Exploiting Augmented Reality for Extrinsic Robot Calibration and Eye-based Human-Robot Collaboration

Daniel Weber
University of Tübingen
Tübingen, Germany
daniel.weber@uni-tuebingen.de

Enkelejda Kasneci
University of Tübingen
Tübingen, Germany
enkelejda.kasneci@uni-tuebingen.de

Andreas Zell
University of Tübingen
Tübingen, Germany
andreas.zell@uni-tuebingen.de

2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI) | 978-1-6654-0731-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/HRI53351.2022.9889538

***Abstract*—For sensible human-robot interaction, it is crucial for the robot to have an awareness of its physical surroundings. In practical applications, however, the environment is manifold and possible objects for interaction are innumerable. Due to this fact, the use of robots in variable situations surrounded by unknown interaction entities is challenging and the inclusion of pre-trained object-detection neural networks not always feasible. In this work, we propose deploying augmented reality and eye tracking to flexibilize robots in non-predefined scenarios. To this end, we present and evaluate a method for extrinsic calibration of robot sensors, specifically a camera in our case, that is both fast and user-friendly, achieving competitive accuracy compared to classical approaches. By incorporating human gaze into the robot's segmentation process, we enable the 3D detection and localization of unknown objects without any training. Such an approach can facilitate interaction with objects for which training data is not available. At the same time, a visualization of the resulting 3D bounding boxes in the human's augmented reality leads to exceedingly direct feedback, providing insight into the robot's state of knowledge. Our approach thus opens the door to additional interaction possibilities, such as the subsequent initialization of actions like grasping.**

***Index Terms*—augmented reality; eye tracking; human-robot collaboration; object detection; robot calibration**

I. Introduction

More and more robots are being used in environments within a close proximity to humans. The possible applications of robots are diverse and possible interactions with humans are multifaceted. Whether as a tour guide in museums [1] or as an assistant in supermarkets [2], each interaction scenario involving robots has its own challenges. Furthermore, successful technical advances in augmented reality (AR) have promoted the interaction and collaboration between humans and robots. Consequently, AR has found application in factories [3] and in imitating assembly processes that a human demonstrates [4].

The long list of possible use cases results in at least as many tasks that need to be solved. Among these tasks, the conveyance of the interaction context, such as the specification of an object to interact with, is particularly challenging. Many tasks, especially object detection, can be accomplished through the benefit of machine learning methods, such as neural networks. While advances in machine learning have had a major impact on the development of human-robot interaction, there are also some drawbacks. Typically, many of these approaches require a sufficient amount of available training data, which cannot always be guaranteed. This data dependency ties the deployment of robots to predefined scenarios and limits interaction with the environment, e.g. with unknown objects that cannot be detected. For example, if a supermarket changes its assortment of products, the robot can usually only interact with the new items if it has learned them beforehand. Our goal is to enable data-independent object detection for cases where no training data is available.

Another even more fundamental problem is the calibration of the robot. In order for a robot to perceive a scene, its sensors, such as a fixed, but adjustable camera, must be properly targeted and its position relative to the robot base must be known. Therefore, the scene or the purpose of the operation needs to be identified in advance, at least to a certain degree. In addition, calibration of extrinsic robot parameters is often laborious [5] since, in most cases, either the existence of a second sensor in the form of a laser scanner or another camera is assumed, or expensive external tools are used. Both make subsequent adjustments in response to changing circumstances difficult. On top of that, the authors of [6] noted that robots in public attract the curiosity of people, especially children. In particular, children tend to touch the robot or exhibit abusive behavior when unobserved. This, in turn, can often lead to misalignments of the robot's sensors and require frequent recalibrations. A less time-consuming calibration method is beneficial in this case.

In this work, we attempt to fill this gap at the intersection of research fields of human-robot interaction, eye tracking, and augmented reality. More specifically, we aim at a flexible deployment of robots, detached from predefined scenarios by leveraging collaboration with humans instead of training data. Our contribution with this work is twofold:

On the one hand, we present a convenient method for determining the transformations between the robot and a

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germanys Excellence Strategy – EXC number 2064/1 – Project number 390727645.

sensor, in our case a camera, as well as between the human and the robot. With our method, time does not have to be spent repeatedly for each calibration run, but only once during the initial setup. Subsequent calibrations can then be performed in a matter of seconds, making the method particularly suitable for situations where frequent recalibrations are required. The calibration can be executed at any time during runtime and allows both the human and the robot to move freely.

On the other hand, after utilizing said calibration, we fuse existing point cloud clustering methods with eye-tracking information to showcase the 3D detection of unknown objects. More precisely, the robot and the human collaborate so that the robot detects which object the human is looking at without knowing the interaction context in advance. Based on our calibration, we can establish a connection for continuous exchange of interaction information. The human continually provides the robot with gaze data and the robot responds with the bounding box of the target object. The human's perception is augmented by integrating the robot's feedback directly into the human's reality. All of this without training and in an online fashion, not after the fact.

In summary, our most important contributions are as follows:

1) We show and evaluate a calibration method via an augmented reality interface that is suitable for the deployment of robots in ever-changing scenarios and allows the robot's capabilities to be further extended by providing it with a new, additional real-time information channel — the human gaze.
2) We are the first to use augmented reality in a human-robot collaboration scenario to segment unknown objects in three-dimensional space without the use of neural networks. We also provide direct feedback to the human, enabling subsequent interactions.

The remaining part of this paper is structured as follows. After a discussion of the related work, in Section III we describe and formalize our approach in detail. Our results and the limitations of our approach are discussed in Section IV. Section V concludes this work and gives an outlook on our future activities.

II. Related Work

Employing gaze information to achieve human-robot interaction with unknown objects requires significant multidisciplinary efforts, which we will discuss in this section. From how 1) robots collaborate with humans, to 2) augmented reality in robotics, 3) robot calibration and 4) 3D object detection, to 5) mapping human gaze to a known frame of reference and 6) previous applications of eye tracking in the context of computer vision.

