Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Robotic Systems Lab
Prof. Dr. Marco Hutter

Master Thesis

Door State Estimation and Parameter Identification in Mobile Manipulation

Autumn Term 2021

Supervised by:
Jean-Pierre Sleiman, Mayank Mittal
Andreea Tulbure, Prof. Dr. Marco Hutter

Author:
Yifei Dong

Declaration of Originality

I hereby declare that the written work I have submitted entitled

Door State Estimation and Parameter Identification in Mobile Manipulation

is original work which I alone have authored and which is written in my own words.[1]

Author(s)

Yifei	Dong

Student supervisor(s)

Jean-Pierre	Sleiman
Mayank	Mittal
Andreea	Tulbure

Supervising lecturer

Marco	Hutter

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette' (`https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetiquette.pdf`). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Zürich , 23/01/2022
Place and date

Yifei Dong
Signature

[1]Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Intellectual Property Agreement

The student acted under the supervision of Prof. Hutter and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.

This agreement regulates the rights to the created research results.

1. Intellectual Property Rights

1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights ("Urheberpersönlichkeitsrechte"), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student's moral rights ("Urheberpersönlichkeitsrechte") shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines "Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich".
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Zürich , 23/01/2022

Place and date

[illegible]

Signature

Contents

Preface

The basis for this research originally stems from my interest in developing better methods of quadruped robot's mobile manipulation. As the world moves further into the automation age, robots are more and more frequently applied in some areas of our daily life and expected to be highly interactive with their surroundings. Robust and efficient mobile manipulation is currently in an immense demand. My passion lies in finding out ways of reliably locating the position of the manipulated object, recursively estimating its parameters and sizes, and effectively controlling articulated floating base system in mobile manipulation.

In truth, I could not have achieved my current level of fulfillment without a strong support group. First of all, Prof. Marco Hutter, Jean-Pierre Sleiman, Mayank Mittal, Andreea Tulbure, and some other colleagues at Robotic Systems Lab, who supported me insightfully with great ideas and full heart. Secondly, I appreciate it that some students working on master's thesis at RSL share their valuable opinions on my project. Most importantly, I am grateful to my parents and friends who always stand behind me. Thank you all for your unwavering support.

Abstract

Recent advances in 3D vision and deep learning have led to marked improvement over the area of perception and thus perceptive manipulation. In this work, we show how to recursively estimate the state and identify the parameters of an unknown door from a camera mounted on a mobile robot.

First, we present a real-time door and handle state estimation framework for door manipulation. The framework makes use of a YOLOv5 object detector, point cloud processing, and Kalman filtering. We deploy the framework on hardware, which runs at 5-7 fps.

Second, we propose a novel recursive approach, which combines RGB-D based detection with odometry, and estimates the door pose in the world coordinate. The approach demonstrates consistent robustness against occlusions and interferential doors.

We evaluate our approaches thoroughly on our dataset with various types of doors, consisting of RGB images, depth images, and robot odometry. The algorithm demonstrates consistent high performance despite of motion blur, changes in lighting condition, or the dimensions and completeness of the door. Preliminary open-loop experiments are done in simulation and hardware of the task-space visual servoing of handle grasping.

Keywords— Door Estimation, Perception for Manipulation, Visual Tracking, Parameter Identification

Symbols

Symbols

$\{\mathcal{W}\}$	World Coordinate
$\{\mathcal{C}\}$	Camera Coordinate
$\{\mathcal{I}\}$	Image Coordinate
$\{\mathcal{D}\}$	Door Coordinate
$b_k^1, b_k^2, ...$	Candidate masks from the YOLO inference of the current frame k
B_k	Selected mask to analyze
$\mathbf{z}_k$	Measurement
$\mathbf{x}_k$	Posteriori State
$\tilde{b}_k$	Re-projective validation mask
${}_D\mathbf{r}_k^{DH}$	Posteriori handle position in $\{\mathcal{D}\}$
${}_W\mathbf{r}_k^{WD}$	CoM position of the door in $\{\mathcal{W}\}$
${}_W\mathbf{n}_k^{W}$	Normal of the door in $\{\mathcal{W}\}$ posteriori
${}_Dh_k$	Posteriori door height
${}_Dw_k$	Posteriori door width
$\mathbf{u}_k$	Control input vector
$\mathbf{F}_k$	Transition matrix
$\mathbf{H}_k$	Observation matrix
$\mathbf{w}_k$	Temporally uncorrelated noise
$\mathbf{v}_k$	Measurement noise
$\mathbf{Q}_k$	Covariance of the process noise
$\mathbf{R}_k$	Covariance of the observation noise
$\mathbf{P}_k$	State covariance matrix
$\tilde{\mathbf{y}}_k$	Measurement residual
$\mathbf{S}_k$	Innovation covariance
$\mathbf{K}_k$	Kalman gain
${}_{\mathcal{W}}\mathbf{M}_k^i$	Corner points of the door
${}_{\mathcal{W}}\mathbf{N}_k$	Center point of the handle
$\mathbf{C}_k$	Camera intrinsics

Acronyms and Abbreviations

RL	Reinforcement Learning
RoI	Region of Interest
IoU	Intersection over Union
CNN	Convolutional Neural Network

RGB-D	RGB and Depth
YOLO	You Only Look Once
GA	Genetic Algorithm
mAP	mean Average Precision
RANSAC	RANdom SAmpling Consensus
LiDAR	Light Detection And Ranging
OBB	Oriented Bounding Box
PCA	Principle Component Analysis
TOPS	Tera Operations Per Second
ROS	Robot Operating System
ToF	Time of Flight
UV	Ultra-Violet
PBVS	Position-Based Visual Servo
PID	A controller consisting of proportional, integral and derivative modules
EOL	Endpoint Open-Loop
PR	Precision-Recall
AUC	Area Under Curve
CSRT	Channel and Spatial Reliability Tracking
CIO	Contact-Implicit Optimization
CoM	Center of Mass
DoF	Degree of Freedom
EE	End-Effector
MPC	Model Predictive Control
WBC	Whole Body Controller
KF	Kalman Filter
COCO	A large-scale object detection, segmentation, and captioning dataset.
ALMA	A poly-articulated quadruped robot with a robotic arm

Chapter 1

Introduction

1.1 Aim and Scope

Nowadays research in the robotics community is putting more emphasis on autonomy, maneuverability and dexterity of mobile robots. In both industrial and social domains, robots are expected to be highly interactive with their surroundings and able to deal with various mobile manipulation tasks, assisted by their mobility and perceptive information. The reliability of the perception routines, as well as the robustness of the motion planner, are both keys to the success of autonomous mobile manipulation. Among the manipulations tasks of articulated object door opening is a fundamental topic that is universally investigated.

There are multiple challenges from the problem, such as: motion blur, occlusions, the sensor sensitivity, ambient lighting, etc. After overcoming the challenges, the robot can autonomously identify and manipulate a door after entering a totally unfamiliar human environment.

In the previous work on door manipulation, the door kinematics, the width and the height of the door, the hinge side, and the handle position are often assumed to be known.

Our aim in this work is to more autonomously implement the door opening task in complicated indoor scenes by recursively estimating the kinematics and dynamics of a new door in an unknown environment (Figure 1.1). The framework does not require prior knowledge of the environment, except some basic assumptions, like a door with a revolute joint that is not locked and not too heavy to open (proper damping and stiffness). The approach to it is designing a perception module that automatically detects doors and handles, filters the wrong ones and analyzes their poses and dimensions for mobile manipulation.

1.2 Related Work

Remarkable works have been done on door detection, estimation and tracking.

1. Door Detection
The earlier work tends to simplify the door or handle detection task by turning it into a semi-autonomous problem. For example, Jain et al. [1] locate the position of RoI by illuminating a door handle using a laser pointer. Several works are limited to specific doors. In [2] [3] the door model is extracted using corner based features and edges from images. Then the door handle is identified by training a set of 1500 instances artificially generated for one handle and extracting Haar-like features. The door-specific algorithm cannot be certainly scaled to various doors.

Detecting a novel door is definitely a more challenging task, which is thoroughly investigated as well. Deep learning based methods, like Mask Region-Based Convolutional Neural

Networks (Mask R-CNN) [4] and YOLO[5], are mainstream techniques for addressing object recognition tasks. Llopart et al. [6]'s approach is to first shrink the search of RoI of the handle by YOLO door detector, and then locate the handle by K-means clustering inside the door mask. Arduengo et al. [7] also train a YOLO network with a custom dataset for door and handle detection, which is subsequently verified in door operation experiments with a Toyota HSR wheeled robot.

There are other solutions to novel door detection besides the learning based methods, like vision based and range sensor based.

Quintana et al. [8] present an integrated approach for the detection, localisation and sizing of doors that are either closed or open. The detection is carried out in coloured 3D laser scanned point clouds. The work contributes the door pose and size in world coordinate, though the sensors are on a static base and the handle is not analyzed. Klingbeil et al. [9] use a camera and a 2D laser scanner to identify a small number of 3D key locations, such as the axis of rotation of the door handle, and the location of the end-point of the door-handle. Novel doors can be detected and 31 out of 34 doors are successfully opened. Kim et al. [10] propose a method to detect doors using context-based object recognition. Robotic context such as the robot's viewpoint and texture features like SIFTs and SURFs are utilized.

Rusu et al. [11] propose a laser-based approach that segments the parts of interest using robust geometric estimators and statistical methods applied on geometric and intensity distribution variations in the scan. The approach is validated by performing multiple experiments on the wheeled PR2 platform. The experimental platforms in some of the works above are based on wheeled robot mounted with a robotic arm. If the stable wheeled base is replaced by a floating-based system, like a legged robot, the motion blur might lead to the failure of the approaches.

In this work, we propose a real-time CNN-based door and handle state estimation and parameter identification framework using color, depth and robot odometry on a floating-based system.

2. State Estimation
To estimate the pose of a sensor-head relative to known objects, Sandy et al. [12] fuse measurements from an inertial measurement unit (IMU) and a monocular camera in a probabilistic moving horizon estimator (MHE) framework, and thus generate smooth, consistent, and accurate estimates. Avots et al. [13] use conditional binary Bayes filters for estimating the dynamic state of the environment, such that the environment changes (e.g., doors are opened or closed) can be tackled.

Kalman filters [14] have received much attention with the increasing demands for robotic vision. It is a standard approach for reducing errors in a least squares sense and in using measurements from different sources [10]. Kalman filter makes it possible to implement measurement integration from multiple sensors, which features marked robustness and flexibility [15]. A Kalman filter can be used to estimate interesting parameters and overcome the temporary occlusion problem [16]. We leverage Kalman filter to gather observations of door and handle pose and size over time.

3. Object Tracking
Most object tracking methods are image-based, leveraging image gradients to discover object edges[17]. Morwald et al. [18] propose extensions for real-time robust object tracking with a Monte Carlo particle filter. Kalal et al. [19] propose Tracking-Modeling-Detection (TMD) that closely integrates adaptive tracking with online learning of the object-specific detector. Crivellaro et al. [20] present an algorithm for estimating the 3D pose of a rigid object in real-time under challenging conditions using 2D images. CSRT tracker is a C++ implementation of Discriminative Correlation Filter with Channel and Spatial Reliability tracking algorithm (CSR-DCF) in OpenCV library [21]. Farhodov et al. [22] combine CSRT tracker with fast R-CNN based object detector to tackle problems of overlapping, camera motion blur, changing object appearance, environmental variations, etc.

The door tracking in our work is novel in that we track the 2D door position on the image plane by re-projecting the 3D estimated door pose using RGB-D data.

1.3 Outline

In this work, we present a door state estimation and parameter identification algorithm, synthesizing both RGB-D images and robot odometry. The main contributions include:

• We propose a reliable framework of door state estimation and parameter identification for mobile manipulation.

The algorithm can be used in various complicated indoor scenes with multiple doors, occlusions, or change of lighting. It overcomes the problem of low estimation accuracy in mobile perception, where the camera mounted on the robotic arm moves constantly and markedly and leads to motion blur.

• We present a novel door tracking strategy based on both RGB-D images and robot odometry.

The target door and its handle can be tracked on both the image plane and the world frame by the combined odometries and the re-projection validation. The re-projection validation efficiently rules out the non-target detections from the object detector, which are significant interference in complicated scenes. The 3D tracking results of the door's pose in world-frame coordinates facilitates door-manipulation tasks, like visual servoing for handle grasping.

• We collect a data on various doors, composed of the RGB-D data, the robot odometry and the ground truth annotation.

The door data collected by time-of-flight cameras and a legged robot is rare in the state-of-the-art research. The novelty of it lies in the robot odometry information, the view of doors from manifold angles and distances, the mixture of blurry and focused RGB images, as well as the existence of interference doors.

In chapter 2, we talk about the formulation of the problem. In chapter 3, experimental results are analyzed to verify the feasibility of the algorithm. In chapter 4, the conclusions are drawn and the potential future work is proposed.

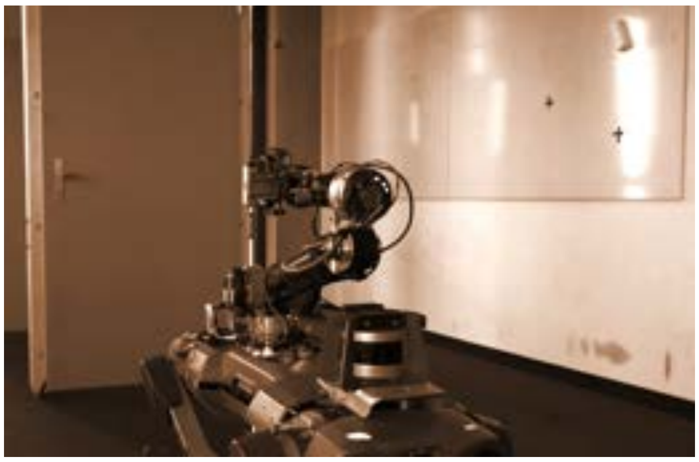


Figure 1.1: The work focus on door and handle perception in mobile locomotion before the robot makes contact with the target door.

Chapter 2

Problem Formulation

In this chapter, we formulate the state estimation and object tracking problem.

2.1 Overview

In this section, we introduce briefly the framework of our work.

The schematic diagram of our door estimation and tracking framework is illustrated in Figure 2.1,

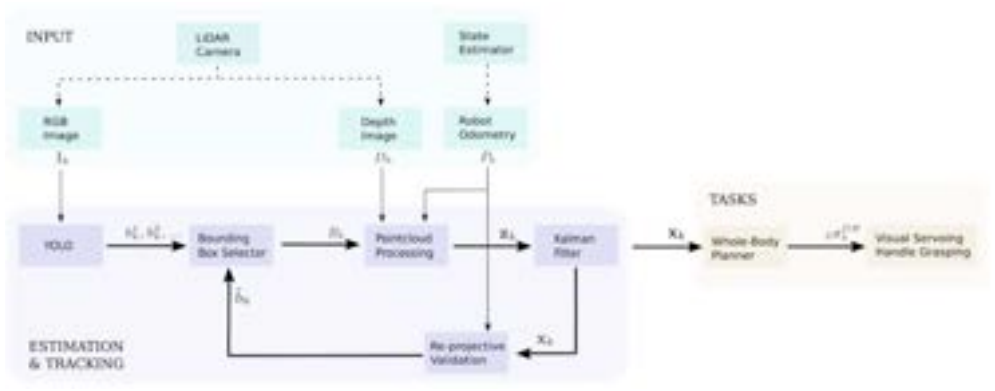


Figure 2.1: Schematic diagram of our 3D door state estimation and parameter identification algorithm (In the following chapters, $\{\mathcal{W}\}$, $\{\mathcal{C}\}$, $\{\mathcal{D}\}$, $\{\mathcal{H}\}$ and $\{\mathcal{I}\}$ indicate the coordinate of World, Camera, Door, Handle and Image, respectively. $b_k^1, b_k^2, ...$: candidate bounding boxes from the YOLO inference of the current frame k; B_k: the selected bounding box to analyze; $\mathbf{z}_k$: measurement; $\mathbf{x}_k$: state posteriori; $\tilde{b}_k$: re-projective validation bounding box; ${}_D\mathbf{r}_k^{DH}$: the posteriori handle position in $\{\mathcal{D}\}$)

We leverage the visual information, as well as the robot odometry, to robustly estimate and track the door and handle instead of the pure vision approach.

