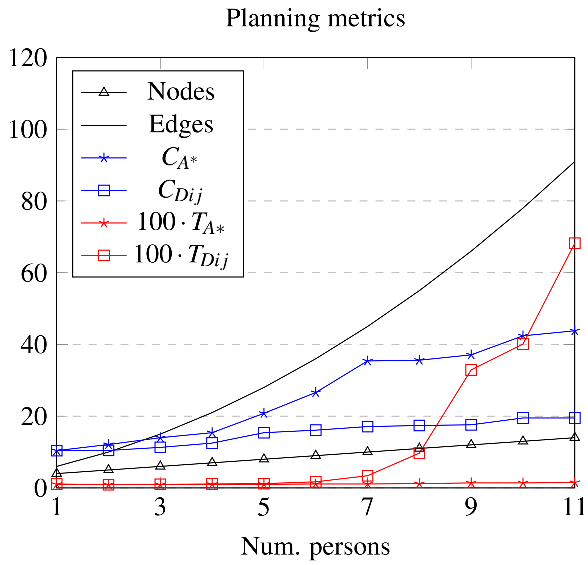

Figure 4 Metrics measurements for planning using the A* search strategy and Dijkstra's algorithm. $C_{\_}$ is the cost in meters of the obtained plans, $T_{\_}$ the plan time in seconds. Nodes and edges indicate the size of the planning instance.

Table 1 Average Precision (AP) for the RGB and Depth-based head detectors, computed on the test set for different IoU thresholds.

IoU	AP for RGB	AP for DEPTH
50%	0.90	0.89
60%	0.81	0.78
70%	0.68	0.58
80%	0.31	0.36

in scenarios of different sizes, up to 11 persons. If Figure 4 we show how A* provides plans that are in the most number of cases sub-optimal. On the other hand, the plans obtained using Dijkstra are always optimal but, as the graph shows, Dijkstra doesn't allow to scale to scenarios with many persons.

In order to test the human detector, we trained the head detector based on YoloV3 to detect people sitting at tables. The dataset utilized to train the classifier contains both RGB and Depth images. To compare and find the optimal modality, we trained different detectors for each modality. For training, we used 3043 RGB and Depth images, and for testing 351 images. Average Precision (AP) for the test set is presented in Table 1. Each row in the table shows the average precision (AP) for different values of *Intersection Over Union* (IoU). IoU quantifies how well the detector's predicted bounding box overlaps with the ground truth bounding box. The results show that both modalities work successfully.

The service robots always come to the proximity of the people, since they need to interact with them. The distance between humans and the robot would decrease less than 1m, and as the range reduces, the shadows in the depth images increases, which makes it harder to detect heads. Also, the dataset may not cover 360 degrees view of the head. Therefore, we create the following test bench. A

Table 2 Close up Head detection results on the robot under real conditions.

	# of Images	Correct Detection	Rate
Trial 1	133	39	0.29
Trial 2	89	28	0.31
Trial 3	80	42	0.525

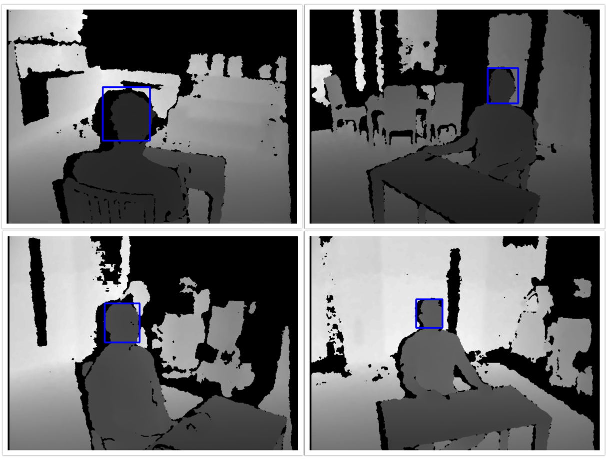

Figure 5 Success cases of the close-up head detection on the robot.

person is sitting on a chair in front of a table, as in Figure 5 and Figure 6. Then, the robot starts moving around the person so that it sees the head from every angle. The distance between the robot and the person is around 1m to 2m throughout the experiment. In Figure 5, there are blue bounding boxes around the head. These are the success cases as opposed to Figure 6.

There are three trials with two participants. In total, the robot collects 302 images. The results are presented in Table 2. As seen, the results are dramatically deviating from the dataset results. The foremost reason is that the head is too close to the camera limits, but at the same time, these are the situations that our service robots face during the care-house tests. That is why we need to test it against these situations as well. The main takeaway would be not to rely on single-frame detection but to rely on the fusion of detection over time.

The navigation stack on the robot calculates the accurate position of the robot by using three lidar sensors. We use this information to evaluate the depth estimation method. That is why we calculate the mean and standard deviation of the head position estimations through the time. The standard deviation on X-axis is 0.49 m and 0.34 m on Y-axis. Although these numbers seem huge, we need to consider that the detection experiments are pushing the limits of the camera, and there is an error caused by navigation. Furthermore, the depth estimation algorithm is a simple, fast, median value-based method. Also, this error can be compensated by an adaptive approaching algorithm presented in [4].