A. Collaborative Settings

In recent years, scenarios in which humans and robots work together side by side have gained attention. Interaction with robots invites interesting possibilities for beneficial collaboration in human everyday life [7]. In [8] a system was presented, that enables a robot to perform cooperative search with a human teammate, where the robot assists the human teammate in navigation to the search target. Collaboration between human and robot is also widespread in industrial environments, such as in assembly tasks [9], surface finishing applications [10] or welding work [11]. In addition to the application in industry, robots have more and more of a social purpose. Due to the lack of medical personnel and rising costs in the health sector, social robots are increasingly being used in the health care system [12]. They are typically deployed for surgical assistance [13], rehabilitation [14], elderly care [15], and as companion robots [16].

B. Augmented Reality in Robotics

With the increasing availability of various augmented reality glasses, the impact of AR in research and industry has also grown. In [17], head orientation and pointing gestures were used to control an industrial robot arm for pick-and-place tasks. However, the arm was fixed in the room to facilitate coordinated transformation by means of a marker attached to the wall and the set of interaction objects was fixed. An AR device was also used by [18] in a multimodal communication setup to help a robot decide which object a human pointed to using gestures, gaze, and speech. In this setup, again, the objects were predefined and their positions were additionally measured accordingly in advance. The authors of [19] visualized sensor data from a robot using AR glasses. All sensors, though, were already calibrated, which additionally allowed for the utilization of a localization algorithm. Following on from this, the same authors recently used a deep learning-based approach in [20] to determine the mutual position of the robot and AR device. Nevertheless, this approach was not suitable for real time scenarios due to the limited computational capacity of the AR glasses. Within a manipulation frame, [21] used pre-trained 2D object detectors to determine 3D bounding boxes. This required a fiducial marker to be in the field of view at all times and was limited to a single object per pass. Such problems of ambiguity we will solve with gaze.

C. Extrinsic Robot Calibration

Modern robots are usually equipped with a large number of sensors, most frequently RGB-D cameras. Ensuring their operability requires the most accurate calibration of extrinsic parameters, i.e. their position on the robot base. A classical approach to this is the use of calibration patterns. By observing the pattern, [22] determined the mutual position between a camera and a 2D laser range finder. With only one image, but several markers, [23] succeeded in calibrating a camera with respect to a second camera or a laser scanner. In both cases, however, the existence of a second sensor was a prerequisite and a common field of view of these two was mandatory.

In [24], a framework for parameter estimation using a motion capture system was built. While such systems, including Vicon [25] or OptiTrack [26] can be very accurate, they require careful calibration beforehand. In addition, they are time-consuming to set up and expensive due to the amount of

hardware involved, such as multiple cameras. We try to close this gap with a fast and universally applicable method.

D. 3D Object Detection

Due to the higher level of difficulty, many 3D object detectors are inspired by detection in 2D. This includes the projection of the point cloud into bird's eye view [27] or cropping on frustums based on 2D bounding boxes [28], [29]. Few also operate on the point clouds directly [30]. What they all have in common, however, is that they rely heavily on the availability of training data and focus predominantly on road scenes or furniture pieces. An approach to instance segmentation of unseen objects was proposed by [31]. While they did not need real world images, they had to generate a large amount of synthetic data for which 3D CAD models were required. As an alternative to neural networks, [32] used a saliency-driven approach to detect unknown objects. Nonetheless, the results were influenced to some extent by a parameter that depended on the size of the objects, and, due to the long calculation time, the system was not suitable for real-time applications.

E. Gaze Mapping

Mapping gaze data from a moving eye tracker to another coordinated frame is still an unsolved challenge and thus ongoing research [33]. One possible solution to this challenge is feature matching. For example, [34] achieved promising results with such a method, however it reaches its limit with diverging camera perspectives. The authors also found that better robustness at less computational cost was achieved with fiducial markers [34]. Such markers were used in recent works by [33] and [35], among others. One disadvantage of this approach is that fiducial markers have to be in the field of view of both cameras, restricting thus movements. With our AR-based approach, we overcome this problem and ensure stable gaze mapping despite free movement and thus independent of the field of view.

F. Eye Tracking and Computer Vision

Although not yet very popular, there are some works that have tried to solve computer vision problems with eye tracking. In [36] for example, features in the neighborhood of human fixations were matched to features of known objects to determine the class of the respective object. Statements about its position could not be made in this way. The authors of [37] reduced the number of superpixels for salient object detection with gaze data. In contrast to our approach, however, this required both multiple gaze points and training data. With only one gaze point, [35] managed to drastically reduce the number of candidate bounding boxes of a region proposal method, but this method is only applicable in 2D.

In this work, we build on existing research to improve human-robot interaction. While speech and gesture are popular channels for communication, gaze is challenging [38] and often neglected. In the following, we link eye tracking and augmented reality to address classical calibration problems as well as data-independent 3D object detection in a collaborative manner.

III. Methods

In this work, we propose finding 3D positions of unknown objects by incorporating human gaze into the robot's segmentation process. For this purpose, we first introduce the interface used to communicate with the robot. Subsequently, we present an extrinsic robot calibration method, which is particularly characterized by its flexibility and ease of execution. In our case, we calibrate a camera's position relative to the robot's base, but in principle the method can be applied to any sensors. Finally, we explain the segmentation process that applies said methods.

A. Augmented Reality Interface

All interaction with the robot is guided via an augmented reality interface and serves as a two-way communication channel between human and robot. In this way, we can, for example, control the movement of the robot, access the robot's camera feed, or perform the extrinsic calibration between robot and its camera. In addition, we can provide the robot with the human gaze data and display the results of the object detection. We use the HoloLens 2 from Microsoft, a head-mounted pair of mixed reality glasses with a built-in eye tracker. For the development of AR applications, Microsoft provides an open-source cross-platform toolkit called Mixed Reality Toolkit (MRTK). The creation and development of our interface takes place in the game development environment Unity. We use the versions MRTK 2.7.2 and Unity 2019.4.29. For the actual communication between the HoloLens' Universal Windows Platform (UWP) and the robot operating system (ROS), we resort to the UWP version of ROS# [39], a set of open source software libraries and tools for communicating with ROS from Unity applications. On system startup, the robot launches ROS#'s file_server package as well as rosbridge_server from the rosbridge_suite. As soon as the AR interface is started on the HoloLens, it immediately establishes a connection with the robot via Wi-Fi. Thereupon, ROS# uses the rosbridge protocol to send JSON based commands via WebSockets, enabling the deployment of custom publishers and subscribers. During runtime, the menu of our interface can be opened by looking at the user's palm. Created virtual objects can then be selected by voice or gestures. For example, menu buttons can be simply pressed with a finger or other virtual objects can be selected by looking at them and pinching the thumb and index finger together or saying "select".

