Noname manuscript No.
(will be inserted by the editor)

Robust and Adaptive Door Operation with a Mobile Robot

Miguel Arduengo[1,2] · **Carme Torras**[2] · **Luis Sentis**[1,3]

Abstract The ability to deal with articulated objects is very important for robots assisting humans. In this work, a framework to robustly and adaptively operate common doors, using an autonomous mobile manipulator, is proposed. To push forward the state-of-the-art in robustness and speed performance, we devise a novel algorithm that fuses a convolutional neural network with efficient point cloud processing. This advancement enables real-time grasping pose estimation for multiple handles from RGB-D images, providing a speed up improvement for assistive human-centered applications. In addition, we propose a versatile Bayesian framework that endows the robot with the ability to infer the door kinematic model from observations of its motion and learn from previous experiences or human demonstrations. Combining these algorithms with a Task Space Region motion planner, we achieve an efficient door operation regardless of the kinematic model. We validate our framework with real-world experiments using the Toyota Human Support Robot.

Keywords Handle Grasping · Door Operation · Kinematic Model Learning · Task Space Region · Service Robot

Miguel Arduengo
E-mail: marduengo@iri.upc.edu

Carme Torras
E-mail: torras@iri.upc.edu

Luis Sentis
E-mail: lsentis@austin.utexas.edu

[1]Human Centered Robotics Lab (UT at Austin)
[2]Institut de Robòtica i Informàtica Industrial (CSIC-UPC)
[3]Department of Aerospace Engineering (UT at Austin)

Fig. 1 The HSR robot assists a person to enter a room.

1 Introduction

Robots are progressively spreading to logistic, social and assistive domains (Figure 1). However, in order to become handy co-workers and helpful assistants, they must be endowed with quite different abilities than their industrial ancestors (Asfour et al., 2008; Schiffer et al., 2012; Torras, 2016). The ability to deal with articulated objects is relevant for robots operating in domestic environments. For example, robots need to open doors when moving around homes and to open cabinets to pick up objects (Mae et al., 2011). The problem of opening doors and drawers with robots has been tackled extensively (Enders et al., 2013; Jain and Kemp, 2009; Kessens et al., 2010; Meeussen et al., 2010; Ott et al., 2005). These approaches usually focus either on a particular type of door and handle mechanism or on a certain aspect of the task.

Handling different types of doors (e.g. drawers, room or refrigerator doors) and handles (e.g. doorknobs, lever handles, drawer pulls) simultaneously remains a challenge. Therefore, our contribution is on devising a more general framework that can incorporate different types of door models and that provides adaptive behavior during door operation. The paper is organized as follows: in Section 2 we review the state-of-the-art in the field; in Section 3 we state the problem addressed; in Section 4 we present our door and handle detection model; in Section 5 we explain our approach for achieving robust real-time estimation of end-effector grasping poses; in Section 6 we describe a method for unlatching door handles; in Section 7 we present a Bayesian approach to learn door kinematic models which allows improving performance by learning from experience as well as from human demonstrations; in Section 8 we discuss the integration of kinematic model inference with a motion planner; in Section 9 we experimentally validate our framework; finally, in Section 10 we draw the main conclusions.

2 Related Work

The detection of doors and handles is a key problem when operating doors with an autonomous robot. A robust algorithm that allows the simultaneous detection of several doors and handles regardless of the shape, color, light conditions, etc, is essential (for instance, see Figure 2). This problem has been explored based on 2D images, depth data, or both. In (Chen et al., 2014), they present a deep convolutional neural network for estimating door poses from images. Although doors are accurately located, the identification of handles is not addressed. In (Banerjee et al., 2015), following the requirements from the DARPA Robotics Challenge, the authors develop an algorithm for identifying closed doors and their handles. Doors are detected by finding consecutive pairs of vertical lines at a specific distance from one another in an image of the scene. If a flat surface is found in between, the door is recognized as closed. Handle detection is subsequently carried out by color segmentation. The paper (Llopart et al., 2017) addresses the problem of detecting room doors and also cabinet doors. The authors propose a CNN to extract and identify the Region of Interest (ROI) in an RGB-D image. Then, the handle's 3D position is calculated under the assumption that it is the only object contained in the ROI and its color is significantly different from that of the door. Although positive results are obtained in these last two works, they rely on too many assumptions limiting the versatility.

The door manipulation problem with robotic systems has also been addressed with different approaches. Some works assume substantial previous knowledge of the kinematic model of the door and its parameters, while others are entirely model-free. Among the works that assume an implicit model, in (Diankov et al., 2008) the operation of articulated objects is formulated as a kinematically constrained planning problem. The authors propose to use caging grasps, to relax task constraints, and then use efficient search algorithms to produce motion plans. Another interesting work is (Wieland et al., 2009). The authors combine stereo vision and force feedback for the compliant execution of the door opening task. In recent work Eppner et al. (2018), the authors propose candidate models that include kinematic and dynamic properties, which are selected using interactive perception. Finally, in Abraham et al. (2020) they propose a model-based path-integral controller that uses physical parameters. Regarding model-free approaches, in (Lutscher et al., 2010) they propose to operate unknown doors based on an impedance control method, which adjusts the guiding speed to achieve two-dimensional planar operation. Another example is the approach presented in (Karayiannidis et al., 2013). Their method relies on force measurements and estimation of the motion direction, rotational axis and distance from the center of rotation. They propose a velocity controller that ensures a desired tangential velocity. Both approaches have their own advantages and disadvantages. By assuming an implicit kinematic model, although in practice a simpler solution is typically achieved, the applicability is limited to a single type of door. On the other hand, model-free approaches release programmers from specifying the motion parameters, but they rely entirely on the compliance of the robot.

Alternatively, other works propose probabilistic methods that do not consider interaction forces. In (Nemec et al., 2017) the authors combine reinforcement learning with intelligent control algorithms. With their method, the robot is able to learn the door-opening policy by trial and error in a simulated environment. Then, the skill is transferred to the real robot. In (Welschehold et al., 2017) the authors present an approach to learn door opening action models from human demonstrations. The main limitation of these works is that they do not allow to operate autonomously unknown doors. Finally, the probabilistic approach proposed in (Sturm, 2013) enables the description of the geometric relation between object parts to infer the kinematic structure from observations of their motion. We have adopted this approach as a basic reference but extended its capabilities to improve the performance by using prior information or human demonstrations.

In this paper, we propose a robust and adaptive framework for manipulating general types of door mechanisms. We consider all the stages of the door opening task in a unified framework. The main contributions of our work are (a) the development of a novel algorithm to estimate the robot's end-effector grasping pose in real-time for multiple handles simultaneously; (b) a versatile framework that provides the robust detection and subsequent door operation for different types of door kinematic models; (c) the analysis of the door kinematic inference process by taking into account door prior information; (d) the testing on real hardware using the Toyota HSR Robot (Figure 1).

3 Problem Statement and Framework Overview

We study the problem of enabling a robot to open doors autonomously, regardless of their form or kinematic model. Performing this task requires the exploitation of the robot's sensorial, actuation and computational capabilities. The following sub-tasks are to be performed sequentially by the autonomous robot:

1. **Door and handle detection** (Section 4): First, the door and handle must be identified and located in the environment where the robot is operating. A vision system that allows the recognition of the corresponding Regions of Interest is required.
2. **Grasping of the handle** (Section 5): Once the handle is located, the robot must position and orient the end-effector adequately in order to grasp it. For inferring this pose, a vision system that also provides information about the 3-dimensional structure of the environment is needed.
3. **Unlatching the handle** (Section 6): Then, the robot must exert an appropriate torque for actuating the handle mechanism. For determining such torque, a sensor that provides force feedback in the robot's end-effector is essential.
4. **Estimating the door kinematic model** (Section 7): The robust operation of doors involves the inference of the required opening motion, i.e. the door kinematic model. This inference can be either performed online while actuating the door, or from observations of the door motion provided by a teacher.
5. **Planning and executing door opening motion** (Section 8): Finally, the control actions for opening the door according to its kinematic model must be planned and executed. For a mobile manipulator robot, this implies tight base-arm coordination under the task constraints.

In this paper, we assume the robot is equipped with a vision system able to capture features in a 3-dimensional space. Additionally, we consider the particular case of a mobile robot with an omnidirectional base and that force/torque feedback in the end-effector is available. Note that these assumptions attempt to be as general as possible, as these requirements are usually met by most service robots nowadays.