In the input section, aligned and synchronized RGB and depth images are passed into the estimator in each frame, as well as the camera intrinsics. Another source is the robot odometry from the state estimator module of the robot.

A classification neural network based on YOLOv5 is trained for door detection (2.2). Several candidate detections are obtained from the object detector. There might be false positive or other inferential doors among the candidates, and thus we need a criterion for filtering. Our approach is to select the target door from the candidates by comparing with a validation bounding box (2.3). The validation bounding box is the re-projection of the synthesis of past estimation results in $\{\mathcal{W}\}$ onto $\{\mathcal{I}\}$. In the module of bounding box selector, we make one-by-one comparisons of the validation box with the candidate boxes from YOLO. The bounding box is selected as a desired one, if it coincides with the validation box.

The target pointcloud is processed subsequently (2.4), aiming to down-sample and remove the outlier points.

After being converted from $\{\mathcal{C}\}$ to $\{\mathcal{W}\}$ using the robot odometry, the observations of door information are gathered in Kalman filter (2.5).

$$\mathbf{x}_k = \begin{bmatrix} {}_{\mathcal{W}}\mathbf{r}_k^{\mathcal{W}\mathcal{D}} \\ {}_{\mathcal{W}}\mathbf{n}_k^{\mathcal{W}} \\ {}_{\mathcal{D}}h_k \\ {}_{\mathcal{D}}w_k \\ {}_{\mathcal{D}}\mathbf{r}_k^{\mathcal{D}\mathcal{H}} \end{bmatrix} \tag{2.1}$$

As is formulated in 2.1, the posteriori state vector $\mathbf{x}_k$ of the Kalman filter is composed of the CoM position of the door in $\{\mathcal{W}\}$, ${}_{\mathcal{W}}\mathbf{r}_k^{\mathcal{W}\mathcal{D}}$; the normal of the door in $\{\mathcal{W}\}$, ${}_{\mathcal{W}}\mathbf{n}_k^{\mathcal{W}}$; the door height ${}_{\mathcal{D}}h_k$; the door width ${}_{\mathcal{D}}w_k$; and the CoM position of the handle in $\{\mathcal{D}\}$, ${}_{\mathcal{D}}\mathbf{r}_k^{\mathcal{D}\mathcal{H}}$.

The measurement $\mathbf{z}_k$ has exactly the same form as the state vector. Besides, the hinge side can be deduced from the y coordinate of ${}_{\mathcal{D}}\mathbf{r}_k^{\mathcal{D}\mathcal{H}}$, since a hinge is always to the opposite side of a handle on a door. The angle rotated can be calculated from ${}_{\mathcal{W}}\mathbf{n}_k^{\mathcal{W}}$.

$\mathbf{x}_k$ from the Kalman filter is projected back to $\{\mathcal{I}\}$ using the camera extrinsics and intrinsics for validation purposes (2.6).

In the task section, the posterior poses of the target door and handle at every iteration, the handle position in the camera coordinate, is passed to the whole-body planner of the robot for visual servoing handle grasping.

2.2 YOLO Object Detection

Our door and handle object detector are customized based on YOLOv5 pre-trained models. YOLOv5 is a family of object detection architectures and models pretrained on the COCO dataset.

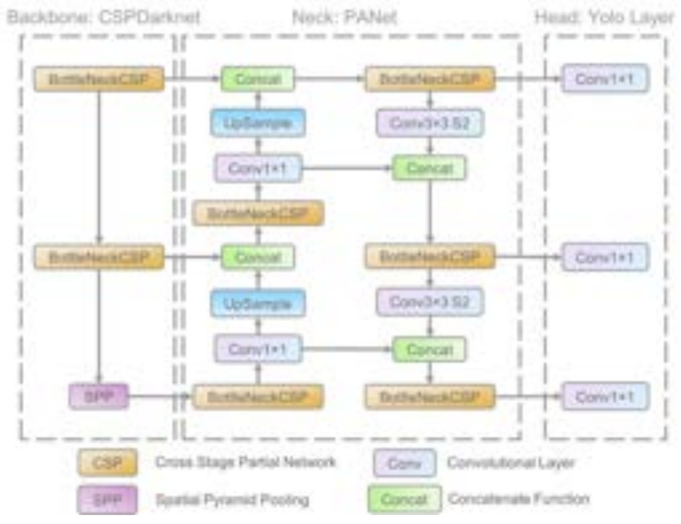


Figure 2.2: Schematic diagram of YOLOv5 one-stage structure ([23]).

2.2.1 Network Structure

The YOLOv5 one-stage structure is composed of three parts (Figure 2.2) besides the input section, backbone (CSPDarknet), neck (PANet) and head (YOLO layer).

The data are first input to CSPDarknet for feature extraction, and then fed to PANet for feature fusion. Eventually, YOLO layer outputs detection results (class, score, location, size).

There are some innovative features in the input stage, like the auto learning bounding box anchors (automatically generate an initial anchor) and the automatic image zooming (minimize the area of zero-padding). In the head part, non-maximum suppression is applied for bounding box filtration, which is an indispensable preliminary step before the bounding box selector module.

In practice, YOLOv5m is adopted considering the trade-off between inference speed and precision (Figure 2.3).

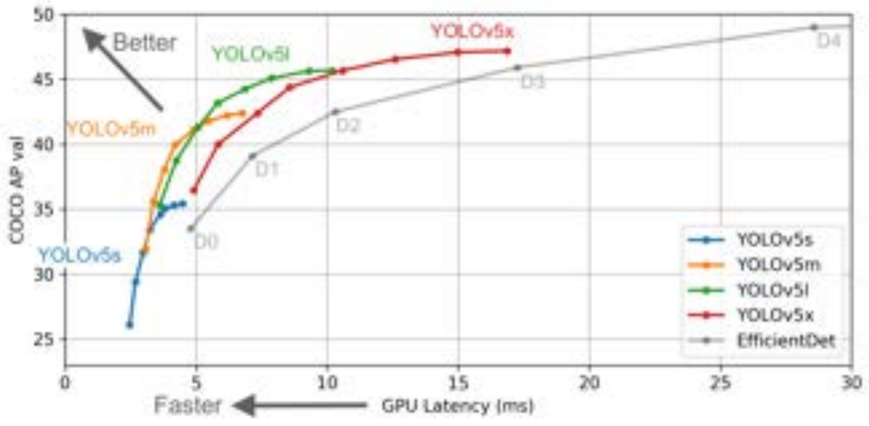


Figure 2.3: The average precision in validation set in terms of GPU speed of several YOLOv5 models tested on COCO dataset. (The figure is provided by the YOLOv5 team.)

2.2.2 Training

We are able to set some hyperparamters in YOLOv5 regarding image augmentation. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned. Hyperparameter evolution is a method of Hyperparameter Optimization using a Genetic Algorithm (GA) for optimization. Hyperparameter optimization is the problem of choosing a set of optimal hyperparameters for a learning algorithm.

The first step of hyperparameter evolution is to initialize the hyperparameters. Afterwards, the fitness that we seek to maximize is defined in the form of a weighted combination of Precision, Recall, mAP@0.5 and mAP@[.5:.95]. The primary step is the evolution. In YOLOv5, mutation is used in genetic operation, with a 90% probability and a 0.04 variance to create new offspring based on a combination of the best parents from all previous generations. At last, we can visualize the results of hyperparameter evolution as follows,

On the other hand, we leverage transfer learning [24] in YOLOv5 to customize a door detector. Transfer learning is a useful way to quickly retrain a model on new data without having to retrain the entire network. Instead, part of the initial weights are frozen in place, and the rest of the weights are used to compute loss and are updated by the optimizer. This requires less resources than normal training and allows for faster training times, though it may also results in reductions to final trained accuracy. In practice, we choose to freeze only the backbone layers (first 10 layers out of the total 24). The way of training is a trade-off, since it greatly saves the GPU utilization and yields model of high inference accuracy.

The dataset [7] we use contains classes of both door and handle. Considering its a medium-sized dataset, we adopt methods of early stop and image augmentation to avoid overfitting.

In terms of the image size, large amount of small objects (handles) in the dataset it can benefit from training at higher resolutions. On the other side, our framework is applied online and thus the inference time cannot be too long. As a consequence, we prefer a medium image size of the model.

In addition, the batch size is the largest that the hardware allows.

2.2.3 Image Augmentation

The dataset, DoorDetect[7], composed of 1,213 images that have been annotated with object bounding boxes. The images come from diverse and complex scenes with often several objects (Figure 2.4). There are four object classes, door, handle, cabinet door and refrigerator door. Each object in the images is labeled with the object class, as well as the center position, the width, and the height of the bounding box.

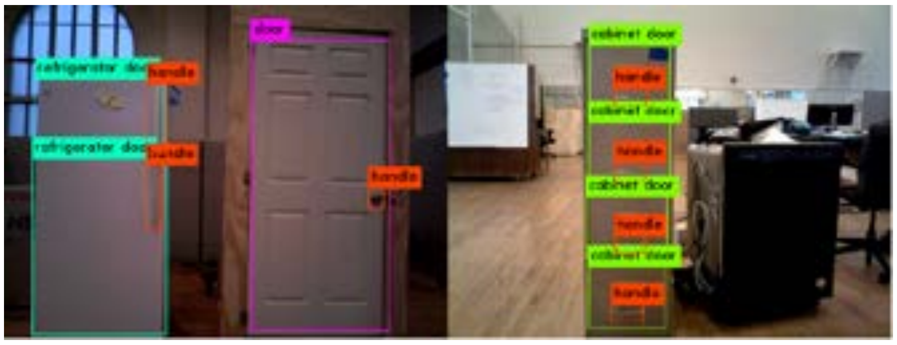


Figure 2.4: Two images in the dataset, annotated with bounding boxes and classes [7].

The original dataset, composed of 1,213 pictures, is not large enough, and thus the neural networks sometimes overfit the training data even if we stop the training process early. Overfitting refers to the phenomenon when a network learns a function with very high variance such as to perfectly model the training data. Image augmentation is a solution to overfitting in neural network training owing to limited training data.

Data augmentation is a ubiquitously used technique to increase both the size and the diversity of the labeled dataset. Besides alleviating the overfitting of deep learning models, image augmentation helps adapt the model into general, complicated, and even extreme scenes, such that it can discover smaller or incomplete objects in various scenes and ambient lighting conditions.

The data augmentation technologies are based on image manipulations, such as flipping, rotation, translation, scale, shear, color space change, and so on. There are some other more advanced and less common ways, like mixing images, kernel filters, random erasing, etc. Aggressive tuning in some of the metrics is especially useful, for example, HSV color space changing and scaling, for the reason that, in our case, the robot might need to approach the door from far away under various lighting conditions. Nevertheless, we should avoid overly exploiting aggressive augmentation, while it possibly influences the performance.

2.3 Bounding Box Selector

We obtain multiple candidate detections from YOLO. In order to correctly choose the target one, the candidates are one-by-one compared with the validation bounding box, until the desired one is found.

A pseudocode of the bounding box selector is shown below,

Algorithm 1 Bounding box selector for the target door (conservative strategy)

```
Require: $k > 0$
  $u \leftarrow 0$
  $L_{thres} \leftarrow 0.5$
  $play \leftarrow False$
  if $U \neq 0$ then
    for $u \leq U$ do
      if $b_k^u$ is not handle then
        function CalculateIOU($b_k^u$, $\bar{b}_k$)
          return $L_{iou}$
        end function
        if $L_{iou} \geq L_{thres}$ then
          $play \leftarrow True$
          $B_k \leftarrow b_k^u$
          break
        end if
        $u \leftarrow u + 1$
      end if
    end for
  end if
  if $play = True$ then
    function RunEstimator($B_k$, $P_k$, $D_k$, $C_k$)
      return $\mathbf{x}_k$
    end function
  else
    $k \leftarrow k + 1$
    import new $P_k$, $D_k$, $I_k$
  end if
```

U indicates the number of YOLO detections in the frame, and L_{thres} the threshold of Intersection over Union (IoU), beyond which the detection is regarded as a valid one.

As is detailed in the Algorithm 1, we calculate the IoU between the validation box and each of the YOLO bounding box. Once the detection is valid, we exit the bounding box selector module and enter the pointcloud processing stage. If there is none of the detections coinciding with the validation box, we enter next frame and import new images and transformation from the robot.

As a matter of fact, this is only a conservative way of the strategies that we can adopt in the selection of bounding box. We will further explain several other strategies in 3.7.

2.4 Pointcloud Processing

After locating the target bounding box on the RGB image, we project the box to the corresponding depth image and thus generate the pointcloud of the target door. The depth image has to be perfectly synchronized and aligned with the RGB image to guarantee the right target is cropped. After the target depth image is obtained, it is converted to pointcloud. The target pointcloud has to be processed to remove the outliers and noise that do not belong to the door. The processing procedure includes voxel grid downsampling, RANSAC, outlier removal and 3D oriented bounding box, respectively, as is illustrated in Figure 2.5.

We leverage the Point Cloud Library (PCL) to process the target pointcloud. The Point Cloud Library is an open-source library of algorithms for point cloud processing tasks[25].

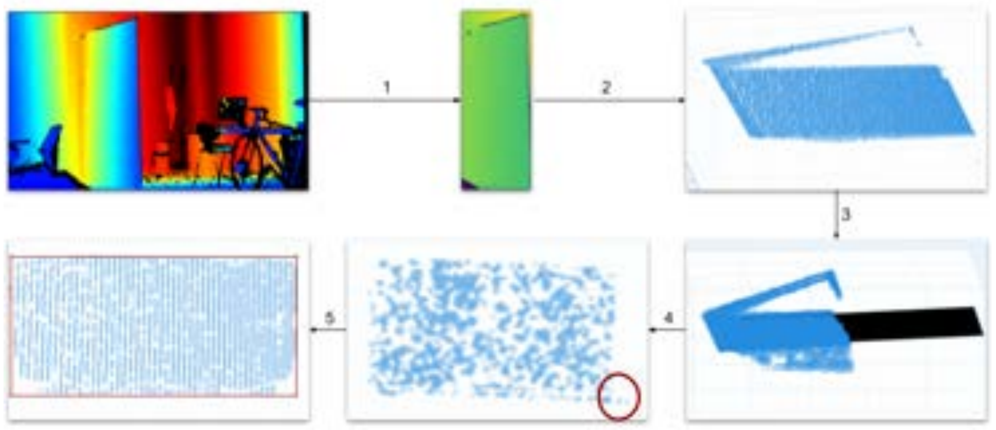


Figure 2.5: The pipeline of pointcloud processing in the estimator. (1: crop the original depth image; 2: convert depth to pointcloud and downsample it; 3: fit a plane using RANSAC; 4: remove outlier; 5: find a oriented bounding box)

2.4.1 Voxel Grid Downsampling

Normally, we obtain an extremely dense target pointcloud from the depth camera. The number of points could reach hundreds of thousand, even after cropping the target out of the original pointcloud. As a result, it would be reasonable to downsampling the pointcloud before further processing, so as to improve the efficiency of the algorithm.

We use voxel grid downsampling to generate a uniform and sparse pointcloud. This technique has been traditionally used in the area of computer graphics to subdivide the input space and reduce the number of points [26, 27, 28]. The primary steps of downsampling are,

- Points are bucketed into voxels.
- Each occupied voxel generates exact one point by averaging all points inside.

The hyperparameter in the downsampling is the voxel grid size (leaf size), which interprets the 3 dimensions of the rectangular voxel.