B. Calibration & Gaze Estimation

The incorporation of the human gaze into the robot's world requires the estimated gaze to be mapped from the reference frame of the human, provided by the HoloLens, into the robot's frame of reference. For this purpose, the transformation can be computed either directly, if the pose of one device in the frame of the other is known, or through indirect co-location by finding corresponding points in the image of the two associated

cameras [35]. The former is often difficult to realize in practice, while the latter has some disadvantages, namely limiting the view of both participants to an overlapping field of view. Furthermore, for the robot to interact with objects in its field of view, the position and orientation of the robot's camera relative to its base must also be known. The solution comes in the form of augmented reality, which we can employ as a bridge. If we create virtual counterparts corresponding to the real poses of the respective frames, we become acquainted with the transformation between frames through the transformation between virtual elements. More precisely, we determine the mutual position of the robot and the robot's camera sensor by aligning them with the corresponding virtual objects and calculating the transformation occurring in between in the virtual space of the HoloLens. The authors of [4], [8] and [19] did something similar to align the coordinate systems of a robot and that of a HoloLens. However, in their case, all the necessary robot sensors had been calibrated beforehand. The advantage of our calibration method is that it splits the usual time-consuming extrinsic sensor calibration into multiple parts. In case of frequent calibrations, only the fast part needs to be repeated.

For us, the approach described means we can intertwine both of our problems: On the one hand, we can calibrate the position of the robot and the camera in relation to each other, and, on the other hand, we can establish a direct transformation between HoloLens and the robot, which means that the robot is aware of the gaze point at all times regardless of the field of view. An overview of the underlying pipeline is shown in Figure 1.

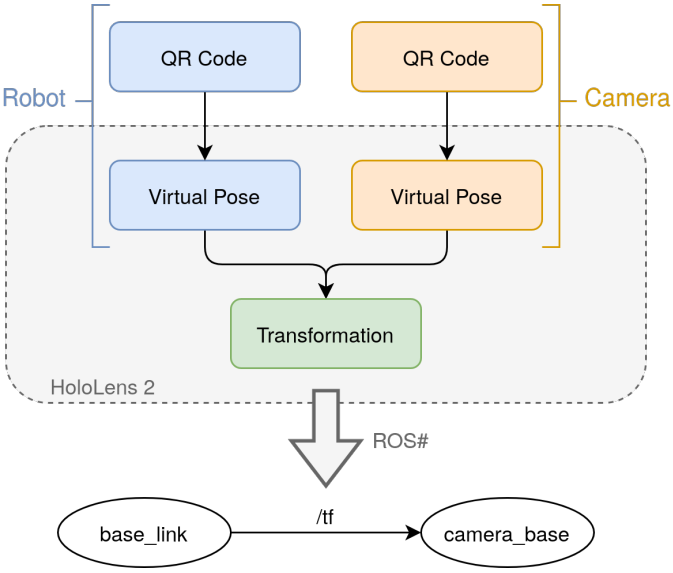


Fig. 1: The QR codes specify the position of the virtual versions of the robot and the camera. The intervening transformation can be determined in the virtual world of the HoloLens 2 and is then published via ROS#.

We start by determining the poses of the two frames of interest. This is, in our case, the so called base_link on the robot side and the camera_base frame on the camera side. In principle, however, any frame can be used whose origin is known relative to a point on the housing. To align real and virtual versions of the robot and its camera, we attach fiducial markers in the form of QR codes (see Figure 2) as they allow for robust and inexpensive detection. The HoloLens 2 is moreover capable of detecting QR codes at the system level in the driver. However, we have to consider that there will be an offset between the pose of the markers and the actual frame. So let $\{b\}$, $\{m_b\}$, $\{c\}$ and $\{m_c\}$ be the coordinate frames of the robot's base (base_link), the QR code on the base, the camera (camera_base), and the marker on the camera, respectively. For two frames $f_1, f_2 \in \{\{b\}, \{m_b\}, \{c\}, \{m_c\}\}$, let the transformation from f_1 to f_2 be denoted by ${}^{f_1}T_{f2} \in \mathrm{SE}(3)$. The connection between the frames can be illustrated by the following transformation graph:

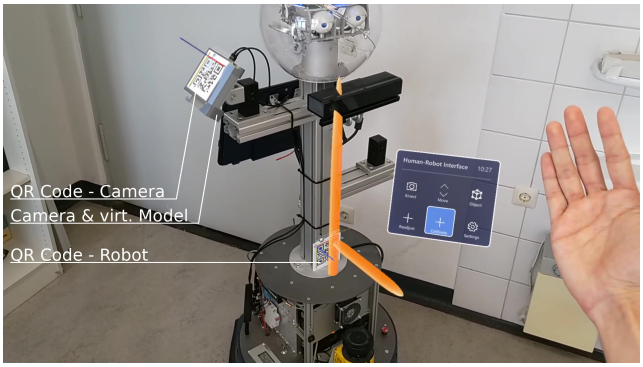
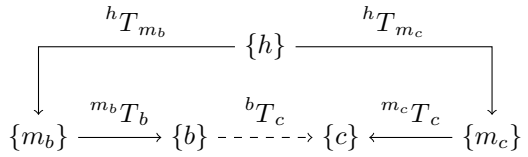


Fig. 2: The AR interface appears when looking at the open palm. The QR code on the camera positions the virtual camera model and the QR code on the robot's torso defines the robot's forward direction and center of rotation (orange).

