Teaching a Robot Where Doors and Drawers Are and How To Handle Them

R. Cupec[1], I. Vidović[1], V. Šimundić[1], P. Pejić[1], S. Foix[2] and G. Alenyà[2]

***Abstract*— We address the problem of teaching a service robot to detect doors and drawers in indoor environments. We propose a robust and accurate method in which a human demonstrates to the robot how to open doors and drawers that the robot is expected to operate in its future use. The proposed algorithm creates a model of a door or drawer from a sequence of RGB-D images and inserts it into an environment map. The model contains information about the size of the door panel or drawer front, as well as the position and orientation of the joint axis. This augmented environment map is then used by the robot to detect the target object in its environment and estimate its state.**

I. INTRODUCTION

Robots are called to become ubiquitous in people's everyday life. Human environments are complex and are not easy to manage by robots. Acceptance and good user experiences depend on the ability to make a useful task, adapt to uncertain environments, and deliver a good experience [1]. The complexity of these environments resides in the variety of objects, the uncertainty in their position, and the possibility or not of interaction [2]. In particular, in this work we focus on understanding the state of doors and drawers, as examples of articulated movable parts with whom a robot can interact.

Doors and drawers are a very common part of furniture and architecture. Robots in a house will have to deal continuously with doors: to go from one room to another, or, for instance, to locate objects within wardrobes. State of the art door detection methods vary in their approach and application, with some focusing on 2D methods using RGB images as input [3], while others use 3D methods and deal with RGB-D images. 3D detection has gained more attention in recent years, with various geometry-based methods [4], [5], [6], [7], [8]. However, many of these methods lack generality, as they can only detect standardized doors for which they have a model. To overcome this limitation, approaches using machine learning [9], [10] and detectors in the form of neural networks, particularly YOLO-based approaches [11], [12], have become popular. These methods use RGB-D data and create bounding boxes around doors, segmenting them based on color clustering and depth point extraction.

Fig. 1. A human demonstrates to a robot the movement of a door. The robot learns where the door is located and which is the direction of movement.

Door detection methods are primarly developed to assist robots in indoor navigation and door opening. For instance, one YOLO-based approach [13] used YOLOv3 to detect doors on a mobile robot navigating through a building simulation. In another study [14], a YOLOv3 model trained on the custom DoorDetect dataset was employed to detect various types of doors and handles, allowing a mobile robot to open them. This custom dataset included annotated images of doors and handles from the Open Images Dataset. In [15], a labeled RGB-D dataset, HoDoor, was created to train a method for classifying different types of door opening, with the aim of enabling robot manipulation in the future.

In [16], an approach is proposed that uses visual information collected during robot manipulation with doors and drawers to estimate their kinematic model. Previous research on the detection and manipulation of articulated objects has been extensively discussed in [17], which compares image-processing-based methods and benchmark datasets for detecting and segmenting articulated objects, determining their joint parameters, and manipulating them with a robot.

In this article, our contribution is a novel method that allows a robot to easily determine the location of doors and drawers within a given environment map and assess their current state. Our approach utilizes human demonstrations to teach the robot the location of doors and drawers (as shown in Fig. 1) within the environment, which are then incorporated into the map using standard

[1]Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, J. J. Strossmayer University of Osijek, Osijek, Croatia {robert.cupec,ivan.vidovic,valentin.simundic, petra.pejic}@ferit.hr

[2]Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain {sfoix,galenya}@iri.upc.edu

*This work has been partially supported by the Croatian Science Foundation under the project IP-2019-04-6819; by MCIN/ AEI /10.13039/501100011033 under the project CHLOE-GRAPH (PID2020-119244GB-I00); by MCIN/ AEI /10.13039/501100011033 and by the "European Union (EU) NextGenerationEU/PRTR under the project ROB-IN (PLEC2021-007859).

localization methods. During the teaching process, a human demonstrates how to open each door and drawer, and our algorithm analyzes the resulting image sequences to create corresponding door and drawer models, which are then positioned within the environment map. By augmenting the map with these models, the robot is able to accurately determine the states of observed doors and drawers (i.e., whether they are open or closed and to what degree). Unlike the approach presented in [16], which also creates a door or drawer model from a 3D image sequence, our approach doesn't rely on information about the robot gripper during the robot action. Instead, we use a generic model of the door panel or drawer front and fit the point clouds captured by a 3D camera to this model. Our method is based on generating candidate models by hierarchical evidence accumulation over a depth image sequence and evaluating these candidate models by projecting them onto the image and calculating hypothesis evaluation cost, similar to that used in [18] and [19]. To our knowledge, this approach has not yet been used to solve the problem under consideration. The model created is used to estimate the state of the door or drawer based on a single depth image.

