Assistive 6-DOF Robotic Grasping of Unknown Objects

Inauguraldissertation zur Erlangung der Doktorwürde der Fakultät für Informatik und Data Science der Universität Regensburg

eingereicht

von
Henry Schaub
aus
München
2025

Erstgutachter: Prof. Dr. phil. Christian Wolff, Universität Regensburg
Zweitgutachter: Prof. Dr. rer. nat. habil. Alfred Schöttl, Hochschule München
Drittgutachter: Prof. Dr.-Ing. Johannes Fottner, Technische Universität München
Tag der mündlichen Prüfung: 27.02.2026

Die Arbeit entstand in gemeinsamer Betreuung durch die Universität Regensburg und die Fakultät für Elektrotechnik und Informationstechnik der Hochschule für angewandte Wissenschaften München.

Regensburg, 2026

Abstract

Picking up an unknown object is a central problem in robotics that has not yet been fully solved. While many approaches operate in a controllable environment, such as the industrial one, we target the assistive context.
Instead of a known and static workspace with clearly defined action constraints, our considered environment is unstructured and unknown. A large number of common, implicit assumptions cannot be made. The initial sensor measurements of imaging sensors might not show a valid grasp solution, motion goals of the manipulator might not be reachable or are in collision and the sensor measurements do not necessarily have to be within the domain of the training data.
In addition, the assistive context offers a variety of unique sensory challenges. The scene might be partially over-illuminated and the surface properties can vary greatly, which implies that measurement noise is not identically distributed. These challenges cannot be overcome by a pre-configured, optimal sensor positioning or software-setting, but rather must be dealt with as is during runtime.
We propose a closed-loop, end-to-end algorithmic pipeline that deals with assistive, robotic grasping in a holistic manner. A probabilistic sensor fusion method is introduced that accurately reconstructs the desired scene segment and enables us to reliably estimate the local reconstruction accuracy. Furthermore, we present an exploration and motion strategy that allows the robot to autonomously search for grasp options while ensuring user safety at controller level. Finally, we propose a probabilistic grasp sampling and evaluation algorithm that does not rely on prior object knowledge or a dataset tailored to the use case but instead utilizes the local estimation uncertainty to generate robust grasps in an online fashion.
The pipeline is real-time capable and extensive simulated as well as real experiments demonstrate that our method is robust and accurate, even with challenging household objects.

Zusammenfassung

Das Greifen eines unbekannten Objekts ist ein zentrales Problem in der Robotik, das noch nicht vollständig gelöst ist. Während sich viele Ansätze mit einer kontrollierbaren Umgebung befassen, z. B. einem industrieellen Setting, zielen wir auf den assistiven Kontext ab.
Anstelle eines bekannten und statischen Arbeitsbereichs mit klar definierten Aktionsbeschränkungen ist die Umgebung unstrukturiert und unbekannt. Häufige implizite Annahmen können nicht getroffen werden. Die initialen Sensormessungen abbildender Sensoren zeigen möglicherweise keine gültige Greifpose für das Objekt. Die Bewegungsziele des Manipulatorarms sind möglicherweise nicht erreichbar oder kollisionsbehaftet, und die Sensormessungen sind potentiell nicht Teil der Grundgesammtheit der Trainingsdaten. Darüber hinaus bietet der assistive Kontext eine Vielzahl von einzigartigen sensorischen Herausforderungen. Die Szene kann teilweise überbelichtet sein und die Oberflächeneigenschaften können stark variieren. Dies bedeutet, dass das Messrauschen nicht identisch verteilt ist. Solche Herausforderungen können nicht durch eine vorkonfigurierte, optimale Sensorpositionierung oder Software-Einstellung überwunden werden, sondern müssen während der Laufzeit behandelt werden.
Für das robotische Greifen von Objekten im assistiven Kontext stellen wir eine algorithmische closed-loop Ende-zu-Ende Pipeline vor. Wir stellen eine probabilistische Sensorfusionsmethode vor, die das gewünschte Szenensegment genau rekonstruiert und es ermöglicht, die lokale Rekonstruktionsgenauigkeit zuverlässig zu schätzen. Darüber hinaus stellen wir eine Explorations- und Bewegungsstrategie vor, die es dem Roboter ermöglicht, autonom nach Greifmöglichkeiten zu suchen und gleichzeitig die Sicherheit des Benutzers auf Controllerebene zu gewährleisten. Außerdem wird ein probabilistischer Sampling- und Bewertungsalgorithmus für das Greifen eingeführt, der sich nicht auf vorheriges Objektwissen oder einen auf den Anwendungsfall zugeschnittenen Datensatz stützt, sondern stattdessen die lokale Schätzungsunsicherheit nutzt, um online robuste Greifposen zu generieren. Die Pipeline ist echtzeitfähig, und umfangreiche simulierte sowie reale Experimente zeigen, dass sie robust und genau genug ist, um Rekonstruktion und Greifen von schwierigen Haushaltsobjekten zu ermöglichen.

Contents

List of Figures

List of Tables

1. Introduction

The focus of this thesis is on assistive robotics for people with severe physical disabilities and the resulting difficulties in interacting with their environment. Due to their special needs, people in this target group involuntarily become dependent on others. They face many different challenges in daily life that can only be overcome with the help of others. The necessary support is often provided by healthcare professionals who ensure both physical and social well-being.

Assistive, robot arms such as the *Kinova Jaco* (*KINOVA Jaco assistive robot, User Guide*, 2021) and the *iArm* (Driessen, Evers, & v Woerden, 2001) help the target group to regain some of their independence and enable them to carry out everyday tasks such as scratching, eating or taking medication. These products are designed as wheelchair extensions and in case of the *Kinova Jaco* a certified, medical device.

They are operated manually, for example, with the aid of a joystick. The manual control unit of an electric wheelchair has a maximum of three degrees of freedom, which can be easily mapped to the joystick input device. However, the control of a manipulator arm must take seven degrees of freedom into account and hence, the manual control of the arm requires frequent switching between several different working modes, i.e. flipping through several pre-defined mappings between input and output space. It has been shown that this is time-consuming and requires a considerable learning effort and mental load to operate this system with the necessary precision (e.g. Al-Halimi & Moussa, 2017; Chung, Wang, & Cooper, 2013; Herlant, Holladay, & Srinivasa, 2016).

Our goal is the development of an intelligent assistive robotic system with the ability to grasp arbitrary, previously unknown objects and present them to the user. The system should have the ability to recognize objects designated by the user and to carry out collision-free path planning and object manipulation, thereby relieving the user of computationally complex tasks in a semi-autonomous manner, which has been shown to perform faster than manual control (Ka, Chung, Ding, James, & Cooper, 2017)).

1.1 Objective and Academic Challenges

From an algorithmic perspective, the overarching research goal is split into four blocks, which can be seen in figure 1. Each of these subsystems requires an individual approach

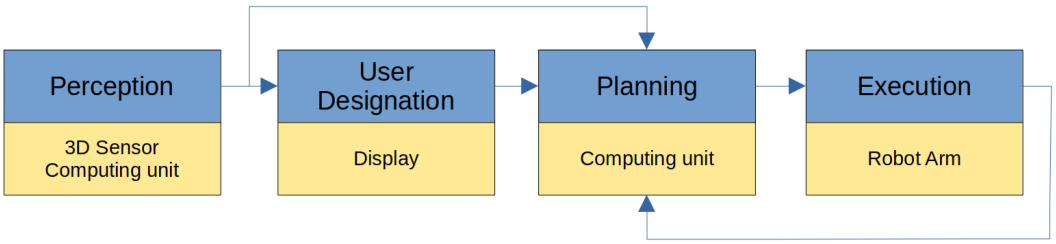

Figure 1: Subsystems and required hardware.

to solving its practical and academic challenges:

- Perception
  To interact with unknown objects, we need 3D-measurements of the object and the surrounding scene. A suitable sensor technology must be selected. The data must be interpreted and fused into a suitable representation to be useful for the following sub-tasks. The scene representation is strongly linked to the question of what constitutes as a good grasp and how corresponding chances of success can be determined.

- User Designation
  A suitable interface must be designed to give the user insight into the algorithm and enable him to command the algorithm at a high semantic level. For example, the user should be able to command the system to give him or her the water bottle without having to deal with the mentally demanding task of collision-free path planning.

- Planning
  The robot must be able to navigate through the initially unknown scene in a collision-free manner. Hence, the current scene representation must be used to derive subsequent motion actions and (re-) evaluate existing ones. If, for example, the current representation does not provide enough information to guarantee a valid, collision-free grasp for the designated object, then the necessary information must be collected. The research on a suitable mapping function from currently known scene to corresponding motion-commands is a central element of our work.