$$\{m_b\} \xrightarrow{{}^{m_b}T_b} \{b\} \dashrightarrow^{{}^{b}T_c} \{c\} \xleftarrow{{}^{m_c}T_c} \{m_c\}.$$

The QR codes on the robot and camera can usually be attached to their housing so that they are either parallel or perpendicular to it. Thus, their orientations and, hence, the rotations to the corresponding frames are known. The same applies to the translation between $\{m_b\}$ and $\{b\}$, since the marker can be placed on the robot according to existing knowledge about other robot frames. If, contrary to expectations, this is not possible, we also managed to approximately estimate the center of rotation of the robot, i.e. where the base_link frame $\{b\}$ is located, as the geometric center of the virtual circle drawn by the marker on the camera as the robot rotates around its own axis. The translation from $\{m_c\}$ to $\{c\}$ can be determined with the help of manufacturer information about the dimensions of the camera. This means ${}^{m_b}T_b$ and ${}^{m_c}T_c$ are known.

We want to determine the transformation ${}^{b}T_c$. The idea is to add a frame $\{h\}$ corresponding to the coordinate system of the HoloLens to close the transformation graph:



After the two QR codes have been detected by the HoloLens, they can be selected via our AR interface and ${}^{h}T_{m_b}$ and ${}^{h}T_{m_c}$ can be estimated. Finally, the transformation ${}^{b}T_c$ from the base of the robot to the camera is given by the following equation:

$$ {}^{b}T_c = {}^{m_b}T_b^{-1}\, {}^{h}T_{m_b}^{-1}\, {}^{h}T_{m_c}\, {}^{m_c}T_c. $$

The result can be published from the HoloLens using ROS# to the transformation topic /tf, making it available to the robot.

Furthermore, we can use $\{h\}$ as a parent frame in which the robot's odometry frame is embedded. This gives us a reference point for the gaze information that we can access via MRTK. Associated with $\{h\}$, we can publish this data on a separate topic. This includes the gaze vector and the hit point of the eye gaze ray with the target.

It should be noted that the fiducial markers are only needed while performing the calibration. Once they have been detected and selected, the user is free from restrictions on the field of view. In addition, contrary to the usual procedure, we do not determine the calibration parameters externally and then store them in configuration files. This means that we can make changes to the camera, such as the tilt, even during runtime. This is a great advantage for use under changing scenarios.

C. Segmentation

We now address the problem of detecting unknown objects in the three-dimensional environment. We tackle this task by enhancing existing segmentation methods on the robot side with gaze information from the human collaborator. The segmentation process can be triggered either on demand by multimodal interaction, such as gestures or speech, or – empowered by the calibration method – continuously in real time. The assistance that the robot receives from the human should be limited solely to the provision of the gaze information. Apart from that, the segmentation should only take place on the robot's side. This makes sense due to the robot's higher resources and computing power compared to head-mounted devices like the HoloLens.

The segmentation process starts with a pass through filter where we assume that all relevant objects are between zero and three meters away from the camera, followed by a voxel grid filter with a leaf size of 0.03 along each axis that downsamples the point cloud we acquire from the robot's camera. This is not mandatory, but it reduces the computation time drastically and allows a segmentation in real time. In most cases, we can assume that the objects to be detected lie on a surface that is reasonably flat. This could be, for example, a table, a shelf, or the floor itself. We can take advantage of the parallelism between all these surfaces. Due to our calibration, we know the orientation of the camera with respect to the robot standing on the floor. This means that we can transform the upward vector from the HoloLens world frame into the camera frame and thus obtain the normal vector of the surface, that is parallel to the floor and on which the objects are located, in the frame of the camera. We can then use RANSAC to search for the largest plane in the robot's field of view, namely the said surface, that is perpendicular to the given normal vector. Thereby, we set the maximum allowed deviation from the normal vector to 30 degrees. All points belonging to this plane are finally removed from the point cloud. In the next step, we let the gaze information flow in. Since the human is looking at the object of interest, we know at least one point on its surface. Starting from this point, we can cluster the point cloud using simple euclidean clustering. That is, we first use a k-d tree to find the point in the point cloud that is closest to the gaze point. Then we cluster the point cloud with respect to the Euclidean distance, a tolerance of $5\,\mathrm{mm}$, and a minimum cluster size of 500. All points that belong to the same cluster as the nearest neighbor of the initial point result in the searched object. Note that without the gaze information we would not be able to distinguish between clusters belonging to objects, clusters of parts of the environment, or noise. *This subtle gaze interaction resolves ambiguities and brings us closer to a natural learning process.*

Finally, we do not only obtain an instance segmentation of an object, but we can also calculate a 3D bounding box from it. The box can be aligned properly in space again due to our calibration and the robot can share the result directly with the human via our AR interface. Thus, the bounding box can be displayed in the human's field of view, providing direct feedback and enabling a natural two-way communication component, as well as an initialization of further interactions of the robot with the object.

IV. Evaluation

In our experiments, a Scitos G5 from MetraLabs [40] was employed as a robotic counterpart. It has been equipped with an Azure Kinect DK, whose relative position to the robot we want to calibrate. The camera also provides image data such as the point cloud on which we perform the object detection. All components communicate with each other using ROS [41].

A. Qualitative Analysis

One of the advantages of our method is already evident when performing a single calibration run. Whereas calibration methods based on data collection are time-consuming and difficult to automate [5], the entire procedure with our variant takes less than a minute. Depending on the user's experience, a single run usually takes only between 15 and 40 seconds. This is especially apparent when the camera needs to be adjusted more frequently, either because it has been unintentionally moved or because the setting has changed.