2.4.2 RANSAC

With the downsampled pointcloud, we extract the target plane of door using RANdom SAmpling Consensus (RANSAC) [29]. The RANSAC algorithm is a learning technique to estimate parameters of a model by random sampling of observed data. RANSAC is capable of interpreting and smoothing data containing a significant percentage of gross errors, and is ubiquitously applied in automated image analysis[29].

In RANSAC plane fitting, we run the algorithm iteratively, and in each iteration, a certain amount of points are randomly selected. The iteration does not stop until a desired model is found within a distance deviation threshold, or the number of the maximum iteration is reached. These two, the distance threshold and the maximum iteration number, are hyperparameters in PCL RANSAC module. The number of iterations k required to obtain a desired target plane is as follows,

$$1 - p = (1 - w^n)^k \tag{2.2}$$

$$k = \frac{\log(1-p)}{\log(1-w^n)} \tag{2.3}$$

with p the desired probability that the RANSAC algorithm provides a useful result after running, w the probability of choosing an inlier each time a single point is selected, and n the number of points chosen in each iteration.

2.4.3 Outlier Removal

RANSAC aims to remove the outliers along the direction perpendicular to the door plane, while the remained outliers inside the plane could be resolved by outlier removal technologies.

1. Statistical Outlier Removal
If we know that the distribution of values in the sample is Gaussian or Gaussian-like, we can use the standard deviation of the sample as a cut-off for identifying outliers. The approach removes points that are further away from their neighbors compared to the average for the point cloud.
The hyperparameters are,

- Number of neighbors: the number of neighbor points taken into account in order to calculate the average distance for a given point.
- Standard deviation ratio: the threshold level based on the standard deviation of the average distances across the point cloud. The lower this number the more aggressive the filter will be.

2. Radius Outlier Removal The approach removes points that have not enough neighbors in a given sphere around them, with hyperparameters,

- Number of points: the minimum amount of points that the sphere should contain.
- Radius: the radius of the sphere that will be used for counting the neighbors.

2.4.4 3D Oriented Bounding Box

In order to find the CoM and dimensions of the target door, we apply a bounding rectangular object to the the downsampled and outlier-removed pointcloud. The position and dimensions of the bounding box are approximately regarded as those of the pointcloud.

1. Axis-aligned Bounding Box
Axis-aligned Bounding Box (AABB) is a rectangular parallelepiped whose faces are each perpendicular to one of the basis vectors. The problem could be easily solved by finding the min-max values along the given axis.

2. Oriented Bounding Box
An oriented bounding box (OBB) is a bounding parallelepiped whose faces and edges are not necessarily parallel to the basis vectors of the frame in which they're defined. AABB is a special case of OBB, in which the orientation of the bounding box is no more fixed.

We leverage eigen-vectors of the pointcloud to align the data to our original cartesian basis vectors, which actually simplifies the problem and makes it an AABB problem. Then the max-min method is once more employed to find the oriented bounding box.

Eigen-vectors are the vectors which do not changes direction upon transformation. Principle Component Analysis (PCA) is often used to find eigen vectors. PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (the first principal component), the second greatest variance on the second coordinate, and so on.

2.5 Kalman Filter

We use a Kalman Filter to recursively estimate the state and parameters of the target door and handle.

2.5.1 Modeling

As is introduced in 2.1, the state vector $\mathbf{x}_k$ of the Kalman filter is composed of the CoM position of the door in $\{\mathcal{W}\}$, the normal of the door in $\{\mathcal{W}\}$, the door height, the door width, and the CoM position of the handle in $\{\mathcal{D}\}$. The state space model and the observation model of the problem are, respectively,

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{2.4}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.5}$$

In our model, considering the target door is static and the observation space is identical to the state space, the control input vector $\mathbf{u}_{k-1}$, the transition matrix $\mathbf{F}_{k-1}$ and the observation matrix $\mathbf{H}_k$ are in a simple form as below,

$$\mathbf{F}_{k-1} = \mathbb{I} \tag{2.6}$$

$$\mathbf{H}_k = \mathbb{I} \tag{2.7}$$

$$\mathbf{u}_{k-1} = \mathbf{0} \tag{2.8}$$

The temporally uncorrelated noise $\mathbf{w}_{k-1}$ and the measurement noise $\mathbf{v}_k$ follow normal distributions with expected value $\mathbf{0}$, shown as below,

$$\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \tag{2.9}$$

$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \tag{2.10}$$

where $\mathbf{Q}_{k-1}$ the covariance of the process noise, and $\mathbf{R}_k$ the covariance of the observation noise.

2.5.2 Filtering

The initialization of the Kalman filtering is,

$$\mathbf{x}_0 = \mathbf{z}_0 \tag{2.11}$$

$$\mathbf{P}_0 = \sigma^2 \cdot \mathbf{I} \tag{2.12}$$

where $\mathbf{P}_k$ the state covariance matrix.

The predict phase uses the state estimate from the previous timestep to produce an estimate of the state at the current timestep. The prior state estimate and its covariance are, respectively,

$$\mathbf{x}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{2.13}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1} \tag{2.14}$$

The measurement update improves estimate based on the current observation. The measurement residual (the innovation),

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k\mathbf{x}_{k|k-1} \tag{2.15}$$

The innovation covariance,

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k \tag{2.16}$$

The Kalman gain,

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{S}_k^{-1} \tag{2.17}$$

The posteriori state and it covariance,

$$\mathbf{x}_k = \mathbf{x}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k \tag{2.18}$$

$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\mathrm{T}} \tag{2.19}$$

2.5.3 Tuning

Intuitively speaking, the dynamic measurement model should lay more confidence on the measurements when the target surface is closer to the camera. In other words, the diagonal terms in the measurement uncertainty matrix should be inversely proportional to the distance between the camera and the target door. This intuition is further proved by Nguyen et al. [30], who build both axis and lateral noise distributions as a function of distance and angle from sensor to observed surface.

2.6 Re-projective Validation

We implement image-level object tracking by re-projecting the estimated door and handle in the world coordinate $\{\mathcal{W}\}$ to the image coordinate $\{\mathcal{I}\}$, followed by filtering the YOLO candidate bounding boxes by the validation box.

The posteriori door and handle poses in world frame, and door dimensions are obtained after Kalman filtering. Theoretically, if we project the information back to the image plane, it would coincide with the target door in the RGB image. Based on YOLO object detection as well as this fact, we implement our object tracking module. The procedure of the re-projective validation is as follows,

1. Discover the four corner points of the door ${}_{\mathcal{W}}\mathbf{r}_{c,i}^{\mathcal{W}\mathcal{D}} (i = 1, 2, 3, 4)$ and the center point of the handle ${}_{\mathcal{W}}\mathbf{r}^{\mathcal{W}\mathcal{H}}$.

2. Transform the corner points to the camera coordinates $\{\mathcal{C}\}$ using the extrinsics parameters from the robot odometry.
The robot odometry provides rotation matrix $\mathbf{R}_k^{WC}$ and translation vector $\mathbf{t}_k^{WC}$ from $\{\mathcal{W}\}$ to $\{\mathcal{C}\}$. It should be noted that the robot odometry should be from the current frame, otherwise the validation box projected back would not coincide with the door if the camera has moved.

$$\mathbf{T}_k^{WC} = \begin{bmatrix} \mathbf{R}_k^{WC} & \mathbf{t}_k^{WC} \\ 0 & 1 \end{bmatrix} \tag{2.20}$$

$$\begin{bmatrix} {}_{\mathcal{C}}\mathbf{r}_{c,i}^{\mathcal{C}\mathcal{D}} \\ 1 \end{bmatrix} = \mathbf{T}_k^{WC} \begin{bmatrix} {}_{\mathcal{W}}\mathbf{r}_{c,i}^{\mathcal{W}\mathcal{D}} \\ 1 \end{bmatrix} \tag{2.21}$$

$$\begin{bmatrix} {}_{\mathcal{C}}\mathbf{r}^{\mathcal{C}\mathcal{H}} \\ 1 \end{bmatrix} = \mathbf{T}_k^{WC} \begin{bmatrix} {}_{\mathcal{W}}\mathbf{r}^{\mathcal{W}\mathcal{H}} \\ 1 \end{bmatrix} \tag{2.22}$$

3. Map the corner points in $\{\mathcal{C}\}$ into the image coordinate $\{\mathcal{I}\}$ using the camera intrinsics $\mathbf{C}_k$,

$$\mathbf{C}_k = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.23}$$

$$w \begin{bmatrix} {}_{\mathcal{I}}\mathbf{r}_{c,i}^{\mathcal{C}\mathcal{H}} \\ 1 \end{bmatrix} = \mathbf{C}_k {}_{\mathcal{C}}\mathbf{r}_{c,i}^{\mathcal{C}\mathcal{D}} \tag{2.24}$$

$$w \begin{bmatrix} {}_{\mathcal{I}}\mathbf{r}^{\mathcal{C}\mathcal{H}} \\ 1 \end{bmatrix} = \mathbf{C}_k {}_{\mathcal{C}}\mathbf{r}^{\mathcal{C}\mathcal{H}} \tag{2.25}$$

where the scale factor w is 1, and the skew coefficient s is zero since the images are rectified.

4. Approximate the target door with a rectangle.
When the corner points of the door are projected in $\{\mathcal{I}\}$, it would commonly form an irregular quadrilateral. It should be approximated with a rectangle, as any bounding box is like.

It should be especially noted that two points, which are symmetric to each other about the origin of the camera coordinate and long one 3D ray, will be projected to the same position on the image. Therefore, it is sometimes necessary to exclude the redundant case. For example, when the robot turns 180 degrees back away from the target door (Figure 3.16), there will be still a projected bounding box if the redundant case is not considered.

Chapter 3

Experimental Results

In this chapter, we validate the estimation and tracking algorithms with a series of experiments. Our proposed methods are applied to different situations, where diverse doors are scanned in different lighting conditions, from different angles and distances.

In this chapter, we first introduce the system (3.1), the assumptions (3.2), the training results of object detector (3.3).

We carry out the following experiments afterwards:

1. Fixed camera setup without re-projective validation (3.4).
2. Offline experiments with data from the robot (3.5).
3. Open-loop experiments online running on the robot (3.6).

Then we introduce other following analysis, including the object tracking results (3.7), the elapsed time (3.8) and the error analysis (3.9).

3.1 System

The system of data collection and processing is composed of the RGB-D camera, ANYmal with a robotic arm, a workstation for neural network deployment, and the middleware.

1. The RGB-D camera
The RGB-D camera used to collect data is Intel RealSense L515 [31]. The Intel RealSense LiDAR Camera L515 is popular in indoor applications that require depth data at high resolution and high accuracy.

2. The robot
The RGB-D camera is mounted on the forearm of ALMA C, an autonomous quadrupedal robot, ANYmal [32] with a 6-DoF robotic manipulator, DynaArm.

3. The workstation
The algorithm is trained and deployed on Nvidia Jetson AGX Xavier[33].

4. The middleware, the software and the programming language
In this work, a Python class structure is deployed along with the use of libraries and tools provided by the Robot Operating System (ROS).

3.2 Assumptions

Although the algorithm of door pose and configuration estimation aims to be generalized to most indoor scenes, there are several basic assumptions in terms of the ambient lighting, the nature of the door, etc.

The assumptions of the algorithm are

1. Ambient lighting
Time-of-Flight (ToF) sensors are sensitive to ambient light, as they can't tell if photons were emitted by the sensor or have arrived from other light sources. ToF sensors tend to work better indoors, where there is little or no ambient light. As a consequence, artificial lighting is the primary source of light in most of the data we collected, with at most slight ambient light casting from windows.

2. Indoor scenes
As is explained above, the datasets we collected are mostly indoors. The indoor scenes can be complicated, where passers-by, non-target doors and handles, cabinet doors randomly appear. It increases the difficulties of estimation and tracking, and demonstrates the robustness of the algorithm.

3. The nature of the door
We presume that the target door is a single door with revolute joint and non-glass surface material. Prismatic joints, double doors (side-by-side doors), glass doors are thus not considered. Generally speaking, glass is transparent to visible light and infrared light but opaque to UV light. Owing to the physical characteristics of the material, the depth information will fail in the case of glass doors.

4. The position of the door hinge
Common sense tells us that the handle lies on the non-hinge side of the door. The detection of hinge side could be thus regarded as a problem of handle detection.

5. The initialization
The initial door and handle detections matter significantly in the algorithm. We have to guarantee that the first ones are correct, otherwise a wrong one would be tracked and it can be hardly corrected. Our approach is to manually select the target door of interest from an user-interactive window (Figure 3.1). The camera view is displayed once the algorithm is initialized, where we can define the RoI containing the target door.

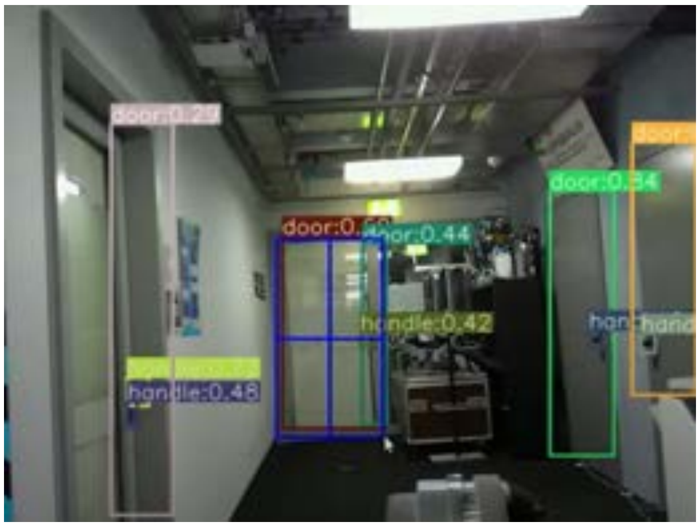


Figure 3.1: RoI manual selection (blue 2×2 blocks)

3.3 Object Detector

Some results related to YOLO network training are detailed in this section. The original Pytorch model is converted to a TensorRT model, which to a certain degree improves the inference efficiency.

3.3.1 Image Augmentation and Hyperparameter Evolution

Several deep learning based object detectors of different input image sizes are customized with a pre-trained YOLOv5m model. Image augmentation is deployed to expand the volume of the dataset (3.2).

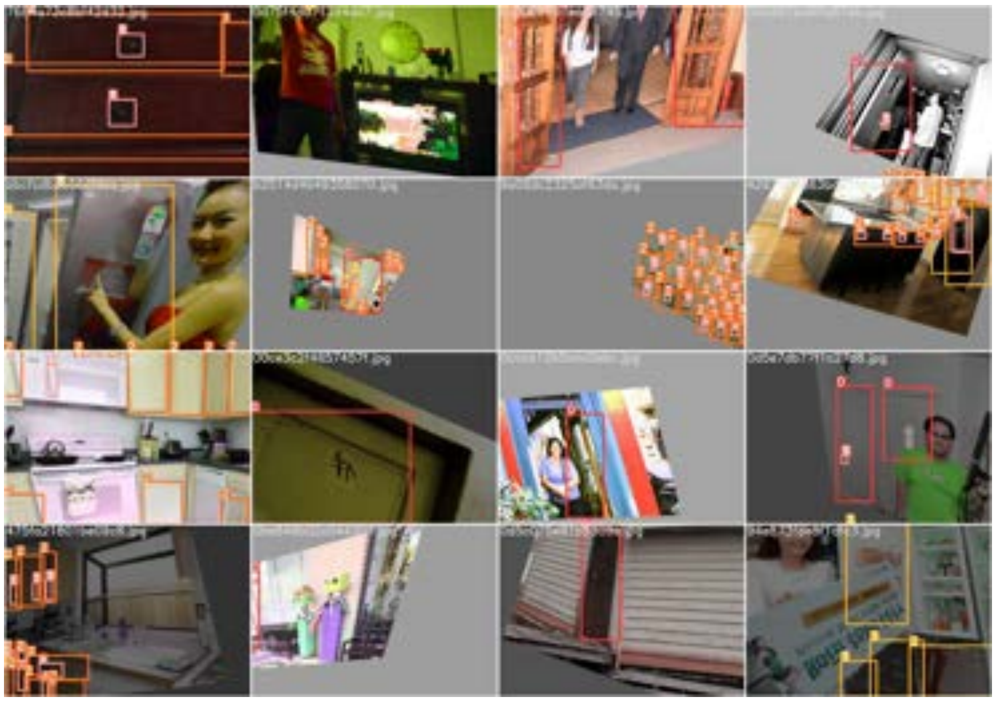


Figure 3.2: Several augmented images with bounding boxes after image augmentation, like rotation, zooming, flipping, ...