The remainder of the paper is structured as follows: Section II outlines our augmented environment map. We provide an overview of our method in Section III. In Sections IV and V, we describe our teaching by demonstration and state estimation algorithms, respectively. Section VI details our experimental results and subsequent discussion. Finally, Section VII presents our conclusions.

II. AUGMENTED ENVIRONMENT MAP

Storage furniture such as cabinets, chests of drawers, or nightstands represent articulated objects whose movable parts are doors and drawers. The algorithm proposed in this paper detects doors and drawers. It augments an arbitrary environment map created by a mobile robot navigation system by inserting door and drawer models (DM) into this map. Each DM consists of information about the size of a door or a drawer and the position and orientation of its joint axis with respect to the environment map reference frame (RF) S_W.

A. Door Model

The door model used in this paper consists of a door panel, which is a movable part connected to the stationary part of a piece of furniture by a revolute joint. We define the door/drawer reference frame S_A centered on the joint axis with its z-axis parallel to the joint axis. The door panel is represented by a cuboid that can rotate about the joint axis, as shown in Fig. 2. It is associated with an RF S_B centered in the center of the cuboid with its axes parallel to the edges of the cuboid. The size of this cuboid is defined by a vector $s = [s_x, s_y, s_z]^\top$, whose elements represent the size of the cuboid along the x, y, and z axes, respectively. The origin of S_A is the orthogonal projection of the origin of S_B onto the joint axis, as shown in Fig. 2. The state of a door is defined by the angle θ, which is the angle between the x-axes of

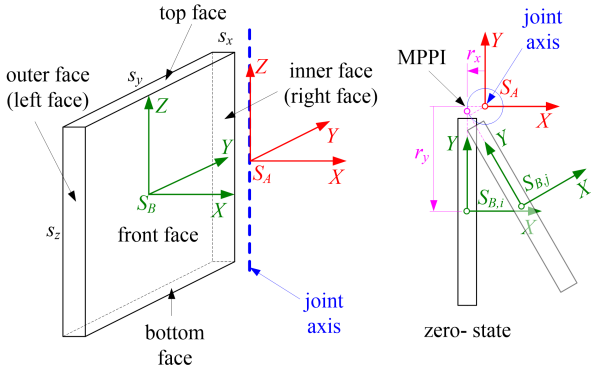

Fig. 2. Door model.

S_A and S_B. The x-axes of S_A and S_B are parallel when the door is closed. This is the zero-state of the door where $\theta = 0$. The relative position of the joint axis and the door panel is defined by the vector $r = [r_x, r_y]^\top$, which represents the x and y coordinates of the panel center in the zero-state with respect to S_A, as shown in Fig. 2. The position and orientation of S_A with respect to S_W are represented by a homogeneous transformation matrix (HTM) ${}^W T_A$.

B. Drawer Model

The drawer front is represented by a cuboid, which is assigned an RF S_B and a size vector s, analogously to the door panel model. The rest of the drawer is not modeled. A drawer is drawn in the direction of the x-axis of S_B. The state of a drawer is its extension, where 0 denotes a closed drawer. The model RF S_A is identical to S_B in the zero-state.

III. OVERVIEW OF THE METHOD

The proposed approach, which we named *Door and Drawer Detector* (DDD), consists of two algorithms. The first algorithm, DDD-THD, where THD stands for *Teaching by Human Demonstration*, detects a door or drawer in a sequence of RGB-D images captured by a 3D camera such as Microsoft Kinect, Asus Xtion, Orbbec, Intel RealSense, or similar devices. This algorithm creates a door or drawer model that can be used to detect a particular door or drawer in a particular environment and for robotic manipulation of that object. The second algorithm, DDD-SE, where SE stands for *State Estimator*, determines the current state of a door or drawer captured in an RGB-D image. A door is opened by rotating the door panel about a joint axis. In the following sections, these two algorithms are described. Due to limited space, some implementation details are omitted.