The proposed framework is structured sequentially following the sub-tasks scheme discussed above. Rather than having a distinct contribution to a specific detailed theory or methodology itself, in this work, we combine several state-of-the-art studies. Therefore, the reader can directly refer to the section of interest, indicated in the aforementioned list. The most novel approaches for addressing the door opening task are those presented for sub-tasks 2, 4 and 5.

4 Door and Handle Detection

Doors and handles present a wide variety of geometries, sizes, colors, etc. Thus, a robust detection algorithm is essential. Additionally, in order to achieve real-time estimation, it must operate at speeds of several frames-per-second (fps). Object detection is the task of simultaneously classifying and localizing multiple objects in an image. In (Redmond et al., 2016) the authors proposed the You Only Look Once (YOLO) algorithm, an open-source state-of-the-art object detector with CNN-based regression. This network uses features from the entire image to predict each bounding box, reasoning globally about the full image and all the objects in the image. It enables end-to-end training and facilitates real-time speeds while maintaining high average precision. For these reasons, we decided to adopt this CNN architecture and train it with a custom dataset for addressing the door and handle detection problem.

4.1 Model Training

Training the YOLO network with a custom dataset allows us to build a handle and door detection model. The simplest classification semantics for our objects of interest are "door" and "handle". However, to increase the detail of the information and also to make our method versatile and extendable to other applications, we propose to split the class door into three classes: "door", which refers to a room door, "cabinet door", which includes all sorts of small doors such as drawers or a locker door, and "refrigerator door". We built a data set using images from the Open Images Dataset (Kuznetsova et al., 2020) and annotated a total of 1213 images containing objects of our desired object classes.

Fig. 2 Examples of annotated images from the training dataset used for building the door and handle detection model. The bounding boxes enclose the objects, with the corresponding label, that should be identified by the model.

A total of 1013 images were used for the training set, and the remaining 200 for the testing set (the dataset is available in (Arduengo, 2019)). Some examples of the annotated images are shown in Figure 2. We also applied data augmentation techniques to improve the generalization capabilities (Taylor and Nitschke, 2018).

4.2 Model Selection

For selecting the CNN weights and assessing the model quality, we applied cross-validation against the test set. As the performance index, we propose to use the mean average precision (mAP). This criterion was defined in the PASCAL VOC 2012 competition and is the standard metric for object detectors (Everingham et al., 2015). Briefly, the mAP computation involves the following steps: (1) Based on the likelihood of the predictions, a precision-recall curve is computed for each class, varying the likelihood threshold. (2) The area under this curve is the average precision. Averaging over the different classes we obtain the mAP. Precision and recall are calculated as:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{TP}{TP+FN} \tag{1}$$

where TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative. True or false refers to the assigned classification being correct or incorrect, while positive or negative refers to whether the object is assigned or not to a category.

5 Grasping the Handle

When a robot moves towards an object, it is actually moving towards a pose at which it expects the object to be. For solving the grasping problem, the handle's 6D-pose estimation is essential. The end-effector grasping goal pose can be then easily expressed relative to the handle's pose, and reached by solving the inverse kinematics of the robot. Perception is usually provided by means of an RGB-D sensor, which supplies an RGB image and its corresponding depth map (Alenya et al., 2014; Elbasiony and Gomaa, 2018). For estimating the 6-D pose in real-time, we propose to: (1) Identify the region of the RGB image where the door and the handle are located. (2) Filter the RGB-D image to extract the Regions of Interest, clean the noise and downsample. (3) From a set of 3D geometric features of the door and the handle, estimate the grasping pose. We explain in detail these steps in this section. The proposed approach is summarized in the algorithm below:

Algorithm 1 End-Effector Grasping Pose Estimation

Input: RGB image $\mathcal{I}$ and point cloud $\mathcal{P} = \{\mathbf{p}_j\}_0^{N_{points}}$
Output: Grasping poses $\mathcal{G} = \{\mathbf{g}_k\}_1^{N_{handles}}$ with $\mathbf{g}_k \in SE(3)$
  Bounding boxes $\mathcal{B} = \{b_l\}_1^{N_{objects}} \leftarrow$ Detect_Objects($\mathcal{I}$)
  foreach $b_l \in \mathcal{B}$ **do**
    $\mathcal{P}_l^{ROI} \leftarrow$ ROI_Segmentation($\mathcal{P}$)
    $\mathcal{P}_l^{denoised} \leftarrow$ Remove_Statistical_Outliers($\mathcal{P}_l^{ROI}$)
    $\mathcal{P}_l^{filtered} \leftarrow$ Downsample($\mathcal{P}_l^{denoised}$)
    if class(b_l) = "handle" **then**
      orientation$_l \leftarrow$ Bounding_Box_Dimensions(b_l)
      $\mathcal{P}_l^{handle} \leftarrow$ RANSAC_Plane_Outliers($\mathcal{P}_l^{ROI}$)
      $\mathbf{O}_l \leftarrow$ Centroid($\mathcal{P}_l^{handle}$)
    else
      Normal $\mathbf{a}_l$; $\mathcal{P}_l^{door} \leftarrow$ RANSAC_Plane($\mathcal{P}_l^{filtered}$)
      $\mathbf{O}_l \leftarrow$ Centroid($\mathcal{P}_l^{door}$)
    end if
  end for
  $k = 1$
  foreach $b_l \in \mathcal{B}$ that class(b_l) = "handle" **do**
    $\mathbf{a}_l \leftarrow$ Assign_Closest_Door($\mathbf{O}_l$)
    $\mathbf{h}_k \in SE(3) \leftarrow$ Handle_Transform($\mathbf{a}_l$; $\mathbf{O}_l$)
    $\mathbf{g}_k \leftarrow$ Goal_Pose($\mathbf{h}_k$; orientation$_l$)
    $k \leftarrow k + 1$
  end for
  return $\mathcal{G}$

5.1 Point Cloud Filtering

Raw point clouds contain a large number of point samples, but only a small fraction of them are of interest. Furthermore, they are unavoidably contaminated with noise. Point cloud data needs to be filtered adequately for achieving accurate feature extraction and real-time processing. We propose the following filtering process:

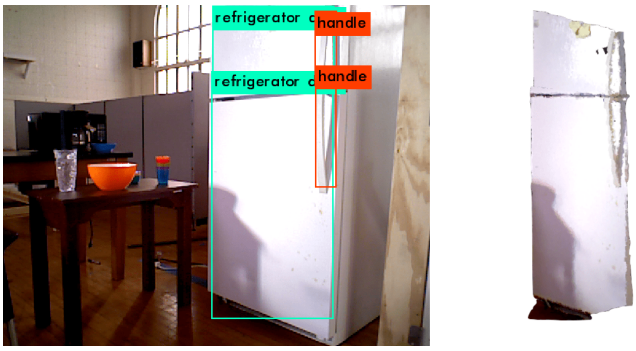

Fig. 3 On the left, the Regions of Interest detected by our door and handle detection model. On the right, the subset of the corresponding point cloud enclosed in the ROIs.

5.1.1 Regions Of Interest (ROIs) Segmentation

The points of interest correspond to the doors and the handles in the scene, which can be defined as those in the bounding boxes of the object detection CNN. By separating the sets of points in each ROI, the amount of data to be processed is reduced significantly (Figure 3). There is a direct correspondence between the pixels in the image and the point cloud indexes if the latter is indexed according to its spatial distribution. As the bounding boxes are usually provided in pixel coordinates, let $\mathcal{P}$ be the raw point cloud. Then, each ROI can be defined as follows:

$$\mathcal{P}^{ROI} = \{\mathbf{p}_j \in \mathcal{P} \,|\, j = \text{width} \cdot y + x\} \tag{2}$$

where j is the point cloud index; width is the image width in pixels, $x \in [x_{min},\ x_{max}]$ and $y \in [y_{min},\ y_{max}]$, being (x_{min}, y_{min}) and (x_{max}, y_{max}) two opposite corners of the bounding box in pixel coordinates.

5.1.2 Statistical Outlier Filtering

Measurement errors lead to sparse outliers, which complicate the estimation of local point cloud features such as surface normals. Some of these irregularities can be solved by performing statistical analysis of each point neighborhood, and trimming those that do not meet a certain criterion. We can carry this analysis at a discrete point level. By assuming that the average distance from every point to all its neighboring points r_j, can be described by a Gaussian distribution, the filtered point cloud can be defined as follows:

$$\mathcal{P}^{denoised} = \{\mathbf{p}_j \in \mathcal{P}^{ROI} \mid r_j \in [\mu_r \pm \alpha \cdot \sigma_r]\} \tag{3}$$

where α is a multiplier, and μ_r and σ_r are the mean distance and the standard deviation, respectively.