- Execution
  Since the scene is initially unknown, the execution of motion commands must be implemented in a dynamic, closed-loop manner instead of the typical sense-plan-act strategy. The robot must therefore be able to process changes in movement within split seconds. The top priority here is always the safety of the user, as the user and manipulator share the same workspace.

For reasons of feasibility of the approach, various constraints must be taken into account. Although the setup of the electric wheelchair leaves plenty of room for customization, it is essentially a mobile application. Hence, computing power is limited and one could, for example, question the usefulness of a system that requires a graphics card (*GeForce RTX™ 4090 GAMING X TRIO 24G*, 2026, $\sim$ 450 W) that needs more than fifteen times the power of the assistive robot arm, (*Specifications of Jaco assistive robotic arm*, 2025, $\sim$ 25 W) . The affordability of the hardware should be taken into account for similar reasons. The perception task must be able to deal with a variety of scenarios as we have to adapt to the environment rather than the other way around. For example the lighting conditions and surface properties vary greatly in the household setting, which can lead to large variations in the quality of the sensor measurements. The algorithm must be able to run robustly despite these variations.
These problems must be taken into account for the object manipulation, which is a central problem for (semi-) autonomous assistants. Many state-of-the-art robotic grasping approaches are tailored to an industrially motivated setting, where they are able to utilize a single scene measurement to predict grasping options for objects (e.g. Liang et al., 2019; Sundermeyer, Mousavian, Triebel, & Fox, 2021). However, the assistive context can lead to (partial-) occlusions and due to the problems mentioned above, the data may be partially corrupted. Instead of this one-shot strategy, an assistive robotic system must operate in a closed-loop manner and take multiple measurements from varying perspectives into account to predict and validate grasping possibilities. Many approaches that follow this close-loop strategy are unsuitable for the assistive context as they assume a benign initial sensor perspective (e.g. Cai et al., 2022), greatly restrict the solution space to match publicly available datasets (e.g. Morrison, Corke, & Leitner, 2018) or assume consistent sensor measurement quality (e.g. Breyer, Chung, Ott, Siegwart, & Nieto, 2022).
The problematic nature of unstructured, unknown environments and the constraints that

are imposed by assistive robotics represent a research gap that is not fully addressed by the current literature. Our research focuses on design and implementation of the first and the last two sub-tasks in figure 1, i.e. the perception and representation of three-dimensional measurements and closed-loop planning and control of a robot arm with the goal of grasping unknown objects.

1.2 Publications

Several sections of this thesis have been previously published and are reprinted here in adapted form. For all subsequent publications, the author is responsible for the algorithmic design, wrote the code, drafted the manuscript and performed the experiments. The co-authors contributed to the conceptualization, proofread the manuscripts and provided valuable feedback.
Apart from translation software [1], no generative tools were used in the writing of the publications or this thesis. The publications that are directly related to this thesis are listed in the following:

- *6-DOF Grasp Detection for Unknown Objects*
  Henry Schaub, Alfred Schöttl; 2020; 10th International Conference on Advanced Computer Information Technologies (ACIT)
- *6-DOF Grasp Detection for Unknown Objects Using Surface Reconstruction*
  Henry Schaub, Alfred Schöttl, Maximilian Hoh; 2021; 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)
- *Probabilistic Fusion of Depth Maps with a Reliable Estimation of the Local Reconstruction Quality*
  Henry Schaub, Alfred Schöttl, Maximilian Hoh, 2022 IEEE Robotics and Automation Letters (RAL), vol. 7, no. 4
- *Probabilistic Fusion of Depth Maps with Local Variance Estimation*
  Henry Schaub, Nico Leuze, Maximilian Hoh, Alfred Schoettl; 2023 IEEE Sensors, Wien

[1] `https://www.deepl.com`

- *Probabilistic Framework for Active 6-DoF Grasping*
  Henry Schaub; CC-Partner Fachtagung, 2023, CCPAF / MLIS, München
- *Probabilistic Closed-Loop Active Grasping*
  Henry Schaub, Christian Wolff, Maximilian Hoh, Alfred Schöttl; 2024 IEEE Robotics and Automation Letters (RAL), vol. 9, no. 4
- *Probabilistic Closed-Loop Active Grasping*
  Henry Schaub, Christian Wolff, Maximilian Hoh, Alfred Schöttl; 2024 International Conference on Intelligent Robots and Systems (IROS), invitation based on the previous RAL publication

Especially the publications at *IEEE Robotics and Automation Letters* [2] are quite influential for this thesis, as the feedback we got from the reviewers as well as the editor was as comprehensive as valuable.

The author contributed as a co-author to the following publications. Although these deal with research areas that are not within the scope of this thesis, the collaboration and exchange with colleagues has formed an academic foundation on which many of the concepts that are presented here are based.

- *A Generative Model for Anomaly Detection in Time Series Data*
  Maximilian Hoh, Alfred Schöttl, Henry Schaub, Franz Wenninger; 2022; Proceedings of the 3rd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2022)
- *Generative Anomaly Detection in Multivariate Time Series*
  Maximilian Hoh, Alfred Schöttl, Henry Schaub, Nico Leuze; 2023; Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI)
- *Rethinking Time Series Anomaly Detection: A Scalable Transformer-based Framework for Large Contexts*
  Maximilian Hoh, Henry Schaub, Nico Leuze, Alfred Schöttl; 2025 International Conference of Control, Automation and Robotics (ICCAR)

[2] Currently ranked second in the field of robotics according to google scholar: `https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_robotics`

- *SSL-VoxPart: A Novel Solid-State LiDAR-Tailored Voxel Partition Approach for 3D Perception*
  Nico Leuze, Henry Schaub, Maximilian Hoh and Alfred Schoettl; 2023 IEEE Sensors, Wien
- *Fostering Sparsity in Sparse Convolution Networks for Efficient 3D Perception via Feature-Guidance*
  Nico Leuze, Henry Schaub, Maximilian Hoh, Samed Dogan, Nicolas R. Peña, Nikolas Voss and Alfred Schoettl; 2024 IEEE Sensors, Kobe

1.3 Thesis Outline

In this thesis we present a novel algorithm for robotic grasping in an assistive context. This context raised issues that have not been fully resolved by the state of the art. The chapters are organized according to the sub-problems we needed to overcome and each of them discusses the problem, the presented solution approach and results. The work is structured as follows:

Chapter - 1 Introduction motivates the research in this thesis and defines the overall objective.

Chapter - 2 Early Approaches for Grasping in Six Dimensions shows our approaches from two publications on robotic grasping where we expanded on low dimensional state-of-the-art methods to achieve 6-DOF grasping. This chapter concludes with our findings on common limitations and provides the rationale behind splitting the proposed pipeline into the following three chapters.

Chapter - 3 Sensor Fusion for Robotic Grasping summarizes sources of measurement errors of depth sensors, proposes a novel sensor noise model and a measurement fusion algorithm and evaluates the results with a public dataset.

Chapter - 4 Active Exploration for Robotic Manipulation utilizes the scene representation of the previous chapter to derive collision-free motion commands for the robot at a controller level in order to gain more information about the object.

Chapter - 5 Grasp Candidate Sampling and Evaluation utilizes the scene representation to sample grasping possibilities, evaluate their feasibility and approximate

their probability of success. The proposed solutions are combined into a closed-loop pipeline. The efficacy of the pipeline is then compared against state-of-the-art algorithms in simulated and real setups.

Chapter - 6 Prototype shows the proof-of-concept setup where we implemented the proposed algorithmic pipeline into a wheel-chair mounted hardware. We equipped the proposed algorithm with a graphical user interface and show the functionality in a proof-of-concept manner.

2. Early Approaches for Grasping in Six Dimensions

In this chapter, we discuss two of our early published approaches on grasping with six degrees of freedom (6-DOF) and the corresponding results. In the course of these studies, we recognized various challenges for the field of assistive robotics that can not be solved adequately with existing approaches. These findings ultimately led to the splitting of the holistic grasping approach into three subsystems that tackle their respective challenges, namely active controlling and exploration, probabilistic sensor fusion with reliability estimation and probabilistic grasping. Combined they enable reliable grasping in unstructured and unknown environments.