IV. TEACHING BY HUMAN DEMONSTRATION

This section describes the algorithm DDD-THD. The input to the algorithm is the current pose of the robot with respect to the environment map RF provided by a robot localization method, and a sequence of RGB-D images representing a human opening a door or drawer. This sequence must be captured by a stationary camera so that the only moving objects in the scene are the human and the moving part of an articulated object, i.e. a door or a drawer. The

algorithm detects and segments the human in the images using TensorMask [20]. The algorithm then removes the human from the images so that the only moving object in the scene is the moving part. The algorithm then generates one or more moving part hypotheses (MPH) for each image and integrates these hypotheses across the sequence to form door/drawer hypotheses (DH). A confidence value is then assigned to these hypotheses. A door/drawer model (DM) is created from the hypothesis with the highest score and inserted into the environment map. The algorithm consists of the following steps applied to a sequence of RGB-D images:

1: Removal of people from the images
2: Generation of MPHs for images in the sequence
3: Integration of MPHs into DHs
4: Assigning a confidence value to the DHs
5: Selection of the DH with the highest confidence value
6: Creating a DM and inserting it into the environment map

A. Moving Part Hypotheses

Moving part hypotheses are generated for each image in the input image sequence by segmenting the scene into approximately planar patches using a suitable segmentation method and determining which of these planar regions is moving. For the detection of planar patches, we use the method proposed in [21]. An example of planar patches detected by the applied algorithm is shown in Fig. 3.

Moving regions in a given image are detected by comparing that image with other images in the sequence within a predefined time window. Two compared images are represented by 3D point clouds, and for each point in one point cloud the closest point in the other one is determined. The difference between the coordinate vectors of these two points represents a *displacement vector*. All the displacement vectors obtained from this image pair are clustered, and the enter of the dominant cluster is considered as the *dominant displacement vector*. The planar patch with the largest number of points whose displacement vectors are sufficiently similar to the dominant displacement vector is considered as the front of a moving part. This planar patch is then used to generate an MPH. The bounding box of the planar patch is computed and an RF S_B is defined as described in Section II-A. The generated MPH is represented by a tuple $h = \left({}^{C}T_B, s, k\right)$, where ${}^{C}T_B$ is the HTM representing the pose of S_B with respect to the camera RF S_C, s is the size vector defined in Section II-A, and k is the index of the image in the input image sequence from which the MPH is generated. An example of an MPH is shown in Fig. 3.

B. Door/Drawer Hypotheses

Door/Drawer hypotheses are generated by integrating MPHs over the image sequence. A DH is represented by a tuple $H = \left(\eta, {}^{C}T_A, s, r, o, \Theta\right)$, where η is the object type: *door* or *drawer*, ${}^{C}T_A$ is the HTM representing the pose of S_A, defined in Section II-A, with respect to S_C, s and r are vectors, defined in Section II-A, $o \in \{-1, 1\}$ defines the *opening direction* and Θ is the *state sequence*.

Fig. 3. Input image (left) and a moving part hypothesis denoted by red lines (right). Colored regions in the right image represent detected planar patches.

The first element of s represents the thickness of a door panel or a drawer front. In the current version of our approach, this element is set to a constant value of $0.018\,m$, which, according to our analysis, fits the majority of real cases within a measurement noise range. It is assumed that the possible states of a given door, except for the zero-state, can have only positive or only negative values. The opening direction o defines the sign of the possible door states. In the case of a drawer hypothesis, o is always 1 and $r = [0, 0]^\top$. Each state in the sequence Θ is represented by a pair (θ, k), where k is the index of the image in which the moving part appears in the state θ.

The integration of MPHs into door hypotheses is done by a hierarchical clustering procedure. First, z-axis hypotheses are generated by forming pairs of MPHs and computing a z-axis candidate for each pair. Let (h_i, h_j) be a pair of MPHs. Each of these two MPHs is associated with an RF S_B, as explained in Section IV-A. Since the z-axis of S_A is perpendicular to the x-axis of S_B in each door state, the z-axis candidate is computed as the unit vector perpendicular to the plane spanned by the x-axes of the RFs $S_{B,i}$ and $S_{B,j}$ associated with the MPHs h_i and h_j. Z-axis hypotheses are then generated by clustering z-axis candidates, where each cluster C^z is represented by a z-axis hypothesis.