Fig. 4 On the left, the raw point cloud. On the right, the downsampled point cloud using a voxelized grid approach.

5.1.3 Downsampling

In order to lighten up the computational load we propose to reduce considerably the amount of data by using a voxelized grid approach (Figure 4). Unlike other subsampling methods, the shape characteristics are maintained. If s is the number of points contained in each voxel A, the set of points in each voxel is replaced by:

$$\bar{x} = \frac{1}{s}\sum_{A} x \qquad \bar{y} = \frac{1}{s}\sum_{A} y \qquad \bar{z} = \frac{1}{s}\sum_{A} z \tag{4}$$

5.2 Grasping Pose Estimation

We have considered three geometric features of the 3D structure of the door and the handle for the grasping pose estimation: the handle orientation, its position and the door plane normal direction.

5.2.1 Handle Orientation

The end-effector orientation for grasping the handle depends on this feature. Since door handles are commonly only oriented vertically or horizontally (for a doorknob, full orientation is not relevant to grasp it), the binary decision can be made by comparing the lengths of the sides of the bounding boxes for the handles in the output of the CNN. If the height is greater than the width, the handle orientation will be vertical and vice versa.

5.2.2 Door Plane Normal

In order to grasp the handle correctly, the normal to the "palm" of the robot's end-effector (which we consider similar to the human hand) must be parallel to the door normal. We propose to use the RAndom SAmple Consensus (RANSAC) algorithm (Rusu, 2013) to compute the normal direction. RANSAC is a numerical method that can iteratively estimate the parameters of a given mathematical model from experimental data that contains outliers, in such a way that they do not influence the values of the estimates.

Fig. 5 The red arrows show the normal direction of the plane defined by each detected door.

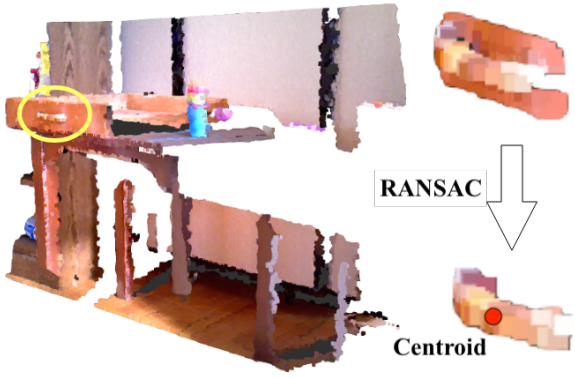

Fig. 6 Computation of the handle position. On the left, the observed scene with the handle highlighted. On the upper-right corner the ROI. On the lower-right corner, the ROI is filtered using RANSAC and the handle position is obtained as the centroid of the resulting point cloud.

A minimal set is formed by the smallest number of points required to uniquely define a given type of geometric primitive. The resulting candidate shapes are tested against all points in the data to determine how many of the points are well approximated by the primitive. RANSAC estimates the model by maximizing the number of inliers (Zuliani, 2017).

Then, in order to compute the door normal direction we fit a planar model to the door point cloud and calculate the coefficients of its parametric Hessian normal form using RANSAC. In Figure 5 we show some examples of the resulting normal vectors obtained with RANSAC.

5.2.3 Handle Position

The proposed approach for estimating the handle position is illustrated in Figure 6. We make the assumption that the handle position can be represented by its centroid. However, it cannot be directly computed from the sub-point cloud associated with the handle ROI, since the defining bounding box usually may include some points from the door in the background. Then, we also use the RANSAC algorithm to separate these points. By fitting a planar model, the ROI points can be classified as inliers and outliers. In this case, the outlier subset corresponds to the handle. The position can then be computed as the centroid of the outliers subset.

5.2.4 Goal Pose Generation

Let $\mathbf{O} = (O_x, O_y, O_z)$ be the handle centroid and $\mathbf{a} = (a_x, a_y, a_z)$ the door plane normal unitary vector, both expressed in an arbitrary reference frame w. The handle pose can be defined as the following transform:

$$\mathbf{T}_w^{handle} = \begin{pmatrix} a_x & \frac{a_y}{a_x^2+a_y^2} & \frac{a_x a_z}{a_x^2+a_y^2} & O_x \\ a_y & -\frac{a_x}{a_x^2+a_y^2} & -\frac{a_y a_z}{a_x^2+a_y^2} & O_y \\ a_z & 0 & -1 & O_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

The grasping pose can then be easily specified as a relative transform to the handle reference frame $\mathbf{T}_{handle}^{grasping}$, taking into account its orientation. Thus, the pose for which the Inverse Kinematics (IK) of the robot must be solved in order to finally grasp the handle, can be computed as:

$$\mathbf{T}_w^{grasping} = \mathbf{T}_w^{handle}\mathbf{T}_{handle}^{grasping} \quad (6)$$

6 Unlatching the Handle

There exists a variety of mechanisms to open a door. Some of them do not require any specific actuation while others generally require a rotation to be applied. A handle usually occupies a small region in the door image. Thus, in order to estimate its kinematic model from visual perception data, the robot camera needs to be placed very close to the handle. This operation would increase considerably the time required to perform the task, making the inference of the handle model using the vision system unappealing.

Instead, we rely on force feedback on the robot's end-effector for inferring how the handle should be actuated. We propose a simple, trial-and-error strategy for operating different types of handle mechanisms, illustrated in Figure 7. The robot tries to turn the handle in both directions, first anti-clockwise and then clockwise. Depending on a torque threshold, either the door is unlatched or no actuation is required. Similarly, as people proceed when opening a door, using the force readings in the direction perpendicular to the door we can determine whether it is required to pull or push to open it by trial-and-error.

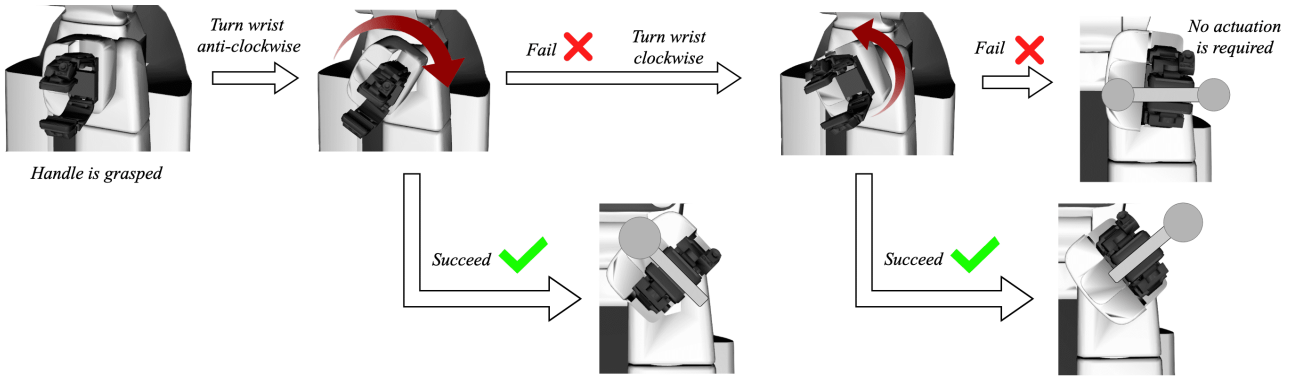

Fig. 7 Proposed handle unlatching strategy. First, the wrist is turned anti-clockwise. If torque feedback is above the allowed threshold, the movement is aborted. Then, the wrist is turned clockwise. If torque feedback is also above the threshold, the handle is identified as "no actuation is required".

7 Learning the Door Kinematic Model

Opening doors in unstructured environments is challenging for robots because they have to deal with uncertainty since the kinematic model of the door is not known a priori. What if a robot has no previous knowledge of the door at the time of taking a decision? And, what if previous knowledge is available? To address these questions, we will present a probabilistic framework that allows inferring the kinematic model of the door when no previous knowledge is available and improve the performance based on previous experiences or human demonstrations.