2.1 6-DOF Grasp Detection for Unknown Objects

2.1.1 Introduction

One of the most fundamental tasks in robotic manipulation is grasping. Several methods achieved remarkable results using reinforcement learning e.g. Pinto and Gupta (2016) and Kalashnikov et al. (2025) or convolutional neural networks (CNN) based on synthetic datasets e.g. Bousmalis et al. (2018). Recent results suggest training convolutional neural networks on human labelled datasets can produce even more reliable grasps across a wide range of different object types Lenz et al. (2015), Mahler et al. (2017), Morrison et al. (2018). These methods rely on 2.5D depth images as input due to practical reasons. The datasets can be labelled easier and the network structure is less complex than in the 3D case. Because of the 2.5D sensor data these methods restrict themselves to simple grasps along the sensor's z-axis. Whether a grasp is successful or not is always related to the interaction of the gripper with the objects surface. By only taking 2.5D images into consideration from a single point of view, significant information about the 3D contact points is lacking. It is very unlikely that the first point of view an object is seen from leads to a satisfactory grasp, especially using mobile robots.

To our best knowledge, there are no publicly available human labelled 3D dataset. We propose an algorithm that uses multiple 2.5D views to allow for a 3D surface representation of the scenery. This concept enables us to evaluate the predictions of state-of-the-art

grasp detection convolutional neural networks with geometric metrics. By considering multiple interpolated viewpoints, we are able to find new grasps candidates, which could otherwise not be found. Compared to other 6-DOF approaches, a significantly less computational power is required. The low computational needs of this approach enables us to run this algorithm on mobile platforms with satisfactory performance. Experiments showed that the increased number of considered grasp candidates and their evaluation based on information about the contact area lead to an improved grasp success rate.

2.1.2 Related Work

The perhaps most similar work is the approach of ten Pas, Gualtieri, Saenko, and Platt (2017). They utilize a point cloud that is fused from several perspectives and sample potential 6-DOF grasp poses randomly across the object's surface. For each grasp sample, only object volume within the closing area of the gripper is considered and represented as a cubic grid of size $m \times m \times m$. The volume is projected onto planes spanned by the axes of the gripper's reference frame. A height map of the observed volume, a height map of the averaged unobserved volume and averaged surface-normals are created for each projection. Hence, each random sample is represented as an image of size $m \times m \times 5$ and a convolutional neural network is utilized to estimate the corresponding chances of success. Due to the random nature of their sampling technique and the computationally intensive pre-processing steps for each sample, the approach requires a significant amount of computational power. Therefore, this approach may not be suitable for mobile applications.
Morrison, Corke, and Leitner (2019a) utilized a six layered convolutional neural network. They scale the incoming depth sensor data to a resolution of 300×300 and estimates grasp quality scores in a pixel-wise manner. Due to the uniform resolution of grasp candidates within the perspective, a prior sampling step is therefore unnecessary. This fact and the small number of parameters a result in low computational costs with typical inference times of 20 milliseconds. The price of this approach is that grasp directions are restricted to the axis perpendicular to the sensor's line of sight.
We build on the work of Morrison et al. (2019a) and propose to use multiple views, a ray-casting algorithm and geometric reasoning to enable six dimensional grasping with little computational effort designed for mobile robotic applications.

2.1.3 Problem Statement

The following chapter is largely based on work which was published as:

> Schaub and Schöttl (2020)
> *"6-DOF Grasp Detection for Unknown Objects"*,
> in 10th International Conference on Advanced Computer Information Technologies (ACIT), 2020

Assume a system with one or more depth sensors yielding depth images $D = \{d_0, ..., d_n\}$ taken from corresponding, known poses $P = \{p_0, ..., p_n\}$ where $p \in SE3$. Let $V \in R^{3 \times n}$ be the combined point cloud with respect to the robots base frame. Given a point cloud V obtained by from two or more perspectives, the problem is to identify a 6-DOF antipodal grasp configuration with regard to some object so that the grasp is reliable and collision-free.

Depending on the depth sensor's resolution, the fused point cloud V represents the discretized visible surface of the object. It potentially consists of several million points. Finding reliable pairs of contact points by brute force is not feasible, especially in the context of mobile robotics. in order to drastically reduces the search space, we propose to use an image-based *CNN*, .

Algorithm 1 Grasp Pose Detection

Input:
two or more depth images D, corresponding sensor poses P

Output:
a valid set of 6-Dof grasp poses G_p

1.) D' $\leftarrow$ preprocessDepthImages(D)
2.) V $\leftarrow$ fuseDepthImagesToPointCloud(D', P)
3.) M $\leftarrow$ convertToMesh(V)
4.) D_{int} $\leftarrow$ calcInterpolatedDepthImages(M, P_{int})
5.) S $\leftarrow$ estimateGraspQualityScores(D_{int})
6.) G $\leftarrow$ generateGraspCandidates(M, S, P_{int})
7.) G', S_g $\leftarrow$ filterAndScoreGraspCandidates(G, M)
8.) G_p $\leftarrow$ computeEndEffectorPoses(G')

The grasp pose detection follows the steps shown in algorithm 1. The pre-processing of the depth images involves non-linear, edge-preserving, and noise-reducing smoothing filter for Gaussian smoothing as well as inpainting in order to remove invalid values. Secondly, we use the known intrinsic and extrinsic parameters of all sensors to fuse the depth images into a single point cloud V. A fast surface reconstruction algorithm Holz and Behnke (2013) is then used to convert the fused point cloud to a mesh M. We uniformly sample virtual viewpoints between the sensor poses and employ a ray-casting algorithm to generate depth images. These images serve as an input to estimate a grasp quality score with a CNN for every visible 3D point. After rejecting geometric infeasible solutions, we generate 6-Dof grasp candidates. Finally, all candidates are weighted by their grasp scores S_g and the best is selected for execution. In the following sections, we discuss steps 4 – 8 of algorithm 1 in detail.

2.1.4 Creating Interpolated Depth Images

Each depth camera creates a depth image D, where each homogeneous pixel coordinate $(u, v, 1)$ stores a measurement that describes the distance d to the next obstacle along the line of sight. Given the projection matrix of the camera, the 3D coordinates (x, y, z) of the measurement d are found via

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{d}{f_x f_y} \begin{bmatrix} f_y & 0 & -c_x f_y \\ 0 & f_x & -c_y f_x \\ 0 & 0 & f_x f_y \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad , \tag{1}$$

where (c_x, c_y) refers to the camera's center in image coordinates and (f_x, f_y) represents the focal lengths in the respective direction. We assume that these parameters are constant across the whole procedure. This is not self-evident since Niedermayr and Wolfartsberger (2022) has shown that measurements can deviate significantly during the warm-up phase of a sensor. Equation 1 is applied to all pixels of the depth image in order to obtain the corresponding point cloud and the fusion V of all n point clouds, represented in homogeneous coordinates, is given by

$$V = \bigcup_{i=0}^{n} P_i V_i \ , \tag{2}$$

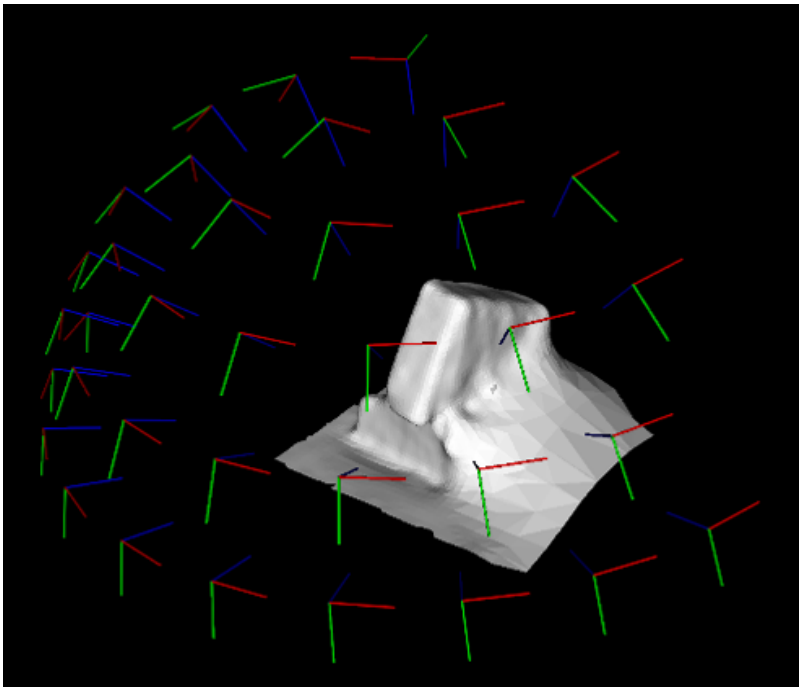

Figure 2: Each coordinate system represents a virtual camera. All virtual cameras are positioned on a semi-sphere pointing inwards. The object's mesh is in the center of the sphere.

where P_i represents the transformation matrix from the i-th sensor pose to the world frame. Elements of the point cloud represent infinitesimally small elements of $\mathbb{R}^3$ and therefore cannot be used to render an image from a different perspective. We employ the Ball-Pivoting algorithm (BPA) originally introduced by Bernardini, Mittleman, Rushmeier, Silva, and Taubin (1999) to circumvent this problem and create a surface representation (mesh) M from V. The resulting mesh consists of a finite number of triangles. The area between each of the three corners is thus represented as a continuous surface and which can be "hit" by view-rays from a different perspective.