The intensity of image augmentation in terms of individual transformation technique is defined in the hyperparameter list before training. Hypermarater evolution is implemented before training the neural network, and some of the results are illustrated in 3.3. In this case, the fitness is defined as follows,

$$f = 0.1 \cdot P_{0.5} + 0.9 \cdot P_{[0.5,0.95]} \tag{3.1}$$

where f, $P_{0.5}$, $P_{[.5,.95]}$ are short for mAP@0.5, and mAP@[0.5:0.95], separately.

From the evolution results, the overall fitness of the neural network is raised from 0.25 to 0.32, which indicates a significant increase of performance.

3.3.2 Evaluation Metrics

The performance of deep learning models can be evaluated by some measures:

- True Positives (TP): the correctly predicted positive values.
- True Negatives (TN): the correctly predicted negative values.
- False Positives (FP) and False Negatives (FN): these values occur when the actual class contradicts with the predicted class.
- Precision (P): the ratio of correctly predicted positive observations to the total predicted positive observations.

$$P = TP/(TP + FP) \tag{3.2}$$

- Recall (R): the ratio of correctly predicted positive observations to the all observations in actual class

$$R = TP/(TP + FN) \tag{3.3}$$

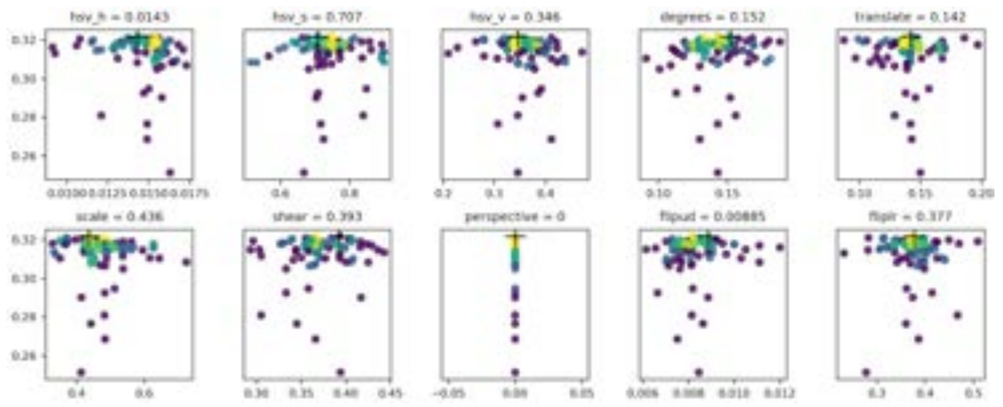


Figure 3.3: A part of hyperparameter evolution results after hundreds of epochs, with y-axis denoting the fitness, x-axis the transformation ratio of each technique (The ratio indicates a lower or upper bound of the transformation, for example, $scale = 0.436$ means the orginal image is scaled with a ratio randomly picked from the interval between $(1 - 0.416)$ and $(1 + 0.416)$). The larger the value is, the more aggressive the transformation is. The hyperparameters are evolved from purple dots to green, and then yellow dots. Final converged values are denoted by black cross signs. (hsv_h: hue; hsv_s: saturation; hsv_v: value; flipud: flip up and down; fliplr: flip left and right.)

- Accuracy (A): the ratio of correctly predicted observation to the total observations.

$$A = (TP + TN)/(TP + FP + FN + TN) \tag{3.4}$$

- F1 Score ($F1$): the weighted average of Precision and Recall.

$$F1 = 2 \cdot R \cdot P/(R + P) \tag{3.5}$$

- Intersection over Union (IoU): A metric that interprets how much the predicted bounding box overlaps with the ground truth, which is calculated from the area of overlapping S_o and that of union S_u.

$$IoU = S_o/S_u \tag{3.6}$$

- Average Precision (AP): the area under the precision-recall curve.
- mAP@0.5: mean AP for each class over IoU threshold of 0.5.
- mAP@[.5 : .95]: average mAP over different IoU thresholds, from 0.5 to 0.95, step 0.05 (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95)
- Confidence: the probability that a box is correctly classified.

3.3.3 Score and Loss

To analyze the score and loss curves of the models, we take an instance of a trained neural network with input image size 640*640, batch size 24, and 80 epochs of iteration.

The plots of validation loss decrease to points of stability and has a small gap with the training loss (Figure 3.4), except that the plot of validation objectness loss decreases to a point and begins increasing again a bit. The overfitting behavior demonstrates a sign of over-complicated model in terms of the objectness. The minor overfitting of the objectness indicates that it is better to end the iteration in advance, while the box and classification plots do not yet completely converge to a stable point. As a consequence, a trade-off has to be made or other hyperparameters should be trialed.

The name of the confusion matrix stems from the fact that it makes it easy to see whether the system is confusing two classes. Each row of the matrix represents the instances in a

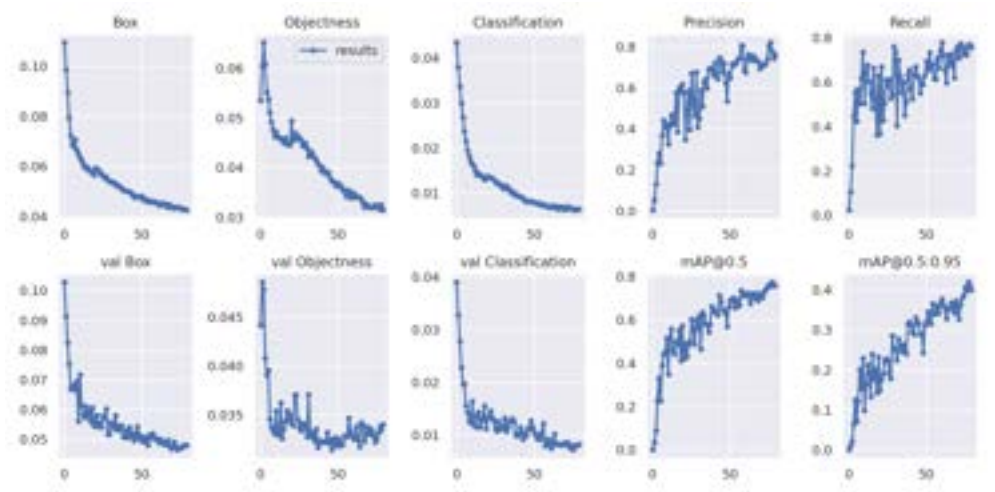


Figure 3.4: The box, objectness and classification loss curves (left 3 columns) in the training set(top row) and the validation set (val, bottom row), as well as the score metrics (right 2 columns). Objectness loss: a loss function related to the prediction probability of the ground truth objectness. Objectness shows the probability that an object exists in an image. The larger the probability is, the smaller the loss is. Class loss: aloss function related to the prediction probability of the ground truth class. Box loss: a loss function related to the prediction probability of the ground truth bounding box.

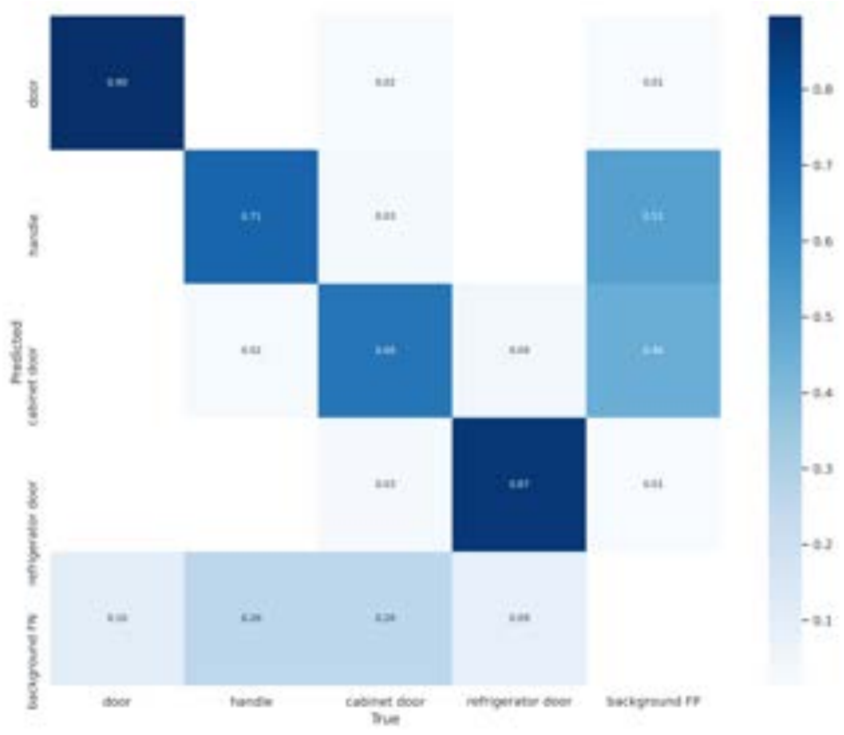


Figure 3.5: The confusion matrix for the four classes (door, handle, cabinet door, refrigerator door).

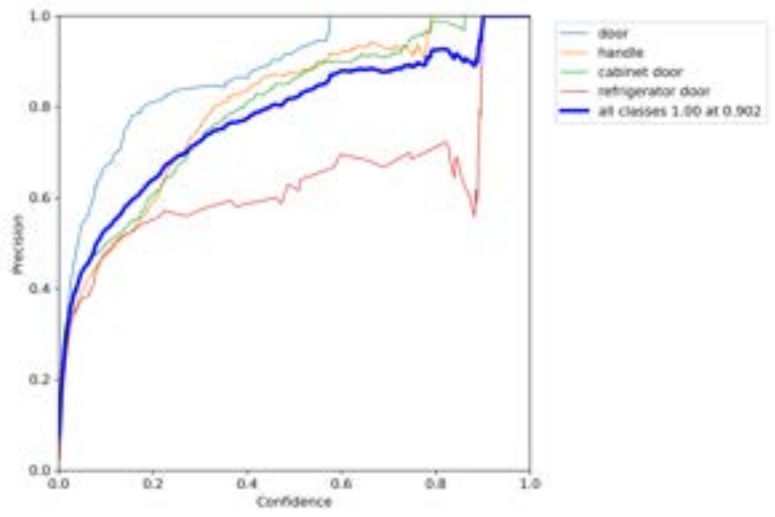


Figure 3.6: The precision curve in terms of the confidence score threshold

predicted class while each column represents the instances in an actual class, as is shown in 3.5. The matrix features every detail of the evaluation of the training results, for instance, the recall and precision of each class. Furthermore, other scores like the PR curve and F1 score can be easily deduced from the matrix.

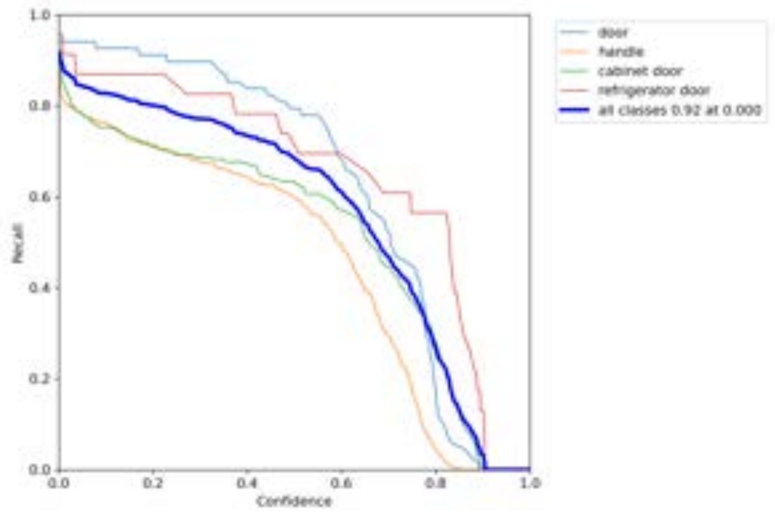


Figure 3.7: The recall curve in terms of the confidence score threshold

The confidence score is the probability of a positive prediction, which lies within $0-1$. The confidence score threshold is a value, beyond which the prediction is regarded as positive, and otherwise as negative. Generally speaking, the higher the threshold is, the more frequently false negative appears, and thus the smaller recall is (Figure 3.7). Similarly, less frequent false positive gives birth to larger precision as the threshold increases, as is illustrated in Figure 3.6.

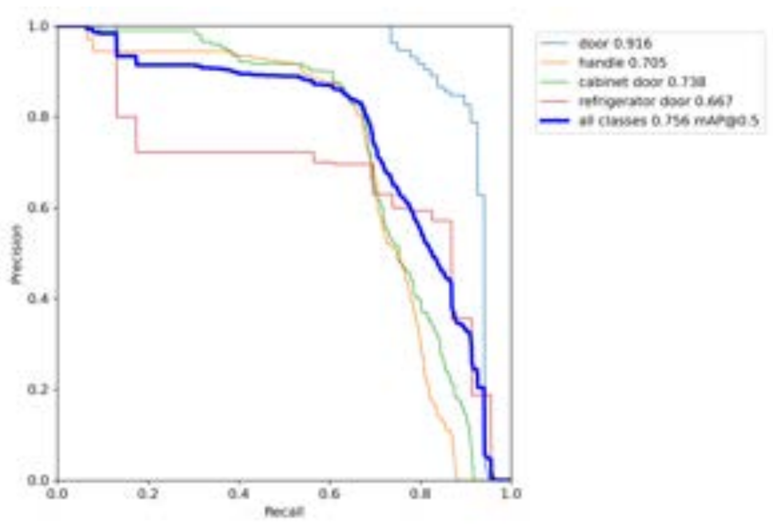


Figure 3.8: The precision - recall curve

Apparently, precision is inversely proportional to recall (Figure 3.8). It is desired that the algorithm should have both high precision, and high recall. However, most machine learning algorithms often involve a trade-off between the two. The area under curve (AUC) of the precision-recall (PR) curve indicates the comprehensive performance of the model. In the figure, the door classification corresponding to the blue line has better performance than the classifier of refrigerator door owing to the larger AUC.