Let $\chi\left(C^{\mathrm{z}}\right)$ be the set of MPHs involved in the formation of a cluster C^{z}. From each such set, one or more joint axis hypotheses are generated. Consider two MPHs $h_i, h_j \in \chi\left(C^{\mathrm{z}}\right)$. A joint axis of a door is usually very close to the line representing the intersection of the yz-planes of RFs $S_{B,i}$ and $S_{B,j}$, as shown in Fig 2. Let us define the *moving part plane intersection* (MPPI) as the point representing the intersection of this line with the plane Π passing through the origin of S_C and perpendicular to the z-axis of S_A. Since the MPPI is usually very close to the joint axis, the position of the joint axis can be estimated by clustering MPPIs computed from pairs of MPHs from the set $\chi\left(C^{\mathrm{z}}\right)$. Let C^{a} be a cluster of MPPIs and let $\chi\left(C^{\mathrm{a}}\right)$ be the set of MPHs involved in the formation of this cluster. The distance between the yz-plane of $S_{B,i}$ and the joint axis should be r_x for all MPHs $h_i \in \chi\left(C^{\mathrm{a}}\right)$. This (signed) distance can be computed by

$$\delta\left(h_i, p\right) = x_{B,i}^{\top}\left(c_i - p\right),$$

where $x_{B,i}$ and c_i are the projections of the x-axis and the origin of $S_{B,i}$, respectively, onto the plane Π and p is the intersection point of the joint axis and the plane Π. The

position of the joint axis is estimated by calculating the point p and the distance r_x that minimize the cost function

$$E\left(p, r_x\right) = \sum_{h_i \in \chi(C^{\mathrm{a}})} \left(\delta\left(h_i, p\right) - r_x\right)^2$$

The joint axis is the line parallel to the z-axis of S_A passing through the point p.

After defining the joint axis, the MPHs $h_i \in \chi\left(C^{\mathrm{a}}\right)$ are used to determine the faces of the cuboid representing the door panel. The supporting plane of the front face of the cuboid is defined by the planar patch used to generate the MPH. The supporting planes of the top and bottom face, cf. Fig. 2, are parallel to the plane Π. Therefore, these planes are defined by their distances from Π. Let d_i^{t} and d_i^{b} be the distances of the top and bottom face of the cuboid corresponding to the MPH h_i. These distances can be calculated from the center and the size of this cuboid. Clustering of the values d_i^{t} and d_i^{b} for all $h_i \in \chi\left(C^{\mathrm{a}}\right)$ is performed, and the centers of the obtained clusters $\bar{d}^{\mathrm{t}}$ and $\bar{d}^{\mathrm{b}}$ are used to define the top and bottom face of the door hypotheses.

The supporting planes of the inner and outer face of the cuboid representing the door panel are parallel to the joint axis and perpendicular to the front face. Therefore, these planes are defined by their distances from the joint axis. The outer face of the cuboid representing the door panel is calculated by clustering the distances d_i^{o} of the outer faces of the MPHs $h_i \in \chi\left(C^{\mathrm{a}}\right)$, analogous to the calculation of the top and bottom faces. In the current implementation of our algorithm, the distance of the inner face from the joint axis is set to 0.

From each MPH $h_i \in \chi\left(C^{\mathrm{a}}\right)$ a pair (θ, k) is added to the sequence Θ, where θ is the door state, computed as the angle between the x-axes of S_A and S_B, and k is the image index of h_i.

The proposed hierarchical clustering procedure is represented by Algorithm 1. The input of the algorithm is the set of all MPHs generated from an image sequence χ_{all} and the output is a set of door hypotheses $\mathcal{H}$ obtained by integrating the MPHs over the image sequence.

The drawer hypotheses are generated by a similar but simpler procedure, which we don't describe here due to space limitations.

Algorithm 1 Generating door hypotheses by hierarchical clustering

```
procedure INTEGRATE(χ_all)
    H ← ∅
    Create z-axis clusters C^z.
    for every cluster C^z do
        Compute z-axis and plane Π.
        Create top face clusters C^t.
        Compute center d̄^t of each cluster C^t.
        Create bottom face clusters C^b.
        Compute center d̄^b of each cluster C^b.
        Compute MPPIs.
        Create joint axis clusters C^a.
        for every cluster C^a do
            Compute p and r_x.
            Create outer face clusters C^o.
            Compute center d̄^o of each cluster C^o.
            for every d̄^o do
                for every d̄^b do
                    for every d̄^t do
                        Generate a door hypothesis H.
                        Put H into H.
    return H
```