We assume that the cameras frustums overlap to a certain degree and represent their poses in spherical coordinates where the coordinate center is the center of the object.

A constant angle threshold is applied to both angular components in both directions. We use this angle segment and a predefined radius to create a spherical surface section on which we sample virtual camera poses uniformly. An example of this sampling can be

seen in figure 2. All perspectives point towards the origin of the sphere and their horizontal axis is aligned with the ground plane. All virtual cameras share a single constant camera calibration matrix K that transforms points on the sensor plane to homogeneous image coordinates $(u,v,w)^T$,

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

For every virtual perspective, the object mesh is transformed into the corresponding camera frame. The intersection of a ray which is "shot" through a pixel (u,v) is given by

$$(u,v) = d\,K \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad \text{with} \quad d > 0 \quad . \tag{4}$$

We then utilize a custom ray-casting algorithm and the PCL library (Rusu & Cousins, 2011), to compute the distance d that corresponds to the intersection of the ray with the transformed mesh for every pixel and populate the interpolated depth images with d. Image positions where no intersection can be computed are inpained using the OpenCV library (Bradski, 2019). It is worth noting that the predefined parameters e.g. the semisphere radius, the camera calibration matrix as well as the image dimensions were chosen to match those of the Cornell grasping dataset (Lenz et al., 2015).

2.1.5 Estimating Grasp Quality

We employ the CNN proposed by Morrison et al. (2019a) to approximate the grasp set G for interpolated depth images D_{int},

$$G(D_{i,int}) = (Q,A) \quad , \tag{5}$$

where Q and A denote images that represent the approximated grasp quality and angle around the axis perpendicular to the image plane for each pixel. An example of Q can be seen in third column of figure 3. We use the network to obtain an estimated quality distribution of grasps parallel to the viewpoint's z-axis. The network was trained using the Cornell grasping dataset (Lenz et al., 2015). It contains 885 real RGB-D images of

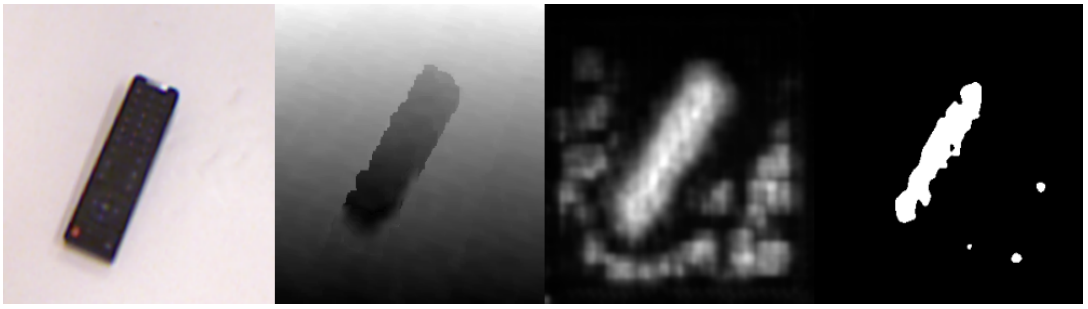

Figure 3: *(Left to right).* Color image of a remote control from the Cornell dataset. The corresponding depth image. The quality output Q of the GGCNN network. Q after filtering out everything but the upper 5% quantile.

Figure 4: Example of the Cornell dataset (Lenz et al., 2015). The left image shows the (cropped) color image of a stapler and human-labelled grasps. The corresponding normalized depth image is on the left. The blue lines represent the contact areas of a two-finger gripper.

objects, with 5110 human-labelled grasps. The images show objects on a planar background and labeled grasps are represented by oriented rectangles as popularized by Jiang, Moseson, and Saxena (2011). An example is shown in figure 4, where the left image shows the RGB-portion of the example together with the grasp labels and the corresponding depth image, represented as grayscale, is on the left.

2.1.6 Generating Grasp Candidates

We discretize the volume under consideration into cubes of equal size and only consider cubes that contain at least one vertex of the mesh M. This uniform surface representation is denoted by S and each voxel element stores its position as well as its surface normal. For

each voxel up to n grasp candidates g are created, where n is the number of interpolated depth images the voxel was visible in,

$$g = \left(v_p, v_s, v_a, v_q\right) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \quad , \tag{6}$$

where v_p is the voxel position, v_s is the unit vector defining the direction the voxel was seen from, v_a the predicted grasp axis perpendicular to v_s and v_q the estimated grasp quality. For the sake of computational time and in order to reject unreliable grasp candidates right away we discard all candidates whose quality estimation v_q is not within the upper fifth percentile of their respective Q. An exemplary visualization of this filtering can be seen in the last column of figure 3, where a lot of questionable grasp candidates on the table are filtered out.

In order for a grasp candidate g to be valid there has to exist at least one pair of antipodal contact points. These points define the points the parallel gripper's fingertips touch the object's surface. For each grasp candidate g, we define a plane E,

$$E(g) = v_p + a\,v_s + b\,v_a, \quad a,b \in \mathbb{R} \tag{7}$$

Let $C(g)$ be the subset of points s on the object's surface whose distance j to $E(g)$ is less than a predefined tolerance ε,

$$C(g) = \{s \in S \ : \ j(s, E(g)) < \varepsilon\} \quad . \tag{8}$$

$C_l(g)$ and $C_r(g)$ denote subsets of $C(g)$ on antipodal sides of the objects surface and represent possible finger contact locations for the grasp candidate g,

$$\begin{aligned} C_l(g) &= \{C(g) \ : \ (s-p)\cdot v_a < 0\} \\ C_r(g) &= \{C(g) \ : \ (s-p)\cdot v_a > 0\} \quad . \end{aligned}$$

We sort all elements c_l, c_r according to their depth along the grasp direction v_s. Elements where the depth exceeds maximum grasp depth (i. e. the length of the gripper fingers) are rejected. We compare all possible combinations of elements within the valid depth range. Figure 5 visualizes this step. We then check all possible pairs of antipodal contact points if all following four conditions are met:

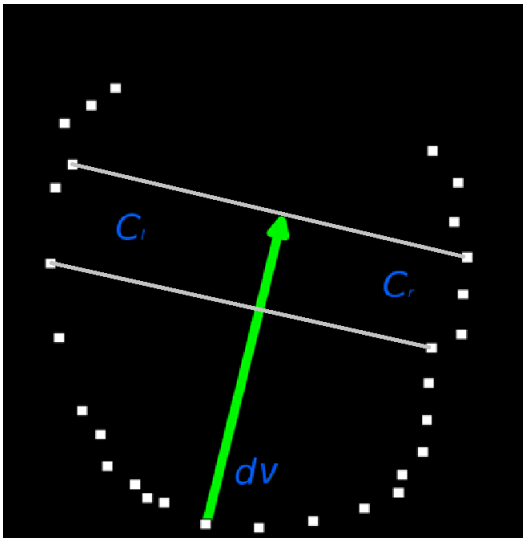

Figure 5: Slice through the object's surface points. The algorithm tries to match points on the left side C_l with points on the right side C_r. In this case, two valid pairs of antipodal contact points were found (grey line).