7.1 Overview of the Probabilistic Framework

Let $\mathcal{D} = (\mathbf{d}_1, \ldots, \mathbf{d}_N)$ be the sequence of N relative transformations between an arbitrary fixed reference frame and the door, observed by the robot. We assume that the measurements are affected by Gaussian noise and, also, that some of these observations are outliers but not originated by the noise. Instead, the outliers might be the result of sensor failures. We denote the kinematic link model as $\mathcal{M}$. Its associated parameters are contained in the vector $\boldsymbol{\theta} \in \mathbb{R}^k$ (where k is the number of parameters). The model that best represents the data can be formulated in a probabilistic context as (Sturm et al., 2010):

$$(\hat{\mathcal{M}}, \hat{\boldsymbol{\theta}}) = \arg\max_{\mathcal{M},\boldsymbol{\theta}} p\left(\mathcal{M}, \boldsymbol{\theta} \mid \mathcal{D}\right) \tag{7}$$

This optimization is a two-step process (MacKay, 2003). First, a particular model is assumed true and its parameters are estimated from the observations:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p\left(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{M}\right) \tag{8}$$

By applying Bayes rule, and assuming that the prior over the parameter space is uniform, this is equivalent to:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}, \mathcal{M}) \tag{9}$$

which shows that fitting a link model to the observations is equivalent to maximizing the data likelihood. Then, we can compare the probability of different models, and select the one with the highest posterior probability:

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \int p\left(\mathcal{M}, \boldsymbol{\theta} \mid \mathcal{D}\right) d\boldsymbol{\theta} \tag{10}$$

Summarizing, given a set of observations $\mathcal{D}$, and candidate models $\mathcal{M}$ with parameters $\boldsymbol{\theta}$, the procedure to infer the kinematic model of the door consists of: (1) fitting the parameters of all candidate models; (2) selecting the model that best describes the observed motion.

7.2 Candidate Models

When considering the set of doors that can be potentially operated by a service robot, their kinematic models belong to a few generic classes (Rühr et al., 2012). We have considered as candidate kinematic models a prismatic model, and a revolute model, shown in Figure 8.

7.2.1 Prismatic model

Prismatic joints move along a single axis. Their motion describes a translation in the direction of a unitary vector $\mathbf{e} \in \mathbb{R}^3$ relative to a fixed origin, $\mathbf{a} \in \mathbb{R}^3$. The parameter vector is $\boldsymbol{\theta} = (\mathbf{a}; \mathbf{e})$ with $k = 6$.

candidate model $\mathcal{M}$	DOFs d	parameters k
prismatic model	1	6
revolute model	1	7

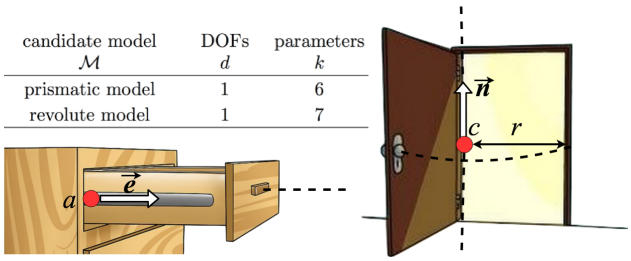

Fig. 8 Prismatic and revolute candidate kinematic models.

7.2.2 Revolute model

Revolute joints rotate around an axis that impose a one-dimensional motion along a circular arc. It can be parametrized by the center of rotation $\mathbf{c} \in \mathbb{R}^3$, a radius $\mathbf{r} \in \mathbb{R}$, and the normal vector $\mathbf{n} = \mathbb{R}^3$ to the plane where the motion arc is contained. This results in a parameter vector $\boldsymbol{\theta} = (\mathbf{c}; \mathbf{n}; r)$ with $k = 7$.

7.3 Model Fitting

In the presence of noise and outliers, finding the parameter vector $\hat{\boldsymbol{\theta}}$ that maximizes the data likelihood is not trivial, as least square estimation is sensitive to outliers. The RANSAC algorithm has proven to be robust in this case and can be modified in order to maximize the likelihood. This is the approach implemented by the Maximum Likelihood Estimation SAmple Consensus (MLESAC) algorithm (Torr and Zisseman, 2000). In this case, the score is defined by the likelihood of the consensus sample. Thus, for estimating the model vector parameter $\boldsymbol{\theta}$, the log-likelihood of a mixture model is maximized (Zuliani, 2017):

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}\left[e(\mathcal{D} \mid \mathcal{M}, \boldsymbol{\theta})\right] \tag{11}$$

$$\begin{aligned}\hat{\mathcal{L}} = \sum_{j=1}^{N} \log \Big(& \gamma \cdot p\left[e(\mathbf{d}_j, \mathcal{M}, \hat{\boldsymbol{\theta}}) \mid \mathrm{j^{th}element} \equiv \text{ inlier}\right] \\ & + (1-\gamma)\, p\left[e(\mathbf{d}_j, \mathcal{M}, \hat{\boldsymbol{\theta}}) \mid \mathrm{j^{th}element} \equiv \text{outlier}\right] \Big)\end{aligned} \tag{12}$$

where γ is the mixture coefficient, which is computed with Expectation Maximization. The first and second term correspond to the error distribution $e(\mathbf{d}_j, \mathcal{M}, \hat{\boldsymbol{\theta}})$ of the inliers and the outliers, respectively. The error statistics of the inliers are modeled with a Gaussian. On the other hand, the error of the outliers is described with a uniform distribution.

7.4 Model Selection

Once all model candidates are fitted to the observations, the model that best explains the data has to be selected (Sturm et al., 2011). Let $\mathcal{M}_m$ $(m = 1, ..., M)$ be the set of candidate models, with vector parameters $\boldsymbol{\theta}_m$. Let $p\left(\boldsymbol{\theta}_m | \mathcal{M}_m\right)$ be the prior distribution for the parameters. Then, the posterior probability of a given model is proportional to (Hastie et al., 2009):

$$p\left(\mathcal{M}_m \mid \mathcal{D}\right) \propto \int p\left(\mathcal{D} \mid \boldsymbol{\theta}_m, \mathcal{M}_m\right) p\left(\boldsymbol{\theta}_m \mid \mathcal{M}_m\right) d\boldsymbol{\theta}_m \tag{13}$$

In general, computing this probability is difficult. Applying the Laplace approximation and assuming a uniform prior for the models, i.e. probability of model M_m of being the true model before having observed any data, it can be estimated in terms of the Bayesian Information Criterion (BIC):

$$p(\mathcal{M}_m \mid \mathcal{D}) \approx \frac{\exp\left(-\frac{1}{2}\triangle BIC_m\right)}{\sum_{m=1}^{M} \exp\left(-\frac{1}{2}\triangle BIC_m\right)} \tag{14}$$

where: $\triangle BIC_m = BIC_m - \min\left\{BIC_m\right\}_1^M$, and:

$$BIC_m = -2 \log\left[\mathcal{L}\left(\mathcal{D} \mid \mathcal{M}_m, \hat{\boldsymbol{\theta}}_m\right)\right] + k \cdot \log N \tag{15}$$

The first term accounts for the likelihood of the fit, and the second term for the model complexity; smaller BIC are preferred. Thus, model selection can be reduced to select the model with the lowest BIC:

$$\hat{\mathcal{M}} = \arg\min_{\mathcal{M}} BIC(\mathcal{M}) \tag{16}$$

7.5 Exploiting Prior Knowledge

A robot operating in domestic environments can boost its performance by learning priors from previous experiences (Calinon, 2016). A small set of representative models can be used as prior information to improve the model selection and parameter estimation in an unknown environment.

Suppose that the robot has previously encountered two doors. We have two observation sequences $\mathcal{D}_1$ and $\mathcal{D}_2$, with N_1 and N_2 samples. We must choose then between two distinct models $\mathcal{M}_1$ and $\mathcal{M}_2$ or a joint model $\mathcal{M}_{1+2}$. In the first case, the posterior can be split as the two models are mutually independent:

$$p\left(\mathcal{M}_1, \mathcal{M}_2 \mid \mathcal{D}_1, \mathcal{D}_2\right) = p\left(\mathcal{M}_1 \mid \mathcal{D}_1\right) \cdot p\left(\mathcal{M}_2 \mid \mathcal{D}_2\right) \tag{17}$$

In the second case, both trajectories are explained by a single, joint model $\mathcal{M}_{1+2}$, which is estimated from the joint data $\mathcal{D}_1 \cup \mathcal{D}_2$. The corresponding posterior probability is denoted $p\left(\mathcal{M}_{1+2} \mid \mathcal{D}_1, \mathcal{D}_2\right)$. In order to determine whether a joint model explains the observed data better than two separate models we can compare the posterior probabilities:

$$p\left(\mathcal{M}_{1+2} \mid \mathcal{D}_1, \mathcal{D}_2\right) > p\left(\mathcal{M}_1 \mid \mathcal{D}_1\right) \cdot p\left(\mathcal{M}_2 \mid \mathcal{D}_2\right) \tag{18}$$

This expression can be evaluated efficiently using the BIC

$$BIC\left(\mathcal{M}_{1+2} \mid \mathcal{D}_1, \mathcal{D}_2\right) < \sum_{i=1}^{2} BIC\left(\mathcal{M}_i \mid \mathcal{D}_i\right) \tag{19}$$

Intuitively, merging two models into one is beneficial if the joint model can explain the data equally well while requiring only a single set of parameters. If we consider more than two trajectories, this should be repeated for all the possible combinations. This can become hard to compute. Thus, instead, we check if merging the new data with each learned model associated with the door class being opened gives a higher posterior. In this way, when opening a refrigerator door, the observations are only going to be compared with the previous refrigerator door openings. In this way, the observed data is more likely to match the recorded data, and trajectories that are not likely to match (i.e. a drawer) are not considered. Finally, we pick the model with the highest posterior and record the new data, which will be used as prior knowledge for future doors. This approach is summarized in algorithm 2.