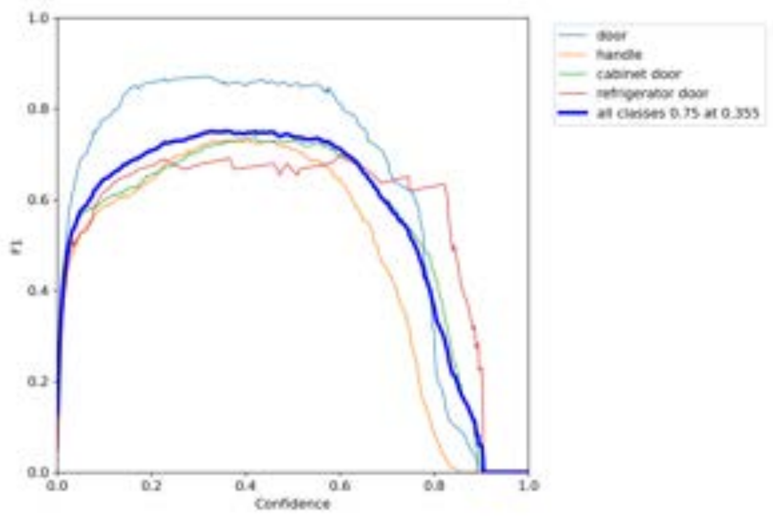
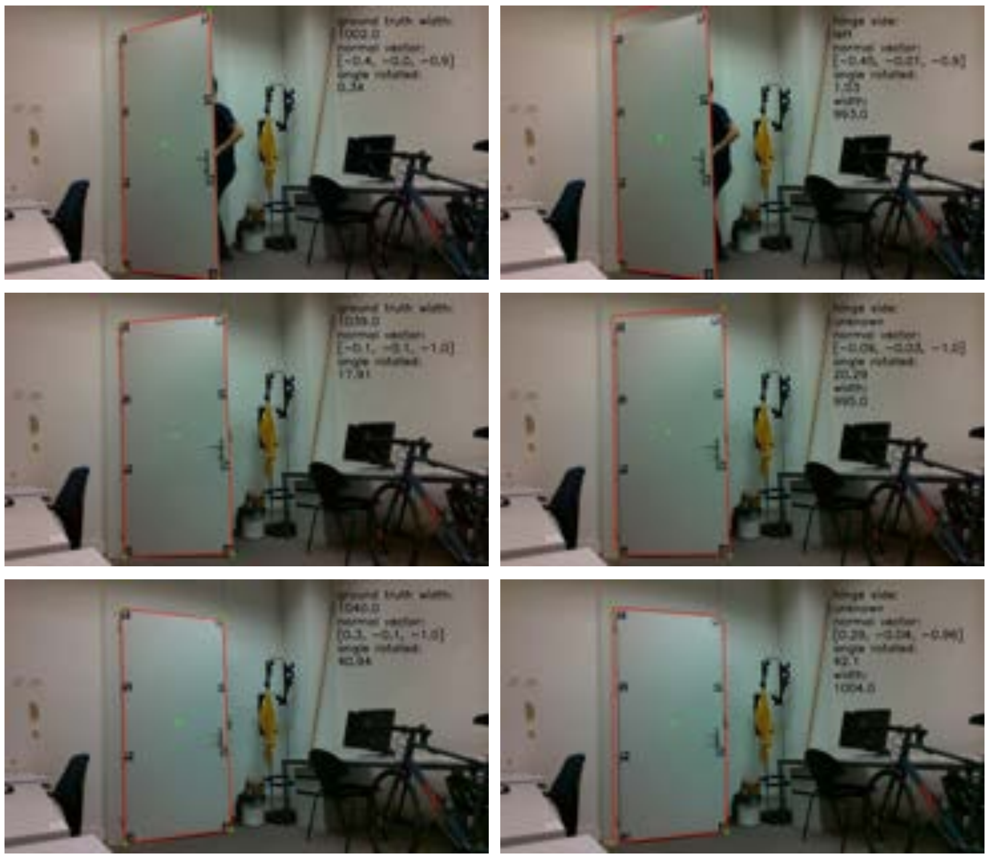


Figure 3.9: The F1 score in terms of the confidence score threshold

The F1 score is the harmonic mean of the precision and recall. The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero. As a consequence, the optimal performance for the average of all classes in Figure 3.9 is attained at confidence score threshold of 0.355 with a peak value 0.75. The F1 score of the door class reaches about 0.85 and thus indicates the best performance.

3.4 Estimation with a Static Camera

For the preliminary test of the estimator, a naive experiment is designed, and various doors are collected using the RGB-D camera. To be specific, the camera is fixed in the world coordinate, such that the camera coordinate coincides with the world coordinate. The purpose of it is to enable door state estimation without camera extrinsics information. The simple configuration without robot odometry, as well as the re-projective validation, is deployed to efficiently test the feasibility of the object detector and the pointcloud processing modules. Several scenarios in the dataset and our test results are shown in A.1 and A.2.



ground truth width:
1002.0
normal vector:
[-0.4, -0.0, -0.9]
angle rotated:
0.34

hinge side:
left
normal vector:
[-0.45, -0.01, -0.9]
angle rotated:
1.53
width:
993.0

ground truth width:
1039.0
normal vector:
[-0.1, -0.1, -1.0]
angle rotated:
17.91

hinge side:
unknown
normal vector:
[-0.06, -0.03, -1.0]
angle rotated:
20.29
width:
995.0

ground truth width:
1040.0
normal vector:
[0.3, -0.1, -1.0]
angle rotated:
40.94

hinge side:
unknown
normal vector:
[0.29, -0.04, -0.96]
angle rotated:
42.1
width:
1004.0

Figure 3.10: A scenario of door state estimation and parameter identification with camera fixed. Screenshots of 3 frames are randomly taken over the streaming of ground truth (left column) and estimation (right column). The red quadrilateral indicates the ground truth or estimated door after back projection from 3D, and the green dots indicate the position of re-projected door corners and the CoM position in the image plane. Some key estimation results are recorded at the top right corner of each image.

Although the camera is fixed, we do not require that the indoor scene is purely static. The target door, either static or not, can be estimated and identified. 3D information of it, including the dimensions, the hinge side, the angle rotated, the normal vector and the CoM position, can be obtained (Figure 3.10).

In order to generate ground truth annotations of the data, an AprilTag based approach is proposed. AprilTags [34, 35] are a specific type of fiducial marker, consisting of a black square with a white foreground that has been generated in a particular pattern. The black border surrounding the marker makes it easier for computer vision and image processing

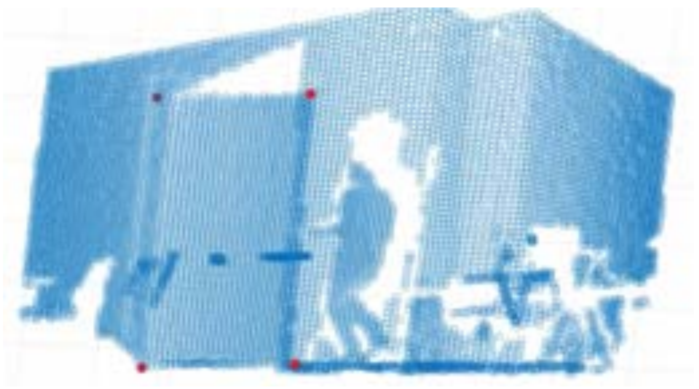


Figure 3.11: The pointcloud converted from the depth image of the scenario shown in Figure 3.10. The pointcloud is downsampled and the four points in red indicates the corner positions of the door. The 3D corners are back projected to 2D points on the image plane, and the latter is colored green and shown in 3.10.

algorithms to detect the AprilTags are universally applied in a variety of scenarios, including variations in rotation, scale, lighting conditions, etc. As is shown in Figure 3.10, several tags are placed along the borders and around the corners of the door. Their rotation and translation w.r.t the camera are recorded in real-time, so that we can track the position of the corners of the door. With the corners, the door geometric and kinematic information mentioned above can be easily deduced by connecting the corners and fitting the door plane. In the example case, redundant AprilTags are placed on the door (8 rather than 4), in order to avoid potential miss-detection of one or some of the tags. The results from the approach can be regarded as the ground truth, while the precision of the fiducial system is so high that the absolute error can be ignored, as is indicated by the yellow curve in Figure 3.12.

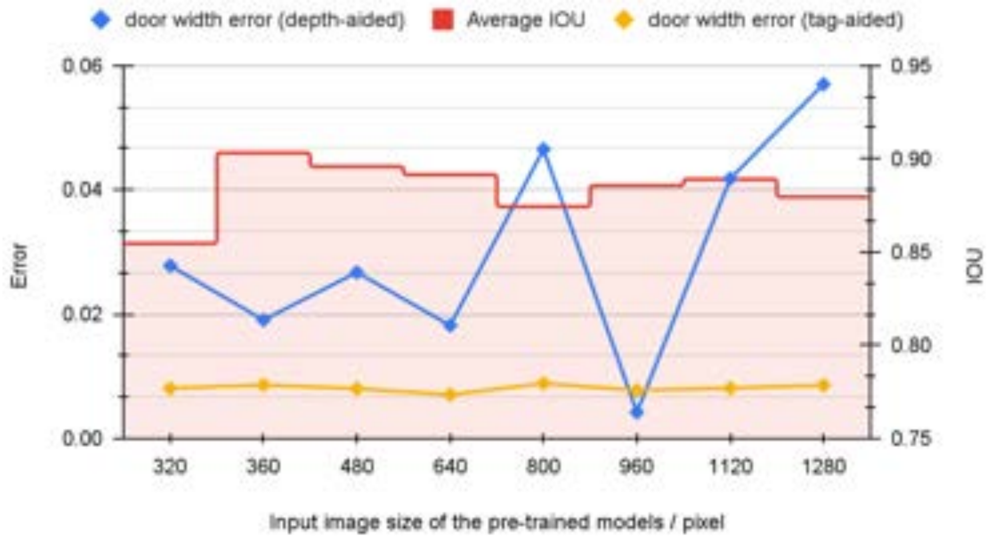


Figure 3.12: Comparison of the tag-aided and the depth-aided approaches with a fixed camera. The target door is non-static though, which is initially open and closed by a human (Figure 3.10). The relative error e (vertical axis) is defined by $e = ({}_{\mathcal{D}}w_{e,k} - {}_{\mathcal{D}}w_{gt,k})/{}_{\mathcal{D}}w_{gt,k}$, where ${}_{\mathcal{D}}w_{gt,k}, {}_{\mathcal{D}}w_{e,k}$ indicate ground truth width and estimated width, respectively. The ground truth width is measured by a measuring tape. The average IOU indicates the intersect between the the depth-aided and the tag-aided bounding boxes over more than 30 frames.

In the figure, YOLO networks of different training image sizes are leveraged to generalize the evaluation. When training a YOLO network, we can manually set its image size. No matter what size the original image is in the training set, it will be resized (down-sampled or up-sampled) to fill in the desired image size. Normally speaking, this hyperparameter influences the precision of the network. Judging by the high IOU values, we can draw the conclusion that the detector and point cloud processor work well in an interference-free scenario.

3.5 Estimation in Mobile Manipulation

With the object detector and the pointcloud processing modules verified in 3.4, we focus on the mobile perception with a non-static camera placed upon a legged robot. In this scenario, the target door is assumed to be absolutely static, since the perception task aims to explore the state and the nature of an unfamiliar door before starting the mobile manipulation task.

3.5.1 Experiment

Compared with the previous experiment, this one receives RGB-D images from an on-board camera mounted on the mobile robot, as well as the robot odometry from the robot navigation stack. The RGB image, the depth image and the robot transformation streaming are required to be well synchronized, since asynchronous data will very possibly lead to tracking inaccuracy.

Several challenging scenarios are explored to evaluate the robustness of the estimator.

1. Door incompleteness (Figure 3.13)
2. Moving human occlusions (Figure 3.14)
3. Door diagonal forward view (Figure 3.15)
4. Door disappearance and reappearance (Figure 3.16)

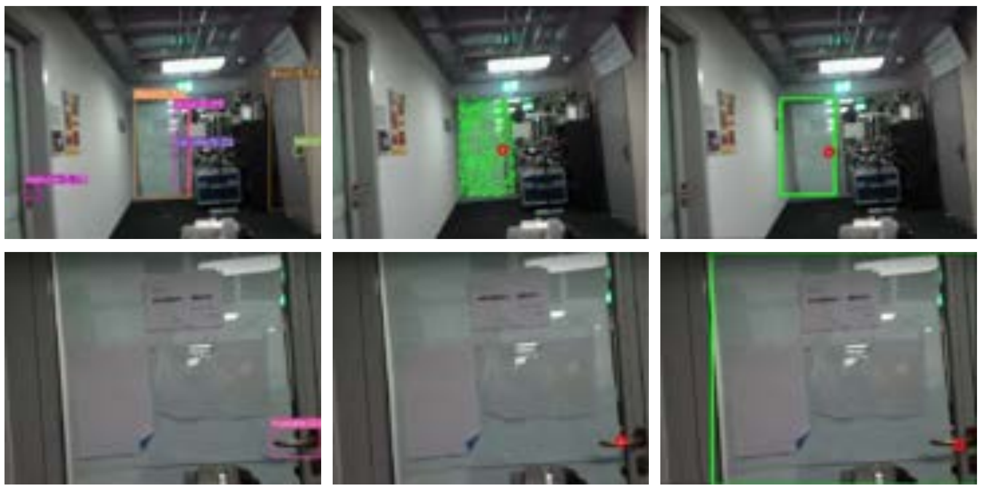


Figure 3.13: Scenario 1: Door incompleteness. The YOLO detections(left column), the measurement (middle column) and validation (right column) bounding boxes when the legged robot approaches the target door. Two frames of screenshots are taken from the streaming of estimation results. Bounding boxes and points are highlighted, with YOLO bounding boxes in various colors, door bounding boxes or points in green and handle bounding boxes or points in red.

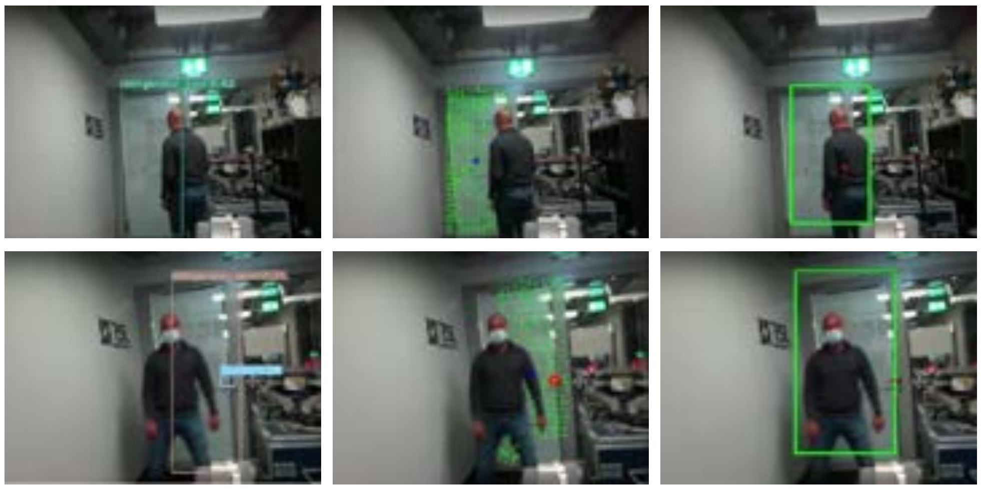


Figure 3.14: Scenario 2: Moving human occlusions. The YOLO detections(left column), measurement (middle column) and validation (right column) bounding boxes when the target door is occluded by a passer-by.

In Figure 3.13, The robot starts estimating the door from far away from the target by selecting a RoI by humans. It is controlled by Joystick and approaches the target until the gripper reaches the handle. As can be seen, the handle and door estimation remains convincing even though the door is not fully in the view. We achieve this by suspending the Kalman filter updates of the door size and position when the re-projected corner points of the door lie outside the image plane. If we keep updating the information otherwise, the estimation would drift since the measurements are wrong when the door is incomplete.

In Figure 3.14, a passer-by appears in front of the door while the robot is exploring and estimating the door. Although the measurement of the two frames is influenced by the occlusion, the posteriori bounding boxes of door and handle remain sound, owing to the past estimation.

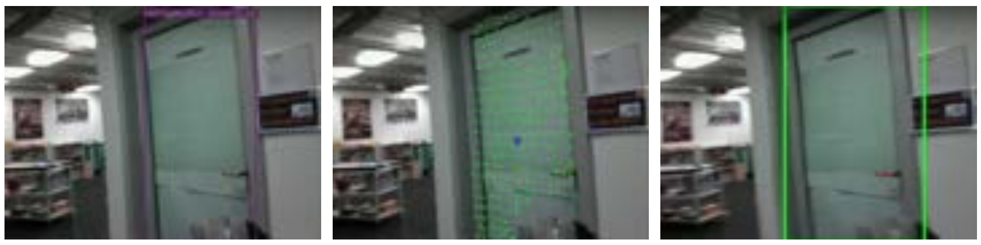


Figure 3.15: Scenario 3: Door diagonal forward view. The YOLO detections(left), measurement (middle) and validation (right) bounding boxes from a diagonally forward view in terms of the target door.