Condition 1: The distance of c_l to c_r is smaller than the maximum gripper width,

$$\|c_l - c_r\|_2 < w_m \quad . \tag{9}$$

Condition 2: The difference of distances of the contact points to the grasp direction line v_s is smaller than a predefined tolerance $\delta > 0$,

$$\left\|(c_l - v_p) \cdot v_s \;-\; (c_r - v_p) \cdot v_s\right\|_2 < \delta \quad . \tag{10}$$

Condition 3: The grasp configuration needs to satisfy the squeezing force closure condition (I.-M. Chen & Burdick, 1993). Force closure has been proven as an effective method to evaluate grasp candidates (e.g. H. Fang, Wang, Gou, & Lu, 2020; V.-D. Nguyen, 1988; ten Pas et al., 2017). This binary metric is the product of geometric considerations under the assumption of the Coulomb friction model. We assume a constant friction coefficient f across the object's surface. Generally, f is a structural property that depends on both contacting materials. It is an experimen-

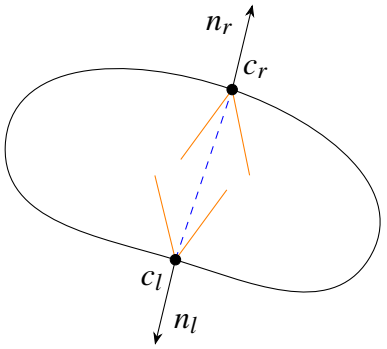

Figure 6: The grasp configuration is in force closure if the vector that connects the contact points (shown in dashed blue) is within both friction cones (shown in orange). The cone's apex angle is defined by the assumed friction coefficient and its rotation axis equal to the inverse surface normal.

tally measured coefficients and $\theta_f = \arctan(f)$ is called by the critical ramp angle. Consider an object on a ramp. If the angle is smaller than θ_f, the object remains in place, and it starts sliding if θ_f is exceeded. This condition is shown schematically in figure 6.

Let $n(c)$ be the surface normal at point a potential contact point c. A squeezing antipodal two-finger grasp is in force closure if the pair of contact points satisfies the squeezing force closure condition (I.-M. Chen & Burdick, 1993),

$$n(c_l) \cdot \frac{c_l - c_r}{\|c_l - c_r\|_2} > \cos\theta_f \quad \wedge \quad n(c_r) \cdot \frac{c_r - c_l}{\|c_l - c_r\|_2} > \cos\theta_f \quad , \tag{11}$$

where θ_f is assumed to be constant across the surface of the object. It is worth noting that Mahler (2024) evaluated several robust analytic grasp quality metrics in a physical system and concluded that they underestimate the probability of success by a significant margin and therefore under-perform in terms of classification accuracy. This is due to dynamic effects that are not modeled by the metrics, e.g. if an object is pushed into alignment by the grasp movement. However, they show a high precision (ratio of predicted successful grasps vs. actually successful grasp) which means that a grasp with high predicted quality is likely to be successful on a physical system. Through our experiments we came to the conclusion that the grasping solution space is highly redundant, i.e. most household objects can be grasped in

numerous ways and seldom one possibility is clearly superior to all others. We argue that due to this over-determination, the underestimation of the success probability is not necessarily bad as long as highly ranked grasp candidates are reliable. Force closure is comparatively easy to implement and one of the few metrics that can be calculated for an unknown object with the limited amount of information of an online method, since the object is not entirely known at any point in time and no physical parameters such as center of gravity or mass or can be determined. Although the friction coefficient is not known either, we found that assuming a low coefficient usually leads to a satisfying performance.

Condition 4: The body of the gripper M_g must not be in collision with the object mesh (including the ground plane) M that is calculated in step 3 of algorithm 1. Let $M_g(g)$ denote the gripper's occupied volume at the gripper configuration g, then

$$M_g(g) \cap M = \varnothing \tag{12}$$

needs to be satisfied. In order to evaluate equation (12) we employ the Fast Collision Library (FCL) (J. Pan, Chitta, & Manocha, 2012) within the MoveIt pipeline (Coleman, Sucan, Chitta, & Correll, 2011).

We score a remaining candidate g that satisfies all conditions according to its inverse grasp energy (Y. Chen & Medioni, 1991) and the quality $q(g)$ as estimated by the GGCNN multiplied with an empirically determined weighting constant w,

$$G_s\big(c_l(g), c_r(g)\big) = \frac{1}{E_g(c_l, c_r)} + w \; q(g) \quad , \tag{13}$$

with

$$E_g(c_l, c_r) = \frac{1}{2}\kappa \|c_l - c_r\|_2 \quad . \tag{14}$$

The grasp energy E_g can be interpreted as the stored energy of a spring with spring constant κ between two antipodal contact points c_l and c_r. The highest scoring grasp candidate is selected for execution. In case no feasible inverse kinematics solution can be found, the next best candidate is selected.

2.1.7 Discussion

The presented method produces reliable grasps for objects with adversarial geometries due to the combination of the semantic information coded by the human-labelled dataset and the geometric analysis of the object. Nonetheless, we are able to identify potential for improvement with regard to the overall goal of assistive robotics.
This approach relies on the benign placement of the cameras which cannot always be assumed. An approach suitable for the context of assistive robotics must take advantage of the mobility of the robot arm to utilize information from multiple perspectives. In addition, the processing of the available data-stream in the sense of a video would also mitigate the sensor noise problems mentioned above, since the estimation variance decreases with increasing number of measurements.
Second, depending on the situation there is a significant number of outliers present in the quality output of the employed CNN. As can be seen in the third column of figure 3, where some outliers are even within the upper 5% quantile (column four). This behavior worsens the further the input images are outside the trained population of the Jaquard dataset (Lenz et al., 2015), which exclusively shows the benign top-down situation. Although most outliers are rejected by the proposed conditions in section 2.1.6, finding an appropriate weighting parameter w for the approximation of grasp quality (equation 13) is non-trivial and depends on the experimental setup.
The proposed algorithm relies on the surface normals, which can be understood as the spacial derivative of the estimated surface points. Therefore, it is susceptible to sensor noise and the proposed fusion of point clouds relies on pre-defined sensor-settings, sensor-internal as well as custom filtering of the data. All corresponding parameters are determined empirically and are inevitably tailored to the experimental setup to some degree. A sensible further approach to obtain estimates with higher reliability is to increase the sample size. Considering that sensors operate at 30 Hz, the logical next step is to merge multiple measurements into a global display.

2.2 6-DOF Grasp Detection for Unknown Objects Using Surface Reconstruction

The following chapter is largely based on work that was published as:

Schaub, Schöttl, and Hoh (2021)
"6-DOF Grasp Detection for Unknown Objects Using Surface Reconstruction",
in 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021

The chapter can be regarded as further development on (Schaub & Schöttl, 2020), that was presented in the previous chapter.

2.2.1 Introduction

The human-centred environment of assistive robotics introduces new challenges which are not fully addressed by current solutions designed for industrial, structured settings. Current approaches usually use only a single sensor image to determine the most promising grasp candidate and therefore heavily rely on a benign initial situation. However, the environment of a service robot is cluttered and unstructured and a benign situation cannot be presumed. The performance grasp-synthesis algorithms that only utilize a single image can be severely impacted by (partial) occlusions, sensor noise or missing data caused by lightning conditions, or reflective surfaces.
In this work, we build upon our previous work, which is discussed in 2.1. We mount a depth sensor on the end effector of a robotic arm and scan the object from multiple views. We propose to use a sensor fusion algorithm to gain a volumetric representation of the object and tackle typical problems like sensor noise, partial occlusion and inaccurate forward kinematics. This approach allows us to use predictions of state-of-the-art grasp detection CNNs to effectively limit the solution space and utilize well-established analytical metrics to find stable grasps. The lightweight nature of our approach makes it suitable for mobile or service robotics with their typical hardware restrictions.

2.2.2 Related work

Although 4-DOF, top-down approaches achieve remarkable results, they are fundamentally restricted in their action space. Various attempts have been made to use multiple viewpoints and combine the best of both worlds. Gregorio, Tombari, and di Stefano (2016) use a RGB-D camera mounted on a robot arm. They assume perfectly known

sensor poses at all times and employ a sensor fusion algorithm to reconstruct the object's surface from multiple viewpoints. A RANSAC-algorithm is used to segment the object from the background. Grasps are found by considering only the contours of the projection of the object onto the three planes orthogonal the principle axes of the object. This procedure effectively restricts the solution space for grasps to the three orthogonal directions.
Lehnert, Sa, McCool, Upcroft, and Perez (2016) employ a similar eye-in-hand setup and fuse multiple colored point clouds while traversing a predefined trajectory. They use color classifiers to find a sweet pepper within the 3D representation and find the orthogonal front, top and side planes by fitting a super-ellipsoid in the corresponding volume. The object is them "grasped" by aligning a suction gripper with the front-axis.
Breyer, Chung, et al. (2022) use a discrete 3D representation of the object and its surrounding volume. They propose a convolutional network to estimate a 6-DOF grasp and corresponding quality for each grid cell. The best grasp is then chosen via quality optimization over all cells. The proposed grid resolution of $40 \times 40 \times 40$ is relatively coarse, which was probably chosen to decrease computational and memory requirements and allow for real-time capability.
Avigal, Paradis, and Zhang (2015) argue that depth cameras are inaccessible and too expensive and use color images instead. They use five simulated, static RGB-cameras and propose to use a *Learnt Stereo Machine* to estimate corresponding depth images. Similar to our approach, the generative grasping CNN proposed by Morrison et al. (2019a) is then utilized to evaluate all five depth images. The best 4-DOF grasp is determined by maximum evaluation over the corresponding five quality outputs. Unfortunately, their approach was only evaluated in a simulated setup which might not accurately reflect the intricacies of a physical, real platform.