After calibration, the whole system, including gaze mapping and the object segmentation, runs in real time. Figure 3 shows a visualization in RVIZ. The gaze ray vector as well as the coordinate of the hit point on the target are published with $59\,\mathrm{Hz}$. Using the default configuration, the Azure Kinect

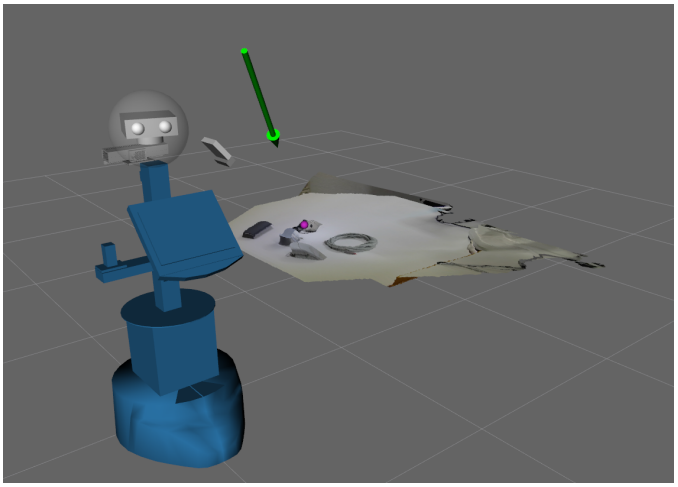


Fig. 3: The robot model with the camera positioned relative to it. The human gaze vector is shown as a green arrow and the gaze hit point as a purple sphere.

provides the point cloud at 4 frames per second. Subsequently, segmentation reduces the rate of the outgoing segmented cloud and thus also that of the bounding box to 2 frames per second. Since the minimal fixation duration is, in most cases, at least $200\,\mathrm{ms}$ [42] and the recommended feedback delay time for manual pointing actions is approximately between $350\,\mathrm{ms}$ and $600\,\mathrm{ms}$ [43], an update every 0.5 seconds is sufficient. Consequently, our method is suitable for human-robot interaction in real time.

Some final example results of segmented objects and the respective bounding box can be seen in Figure 4. For simplicity, we have chosen common household objects and office utensils, which we have placed on a table in front of the robot. In principle, both humans and robots can move freely around the table, since the position of both is known in the HoloLens based parent frame. However, to ensure that the robot's movements are tracked as precisely as possible, an additional localization procedure is required, which is beyond the scope of this work, as solving such a problem has already been extensively researched, and possible solutions can be found in the literature [44], [45]. Naturally, the current position can be manually repositioned at any time via our interface.

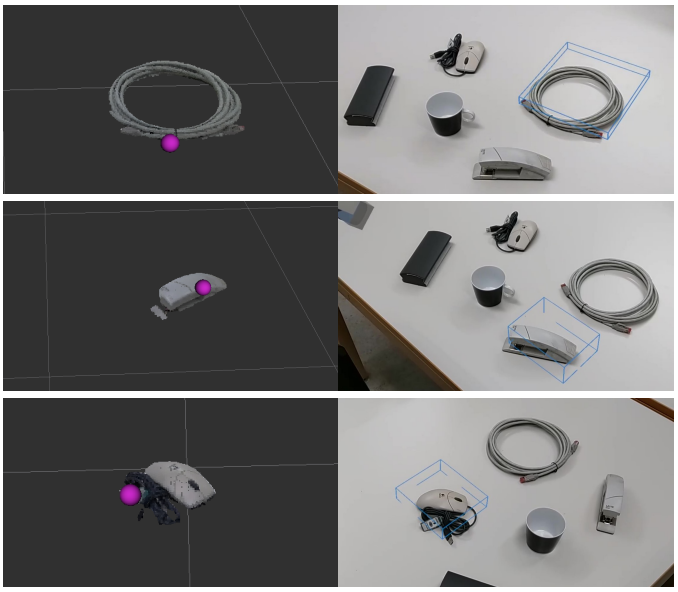


Fig. 4: The segmentation with the gaze point (left) and the resulting bounding box as seen from the human (right). The box is given in world coordinates, therefore tracking of already detected objects during movements of the robot is superfluous.

TABLE I: The translation in meters determined by the calibration with OptiTrack as well as our AR interface.

	Vertical		Horizontal		Inclined	
Axis	OptiTrack	mARC[a]	OptiTrack	mARC	OptiTrack	mARC
x[b]	-0.081	-0.080	-0.079	-0.079	-0.077	-0.079
y	-0.295	-0.295	-0.324	-0.327	-0.331	-0.332
z	0.973	0.973	1.072	1.071	1.033	1.035
ØDist.[c]	0.003		0.004		0.003	

[a] Average value of our AR-based calibration
[b] Coordinate axes refer to the ROS coordinate system
[c] Average spatial distance of all runs calculated with the euclidean norm

B. Quantitative Analysis

First we start with the evaluation of the calibration part. To establish a reference ground truth, we utilize the OptiTrack motion capture system [26]. We place multiple reflective markers on both the robot and the camera. These can be tracked by the Optitrack system with an accuracy of $1\,\mathrm{mm}$. Given these point observations, we can calculate the robot and the camera poses with respect to the coordinate system of the motion capture system, and then compute the camera pose of interest relative to the coordinate system of the robot. Based on the deviations we have observed in several test trials, we estimate that this post-processing decreases the accuracy to about $3\,\mathrm{mm}$. In this way, we determine the ground truth of the transformations from the robot frame to the camera frame for three different poses of the camera. Once horizontally, i.e. parallel to the floor, once vertically, i.e. perpendicular to the floor, and once in an inclined position at about 45 degrees. Without moving the camera in between, one of the system's designers performed the calibration 20 times per tilt using the method we presented in Section III. For each tilt, we evaluate the translation and rotation components separately.

Table I shows the result of the translation part of our AR-based calibration compared to the calibration using OptiTrack. In the table, the translation in each direction is given with respect to the ROS coordinate system. The difference between the result of the OptiTrack system and the mean result from our calibration varies, but is not noticeably pronounced with respect to any direction. The largest difference is observed with $3\,\mathrm{mm}$ in the direction of the y-axis in the case of horizontal orientation. All other values do not differ at all or only 1 to $2\,\mathrm{mm}$. Our analyses have shown that the same is true for the average of all individual differences to the ground truth.