C. Hypothesis Evaluation and Selection

For a given image sequence, a set $\mathcal{H}$ of door and drawer hypotheses is generated. Each hypothesis $H \in \mathcal{H}$ is evaluated by computing its *hypothesis evaluation cost*

$$\Psi\left(H, k\right) = \zeta\left(H, k\right)\left(-\Omega\left(H, k\right) + \left|\Phi\left(H, k\right)\right|\right)$$

which represents the sum of the *scene fitting score* Ω and the *transparency cost* $|\Phi|$, as proposed in [18] multiplied by the *zero-state factor* ζ. The zero-state factor, defined as

$$\zeta\left(H, k\right) = 1 + e^{-(\theta/\sigma)^2},$$

assigns greater weight to hypotheses that are close to the zero-state, because the zero-state is a priori more likely than other states. The value σ is a user-defined constant set to 5° for doors and 0.05 m for drawers.

Cost Ψ is computed for each pair (θ, k) in the sequence Θ. The pose of the cuboid representing a door panel or drawer front with respect to the camera is computed using ${}^{C}T_A$, r and θ. This pose is represented by an HTM ${}^{C}T_B$. The surface of the cuboid is uniformly sampled and the obtained point cloud is transformed into the scene using ${}^{C}T_B$. By comparing this transformed point cloud with the scene point cloud computed from the k-th depth image of the input image sequence, $\Omega\left(H\right)$ and $\Phi\left(H\right)$ are computed as described in [18].

Furthermore, for each hypothesis H, the temporally consistent subsequence $\Theta^* \subseteq \Theta$ is determined, i.e. the subsequence with the lowest total hypothesis evaluation cost which satisfies a temporal constraint. The temporal constraint requires that the change in state between two successive states be within a predefined threshold. The total hypothesis evaluation cost of a sequence Θ' is computed by

$$\Psi_\Sigma\left(H, \Theta'\right) = \sum_{(\theta,k)\in\Theta'} \min\left\{\Psi\left(H, k\right), 0\right\}.$$

Finally, the hypothesis H^* with the least cost $\Psi_\Sigma\left(H^*, \Theta^*\right)$ is chosen as the final solution. From this hypothesis, a DM is created and inserted into the augmented environment map. The pose of this DM with respect to the environment map RF S_W is calculated by

$${}^{W}T_A = {}^{W}T_C \cdot {}^{C}T_A,$$

where ${}^{W}T_C$ is provided by the mobile robot localization system.

V. STATE ESTIMATION

After the robot learns the location and movement direction of doors and drawers, it can estimate their state when it stops in front of them and captures an RGB-D image. The following procedure is used:

1: Predict the target object's location in relation to the camera.
2: Detect one or more planar patches within a region of interest (RoI).
3: Generate hypotheses about the moving part's state from the planar patches in the RoI.
4: Evaluate the hypotheses and select the most likely one.
5: Compute the object state, which is the angle of the door panel or drawer extension relative to the zero-state.

We assume that the robot localization system can provide information about the camera pose with respect to the environment map RF S_W in the form of an HTM ${}^{W}T_C$. This is a very reasonable assumption. The augmented environment map (described in Sect. II) contains the information about the doors and drawers in the robot's environment in the form of DMs, where each DM is associated with an HTM ${}^{W}T_A$ describing the pose of a door or drawer with respect to S_W. The matrices ${}^{W}T_C$ and ${}^{W}T_A$ can be used to compute the HTM ${}^{C}T_A$ defining the pose of a particular DM with respect to the camera RF S_C.

Given the pose of a DM, a RoI is computed in the form of a 3D bounding box aligned with the axes of S_A. The size of this bounding box in the y and z-directions is equal to the second and third element of the size vector s, defined in Sect.II-A extended by 10%, while its size in the x direction is a constant value large enough for the RoI to contain a door or drawer in any state. We set this value to 1.1 m.

For each planar patch with at least 1000 points within the RoI, a DH is generated with the x-axis of S_B parallel to the planar patch normal. The door state is computed as the angle between the x-axes of S_B and S_A and the drawer state is computed as the distance between the planar patch and the drawer front in the zero-state. Each DH is evaluated using the approach described in Sect. IV-C, and all hypotheses with positive hypothesis evaluation costs are rejected. In some cases, a door panel is oriented in such a way that it is aligned with the camera optical rays, i.e. the angle between its surface and the camera optical rays pointing to that surface is very small. In such cases, the door panel is poorly visible in the image, and there is no suitable planar patch representing this surface. An example of this is the bottom-right image in Fig 4. Thus, if no hypothesis with a negative hypothesis evaluation cost is generated, the door panel is assumed to be perfectly aligned with the optical rays of the camera, and its state is calculated accordingly.