In Figure 3.15, the target door is approached from a diagonally forward direction rather than from the right front of it. The door appears to be a parallelogram instead of a rectangle as usual because of the perspective. Considering the rectangular YOLO bounding box, the imperfect overlapping might lead to more outlier points in the pointcloud and failure of estimation. As far as we can see in the frame, the measurement of the target remains reasonable in this case.

In Figure 3.16, the robot turns 180 degrees around such that the target door disappears from the view for 20 seconds. The non-target doors are not mistaken as the target one though the dimensions of them might be similar. When it turns back, the previous estima-

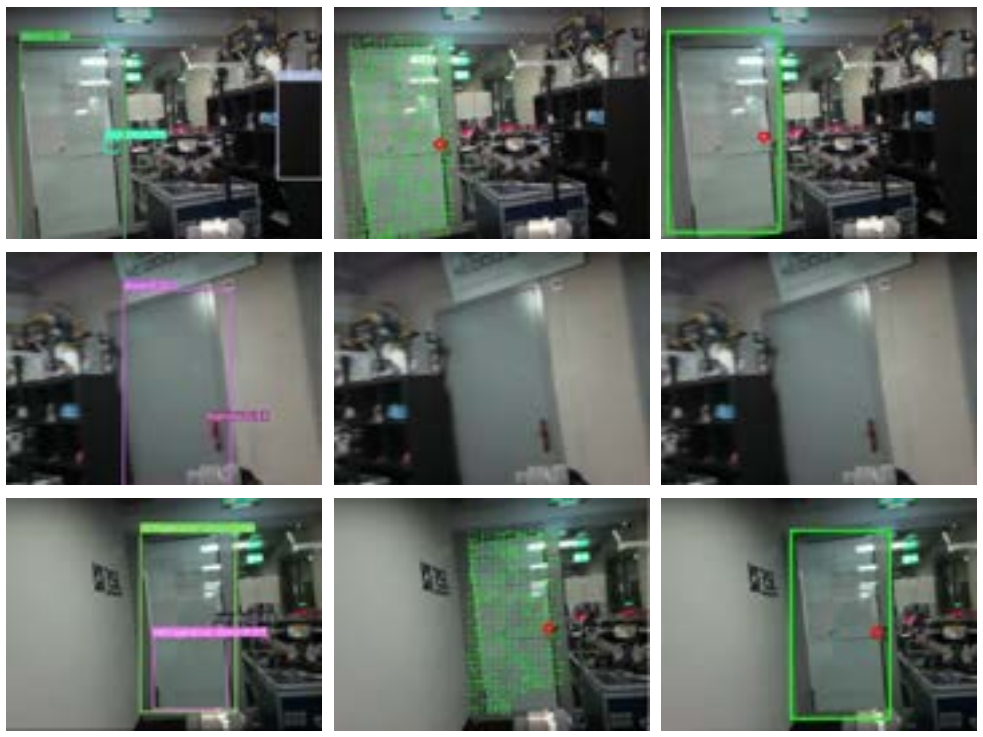


Figure 3.16: The YOLO detections(left column), measurement (middle column) and validation (right column) bounding boxes when the target door disappears in the view for about 20 second (middle row) and then reappears (bottom row).

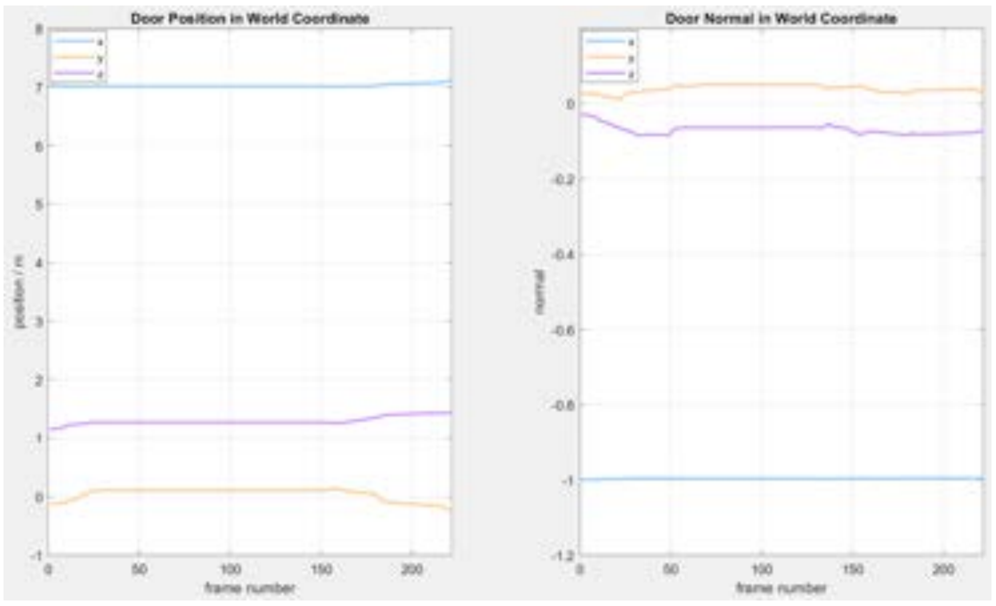


Figure 3.17: The estimated door pose in $\{\mathcal{W}\}$

tion is proved to be solid enough that the re-projected bounding box accurately outlines the target.

3.5.2 Door Estimation Results

In the offline mobile manipulation experiment above, some key estimation results are recorded related to Kalman filtering. In the experiment, the legged robot approaches the static door until a reachable distance, and trots backward afterwards.

The target door position and normal vector in terms of frames are illustrated in Figure 3.17. The posterior pose of the door remains relatively stable across the frames. The update of door CoM position is suspended during frame 40-150, because the target door is too close to the camera that it is only partially in the view. The fluctuation of the plots after frame 180 is due to the fact that the legged robot quickly trots backward and thus error occurs.

The normal vector of the door lies along the x-axis in $\{\mathcal{W}\}$. Since the incomplete door still offers solid normal information, the update of door normal is not suspended until too few points are included in the pointcloud (frame 70-130).

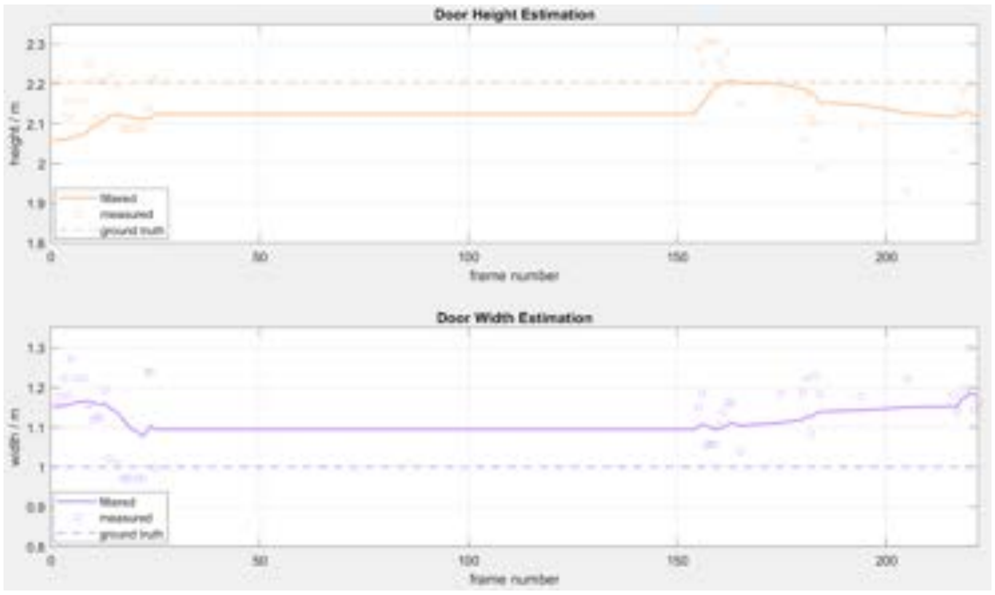


Figure 3.18: The estimated door width and height (measurement and posteriori), as well as the ground truths

As we can see in Figure 3.18, the door width and height estimation remains close to the ground truths. The error of the width is about 12 centimeter, while the door remains tightly closed. The residual points belonging to the wall cannot be removed from the door by RANSAC, since they are in the same plane and the difference in depth is not distinguishable. The error of the bounding box in the RGB image thus leads to the width estimation error. By comparing the measurement and filtered data in the plots, conclusion can be drawn that Kalman filter effectively alleviates the influence of some outlier measurements.

3.5.3 Handle Estimation Results

In terms of handle position estimation (Figure 3.19), difficulties frequently arise. First, the handles are smaller and not easily detected. Second, the handles are usually made of metal, which has a property of specular reflection and might cause the failure of depth sensing. However, the performance of our estimator on handles is really satisfying, in that

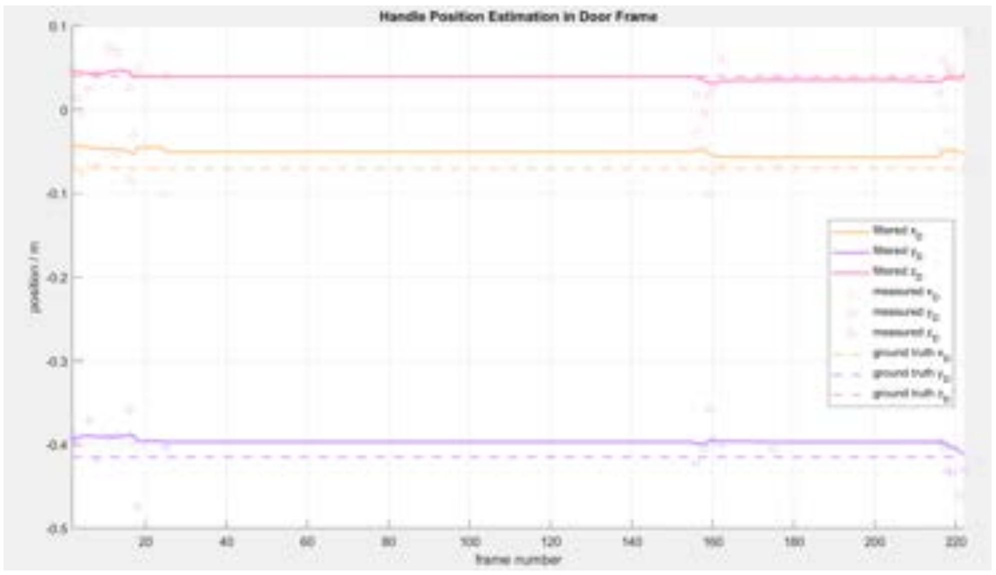


Figure 3.19: The estimated handle position in $\{\mathcal{D}\}$ (measurement and posteriori), as well as the ground truth

the absolute error along each axis after filtering is no more than 2 cm. When the door is not fully in the view as is mentioned above, we stop updating the handle position in $\{\mathcal{D}\}$ in Kalman Filter since the position is related with the door position. There are fewer measurements of the handle than the door, since the handle is less detectable and its depth information, even detect, might not suffice its position analysis.

3.6 Visual Servoing Handle Grasping

The task space experiment of visual servoing effectively evaluates the feasibility of the estimator. Visual servoing is closed-loop position control for a robot end-effector provided by machine vision [36].

The visual servoing experiment of handle grasping is introduced from simulation to real hardware.

3.6.1 Simulation

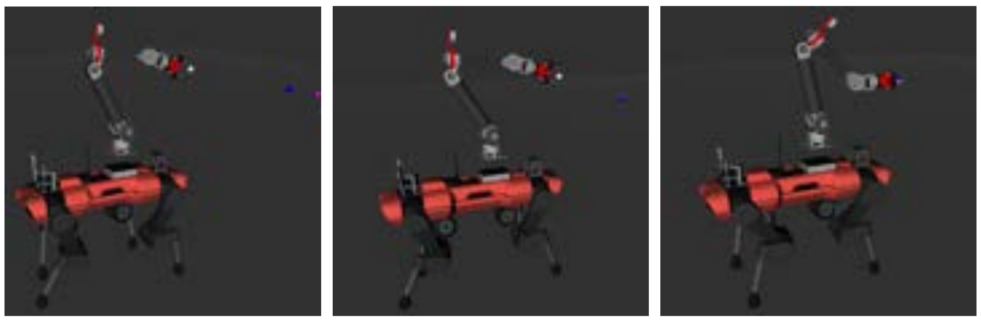


Figure 3.20: The visual servoing handle grasping task in simulation. the pink point in the video indicates the estimated position of the handle, the blue point is the end-effector reference position, and the white point implies the gripper position.

We test the visual servoing task in simulation before running it on hardware. The position of the target handle is determined by the estimator, and passed to the visual servoing module. In the module, either a PID controller or a MPC controller is implemented to generate a desired end-effector reference from the target position.

In Figure 3.20, the quadrupedal robot is able to gradually approach the target, stop trotting, transit from base mode the end-effector mode, and eventually grasp the target using the gripper.

3.6.2 Hardware Experiment

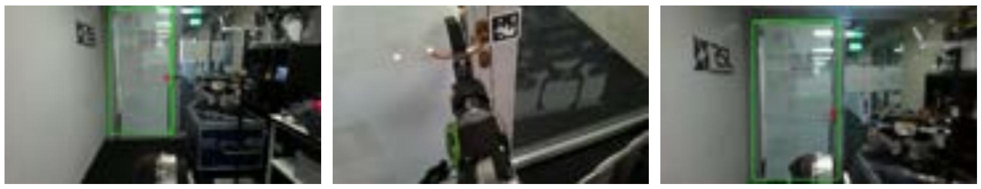


Figure 3.21: The preliminary open-loop visual servoing handle grasping experiment on hardware. The legged robot is controlled by Joystick to approach the door (left), grasp the handle manually (middle) and retreat (right). Intensive motion of trotting leads to motion blurred images with re-projected target door (green bounding box) and handle (red dot).

A preliminary experiment is designed for trial and error before the real visual servoing experiment is implemented (Figure 3.21). In the trial experiment, the legged robot is controlled by the joystick to perform the desired handle grasping actions. A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling. Meanwhile, the estimator runs on the onboard Jetson workstation and outputs the door estimation results, which are monitored in the visualization tool on a distant PC.

As is plotted in Figure 3.22, either the posteriori door CoM position or the normal vector in $\{\mathcal{W}\}$ along 3 axes remains close to the ground truths, which indicates that the estimation pipeline remains robust on hardware.

The prior experiment serves for debug purposes and is a necessary step, though the estimation results are not yet passed to the planner as references.

3.7 Object Tracking

In this section, the effectiveness of object tracking are evaluated from perspectives of the real-time estimation in mobile manipulation, the comparison with state-of-the-art tracking, and the error analysis of different re-projective validation strategies.

3.7.1 Comparison

In the algorithm, the object is localized by summing the responses of the learned correlation filters and weighted by the estimated channel reliability scores. On the contrary to our visual and robot odometry based tracking approach, CSRT tracker localizes the target purely on the RGB image level.

As is shown in 3.23, a robot is approaching the target door while estimating the handle position in real-time. The CSRT tracker easily loses track of the target, when the gripper partially occludes it. In contrast, the position of the handle is correctly found by our tracker even when it is totally occluded.