Since the deviation in individual directions is less relevant than the spatial distance, we also want to take this into account. We measured the euclidean distance of the translation of each individual calibration run from the ground truth translation. The results are reported in the last row of Table I. One can see that the spatial error does not exceed $4\,\mathrm{mm}$ on average. This is comparable to the accuracy of the extrinsic calibrations evaluated in [22] and [46]. The distribution of the individual euclidean distances to the ground truth are shown in the box plot in Figure 5. Although the error in the vertical setting is

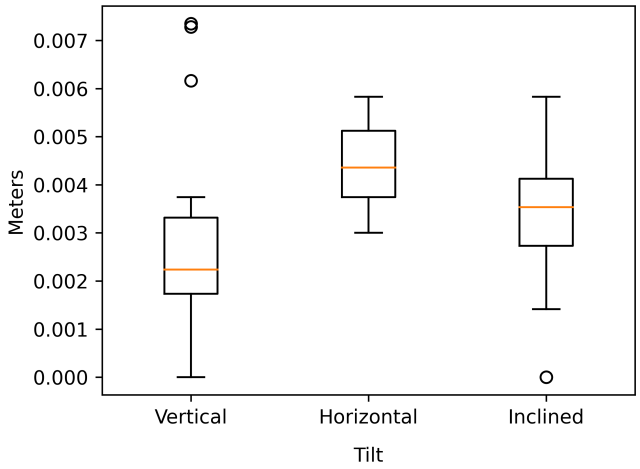


Fig. 5: The box plot represents the distribution of the translation errors with respect to the euclidean norm.

generally the smallest, there were also some outliers. Basically, in all three scenarios the vast majority of errors were below $5\,\mathrm{mm}$. The medians lie between $2\,\mathrm{mm}$ and $4.5\,\mathrm{mm}$. Note that this is only slightly above the accuracy range of the reference ground truth estimated via OptiTrack.

We now examine the rotational error of the transformation. In general, each rotation can be expressed by an axis of rotation and an angle of rotation. This rotation angle can be considered a measurement of the similarity of two orientations. This means that for each rotation component determined by our calibration, we calculate the difference rotation, which transforms the obtained rotation into the ground truth rotation. The smaller the angle of rotation, the more similar the two rotations. The angles of all difference rotations are plotted in Figure 6. Although there are, again, a few outliers, most values are below 2 degrees with medians ranging from 1.6 to 1.9 degrees. The same applies to the average rotation error. Thus, the rotation error is of the same order of magnitude as that of classical approaches like [22]. All in all, the accuracy meets the requirements of most applications, including our gaze segmentation, while being flexible and fast.

Let us now have a closer look at the evaluation of the segmentation part. Although the performance of current 3D object detectors lags behind the state of the art in 2D object detection, there are 3D object detectors that promise good results on indoor datasets such as SUN RGB-D [47]. However,

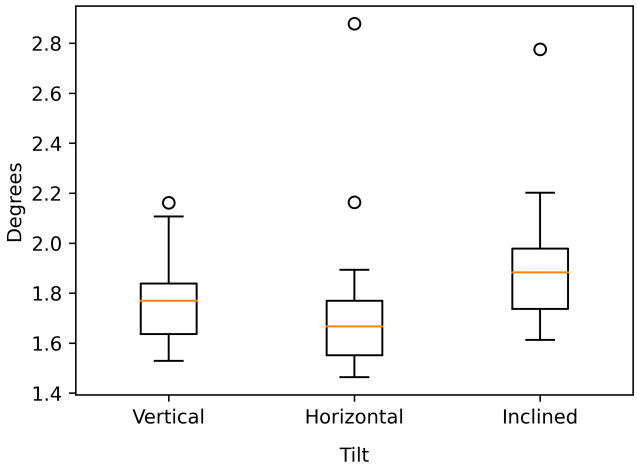


Fig. 6: The rotation errors displayed in a box blot.

our experiments have shown that the claimed results are difficult to achieve in practical applications. One possible reason could be that, due to comparability, the evaluations are usually conducted on the same few categories [28], [29], [30]. As a result, performance on other classes is often significantly worse or remains unknown.

We trained several neural networks, such as VoteNet [30] and Frustum ConvNet [29] on the classes book, bottle, bowl, cup, keyboard, laptop, mouse, paper, plant, and telephone from the Sun RGB-D dataset. These objects were more appropriate for our setup although smaller than the furniture used in the original papers. It turned out that all state-of-the-art networks performed very poorly on our set of objects and could not serve as reasonable reference ground truth. To put this in numbers: Whereas the mean average precision with a 3D Intersection over Union (IoU) threshold of 0.25 was only 27.8% for Frustum ConvNet, this value was even less than 1% for VoteNet. Thus, almost none of the available test objects were successfully detected by the neural networks and a meaningful comparison was therefore not possible. For this reason, we devised an alternative evaluation strategy and eventually conducted two different approaches. In the first one, we labeled the 3D bounding boxes of the objects in the acquired point cloud of the scene by hand and calculated the 3D IoU (with regard to the volume) for ten test objects. In the second one, we used a pretrained 2D object detector to avoid vulnerability regarding a bias in labeling. While modern 3D detectors are still far from being able to serve as ground truth, 2D detectors certainly are capable of doing so. Hence, we projected the points segmented by our method onto the 2D image plane and compared the resulting 2D bounding box with Faster R-CNN [48] (ResNet-101 backbone) trained on Microsoft COCO [49]. This dataset was also the criterion by which the ten test objects were selected. The results of both evaluations are shown in Table II. In the 2D case, all values are above 0.5 and thus all objects can be considered correctly

TABLE II: The IoU between the bounding boxes obtained by our method and the respective ground truth.