An accurate estimate of the state of a door or drawer requires an accurate pose ${}^{C}T_A$. Since this pose is computed using the camera pose ${}^{W}T_C$ provided by the robot localization system, an error in the robot localization will result in an error in the state estimate. To compensate for the inaccuracy of the robot localization, we store the local environment model, i.e. a set of planar patches in the vicinity of a door or a drawer detected by the DDD-THD algorithm, and use it to align the query image with this local environment model. An ICP-based algorithm can be used for this alignment, using the pose ${}^{W}T_C$ provided by the robot localization system as the initial solution. In the experiments reported in this paper, we use our alignment algorithm specifically designed for aligning rectangular structures, but this is beyond the scope of this paper and therefore not described due to space limitations.

Fig. 4. Examples of correct (green) and incorrect (red) state estimates.

VI. EXPERIMENTS

We conducted two experiments to evaluate our approach. The goal of the first experiment was to test the ability of DDD-THD to recognize doors or drawers in sequences of RGB-D images in which a human demonstrates opening a door or drawer. The second experiment aimed to measure the accuracy of the state estimation achieved by DDD-SE. The test images were captured in a laboratory at the *Institut de Robòtica i Informàtica Industrial* in Barcelona, which replicated a typical household setting with furniture. Four different test objects were used in the experiments: a room door, a kitchen cabinet with two doors and one drawer, a nightstand with one door, and a wardrobe with two door leafs, see Fig. 5.

For the experiments, we utilized a TIAGo robot mobile platform. This robot has the ability to autonomously create a map of its environment and navigate without colliding with obstacles thanks to using the ROS Advanced Navigation Stack from PAL Robotics©. Additionally, it possesses a 7-DOF robotic manipulator attached to its chest, enabling it to grasp objects effectively. This feature is particularly advantageous as it will enable us to manipulate objects in future. To improve its visual capabilities, we installed an Intel RealSense Lidar Camera L515 on top of its head. This addition enables us to capture highly detailed RGB-D images of the objects. This is very important to get good estimates of their state.

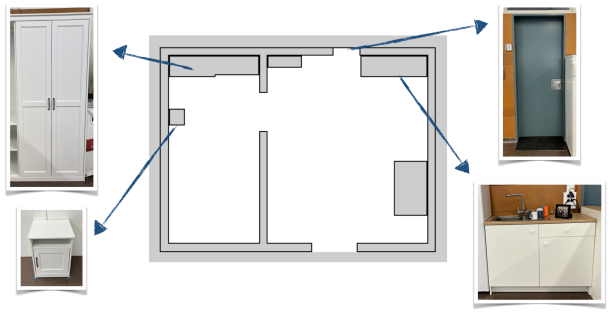

Fig. 5. Location of the different pieces of furniture within the map. For clarity, the rest of the furniture is not represented.

TABLE I

INTERSECTION OVER UNION FOR EACH MOVING PART.

Moving part	min IoU	max IoU	mean IoU
Room door	85.21%	94.52%	88.33%
Drawer	58.66%	82.74%	70.70%
Small cabinet door	78.94%	91.13%	85.04%
Big cabinet door	81.59%	89.57%	85.58%
Nightstand	78.79%	79.21%	79.00%
Wardrobe left door	72.58%	90.50%	81.59%
Wardrobe right door	84.91%	95.38%	92.25%

A. Door/Drawer Detection

To test the accuracy of door/drawer detection, we captured 38 sequences of RGB-D images in which a human opens doors or drawers of the test objects and applied the algorithm DDD-THD to these image sequences. For each image sequence, DDD-THD generated a DM. In addition, we manually annotated the images of the four test objects by outlining the moving parts to obtain the ground truth. The ground truth annotations are compared to the image projections of the front faces of the DMs in the zero-state (closed doors and drawers) by computing Intersection over Union (IoU) as the detection performance index. The results of the described experiment are shown in Table I. Of the 38 captured sequences, 6 belong to a room door, 12 to the left door of a wardrobe, 12 to the right door of the wardrobe, and 2 to each of the following moving parts: drawer, small cabinet door, large cabinet door and nightstand. The drawer has the lowest mean IoU of the tested parts. In one case, the hand of the human opening the drawer covers a large part of the front surface of the drawer. This causes the algorithm to incorrectly estimate the size of the DM. The average IoU value accross all objects is 83.62%. The results of the door and drawer detection experiment are shown in Fig. 6.