Algorithm 2 Model Selection Using Prior Knowledge

Input: New observed trajectory $\mathcal{D}_{\text{new}} = \left\{\mathbf{d}_j^{\text{new}}\right\}_1^N$;
door class $c \in \{\text{door, cabinet door, refrigerator door}\}$;
previously observed trajectories $\mathbb{D}_c = \{\mathcal{D}_s\}_1^S$
Output: Best model $\mathbb{M}_{\text{best}}$ and prior knowledge updated $\mathbb{D}_c$
$\mathcal{M}_{\text{new}} \leftarrow \text{Kinematic\_Model}\left(\mathcal{D}_{\text{new}}\right)$
$\mathbb{M}_{\text{best}} \leftarrow \{\mathcal{M}_{\text{new}}\}$, $\mathbb{D}_c \leftarrow \mathbb{D}_c \cup \{\mathcal{D}_{\text{new}}\}$, $p_{\text{best}} \leftarrow 0$
foreach $\mathcal{D}_s \in \mathbb{D}$ **do**
$\mathcal{M}_s \leftarrow \text{Kinematic\_Model}\left(\mathcal{D}_s\right)$
$\mathcal{M}_{\text{new+s}} \leftarrow \text{Kinematic\_Model}\left(\mathcal{D}_{\text{new}} \cup \mathcal{D}_s\right)$
if $p\left(\mathcal{M}_{\text{new+s}} | \mathcal{D}_{\text{new}}, \mathcal{D}_s\right) > p\left(\mathcal{M}_{\text{new}} | \mathcal{D}_{\text{new}}\right) p\left(\mathcal{M}_s | \mathcal{D}_s\right)$ & $p\left(\mathcal{M}_{\text{new+s}} | \mathcal{D}_{\text{new}}, \mathcal{D}_s\right) > p_{\text{best}}$ **then**
$\mathbb{M}_{\text{best}} \leftarrow \{\mathcal{M}_{\text{new}}, \mathcal{M}_s\}$
$\mathbb{D}_c \leftarrow \{\mathcal{D}_1, \ldots, \mathcal{D}_{\text{new}} \cup \mathcal{D}_s \ldots, \mathcal{D}_S\}$
$p_{\text{best}} \leftarrow p\left(\mathcal{M}_{\text{new+s}} \mid \mathcal{D}_{\text{new}}, \mathcal{D}_s\right)$
end if
end for
return $\mathbb{M}_{\text{best}}$ and $\mathbb{D}_c$

7.6 Learning from Human Demonstrations

If robots can learn from demonstration, this can boost the scale of the process, since nonexperts would be able to teach them (Lee, 2017). With our probabilistic framework, the only necessary input we need is a set of observations of the door's motion. Using our 6D-pose estimation approach, this tracking behavior can be efficiently achieved. Thus, the robot's prior knowledge can be provided by human demonstrations (Figure 9).

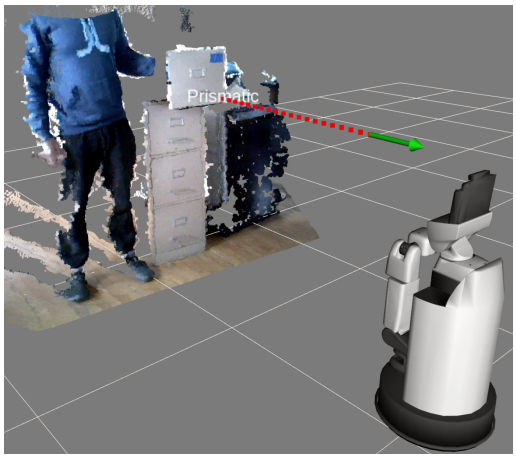

Fig. 9 Observations can be provided by executions of the task by a human teacher. In this case, the robot infers the motion of the cabinet is described by a prismatic model.

8 Execution of the Door Opening Motion

Computing the motion that enables a mobile manipulator to open a door is challenging because it requires tight coordination between arm and base. This makes the problem high-dimensional and thus hard to plan. In the previous section, we have discussed how to learn the door kinematic model from observations of its motion. In order to achieve full autonomy, although observations are not provided before-hand, the robot must also be able to operate the previously unseen door. In this section, we will discuss how these issues can be addressed through a suitable motion planning framework and an effective door opening strategy.

8.1 Task Space Region (TSR)

Task Space Region is a constrained manipulation planning framework presented in (Berenson et al., 2011). The authors propose a specific constraint representation, that has been developed for planning paths for manipulators with end-effector pose constraints. The framework unifies an efficient constraint representation, constraint satisfaction strategies, and a sampling-based planner, to create a state-of-the-art whole-body manipulation planning algorithm.

The sampling-based planner is based on rapidly exploring random trees (RRTs). Thus, it inherits many of the limitations of sampling-based methods for planning. For instance, it is very difficult to incorporate non-holonomic constraints and dynamics because these constraints would disrupt the distance metric used by the RRT. For these reasons, the applicability of this framework is limited to robots without non-holonomic constraints.

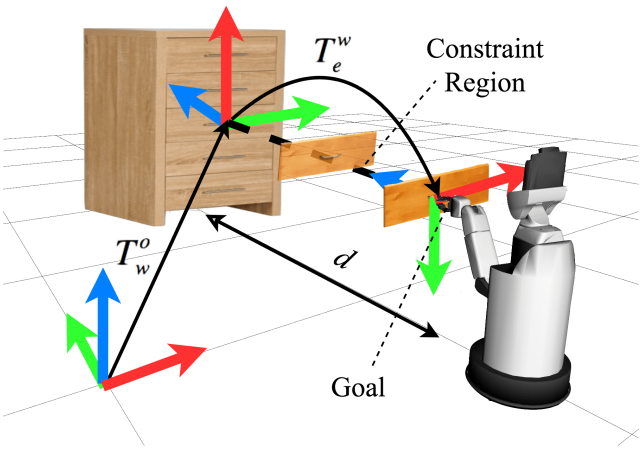

Fig. 10 TSR representation for operating prismatic doors. The x, y and z axis of each reference frame are red, green and blue respectively.

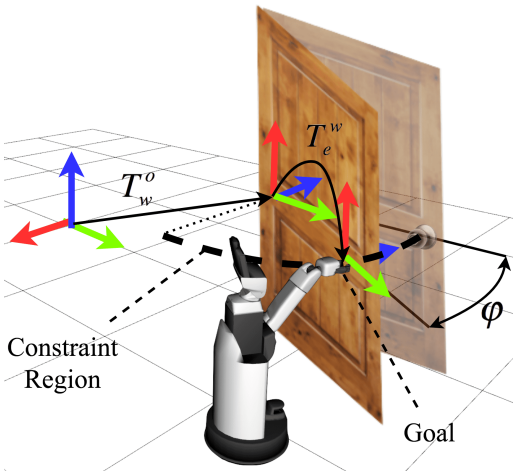

Fig. 11 TSR representation for operating revolute doors. The x, y and z axis of each reference frame are red, green and blue respectively.

TSRs describe end-effector constraint sets as subsets of $SE(3)$ (Special Euclidean Group). These subsets are particularly useful for specifying manipulation tasks. Once the end-effector pose restrictions are specified in terms of a TSR, the algorithm finds a path that lies in the constraints manifold. To define a TSR, three elements are required:

- $\boldsymbol{T}_w^o$: Transform between the origin reference frame o and the TSR frame w.
- $\boldsymbol{T}_e^w$: End-effector offset transform.
- $\boldsymbol{B}^w$: 6×2 matrix that defines the end-effector constraints, expressed in the TSR reference frame

$$(\boldsymbol{B}^w)^T = \begin{pmatrix} x_{\min} & y_{\min} & z_{\min} & \varphi_{\min} & \theta_{\min} & \psi_{\min} \\ x_{\max} & y_{\max} & z_{\max} & \varphi_{\max} & \theta_{\max} & \psi_{\max} \end{pmatrix} \tag{20}$$

where the first three columns bound the allowable translation along the x, y and z axes, and the last three columns bound the allowable translation assuming the Roll-Pitch-Yaw angle convention. Thus, the end-effector constraints for the considered kinematic models can be easily specified as follows.