2.2.3 On the Choice of the Volume Representation

Many robotic applications fuse sensor measurements to obtain a representation of the environment, which are in turn used to guide the robots actions. Many types of environmental representations exist and are used in different robot application domains. A common feature among them is the requirement for online processing, meaning that measurements can be fused in a real-time manner enabling closed loop control of the robot. In the following, we present three of these representations.

2.2.3.1 Occupancy Grids

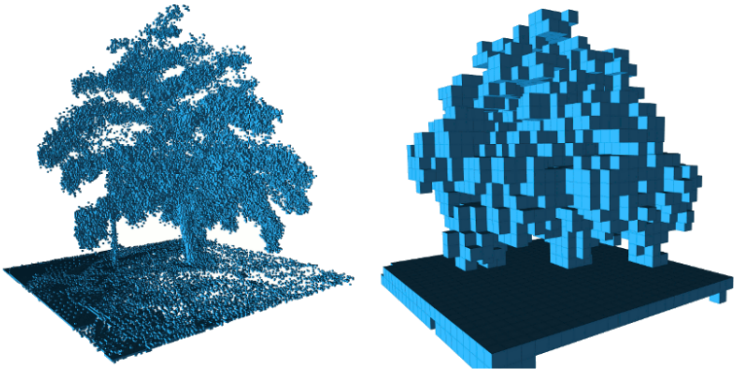

Figure 7: Example of a 3d *occupancy grid* of a tree at resolutions 0.08 m and 0.64, taken from (Hornung et al., 2013)

Occupancy Grids are a volumetric representation of the scene and were first proposed by Moravec and Elfes (1985). They divide the three-dimensional space into cubes of equal size, where the spatial resolution and the size of the considered volume are preset parameters. Each cube represents the state of the volume it encloses, where their state can be either *occupied* or *free*. Often probabilistic methods are used to estimate the posterior probability of a cube being either *occupied* or *free* given a set of corresponding measurements and a user defined initial prior probability which is often set to 0.5. Their most common application is mapping applications (e.g. Ferri, Tesei, Stinco, & LePage, 2019; Rogers, Eshaghi, Nejat, & Benhabib, 2023; Wijaya, Purnomo, Utomo, & Anandito, 2019)) and they are integrated in the *MoveIt* perception pipeline (Hornung et al., 2013). Interesting for service-robot applications is the approach of Dengler et al. (2023). In an effort to gather information about heavily occluded objects in shelves they propose to use a robot arm with eye-in-hand system. They employ a top-view two-dimensional occlusion map and use it to either compute promising next-best-views via entropy considerations or move occluding objects out of the viewing area.

The frequent use of *occupancy grids* in mapping is due to the fact that this representation is suitable for path planning where the knowledge that space is *free* is just as valuable as known obstacles and the binary distinction and often low resolution is sufficient to avoid

collisions.
Occupancy grids are sometimes used in grasping applications (e.g. Varley, DeChant, Richardson, Ruales, & Allen, 2017; Yan et al., 2018). However, in both cases no measurement fusion is applied, but instead a convolutional neural network is used to estimate the three-dimensional *occupancy grid* of the object from a single point cloud. This estimated map is then used to predict possible grasping candidates.
Because they division of space into *free* and *occupied* grid cells, following applications are inherently limited to the resolution of the grid. This means that the complexity of some objects may be difficult to represent since the memory requirement scales cubically with the resolution of the grid. Another disadvantage is that the traditional occupancy description of a grid cell does not allow specifying the variance of the estimate. A surface section that has been measured many times with strongly fluctuating results cannot be distinguished from a surface section that has been measured only a small number of times.

2.2.3.2 Surfel Clouds

The term surfel is an abbreviation for surface voxel or element in the volume rendering and discrete topology literature (Pfister, Zwicker, van Baar, & Gross, 2007). Similar to point clouds, *surfel clouds* are a data structure where each element stores the position, surface normal vector and sometimes the radius and color of a disc, which is called *surfel*. *Surfel clouds* are flexible and can be updated very efficiently. Since they only represent surface elements, they require significantly less memory compared to volumetric representations, which is especially useful for graphics card implementations.
As an example, Whelan, Leutenegger, Salas-Moreno, Glocker, and Davison (2015), propose a self localization and mapping algorithm which is capable to be run on GPU. Due to their restriction to the surface, it is difficult to retrospectively distinguish known from unknown, and free from occupied space. Monica and Aleotti (2018) proposes an active exploration approach using an eye-in-hand system. They represent the environment using a *surfel cloud* but have to resort to a *truncated signed distance function* to differentiate between known and unknown space in order to identify promising next-best-view in an exploration application.
Surfel clouds are loosely managed with no topological connections and often require addi-

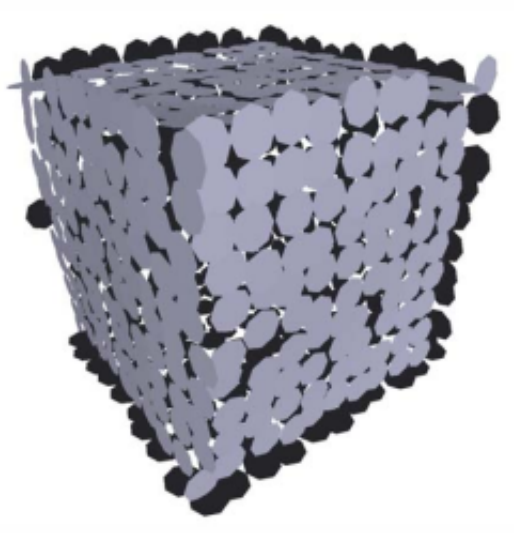

Figure 8: Example of a *surfel* model of a cube from Dahl et al. (2010)

tional modules for efficient indexing and rendering (Botsch & Kobbelt, 2003). Additionally, they are relatively prone to noisy sensor input (Dong, Wang, Wang, & Zha, 2018). Applications of *surfel clouds* that are related to robotic grasping include the work of Holz, Topalidou-Kyniazopoulou, Stückler, and Behnke (2015) and Stückler and Behnke (2012). Both approaches use a multi-resolution surfel map to match known object models with incoming sensor data. For grasp selection Holz et al. (2015) rely on a set of predefined grasps whereas Stückler and Behnke (2012) use geometric reasoning based on the estimated principal axes of the object.

2.2.3.3 Truncated Signed Distance Function

Similar to occupancy grids the *truncated signed distance function* (TSDF) is a volumetric representation of the environment. The signed distance function describes the distance of any point in space to the nearest point on a given surface. The distance is positive if the point is outside of obstacles and negative if within. In most practical applications a predefined volume discretized uniformly into cubes of equal size. A schematic two-dimensional representation of the TSDF grid is shown in 9 where the object surface boundary is depicted via the dark red line. Cells outside the object receive positive distances and are depicted in green, whereas grid cells inside the object receive negative distance measure-

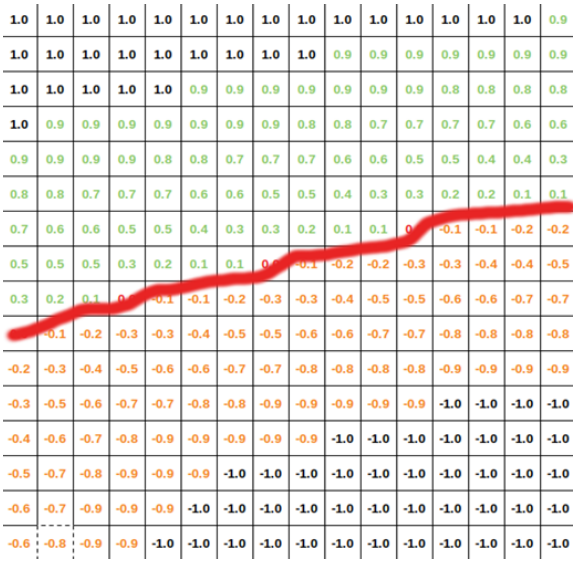

Figure 9: Example of a two-dimensional TSDF-Grid.

ments, which are depicted in red. Distances are truncated to a constant. In most applications the distances are scaled to this constant. Hence, the distance estimations are often in the range $[-1, 1]$.