B. State Estimation

The accuracy of the state estimation of the proposed approach is evaluated by applying the DDD-SE algorithm to RGB-D images of the considered test objects in different states and comparing the estimated state values with manually measured ground truth data. The results of this experiment are presented in Table II. For each moving part of the test objects, the total number of occurrences, the number and percentage of correct measurements, and the average

Fig. 6. Door/drawer detection results for the test objects. The ground truth data is shown with a green bounding box, while the detected parts are shown with red bounding boxes.

absolute error are given. We assume that a measurement is correct if the absolute difference between the estimated and true state is at most 5° for doors or 0.05 m for the drawer. These values were obtained by camera measurement error analysis. We analyzed a scene with a dominant planar surface by fitting a plane to the points on that surface detected by the 3D camera and calculating the standard deviation of those points from the plane. This standard deviation was ε = 0.0068 m. Assuming a normal distribution of the camera measurement error, almost all points on the surface under consideration are within 3ε of the plane. Therefore, we assume that the maximum error in estimating the position of a plane is 3ε. The state of the drawer is measured by calculating the distance between the drawer front and the static part of the furniture to which this drawer belongs. If we assume the worst case, where both the drawer front and the furniture front are estimated with the maximum error of 3ε, then the error in the state measurement is 6ε = 0.0408 m. Rounding up this value to the first larger integer value in centimeters, we obtain the tolerance of τ_d = 0.05 m. We determined the tolerance for the door state by assuming that the maximum error in measuring the distance γ between the point of the door panel farthest from the door axis and the static part of the furniture is 6ε. Assuming a reference door width of w_{door} = 0.5 m, the door angle corresponding to the distance γ is $6\varepsilon / w_{\text{door}} = 4.68^\circ$. The nearest integer value in degrees is used as the tolerance for the door state $\tau_\theta = 5^\circ$.

A few examples of correct state estimates are shown in Fig. 4. Most false measurements occur when the door panel is oriented so that the camera's optical rays fall on its surface at a small angle. In this case, the door panel is almost invisible in the depth image. For some surfaces, this effect occurs at larger angles than for others, depending on the surface finish. In the case of the small cabinet door, there is a wall to the right of the cabinet that is a flat vertical surface barely different from the door panel open at 90°, which is another cause of incorrect estimates. An example of such a case is

TABLE II

STATE ESTIMATION ACCURACY FOR EACH MOVING PART.

Moving part	total	correct	%	avg. err.
Room door	30	19	63.33	1.35°
Drawer	78	77	98.72	6.0 mm
Small cabinet door	78	67	85.90	0.71°
Big cabinet door	78	78	100.00	0.53°
Nightstand	30	21	70.00	2.70°
Wardrobe left door	54	43	79.63	0.61°
Wardrobe right door	54	44	81.48	1.74°

shown in Fig. 4 (bottom right), where the misidentified door panel is outlined in red.

Furthermore, since the proposed method relies on the alignment of furniture surfaces and walls, it fails in cases where the scene has a simple geometry with few surfaces available for matching and some of these surfaces cannot be reconstructed by the RGB-D camera used due to specular reflection. This is the cause of the lower performance of our algorithm for the nightstand shown in Fig. 4 (bottom left).

VII. CONCLUSIONS

In this paper, we have presented a novel method for teaching robots to detect and retrieve the state of drawers and doors, a crucial ability for robots operating in household environments. Our approach utilizes human demonstrations, making it easily attainable for non-experts. Although the method is effective in most cases, certain degenerate cases, such as when the door is aligned with the camera axis, may not be accurately detected. This problem could be mitigated by determining that the door is not detected and using a next best view strategy to move the robot to a position from which the door panel is clearly visible. Another approach would be to use RGB information, since depth information is not reliable in such cases. For this purpose, a machine learning algorithm could be used that could be trained with a set of images of doors in different states captured from two different viewpoints, where the door would be clearly visible from at least one of these two viewpoints.

We have successfully validated our approach in a realistic environment using doors of varying sizes and opening directions. In the future, we aim to further develop the robot's physical interaction capabilities with the environment, particularly for the task of looking for a particular object in the context of assistive robotics.

ACKNOWLEDGMENT

We would like to thank Pablo Salido for his dedicated efforts in acquiring the required datasets for this research.