8.1.1 TSR representation for Prismatic Doors

By fitting a prismatic model to the observations we can estimate the axis along which the door moves. If this axis is determined, we can specify the TSR reference frame as shown in Figure 10, and define the end-effector pose constraints as:

$$(\boldsymbol{B}^w)^T = \begin{pmatrix} 0 & 0 & -d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{21}$$

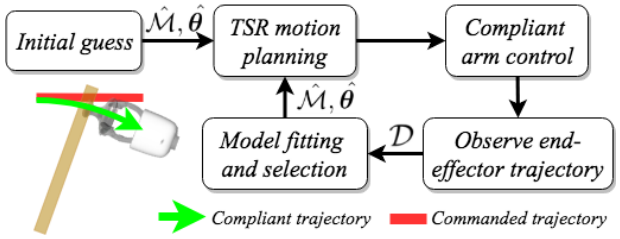

Fig. 12 Adaptive door opening procedure scheme. The robot opens the door following these steps iteratively.

8.1.2 TSR representation for Revolute Doors

In the case of fitting a revolute model to the observations, we can estimate the center of rotation, the radius and the normal axis. With these parameters, we can specify the TSR reference frame as shown in Figure 11, and define the end-effector pose constraints as:

$$(\boldsymbol{B}^w)^T = \begin{pmatrix} 0 & 0 & 0 & -\varphi & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{22}$$

8.2 Door Opening Procedure

To define the TSR reference frame, observations of the door motion are also required. Instead of using visual perception, it can also be inferred by direct actuation. Once the handle is grasped, the position of the end-effector directly corresponds with the position of the handle. As a result, the robot can make observations of the motion by solving its forward kinematics. Thus, $\mathcal{D}$ can be obtained by sampling the trajectory. We execute the door opening motion repeating iteratively a series of sequential steps, shown in Figure 12. After each iteration, we re-estimate the kinematic model of the door and its parameters adding the new observations to $\mathcal{D}$.

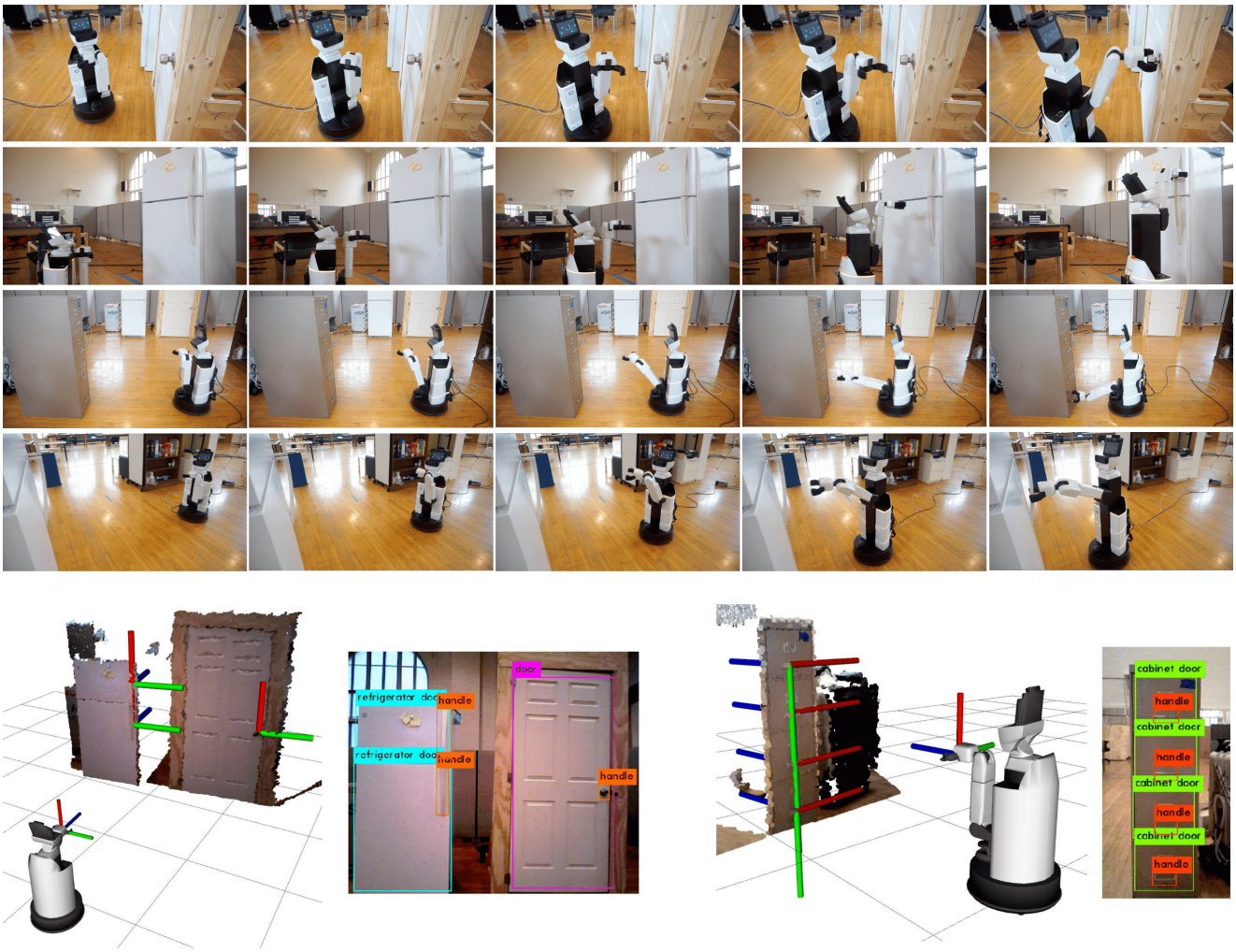

Fig. 13 On top, a series of pictures of the HSR robot grasping different handles, starting from various relative positions. Below, the estimated grasping pose for the handles in the scene, as well as the corresponding detections provided by our CNN. The estimated grasping pose is illustrated through the red, green and blue axis (x, y and z respectively). Note the end-effector reference frame is shown at the HSR gripper. Video demonstrations are available at https://www.youtube.com/watch?v=LbDfKPpxEss.

To start the opening process, when no observations are available, we make the initial guess that the model is prismatic. Using a compliant controller, the robot's end-effector trajectory is also driven by the forces exerted by the door, adapting its motion to the true model. Thus, a certain error margin is allowed, enabling the robot to operate the door correctly even if the estimation is biased when only a few observations have been acquired.

9 Experimental Evaluation

In order to validate experimentally the proposed door operation framework, we implemented it on the Toyota HSR robot, a robot designed to provide assistance. It is equipped with an omnidirectional wheeled base, an arm with 4 degrees-of-freedom, a lifting torso and a two-fingered gripper as an end-effector.

In this work, we take advantage of the sensorial feedback provided by a 6-axis force sensor, located on the wrist, and an RGB-D camera, located on its head. We conducted a series of real-world experiments to test the performance of the presented grasping pose estimator and the proposed kinematic model inference process. We tested the latter in two different scenarios: with and without exploiting prior knowledge. To assess the robustness of our framework, we used different doors such as cabinet, refrigerator, and room doors with their variety of handles. Video demonstrations are available at the following hyperlink.