Many surface reconstruction approaches build upon the work of Newcombe et al. (2011), which is to our knowledge the first real-time capable GPU implementation of a TSDF. Due to the continuous nature of the function, its readily available topology connections and the implicit de-noising effect of the weighed average, the TSDF is widely used in robotic applications, such as self localization and mapping (SLAM) applications, (e.g. Fehr et al., 2017), (Eisoldt et al., 2022), object tracking (Grinvald, Tombari, Siegwart, & Nieto, 2021) and grasping (Gregorio et al., 2016; Zhao, Yu, Wu, & Zhang, 2024). Taking signed distances as random variables, Newcombe et al. (2011) suggest that the weighted average can be regarded as the maximum likelihood estimation of a Gaussian distribution. However, the weighted average does not consider the varying sensor conditions and is very sensitive against outliers. Although the weight is sometimes treated as a measure of confidence of the corresponding signed distance function (SDF) - estimate (e.g. F. Li, Du, & Liu, 2016), it does not reflect the underlying distribution and hence the estimated quality can hardly be used for subsequent algorithms and a number of probabilistic ap-

proaches were presented.

A number of approaches were presented which take these changing sensor conditions into account by weighting measurements according to probabilistic considerations. Only some of which are addressed here. Dong et al. (2018) developed a framework to allow for varying variances of measurements. They employ a sensor model originally presented by Vogiatzis and Hernández (2011), which is defined as mixture of a Gaussian and a uniform distribution. In our experiments, we were unable to find data that supports the assumption of an additional uniformly distributed noise component. J. Yang, Li, and Waslander (2021) employ the disparity map of an active stereo camera to calculate a photometric confidence score and local 3D-features of the incoming point cloud to estimate the geometric uncertainty. The authors use a training-based approach to map the photometric confidence score to a pixel-wise inlier probability and perform probabilistic updates of the volume frame by frame. When working in an inconsistent environment however, it can be difficult to acquire the required training data. Saulnier, Atanasov, Pappas, and Kumar (2020) propose an active exploration algorithm and use a mobile robot equipped with a 2D range sensor. The environment is represented as a two-dimensional truncated signed distance field where each surface element is modeled to be independently Gaussian distributed. New measurements are integrated into the map via Kalman filter update equations, where the expected sensor noise variance is modelled to be proportional to the measured depth to the power of four. They show that various simulated environments can be actively reconstructed by computing the Shannon mutual information over the TSDF, but do not provide a measure of the reconstruction accuracy and have not tested their algorithm under real conditions and the varying noise behavior that goes along with it.

The TSDF offers decisive advantages for our project. The sub-grid resolution allows for a more accurate determination of the surface gradients compared to other volumetric approaches. Secondly, the distinction between unknown, free and occupied volumes is given by the implicit topology and does not require any further computational steps which is very useful for e.g. collision avoidance and grasp filtering. Its main drawbacks, namely its lack of flexibility, comparatively huge memory consumption can be overlooked due to the strong performance improvements of graphics cards in the past years. In addition, these disadvantages are hardly noticeable especially when only small volumes need to be considered, as in the case of robotic grasping. Hence, we opt for the TSDF representation.

2.2.4 Sensor Fusion

In this work we mount a 3D-sensor on the robot arm's end effector and build a 3D representation of the considered volume. We move the sensor in a circular trajectory above the object and fuse all resulting depth measurements into a TSDF-volume representation, where we use an implementation [3] of the algorithm that was originally proposed by Newcombe et al. (2011). We consider the problem of finding collision-free, parallel-jaw grasps for unknown objects within the robot's workspace.
The surface reconstruction is used to consider grasps from multiple positions and orientations and to provide a robust quality metric. Our goal is to find a lists of tuples (g,q), where g is a 6-DOF the grasp configurations and q the corresponding quality. We use a state-of-the-art CNN to vastly reduce the solution space and employ well-established metrics to calculate q for each g. An overview of the proposed sensor fusion pipeline is given in figure 10. The fusion of measurements be understood as a simultaneous localiza-

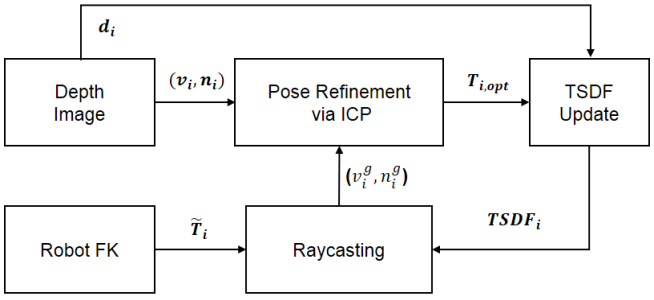

Figure 10: System flow of the surface reconstruction algorithm.

tion and mapping (SLAM) problem where even small pose estimation errors (especially the orientation) of the sensor could have a large effect on the precision of the map. We evaluate grasp possibilities in a point-wise manner and therefore an insufficient map inevitably leads to diminished grasping successes rates. Hence, the accuracy of the sensor pose estimation ${}^{bf}T_{cam}$ with respect to the robot's base frame is of high priority. It can be described by

$$ {}^{bf}T_{cam} = {}^{bf}T_{eef} \ {}^{eef}T_{cam} \ , \tag{15}$$

[3] https://github.com/AustinDeric/yak

where the transformation from the robots base frame to the end effector ${}^{bf}T_{eef}$ can be conveniently retrieved through the known forward kinematics of the robot. The pose of the camera in the end effector's frame ${}^{eef}T_{cam}$ is approximated during an offline calibration step where we use an implementation [4] of the algorithm originally proposed by Tsai and Lenz (1989). An auxiliary marker is placed in the workspace of the robot and photos are taken from different perspectives. The known motion of the robot arm is then utilized to find the ${}^{eef}T_{cam}$ that minimizes the pose difference of the marker over all the perspectives. Despite multiple calibrations of camera intrinsic and extrinsic, we found that the 3D point cloud of a static scene experiences small motions that depend on the trajectory of the sensor. These displacements are not erratic but systematic. Currently, we can only explain them by small errors in the forward kinematics and therefore the end-effector pose ${}^{bf}T_{eef}$. Often robots that are used in AAL (Ambient Assisted Living) applications are manually operated. Therefore, the user performs the task of the controller and is responsible for the accuracy of the movement. Hence, highly precise forward kinematics might not be the main focus of during the development of the arm. For instance, Iturralde, Kinoshita, and Bock (2019) report positional errors in the order of several centimeters for the *Kinova Jaco*, which we use for the prototypical construction in the second to last section 6.
Robot arms that are designed for industrial applications often report repeatability errors in the sub-millimeter range in accordance with ISO 9283 9283:1998(E) (2003). However, information about the accuracy of the forward kinematics and pose deviations is sparser compared to the repeatability error. This might be related to the typical task of the industrial context where a single trajectory is taught once and then must be executed repeatedly with a high degree of reliability. In their experiments, Žlajpah and Petrič (2022) measured positional errors of up to 1.5 cm for the *Franka Panda* which is the same robot arm that we use to evaluate our approach.
We therefore refine the available forward kinematics of the robot with a point-to-plane iterative closest point (ICP) algorithm originally proposed by Low (2021). An overview of the proposed pipeline is provided in figure 10. Let P_s be the current point cloud measurement represented in the sensor frame and let P_g be the point cloud created via ray casting the current scene reconstruction. For each frame we want to find the transformation $T_{i,opt}$

[4] `https://github.com/IFL-CAMP/easy_handeye`

that minimizes the point-to-plane error metric and overlaps both clouds,

$$T_{opt} = \underset{T}{\operatorname{argmin}} \sum_{u} \left\| (T \cdot p_s(c) - p_g(c)) \cdot n_g(c) \right\|^2 \quad , \tag{16}$$

where c represents the index of point-wise correspondences between $p_s = P_s(c)$ and $p_g = P_g(c)$ that are found via projective, pixel-to-pixel association. n_g is the surface normal at p_g and is found by examining the neighboring gird elements of the TSDF. Note that all correspondences where either the Euclidean distance $\left\| p_s - p_g \right\|$ or the angle between their normals $\angle(n_g, n_s)$ exceeds a predefined threshold are discarded.
The minimization of equation 16 is linearized using the small angle approximation (Low, 2021) to allow for real-time capabilities

$$T_{opt}(\alpha, \beta, \gamma, t_x, t_y, t_z) \approx \begin{bmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad . \tag{17}$$

By arranging the $\left(\hat{T} \cdot p_s(u) - p_g(u)\right)$ into a linear system $Ax = b$, where $x = (\alpha, \beta, \gamma, t_x, t_y, t_z)^T$, the normal equation $A^T Ax = A^T b$ is used to solve the linear least squares problem for an optimal x.
Although this approach does not eliminate the problem of inaccurate sensor poses, we are able to calibrate the initial pose and reference subsequent frames to this "accurate" pose in a frame-to-map manner. We found that the residual $r = \|Ax - b\|_2^2$ typically converges after a single iteration. In our setup translational refinements up to 4 mm and significant visual improvements of the reconstruction were observed.