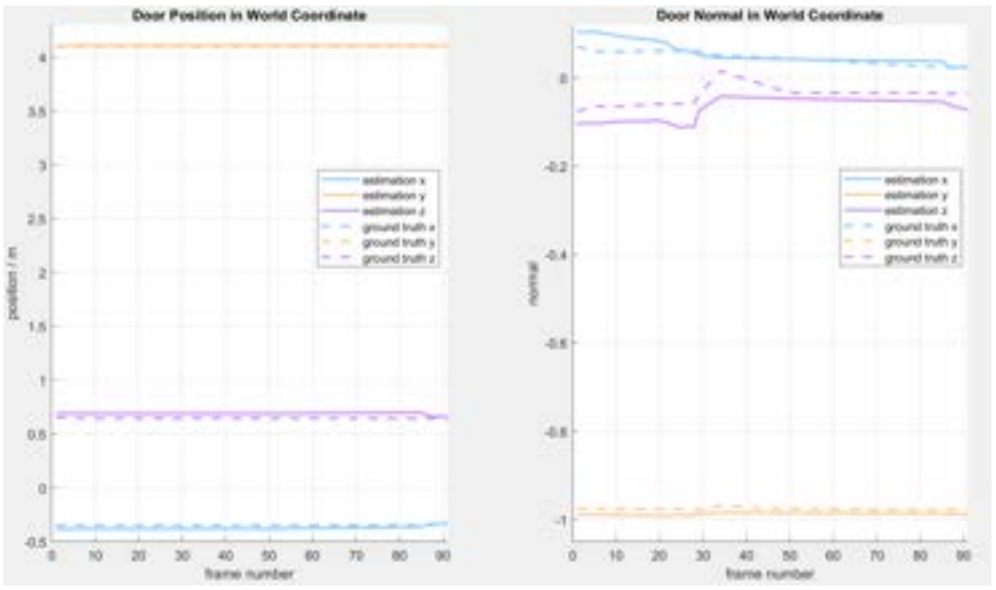


Figure 3.22: Door pose estimation and ground truth in World coordinate in the preliminary visual servoing test. The ground truth is obtained from Apriltags placed on the corners of the door. Error occurs in the Apriltag-aided odometry, which explains the fluctuation of the ground truth normal curves along x axis and z axis. The values of the ground truth are thus processed by an averaging filter in order to minimize the error.

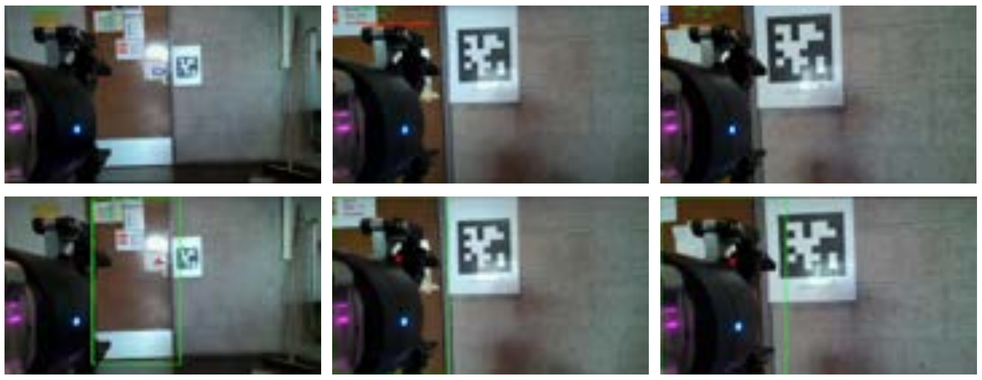


Figure 3.23: The comparison between CSRT tracking (first row) and our tracking (second row) over three randomly sampled frames (from left to right). The handle position is illustrated by a blue box in CSRT tracker scenario, while it is highlighted by a red point in the scenario using our tracker. The green box in the second row indicates the posteriori bounding box of the target door.

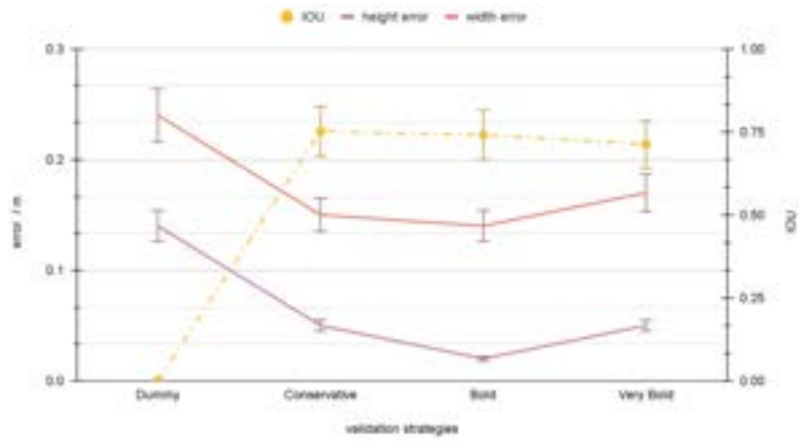


Figure 3.24: The evaluation of strategies in re-projective validation (the IoU between the re-projected bounding box and the YOLO bounding box of the highest score; the data are average results over frames)

In the occlusion case, the YOLO often fails in detecting the target door and handle, and we thus simply skip this frame. However, there are sometimes successful target detection even with a person in between (Figure 3.14). The point cloud of a person is not strictly on a plane as a door is. In order to avoid the case that a plane is fitted around the person, we set a small standard deviation threshold for RANSAC. On the other hand, the frame will be skipped if there are not sufficient RANSAC inlier points. In this way, if the target door is mostly occluded and the door behind is barely visible, the frame contains too few information to analyze.

3.7.2 Re-projection Strategies

Several strategies can be applied in re-projective validation.

- The conservative strategy
  As shown in the Algorithm 1, it exits the current frame if no valid YOLO box is found.
- The bold strategy
  Instead of exiting the frame when the selection fails, the validation box $\tilde{b}_k$ is taken as B_k.
- The aggressive strategy
  $\tilde{b}_k$ is selected as B_k, even if YOLO detects nothing in the frame ($U = 0$).
- The dummy strategy
  The estimator is simplified by removing the re-projective validation, and the YOLO detection is unconditionally relied upon. To be specific, the door detection from YOLO with the highest confidence would be B_k.

From Figure 3.24, the dummy method skips re-projective validation and thus the IoU is zero. The conservative and bold strategies lead to more precise door size estimation and better consistency with the YOLO bounding boxes than the others. Considering the algorithm efficiency and precision, a trade-off decision should be the conservative or the bold strategy. Theoretically speaking, the conservative strategy gives up quit a few frames when the IoU is smaller than the threshold compared with the bold strategy, which decreases its number of measurement and makes estimation inaccuracy possible. However, in terms of individual measurement, a safe choice definitely leads to higher probability of precision.

3.8 Elapsed Time

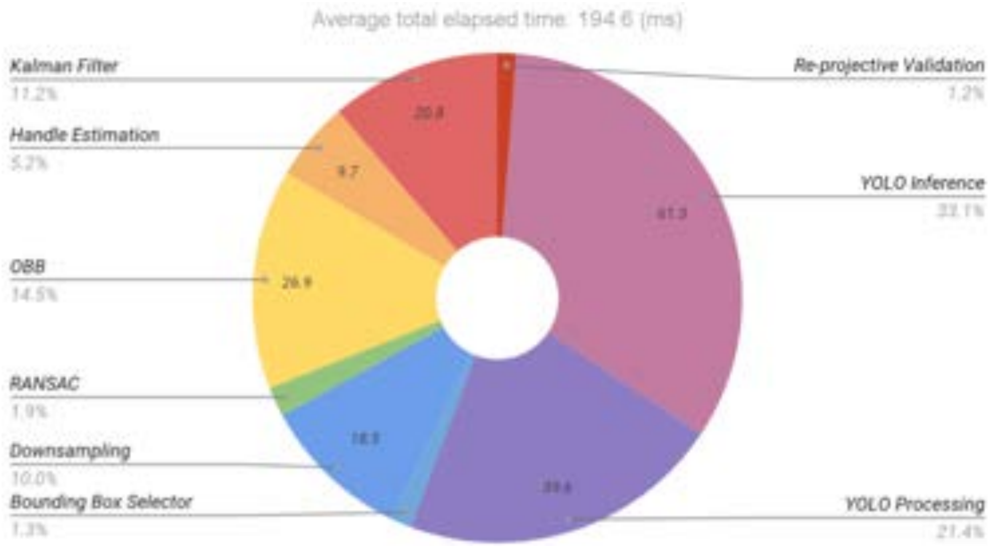


Figure 3.25: The average elapsed time of the door state estimation and parameter identification algorithm over hundreds of frames in real-time mobile manipulation.

The complete estimation and tracking algorithm runs at about 5-7 fps, as is illustrated in Figure 3.25, which depends on the performance of CPU and GPU. The speed of the algorithm is adequate for the real-time state estimation of static doors. More than half of the total elapsed time is invested in YOLO inference, processing, and filtration. The YOLO inference alone is fast enough though, which reaches about 20 fps on the Xavier GPU with a TensorRT structure. Pointcloud processing, including oriented bounding box, RANSAC, and downsampling, takes up about 25% of the total elapsed time.

3.9 Error Analysis

3.9.1 Results

As is illustrated in Figure 3.26, the overall error of the metrics can be limited within $0.1m$ in most cases. The door sizes error appears to be larger than the handle position error in magnitude, in that the target door remains tightly closed and there is thus no distinction between the wall and the door in terms of depth in the depth image. After further optimization of the algorithm, the handle position error is further alleviated and satisfies the precision requirements of the visual servoing experiment.

3.9.2 Sources of Error

The estimation error above might result from the following reasons,

1. Ambient lighting
The depth images are sometimes corrupted by ambient lighting. Ambient lighting could be a serious problem for structured light cameras. Time-of-flight cameras like RealSense L515 shows better robustness against it. However, when the scene is too bright, the imaging sensor saturates and all information is lost [37]. In our indoor robotic applications, the camera performs well in the pure artificial lighting environment. When strong sunlight casts in from windows behind the door in Figure 3.27(a), nevertheless, the camera cannot distinguish its own infrared light and ambient light, and depth artifacts thus, like holes, might appear.

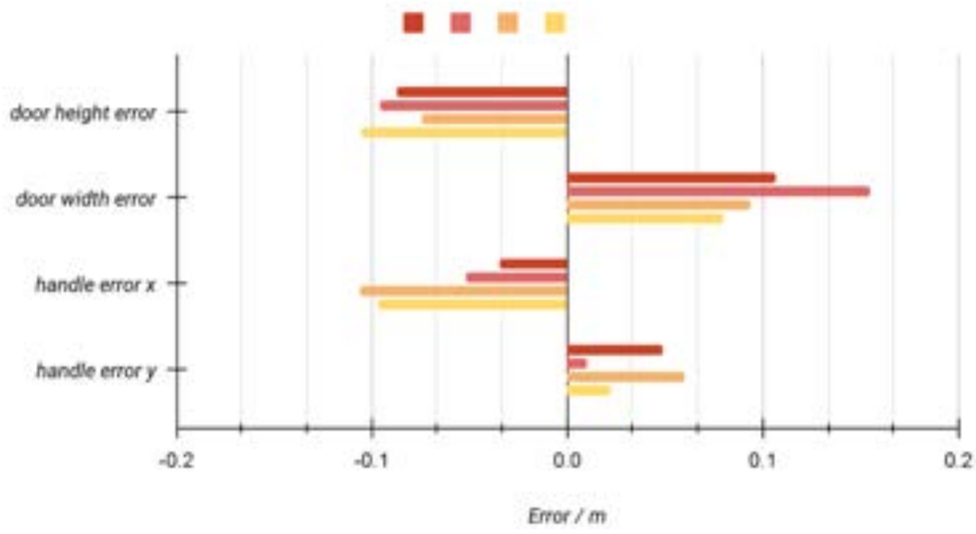


Figure 3.26: Door and handle estimation average absolute error column chart. Each column in the chart indicates the average result from the average of about 150 frames in the experiment of 3.13.

2. The valid range of detection
The nominal detection range of RealSense L515 LiDAR camera is 0.25 - 9 meter, but its practical range would shrink. In our experiment, the depth artifacts are tolerable when the distance between the door and the camera lies within 0.5-6 meter. When the robot initially discovers a door from far away or starts manipulating it with the gripper at the handle, the depth image might not function as expected.

3. The material of the target
The texture and material of the target door and handle also influence the time-of-flight camera. Smooth and specular surfaces appear overexposed in the infrared image, generating holes in the depth map. As a result, the 3D oriented bounding box does not perfectly fit the shape of the door in such scenarios.

4. Time synchronization
The robot odometry and the image data are not always perfectly synchronized because of the inconsistency of the timestamps. When the extrinsics does not correspond to the images, the re-projected validation bounding box could mismatch the ground truth.

5. YOLO false positive detection
Although the majority of the false positive bounding boxes from YOLO are filtered in the bounding box selector module of the algorithm, some might, though very unlikely, remain and affect the precision of the target estimation and tracking.

6. Rectangle 2D bounding boxes
The 2D bounding boxes from YOLO inference are always rectangles. In the cases of target observation from diagonally forward directions, a rectangle bounding box might encompass some points (from door frames and wall) not belonging to the door, while excludes points (from door corners) belonging to it. The oriented bounding box might fail this case, though the pointcloud processing module should remove the outlier. Compared with rectangle bounding boxes, rregular quadrilateral bounding boxes would be a more reasonable alternative.

7. 3D oriented bounding box
OBB is sensitive to the outliers of the target door pointcloud. Aggressive outlier removal sometimes cuts the target pointcloud off from the middle, especially when there are artifacts of holes in the pointcloud. For example, the bottom part of the door pointcloud in Figure 3.27(b) is sometimes sparse because of the interference of the ambient lighting. In this case, we obtain a smaller oriented bounding box of the door than expected, since part of the points belonging to the door are missing. On the other hand, some outliers cannot

be completely removed by the pointcloud processing module, like the points of door frames when the door is tightly closed. It would yield a larger bounding box, and the estimated door width and height are larger than expected.

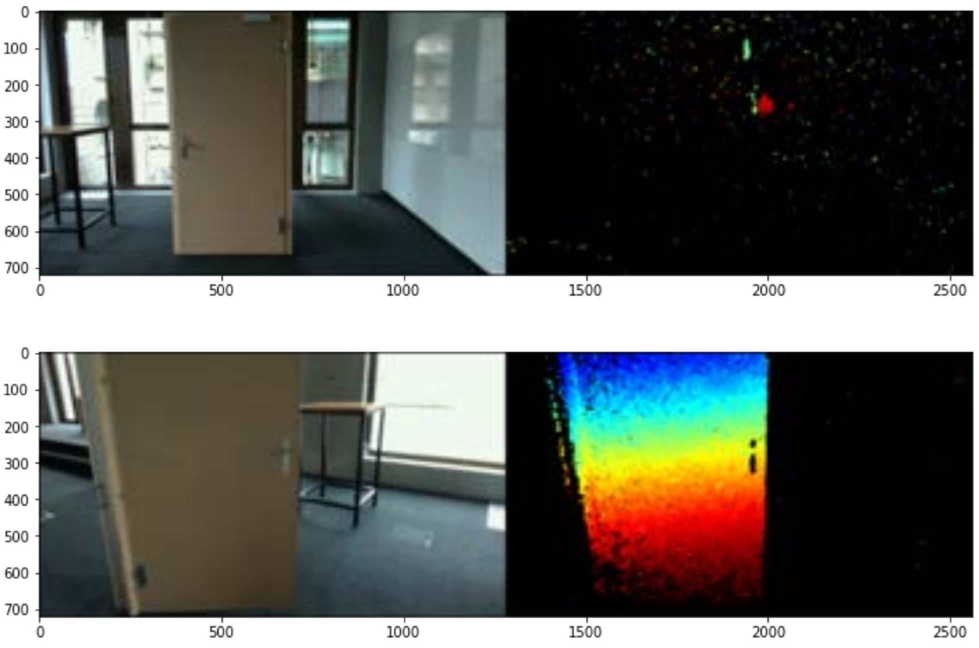


Figure 3.27: Two modes of depth image artifacts, with the RGB images (left) and depth images (right): Global invalidity (top) and Local erosion (bottom)

Chapter 4

Conclusion and Future Work

4.1 Conclusion

In this work, we devise a real-time algorithm for recursive door state estimation, parameter identification, and object tracking with a mobile RGB-D camera mounted on the robot. We combine the robot odometry from the robot state estimator and the visual odometry from the RGB-D camera, and demonstrated robustness against occlusions, door incompleteness, etc.