9.1 Grasping Pose Estimation

For evaluating the performance of the grasping pose estimation, we focused on accuracy and speed. Regarding the door and handle detection model, due to the

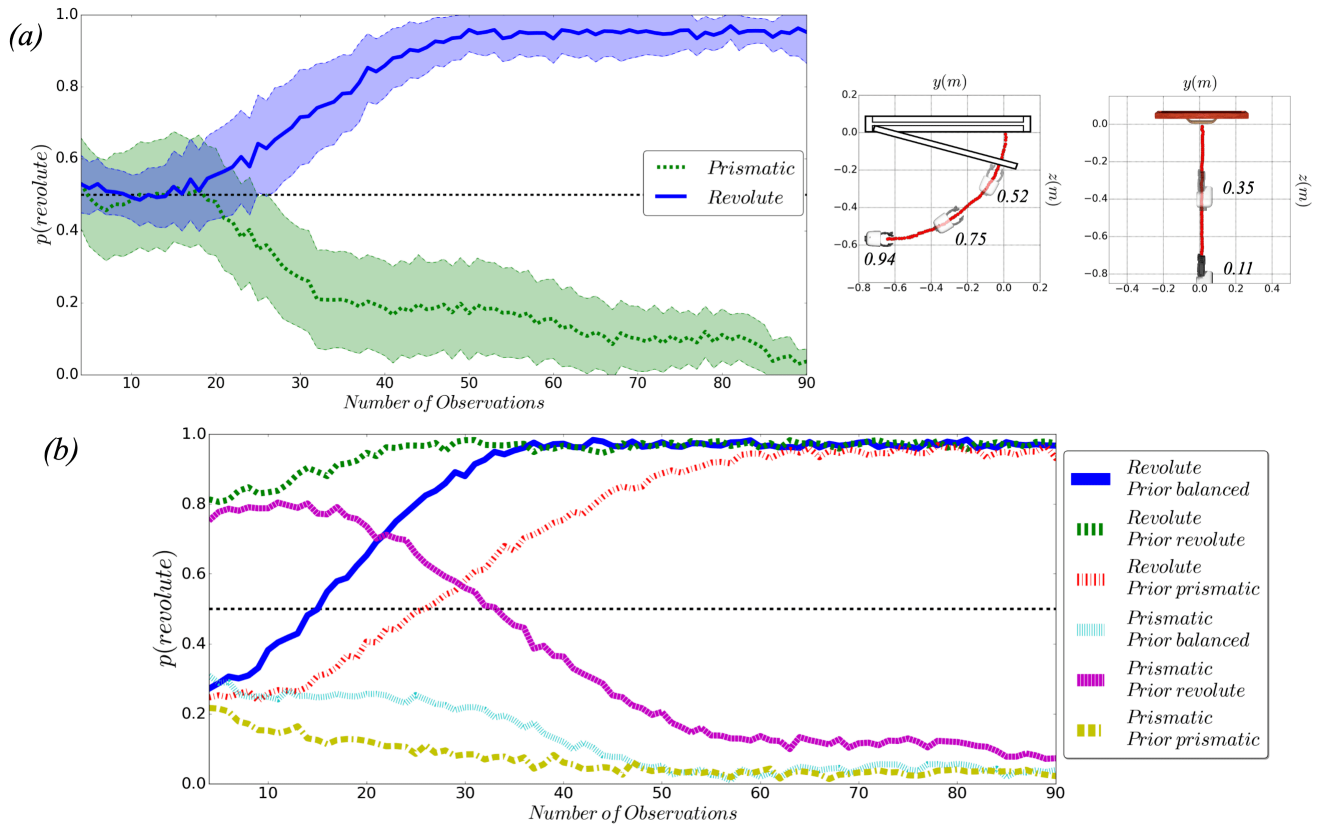

Fig. 14 (a) The posterior of the revolute model vs the number of observations. In the legend, the door true models are indicated. The means of the executions are displayed as continuous lines. The shaded areas represent a margin of two standard deviations. Next to the plot, the evolution of the posterior along the opening trajectory is shown graphically. (b) Evolution of the revolute posterior mean against the number of observations. The legend indicates the true model of the doors being opened and the predominant prior during the realization.

limited range of different doors available in the laboratory, its accuracy is best assessed, as discussed in Section 4, by computing the mAP on the test set, with a wide variety of doors and handles. The resulting mAP of the selected model, as well as some reference values for comparison, are shown in Table 1. We can see that our model's mAP is just 10% lower. This performance value is close to that obtained by state-of-the-art object detectors in high-quality image datasets.

Qualitatively, testing the model in the laboratory, the available doors and handles were effectively detected from different viewpoints. Given a successful detection, the algorithm always computed the grasping pose of the handles present in the image correctly. By solving the IK, if the handle was located within the reachable workspace, the robot was always able to grasp the handle. Using an Nvidia Geforce GTX 1080 GPU, we obtained a computation rate of 6fps. This shows an efficient behavior of the presented real-time grasping pose estimator. In Figure 13 we show a series of pictures that illustrate how the HSR robot reaches the handle in different scenarios, after inferring the grasping pose with our method. An effective grasping is achieved from several starting positions and types of doors. We also show some examples of the estimated goal pose from RGB-D data. We can observe that is accurately located in the observed point cloud for all the handles simultaneously.

Table 1 mAP comparison

	mAP
YOLO on COCO dataset	55%
YOLO on VOC 2012	58%
YOLO on our custom dataset	45%

9.2 Kinematic Model Inference

In order to evaluate the door kinematic model inference process, when no prior knowledge is available, we opened three different types of doors ten times: a drawer, a room and a refrigerator door. The task of the robot was to grasp the handle and open the door while it learned its kinematic model. The robot succeeded 26 times out of 30 trials (87%). All four failures were due

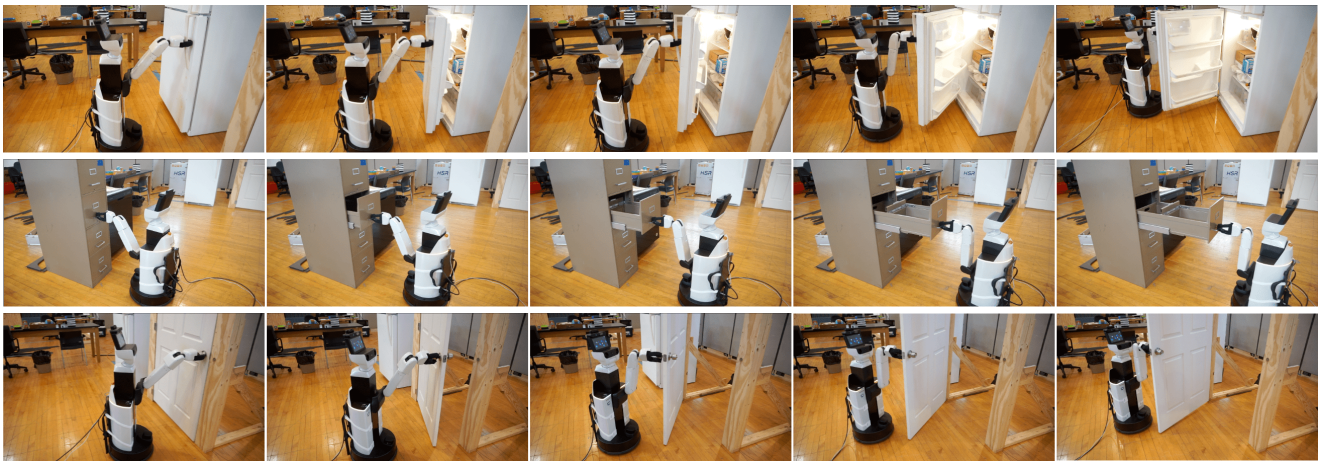

Fig. 16 The HSR robot successfully opens different types of doors, without a priori knowledge of their kinematic model.

to the gripper slipping from the doorknob, most likely caused by the design of the gripper which is not very suitable to manipulate this kind of object. No errors were observed during model learning.

We also studied the convergence of the estimators versus the number of training samples. We considered ten successful openings for each of the two considered kinematic models. Results are shown in Figure 14(a). During the task, the evolution of the candidate posterior model was evaluated against the number of observations. It can be seen that the posterior probability for both cases converges towards the true model as the number of observations increases. When a observations are acquired, the probability oscillates around 0.5, which is consistent with considering equal priors. However, they soon diverge from this value, showing an effective behavior regarding the decision criterion. A more convergent behavior is visible in the case of a revolute door. This is due to the difference in complexity between both models. When a prismatic door is opened, the revolute model can fit the data, which does not happen in the opposite case.

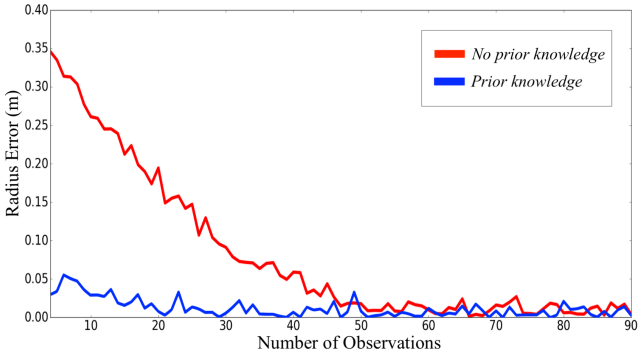

Fig. 15 Evolution of the estimation error of the radius of a revolute door during its operation. Two scenarios are compared: without and with previous knowledge.