Class name	apple	backpack	book	bowl	clock	cup	keyboard	mouse	remote	tennis ball	mIoU
2D IoU	0.78	0.78	0.79	0.86	0.68	0.84	0.88	0.79	0.80	0.87	0.81
3D IoU	0.70	0.66	0.72	0.84	0.62	0.73	0.66	0.59	0.64	0.71	0.69

detected [50], [51]. Furthermore, almost all values are even above 0.7 with a mean IoU of 0.81. In contrast, the 3D IoU values are naturally smaller. Nevertheless, all objects are again considered to be detected, using the usual 3D threshold of 0.25 as reference [28], [47]. The mean 3D IoU is 0.69. Figure 7 shows the recall as a function of the IoU threshold at which a bounding box is classified as true positive. Note that even

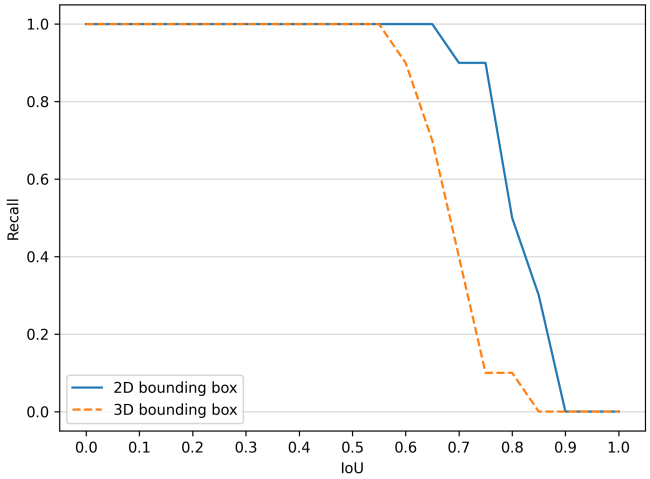


Fig. 7: The recall as a function of the IoU threshold at which the objects are considered to be detected.

with a 3D IoU threshold of 0.5, which is twice as large as that used by the authors of VoteNet and others [29], [47], the recall is still $100\,\%$.

Overall, our method hints at going far beyond the practical applicability of state-of-the-art neural network-based 3D object detectors, illustrating the importance of diverse solution strategies along with neural networks.

C. Limitations

Although our method of calibration is remarkably fast and user-friendly, the initial setup takes some time. While less in-depth expert knowledge is required compared to other methods, care must be taken to ensure that the markers are positioned accurately and that the distances to the corresponding frames can be determined. However, since this is a one-time step, this time expenditure is not of any significance compared to the time saved in each subsequent calibration run.

Furthermore, as with any other existing method, our segmentation and the calculated bounding box strongly depend on the quality of the original point cloud provided by the depth sensor. In this regard, the perspective of the robot's camera on the object also plays a role and whether the depth sensor can correctly determine the distance at the edges of the objects. However, the fact that the image quality has an influence on the result is in the nature of things and could be resolved by using multiple cameras or additional angles.

For objects that are too close to each other, it is not possible to keep them apart by extracting euclidean clusters. In this case, one could, for instance, resort to a min-cut based segmentation algorithm, also generally suitable, since a point must be given in the center of the object, which can be provided by the gaze point. In our tests, min-cut segmentation indicated promising results, but also required the approximate size of the respective object as an additional input argument.

V. Conclusion

In this work, we presented a novel method that allows for the deployment of robots under changing and non-predefined conditions. In this course, we combined robotics, augmented reality, and eye tracking to improve human-robot collaboration. Merely by receiving gaze information from its human partner, the robot was capable of detecting and segmenting unknown objects.

While most existing methods for extrinsic robot calibration are time consuming and often quite complicated to conduct, we have developed a method that is user-friendly, customizable at runtime, and takes only a few seconds to complete. At the same time, our evaluation has shown that we still achieve competitive accuracy compared to classical methods.

In addition, we bridge the two worlds of human and robot through the use of head mounted augmented reality glasses, giving the robot access to another persistent information channel — the human gaze. Just by having a human look at an object, the robot was able to segment objects it has not seen before and calculate associated three-dimensional bounding boxes. This goes beyond the capabilities of some state-of-the-art 3D object detectors and we found that our method works in situations where current existing neural networks have failed. Through direct feedback in the augmented human reality, the human is continuously informed about the results and the initialization of further interactions between the robot and the object is possible. This could be especially relevant for physically disabled people who are limited to movements in the head or neck area, in combination with a robotic arm that helps them grasp or manipulate objects.

In summary, our proposed method is versatile and facilitates general human-robot collaboration, as well as unknown object detection in the context of such scenarios in particular. However, there remains a significant amount of future work as we seek to investigate our segmentation in more challenging scenarios and realize a subsequent interaction between the robot and the objects.

REFERENCES

[1] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte *et al.*, "Minerva: A second-generation museum tour-guide robot," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 3. IEEE, 1999.

[2] H.-M. Gross, H.-J. Boehme, C. Schröter, S. Müller, A. König, C. Martin, M. Merten, and A. Bley, "Shopbot: Progress in developing an interactive mobile shopping assistant for everyday use," in *2008 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2008, pp. 3471–3478.

[3] A. Y. Nee, S. Ong, G. Chryssolouris, and D. Mourtzis, "Augmented reality applications in design and manufacturing," *CIRP annals*, vol. 61, no. 2, pp. 657–679, 2012.

[4] S. Blankemeyer, R. Wiemann, L. Posniak, C. Pregizer, and A. Raatz, "Intuitive robot programming using augmented reality," *Procedia CIRP*, vol. 76, pp. 155–160, 2018.

[5] A. Elatta, L. P. Gen, F. L. Zhi, Y. Daoyuan, and L. Fei, "An overview of robot calibration," *Information Technology Journal*, vol. 3, no. 1, pp. 74–78, 2004.

[6] D. Brščić, H. Kidokoro, Y. Suehiro, and T. Kanda, "Escaping from children's abuse of social robots," in *Proceedings of the tenth annual acm/ieee international conference on human-robot interaction*, 2015, pp. 59–66.

[7] B. Chandrasekaran and J. M. Conrad, "Human-robot collaboration: A survey," in *SoutheastCon 2015*. IEEE, 2015, pp. 1–8.

[8] C. Reardon, K. Lee, and J. Fink, "Come see this! augmented reality to enable human-robot cooperative search," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2018, pp. 1–7.