A supervised learning based door and handle detector is customized, using an annotated door dataset and image augmentation techniques. Problems arise related to the filtering of the false positive detection from the detector. Compared with the state-of-the-art approaches, we leverage the robot odometry and past estimation results to generate a re-projected validation bounding box, which indicates the 2D position and dimension of the target. A bounding box is selected from the candidate pool by comparing it with the validation bounding box. We get rid of the outlier points from the target pointcloud and obtain a new door state measurement, which is taken into account by Kalman filter. The posteriori door and handle information over time provides reliable ground truth for the re-projective validation.

We collect data consisting of various doors with either a static camera or a camera mounted on the mobile quadrupedal robot, ANYmal with arm. The dataset, including RGB images, depth images as well as calibrated camera intrinsic and extrinsic, validates the feasibility of the algorithm

The visual servoing handle grasping is already tested in simulation, as well as a preparatory open-loop visual servoing hardware experiment on the legged robot.

4.2 Future Work

When it comes to the future work, our first priority is to implement closed-loop visual servoing handle grasping on hardware. We have already tested it in simulation, and tested the estimator on hardware in an open-loop fashion.

In our previous work on ANYmal, the robot keeps grasping the handle and thus always knows the angle and velocity of the door. Once the gripper slips from the handle, the manipulation task might fail. Based on this framework and with some further adjustment and improvements, door state estimation after loss of contact could be tackled as well. The robot can thereby estimate the position of the slipped handle and re-grasp it.

Edge and corner detection could also be introduced to supplement the depth information. The current estimator might suffer from large door width or height error when the door is closed. In this case, RANSAC cannot remove the points belonging to the wall from those of the door, since they are basically in the same plane. The minor error of the bounding box in the RGB image could thus lead to potentially significant estimation error of door dimensions.

We could augment the object detector's training dataset and make it more robust to the motion blur. Blurring of images could have great impacts on image classification, especially in our legged robot applications, where the camera constantly moves around. Blur particularly obscures convolution's ability to locate edges in early levels of feature abstraction, causing inaccurate feature abstraction early in a network's training. We could either augment the training dataset with out-of-focus blurred door images collected from the robot, or create Gaussian blur to the images that already exist.

In several works on model-based manipulation, kinematics models of articulated objects, with either a revolute joint or a prismatic joint, are deduced from sensor data [7, 38, 39]. If we can endow the robot with the ability to infer the target kinematic model, our work can be generalized to different objects besides doors, like prismatic-joint articulated objects.

After door perception, subsequent works on door operation are closely relevant to this work as well. Reinforcement learning based approaches are drawing more attention in door manipulation [40, 41, 42, 43] to acquire manipulation skills. Besides the learning based approaches, model predictive control (MPC) is another popular method of generating motion and force policies. Sleiman et al. [44] design a unified model predictive control (MPC) framework for whole-Body dynamic locomotion and manipulation, which is verified by pushing/pulling a heavy resistive door.

We also consider global mapping construction using LiDAR, so that we could select the RoI more automatically. For example, we could segment the pointcloud by pixel-wise semantic segmentation [45, 46], and select the target door of interest using behavior tree and state machine.

After successfully estimating the state and parameters of the target door, the handle position in $\{\mathcal{C}\}$ helps in handle grasping. The door width also plays an important role in the unified MPC framework [44, 47]. We expect to implement the door opening experiment combining the estimation algorithm and a contact-implicit MPC approach.

Bibliography

[1] A. Jain and C. C. Kemp, “Behaviors for robust door opening and doorway traversal with a force-sensing mobile manipulator.” Georgia Institute of Technology, 2008.

[2] A. Andreopoulos and J. K. Tsotsos, “A framework for door localization and door opening using a computer controlled wheelchair for people living with mobility impairments,” in *RSS 2007 Manipulation Workshop*, 2007.

[3] ——, “Active vision for door localization and door opening using playbot: A computer controlled wheelchair for people with mobility impairments,” in *2008 Canadian Conference on Computer and Robot Vision*. IEEE, 2008, pp. 3–10.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[6] 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)*. IEEE, 2017, pp. 144–149.

[7] M. Arduengo, C. Torras, and L. Sentis, “Robust and adaptive door operation with a mobile robot,” *Intelligent Service Robotics*, May 2021.

[8] B. Quintana, S. A. Prieto, A. Adán, and F. Bosché, “Door detection in 3d coloured point clouds of indoor environments,” *Automation in Construction*, vol. 85, pp. 146–166, 2018.

[9] E. Klingbeil, A. Saxena, and A. Y. Ng, “Learning to open new doors,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2751–2757.

[10] S. Kim, H. Cheong, D. H. Kim, and S.-K. Park, “Context-based object recognition for door detection,” in *2011 15th International Conference on Advanced Robotics (ICAR)*. IEEE, 2011, pp. 155–160.

[11] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz, “Laser-based perception for door and handle identification,” in *2009 International Conference on Advanced Robotics*. IEEE, 2009, pp. 1–8.

[12] T. Sandy and J. Buchli, “Object-based visual-inertial tracking for additive fabrication,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1370–1377, 2018.

[13] D. Avots, E. Lim, R. Thibaux, and S. Thrun, “A probabilistic technique for simultaneous localization and door state estimation with mobile robots in dynamic environments,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 2002, pp. 521–526.

[14] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[15] N. Kyriakoulis and A. Gasteratos, "Color-based monocular visuoinertial 3-d pose estimation of a volant robot," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 10, pp. 2706–2715, 2010.

[16] B. P. Larouche, Z. H. Zhu, and S. A. Meguid, "Development of autonomous robot for space servicing," in *2010 IEEE International Conference on Mechatronics and Automation*. IEEE, 2010, pp. 1558–1562.

[17] P. Bouthemy, "A maximum likelihood framework for determining moving edges," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 5, pp. 499–511, 1989.

[18] T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, "Advances in real-time object tracking," *Journal of real-time image processing*, vol. 10, no. 4, pp. 683–697, 2015.

[19] Z. Kalal, J. Matas, and K. Mikolajczyk, "Online learning of robust object detectors during unstable tracking," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, 2009, pp. 1417–1424.

[20] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "Robust 3d object tracking from monocular images using stable parts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1465–1479, 2017.

[21] A. Lukezic, T. Vojir, L. ˇCehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6309–6318.

[22] X. Farhodov, O.-H. Kwon, K. W. Kang, S.-H. Lee, and K.-R. Kwon, "Faster rcnn detection based opencv csrt tracker using drone data," in *2019 International Conference on Information Science and Communications Technologies (ICISCT)*. IEEE, 2019, pp. 1–3.

[23] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, 2021.

[24] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[25] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[26] C. Connolly, "Cumulative generation of octree models from range data," in *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1984, pp. 25–32.

[27] R. Martin, I. Stroud, and A. Marshall, "Data reduction for reverse engineering," *RECCAD, Deliverable Document 1 COPERNICUS project, No 1068*, p. 111, 1997.

[28] S. Orts-Escolano, V. Morell, J. García-Rodríguez, and M. Cazorla, "Point cloud data filtering and downsampling using growing neural gas," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–8.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[30] C. V. Nguyen, S. Izadi, and D. Lovell, “Modeling kinect sensor noise for improved 3d reconstruction and tracking,” in *2012 second international conference on 3D imaging, modeling, processing, visualization & transmission.* IEEE, 2012, pp. 524–530.

[31] “Intel realsense lidar camera l515,” https://www.intelrealsense.com/lidar-camera-l515/, accessed: 2022-01-19.

[32] “Anymal,” https://www.anybotics.com/anymal-autonomous-legged-robot/, accessed: 2022-01-19.

[33] “Jetson agx xavier developer kit,” https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit, accessed: 2022-01-19.

[34] E. Olson, “Apriltag: A robust and flexible visual fiducial system,” in *2011 IEEE International Conference on Robotics and Automation.* IEEE, 2011, pp. 3400–3407.

[35] J. Wang and E. Olson, “Apriltag 2: Efficient and robust fiducial detection,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2016, pp. 4193–4198.

[36] F. Chaumette and S. Hutchinson, “Visual servo control. i. basic approaches,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[37] A. Kadambi, A. Bhandari, and R. Raskar, “3d depth cameras in vision: Benefits and limitations of the hardware,” in *Computer vision and machine learning with RGB-D sensors.* Springer, 2014, pp. 3–26.

[38] R. M. Martin and O. Brock, “Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2014, pp. 2494–2501.

[39] V. Zeng, T. E. Lee, J. Liang, and O. Kroemer, “Visual identification of articulated object parts,” *arXiv preprint arXiv:2012.00284*, 2020.

[40] Y. Urakami, A. Hodgkinson, C. Carlin, R. Leu, L. Rigazio, and P. Abbeel, “Doorgym: A scalable door opening environment and baseline agent,” *arXiv preprint arXiv:1908.01887*, 2019.

[41] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, “Learning force control policies for compliant manipulation,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2011, pp. 4639–4644.

[42] B. Nemec, L. Žlajpah, and A. Ude, “Door opening by joining reinforcement learning and intelligent control,” in *2017 18th International Conference on Advanced Robotics (ICAR).* IEEE, 2017, pp. 222–228.

[43] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *2017 IEEE international conference on robotics and automation (ICRA).* IEEE, 2017, pp. 3389–3396.

[44] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, “A unified mpc framework for whole-body dynamic locomotion and manipulation,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[45] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, “Understanding convolution for semantic segmentation,” in *2018 IEEE winter conference on applications of computer vision (WACV).* IEEE, 2018, pp. 1451–1460.

[46] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[47] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, "Contact-implicit trajectory optimization for dynamic object manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6814–6821.

Appendix A

Complementary Material

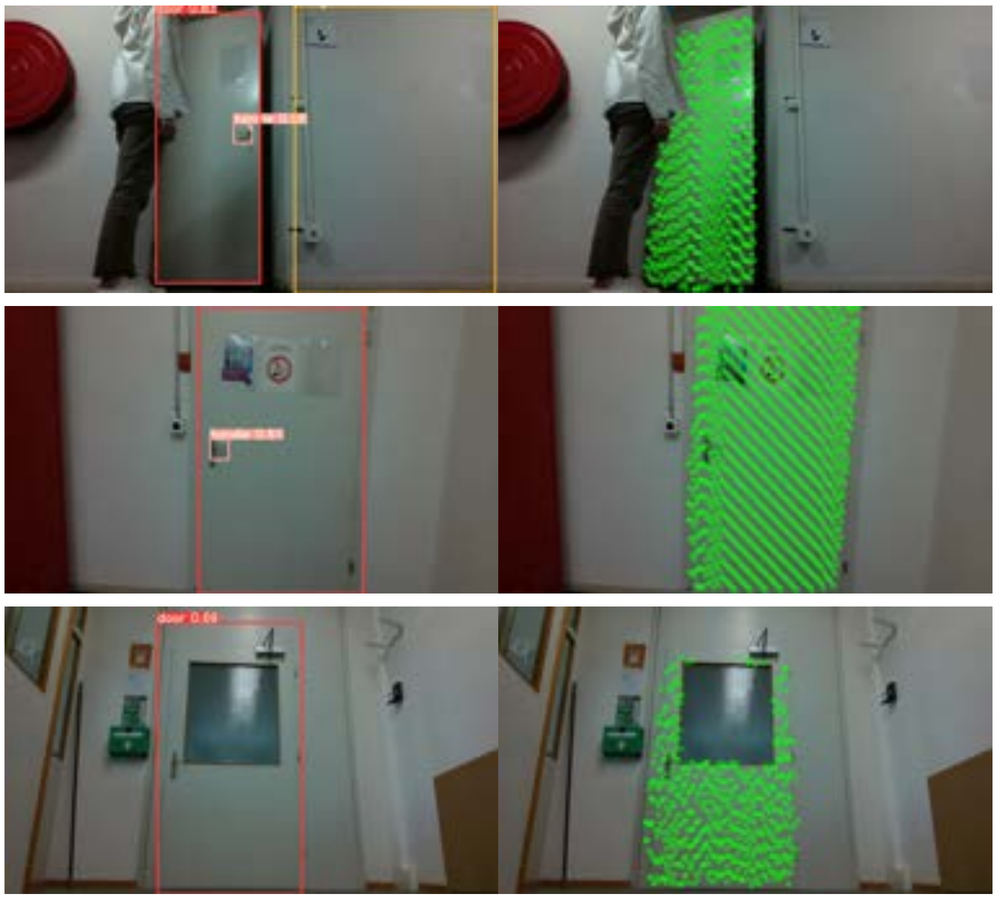


Figure A.1: YOLO masks (left column) and measurement (right column) in various scenarios of the naive experiment with camera fixed. (Part I)

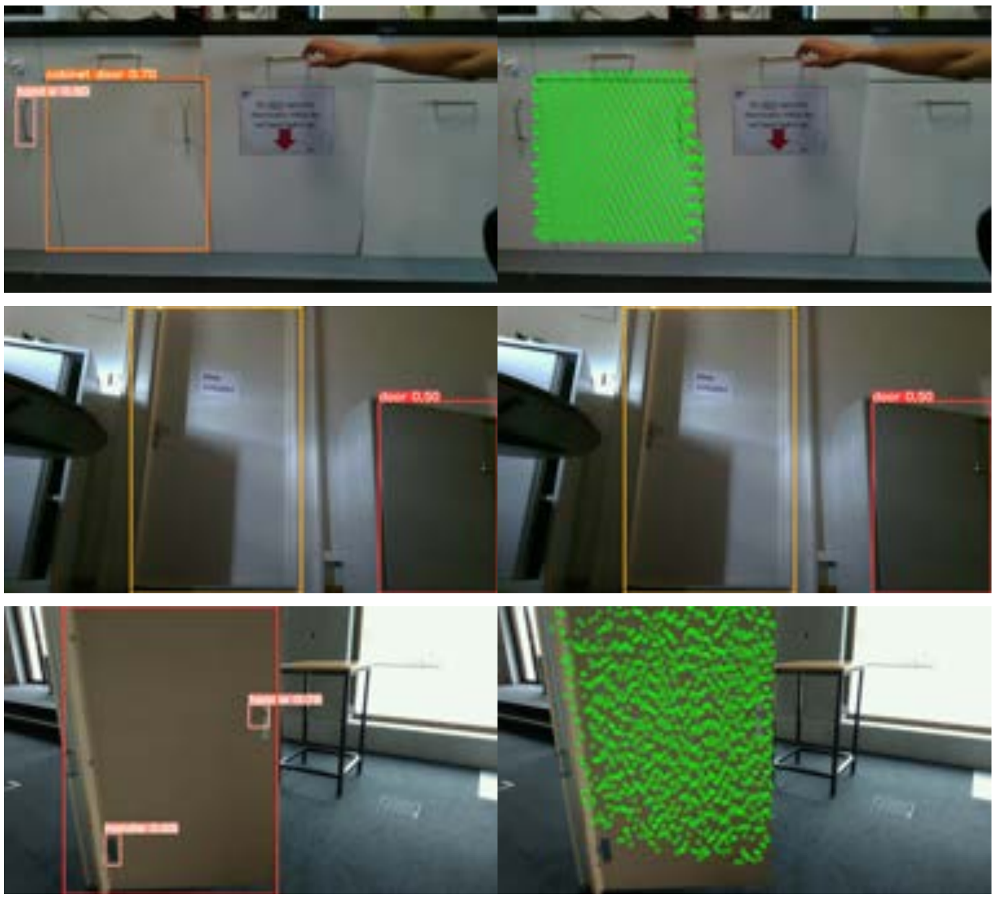


Figure A.2: YOLO masks (left column) and measurement (right column) in the naive experiment with camera fixed. (Part II)