Then, we analyze our approach for exploiting prior knowledge. We reproduced the same experiments when no prior knowledge is available but for three different situations: when the prior is predominantly revolute or prismatic, and when both are balanced. Results are shown in Figure 14(b). It can be observed that the behavior depends on the predominant prior. In the case it matches the true model, the posterior converges quickly. If the prior is balanced, the evolution depends on the true model. When few new observations are available, the posterior tends to converge to the simplest model which is prismatic. This is reasonable since the trajectory is very similar for both models at this point but the complexity is penalized. However, at a relatively low number of observations, the posterior rapidly converges to the true model proving, therefore, an improvement in performance. Note that priors start around 0.3, this is because the new observations are matched to a previous model already from the starting point. Finally, in the case the prior does not match the true model, the behavior is symmetric for both doors. At the beginning, the observations converge with the predominant prior model. However, when the number of observations is sufficiently large, they converge towards the true model. A numerical evaluation of the advantage of exploiting prior knowledge is shown in Figure 15, where we can observe the evolution of the radius estimation error when opening a revolute door with and without providing demonstrations of its motion. By exploiting prior knowledge, we can see that the estimation error is almost null from the initial stages of the opening process, which does not occur in the other scenario. Finally, in Figure 16 we show a series of pictures that illustrate how the HSR successfully opens different doors. Combining the proposed probabilistic approach, with the TSR manipulation framework, the robot can operate doors autonomously in an unknown environment.

10 Conclusion

In this work, our objective is to push the state-of-the-art towards achieving autonomous door operation. The door opening task involves a series of challenges that have to be addressed. In this regard, we have discussed the detection of doors and handles, the handle grasp, the handle unlatch, the identification of the door kinematic model, and the planning of the constrained opening motion.

The problem of rapidly grasping door handles leads to the first paper contribution. A novel algorithm to estimate the required end-effector grasping pose for multiple handles simultaneously, in real-time, based on RGB-D has been proposed. We have used a CNN, providing reliable results, and efficient point cloud processing to devise a high-performance algorithm, which proved robust and fast in the conducted experiments. Then, in order to operate the door reliably and independently of its kinematic model, we have devised a probabilistic framework for inferring door models from observations at run time, as well as for learning from robot experiences and from human demonstrations. By combining the grasp and model estimation processes with a TSR robot motion planner, we achieved a reliable operation for various types of doors.

Our desire is to extend this work to include more general and complex kinematic models (Barragan et al., 2014; Hoefer et al., 2014). This would enable robots, not only to achieve robust door operations but would ultimately achieve generally articulated object manipulation. Furthermore, the use of non-parametric models, such as Gaussian processes, would allow the representation of even more complex mechanisms. Also, we would like to explore in more depth the possibility of integrating our system in a general Learning from Demonstration (LfD) framework.

References

I. Abraham, A. Handa, N. Ratliff, K. Lowrey, T. D. Murphey, and D. Fox. Model-based generalization under parameter uncertainty using path integral control. *IEEE Robotics and Automation Letters*, 5(2): 2864–2871, 2020.

G. Alenya, S. Foix, and C. Torras. Using ToF and RGBD cameras for 3D robot perception and manipulation in human environments. *Intelligent Service Robotics*, 7:211–220, 2014.

M. Arduengo. Labelled image dataset for door and handle detection. https://github.com/MiguelARD/DoorDetect-Dataset, 2019.

T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann. Toward humanoid manipulation in human-centred environments. *Robotics and Autonomous Systems*, 56(1):54–65, 2008.

N. Banerjee, X. Long, R. Du, F. Polio, S. Feng, C. Atkeson, M. Gennert, and T. Padir. Human-supervised control of the ATLAS humanoid robot for traversing doors. *15th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 722–729, Nov 2015.

P. R. Barragan, L. P. Kaelbling, and T. Lozano-Perez. Interactive bayesian identification of kinematic mechanisms. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2013–2020, May 2014.

D. Berenson, S. Srinivasa, and J. Kuffner. Task Space Regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, 30(12):1435–1450, 2011.

S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9: 1–29, 2016.

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

R. Diankov, S. Srinivasa, D. Ferguson, and J. Kuffner. Manipulation planning with caging grasps. *8th IEEE-RAS International Conference on Humanoid Robots*, pages 285–292, Dec 2008.

R. Elbasiony and W. Gomaa. Humanoids skill learning based on real-time human motion imitation using Kinect. *Intelligent Service Robotics*, 11:149–169, 2018.

F. Enders, J. Trinkle, and W. Burgard. Learning the dynamics of doors for robotic manipulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3543–3549, Nov 2013.

C. Eppner, R. Martín-Martín, and O. Brock. Physics-based selection of informative actions for interactive perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7427–7432, 2018.

M. Everingham, S.-M. Eslami, L. Gool, C.-K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan 2015.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Verlag, 2009.

S. Hoefer, T. Lang, and O. Brock. Extracting kinematic background knowledge from interactions using task-sensitive relational learning. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4342–4347, May 2014.

A. Jain and C.-C. Kemp. Pulling open novel doors and drawers with equilibrium point control. *9th IEEE/RAS International Conference on Humanoid Robots*, pages 498–505, Dec 2009.

Y. Karayiannidis, C. Smith, F. Vina, P. Ogren, and D. Kragic. Model-free robot manipulation of doors and drawers by means of fixed-grasps. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4485–4492, May 2013.

C.-C. Kessens, J. Rice, D. Smith, S. Bigss, and R. Garcia. Utilizing compliance to manipulate doors with unmodeled constraints. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 483–489, Oct 2010.

A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 2020.

J. Lee. A survey of robot learning from demostrations for Human-Robot collaboration. *ArXiv e-prints*, 2017. arXiv:1710.08789v1.

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. *3rd International Conference on Control, Automation and Robotics*, pages 144–149, April 2017.

E. Lutscher, M. Lawitzky, G. Cheng, and S. Hirche. A control strategy for operating unknown constrained mechanisms. *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

Y. Mae, H. Takahashi, K. Ohara, T. Takubo, and T. Arai. Component-based robot system design for grasping tasks. *Intelligent Service Robotics*, 4, 2011.

W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Erhimov, T. Foote, J. Husu, R. Rusu, B. Marthi, G. Bradski, K. Konolige, B. Gerkey, and E. Berger. Autonomous door opening and plugging in with a personal robot. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 729–736, May 2010.

B. Nemec, L. Zlajpah, and A. Ude. Door opening by joining reinforcement learning and intelligent control. *18th International Conference on Advanced Robotics (ICAR)*, pages 222–228, July 2017.

C. Ott, B. Bauml, C. Borst, and G. Hirzinger. Employing cartesian impedance control for the opening of a door: A case study in mobile manipulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Workshop on mobile manipulators*, Jan 2005.

J. Redmond, S. Divvala, R. Grirshick, and A. Farhadi. You only look once: Unified, real-time object detection. *ArXiv e-prints*, 2016. arXiv:1506.02640v5.

R. Rusu. *Semantic 3D object maps for everyday robot manipulation*. Springer Verlag, 2013.

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

S. Schiffer, A. Ferrein, and G. Lakemeyer. Caesar: an intelligent domestic service robot. *Intelligent Service Robotics*, 5:259–273, 2012.

J. Sturm. *Approaches to probabilistic model learning for mobile manipulation robots*. Springer Tracts in Advanced Robotics (STAR), volume 89. Springer, 2013.

J. Sturm, A. Jain, C. Stachniss, C. Kemp, and W. Burgard. Operating articulated objects based on experience. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2010.

J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011.

L. Taylor and G. Nitschke. Improving Deep Learning using generic Data Augmentation. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547, 2018.

P. Torr and A. Zisseman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78 (1):138–156, 2000.

C. Torras. Service Robots for Citizens of the Future. *European Review*, 24(1):17–30, 2016.

T. Welschehold, C. Dornhege, and W. Burgard. Learning mobile manipulation actions from human demonstrations. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3196–3201, Sept 2017.

S. Wieland, D. Gonzalez-Aguirre, N. Vahrenkamp, T. Asfour, and R. Dillmann. Combining force and visual feedback for physical interaction task in humanoid robots. *9th IEEE-RAS International Conference on Humanoid Robots*, pages 439–446, Dec 2009.

M. Zuliani. RANSAC for dummies. Technical report, UCSB Vision Research Lab, 2017.