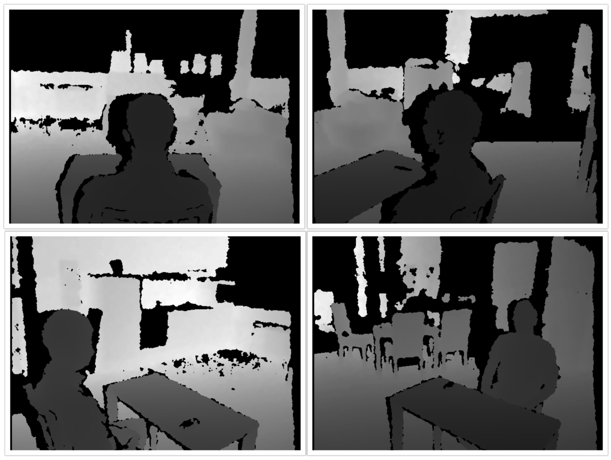

Figure 6 Failures of the close-up head detection on the robot.

5 Conclusions and Future Work

This paper describes ongoing work to design a service robot that efficiently serves drinks to the people in a room. We realized a human detection system based on depth and video cameras, and a planning systems suitable for the envisioned application. While their integration is still missing, we tested their performances separately, as shown in Section 4. The planning subsystem can be optimized in two ways. The first is by implementing a heuristic that is suitable for TSP problems. This would allow to use A* instead of Dijkstra, thus improving how the plan computation times scale in the number of graph nodes. The second way is to prune the search graph as right now the graph has several edges that scales quadratically in the number of nodes. Pruning the graph would allow to remove suboptimal edges, such as connecting nodes on the other side of the room with each other.

We tested the planning system in a real environment and verified its functioning. Preliminary results in real scenarios proved to be promising. However, at the moment the proposed system makes use of some simplifying assumptions. For instance, we explicitly assume that possible sitting locations are known and fixed. This is believed to be a realistic assumption since it is the case for most cafes, restaurants, and canteens. Future versions of the proposed system can remove this assumption to be able to serve also people standing at any position in the room. Furthermore, we identified cases where clustering persons and serving the clusters altogether can be a better solution rather than always serving individual persons. This can be the case when several persons are seated very closely. We also assume that no persons are walking during serving, and all persons in the room are sitting down by a table. Additional safety systems can be added to allow co-occupation of the space in the room between robot and persons, for example, [9] introduces six safety-related rules that the robot should follow when navigating spaces that can be occupied.

6 Literature

[1] Simon Baumgarten, Theo Jacobs, and Birgit Graf. 2018. The Robotic Service Assistant – Relieving the Nursing Staff of Workload. *ISR 2018 - 50th International Symposium on Robotics*.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. 3464–3468.

[3] Michael Decker, Martin Fischer, and Ingrid Ott. 2017. Service Robotics and Human Labor: A first technology assessment of substitution and cooperation. *Robotics and Autonomous Systems* 87 (2017), 348–354.

[4] Florenz Graf, Çağatay Odabaşı, Theo Jacobs, Birgit Graf, and Thomas Födisch. 2019. MobiKa-Low-Cost Mobile Robot for Human-Robot Interaction. *International Symposium on Robot and Human Interactive Communication (RO-MAN), New Delhi* (2019).

[5] M Hans, B Graf, and RD Schraft. 2002. Robotic home assistant care-o-bot: Past-present-future. In *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*. IEEE, 380–385.

[6] Jörg Hoffmann and Bernhard Nebel. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14 (2001), 253–302.

[7] Chandimal Jayawardena, I Han Kuo, Ulrike Unger, Aleksandar Igic, Richie Wong, Catherine I Watson, RQ Stafford, Elizabeth Broadbent, Priyesh Tiwari, Jim Warren, et al. 2010. Deployment of a service robot to help older people. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5990–5995.

[8] Minseong Kim, Suntae Kim, Sooyong Park, Mun-Taek Choi, Munsang Kim, and Hassan Gomaa. 2009. Service robot for the elderly. *IEEE robotics & automation magazine* 16, 1 (2009), 34–45.

[9] Chi-Pang Lam, Chen-Tun Chou, Kuo-Hung Chiang, and Li-Chen Fu. 2010. Human-centered robot navigation—towards a harmoniously human–robot coexisting environment. *IEEE Transactions on Robotics* 27, 1 (2010), 99–112.

[10] Drew McDermott. 1998. PDDL-the planning domain definition language.

[11] Nils J Nilsson. 1984. *Shakey the robot*. Technical Report. SRI INTERNATIONAL MENLO PARK CA.

[12] Maribel Pino, Mélodie Boulay, François Jouen, and Anne Sophie Rigaud. 2015. "Are we ready for robots that care for us?" Attitudes and opinions of older adults toward socially assistive robots. *Frontiers in aging neuroscience* 7 (2015), 141.

[13] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, Vol. 3. Kobe, Japan, 5.

[14] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[15] Luciano Spinello and Kai O Arras. 2011. People detection in RGB-D data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3838–3843.

[16] Tapio Taipalus and Juhana Ahtiainen. 2011. Human detection and tracking with knee-high mobile 2D LIDAR. In *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 1672–1677.

[17] Tapio Taipalus and Kazuhiro Kosuge. 2005. Development of service robot for fetching objects in home environment. In *2005 International Symposium on Computational Intelligence in Robotics and Automation*. IEEE, 451–456.

[18] Department of Economic United Nations and Population Division Social Affairs. 2019. World Population Ageing 2019: Highlights.

[19] Kazuyoshi Wada and Takanori Shibata. 2007. Living with seal robots—its sociopsychological and physiological influences on the elderly at a care house. *IEEE transactions on robotics* 23, 5 (2007), 972–980.

[20] Torsten Wilhelm, Hans-Joachim Böhme, and Horst-Michael Gross. 2002. Sensor fusion for vision and sonar based people tracking on a mobile service robot. In *Proceedings of the International Workshop on Dynamic Perception*. 315–320.