REFERENCES

[1] S. Forgas, R. Huertas, A. Andriella, and G. Alenyà, "Social robot-delivered customer-facing services: an assessment of the experience," *The Service Industries Journal*, vol. 43, no. 3-4, pp. 154–184, 2023.

[2] C. Angulo, S. Pfeiffer, R. Tellez, and G. Alenyà, "Evaluating the use of robots to enlarge aal services," *Journal of Ambient Intelligence and Smart Environments*, vol. 7, no. 3, pp. 301–313, 2015.

[3] W. Chen, T. Qu, Y. Zhou, K. Weng, G. Wang, and G. Fu, "Door recognition and deep learning algorithm for visual based robot navigation," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, Dec. 2014, pp. 1793–1798.

[4] S. Meyer Zu Borgsen, M. Schopfer, L. Ziegler, and S. Wachsmuth, "Automated door detection with a 3d-sensor," in *2014 Canadian Conference on Computer and Robot Vision*. Montreal, QC: IEEE, May 2014, pp. 276–282. [Online]. Available: https://ieeexplore.ieee.org/document/6816854/

[5] M. I. Habib, "Detecting doors edges in diverse environments for visually disabled people," *International Journal of Computer Science and Network Security*, vol. 21, no. 5, p. 9–15, May 2021.

[6] M. Vlaminck, L. H. Quang, H. Van Nam, H. Vu, P. Veelaert, and W. Philips, "Indoor assistance for visually impaired people using a RGB-D camera," in *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, Mar. 2016, pp. 161–164.

[7] P. Skulimowski, M. Owczarek, and P. Strumillo, "Door detection in images of 3D scenes in an electronic travel aid for the blind," in *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, Sep. 2017, pp. 189–194.

[8] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. G. Atkeson, M. Gennert, and T. Padir, "Human-supervised control of the ATLAS humanoid robot for traversing doors," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. Seoul, South Korea: IEEE, Nov. 2015, pp. 722–729.

[9] J. G. Ramôa, L. A. Alexandre, and S. Mogo, "Real-Time 3D Door Detection and Classification on a Low-Power Device," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Apr. 2020, pp. 96–101.

[10] J. G. Ramôa, V. Lopes, L. A. Alexandre, and S. Mogo, "Real-time 2d–3d door detection and state classification on a low-power device," *SN Applied Sciences*, vol. 3, no. 5, p. 590, May 2021.

[11] A. Llopart, O. Ravn, and N. A. Andersen, "Door and cabinet recognition using Convolutional Neural Nets and real-time method fusion for handle detection and grasping," in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, Apr. 2017, pp. 144–149.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255.

[13] T. Kim, M. Kang, S. Kang, and D. Kim, "Improvement of Door Recognition Algorithm using Lidar and RGB-D camera for Mobile Manipulator," in *2022 IEEE Sensors Applications Symposium (SAS)*, Aug. 2022, pp. 1–6.

[14] M. Arduengo, C. Torras, and L. Sentis, "Robust and adaptive door operation with a mobile robot," *Intelligent Service Robotics*, vol. 14, no. 3, pp. 409–425, Jul. 2021.

[15] V. Šimundić, M. Džijan, P. Pejić, and R. Cupec, "Introduction to door opening type classification based on human demonstration," *Sensors*, vol. 23, no. 6, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/6/3093

[16] T. Rühr, J. Sturm, D. Pangercic, M. Beetz, and D. Cremers, "A generalized framework for opening doors and drawers in kitchen environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 3852–3858.

[17] P. Pejić, V. Šimundić, M. Džijan, and R. Cupec, "Articulated objects: From detection to manipulation—survey," in *Intelligent Autonomous Systems 17*, I. Petrovic, E. Menegatti, and I. Marković, Eds. Cham: Springer Nature Switzerland, 2023, pp. 495–508.

[18] R. Cupec, I. Vidović, D. Filko, and P. Đurović, "Object recognition based on convex hull alignment," *Pattern Recognition*, vol. 102, 2020.

[19] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypothesis verification framework for 3d object recognition in clutter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1383—-1396, 2016.

[20] X. Chen, R. Girshick, K. He, and P. Dollar, "Tensormask: A foundation for dense object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[21] R. Cupec, D. Filko, and E. Nyarko, "Segmentation of depth images into objects based on local and global convexity," in *Proc. European Conference on Mobile Robots (ECMR)*, 2017.