[9] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, "Gestures for industry intuitive human-robot communication from human observation," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2013, pp. 349–356.

[10] A. D. Wilbert, B. Behrens, O. Dambon, and F. Klocke, "Robot assisted manufacturing system for high gloss finishing of steel molds," in *International Conference on Intelligent Robotics and Applications*. Springer, 2012, pp. 673–685.

[11] R. Müller, M. Vette, and A. Geenen, "Skill-based dynamic task allocation in human-robot-cooperation with the example of welding application," *Procedia Manufacturing*, vol. 11, pp. 13–21, 2017.

[12] I. Olaronke, O. Oluwaseun, and I. Rhoda, "State of the art: a study of human-robot interaction in healthcare," *International Journal of Information Engineering and Electronic Business*, vol. 9, no. 3, p. 43, 2017.

[13] A. Kapoor, M. Li, and R. H. Taylor, "Constrained control for surgical assistant robots." in *ICRA*, 2006, pp. 231–236.

[14] H. I. Krebs, J. J. Palazzolo, L. Dipietro, M. Ferraro, J. Krol, K. Rannekleiv, B. T. Volpe, and N. Hogan, "Rehabilitation robotics: Performance-based progressive robot-assisted therapy," *Autonomous robots*, vol. 15, no. 1, pp. 7–20, 2003.

[15] J. Broekens, M. Heerink, H. Rosendal *et al.*, "Assistive social robots in elderly care: a review," *Gerontechnology*, vol. 8, no. 2, pp. 94–103, 2009.

[16] H. Robinson, B. A. MacDonald, N. Kerse, and E. Broadbent, "Suitability of healthcare robots for a dementia unit and suggested improvements," *Journal of the American Medical Directors Association*, vol. 14, no. 1, pp. 34–40, 2013.

[17] D. Krupke, F. Steinicke, P. Lubos, Y. Jonetzko, M. Görner, and J. Zhang, "Comparison of multimodal heading and pointing gestures for co-located mixed reality human-robot interaction," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[18] E. Rosen, D. Whitney, M. Fishman, D. Ullman, and S. Tellex, "Mixed reality as a bidirectional communication interface for human-robot interaction," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 431–11 438.

[19] L. Kästner and J. Lambrecht, "Augmented-reality-based visualization of navigation data of mobile robots on the microsoft hololens-possibilities and limitations," in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, 2019, pp. 344–349.

[20] L. Kästner, V. C. Frasineanu, and J. Lambrecht, "A 3d-deep-learning-based augmented reality calibration method for robotic environments using depth sensor data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1135–1141.

[21] F.-J. Chu, R. Xu, Z. Zhang, P. A. Vela, and M. Ghovanloo, "The helping hand: An assistive manipulation framework using augmented reality and tongue-drive interfaces," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 2158–2161.

[22] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[23] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3936–3943.

[24] G. Chen, G. Cui, Z. Jin, F. Wu, and X. Chen, "Accurate intrinsic and extrinsic calibration of rgb-d cameras with gp-based depth correction," *IEEE Sensors Journal*, vol. 19, no. 7, pp. 2685–2694, 2018.

[25] Vicon, https://www.vicon.com/, accessed: 2021-02-24.

[26] OptiTrack, https://optitrack.com/, accessed: 2021-02-17.

[27] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[28] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.

[29] Z. Wang and K. Jia, "Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1742–1749.

[30] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9277–9286.

[31] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "Unseen object instance segmentation for robotic environments," *arXiv preprint arXiv:2007.08073*, 2020.

[32] J. Bao, Y. Jia, Y. Cheng, and N. Xi, "Saliency-guided detection of unknown objects in rgb-d indoor scenes," *Sensors*, vol. 15, no. 9, pp. 21 054–21 074, 2015.

[33] V. Peysakhovich, F. Dehais, and A. Duchowski, "ArUco/gaze tracking in real environments," in *Eye Tracking for Spatial Research, Proceedings of the 3rd International Workshop*. ETH Zurich, 2018.

[34] M. Kalash, K. Singh, R. Eskicioglu, and N. D. Bruce, "Gaze-contingent interactive visualization of high-dynamic-range imagery," in *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*. IEEE, 2016, pp. 16–20.

[35] D. Weber, T. Santini, A. Zell, and E. Kasneci, "Distilling location proposals of unknown objects through gaze information for human-robot interaction," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 086–11 093.

[36] T. Toyama, T. Kieninger, F. Shafait, and A. Dengel, "Gaze guided object recognition using a head-mounted eye tracker," in *Proceedings of the 2012 ACM Symposium on Eye Tracking Research & Applications*. ACM, 2012, pp. 91–98.

[37] F. Xiao, L. Peng, L. Fu, and X. Gao, "Salient object detection based on eye tracking data," *Signal Processing*, vol. 144, pp. 392–397, 2018.

[38] K. Holmqvist, S. L. Örbom, I. T. Hooge, D. C. Niehorster, R. G. Alexander, R. Andersson, J. S. Benjamins, P. Blignaut, A.-M. Brouwer, L. L. Chuang *et al.*, "Eye tracking: empirical foundations for a minimal reporting guideline," *Behavior Research Methods*, 2021.

[39] M. Bischoff, "ROS#," Jun. 2019. [Online]. Available: https://github.com/siemens/ros-sharp

[40] MetraLabs, https://www.metralabs.com/mobiler-roboter-scitos-g5/, 2019, accessed: 2021-02-17.

[41] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[42] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2011.

[43] C. Müller-Tomfelde, "Dwell-based pointing in applications of human computer interaction," in *IFIP Conference on Human-Computer Interaction*. Springer, 2007, pp. 560–573.
[44] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," *Aaai/iaai*, vol. 593598, 2002.
[45] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
[46] S. Bi, D. Yang, and Y. Cai, "Automatic calibration of odometry and robot extrinsic parameters using multi-composite-targets for a differential-drive robot with a camera," *Sensors*, vol. 18, no. 9, p. 3097, 2018.
[47] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.
[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.
[49] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.
[50] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
[51] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*. Springer, 2014, pp. 391–405.