2.2.5 Finding Grasp Possibilities

We move the depth sensor, mounted on the robots end-effector, in a circular trajectory above the object and fuse all incoming sensor frames into a unified TSDF. Depending on the chosen TSDF resolution the reconstructed surface may consist of several million vertex points. Evaluation of a 6-DOF quality function by brute force is time-consuming, especially in the context of mobile robotics.
Depth images are rendered from multiple views that are positioned on a semi-sphere cen-

tered around the object and use the CNN proposed by Morrison et al. (2019a) in order to vastly reduce the search space for grasp candidates. The measurements taken along the trajectory show the object from all sides which leads to a complete reconstruction of the object. Hence, we are able to utilize the entire semi-spheres surface and evaluate all grasp approach directions.

We discretize the upper polar angle range as well as the azimuth angle range and leave the radius fixed at a value that corresponds to the typical distance in the Jacquard grasping dataset. Each sampling point represents a view, where the view direction points toward the spheres center and the horizontal axis parallel to the ground plane. We render depth images from all the views but due to the TSDF implementation, we are able to use a build-in GPU-accelerated ray-casting algorithm which enables the generation of depth images significantly more efficient than the previous custom implementation. Similar to our previous work, presented in section 2.1, we utilize the CNN of Morrison et al. (2019a) to generate pixel-wise quality estimations in order to reduce the search space for grasps. We drop all quality-estimations that are not within the upper 10 % quantile and find face pixel correspondences via projective association. The view direction as well as the angle approximation of the highest scoring quality prediction over all views is stored for each face. Figure 11 shows a visualization of a typical result of this process, where the faces are colored according to the highest quality of adjacent vertices.

2.2.6 Rejection and Evaluation of Grasp Poses

Let a grasp candidate by denoted by $g = (p, v, \alpha) \in \mathbb{R}^3 \times \mathbb{R}^3 \times [0, \pi)$, where p is the center point of the center point of the face, v the highest scoring direction the face was seen from and α the predicted angle. The rejection for all g follows the same for geometric feasibility of grasp candidates follows the same first three steps listed in section 2.1.6. Although the implementation has been significantly improved compared to our previous work in section 2.1.6, the identification of infeasible grasp candidates follows the same logic and is not presented again. We represent the occupied volume of the end effector simplified by a box and discretize it into the set of uniformly distributed 3D coordinates $p_{eef} \in P_{eef}$ with a step size equal to the resolution of the TSDF r_t. For each grasp candidate, a corresponding affine transformation matrix $T(g)$ is used transform the end-effector points into the coordinate frame of the TSDF. The collision can then be simplified to a computationally

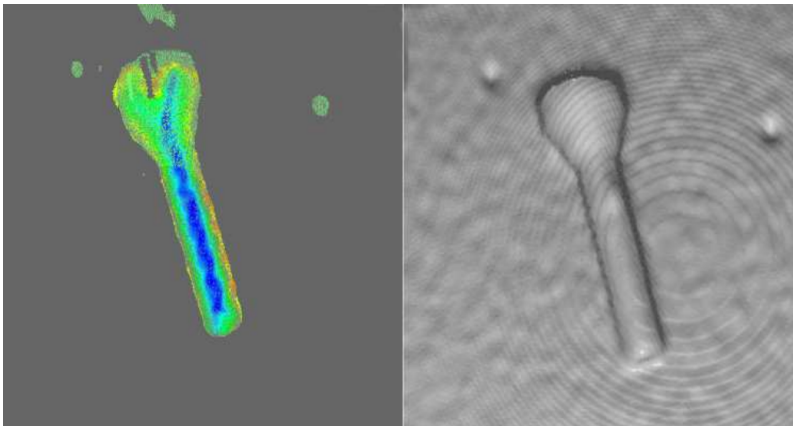

Figure 11: The predicted back-projected grasp quality of a flashlight. Highest scoring candidates are depicted in blue, low scoring candidates in red. Candidates not within the 10%-quantile are uncolored (grey). The right side shows the corresponding rendering of the reconstruction from the top view for reference.

efficient look-up table operation that is denoted by

$$0 \;\leq\; \underset{p_{eef}}{\operatorname{argmin}}\; \mathrm{TSDF}\left(\mathrm{Int}\left(\frac{1}{r_t}\, T(g)\;\; P_{eef}\right)\right) \quad , \tag{18}$$

where $\mathrm{Int}(\cdot)$ denotes rounding to the nearest integer in order to use the points as indices for the lookup operation $\mathrm{TSDF}(\,\cdot\,)$. The equation can be conveniently written in matrix form for all grasp candidates and evaluated in a highly efficient manner. This evaluation reduces the number of high quality but geometrically impossible grasps to consider during the following path planning and is schematically depicted in figure where P_{eef} is represented by the corresponding box.

If a candidate meets all four conditions (see section 2.1.6), it is considered valid and its grasp score is computed. Due to the issues mentioned in the previous discussion section 2.1.7, we opt for a different grasp quality metric. As can be seen in figure 5, we identify the two sets C_l, C_r of possible contact points on antipodal sides of the object. Both sets are sorted according to their depth along the grasp direction and elements outside the reachable depth of the gripper were rejected. Let (c_l, c_r) be an antipodal pair of contact points and (n_l, n_r) be their corresponding normal vectors. Inspired by H. Fang et al. (2020) who describe grasp robustness as the lowest friction coefficient that still leads to a valid

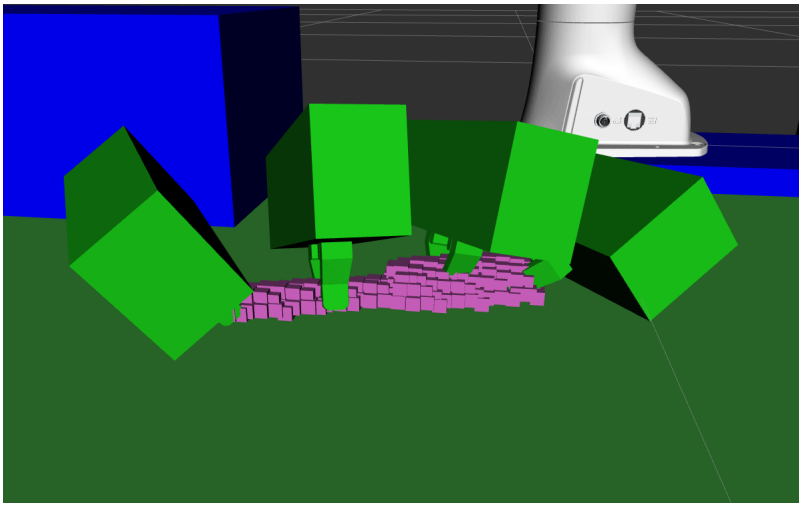

Figure 12: A selected set of grasp candidates for a screwdriver is visualized. The bounding box representation of the end effector is used to eliminate geometrically infeasible solutions.

grasp, we evaluate the score $s(c_l, c_r)$ of the pair with

$$s(c_l, c_r) = \frac{1}{2}\left(n_l \cdot \frac{c_l - c_r}{\|c_l - c_r\|_2} + n_r \cdot \frac{c_r - c_l}{\|c_l - c_r\|_2}\right) \tag{19}$$

We partition C_l and C_r according to their depth along the grasping direction and use a window size equal to the finger contact length with stride equal to half the window size. Each cluster D_k is a set of all possible pairings of $c_{l,k}$ and $c_{r,k}$ within the depth window k. We found that considering $\|D_k\|$ reduces the impact of sensor noise and provides an additional safety margin. We model the quality of a cluster $q(D_k)$ to be the average pairwise score weighted by an exponential function,

$$q(D_k) = 1 - e^{-\sigma \|D_k\|} \quad \frac{1}{\|D_k\|} \sum_{(c_{l,k}, c_{r,k}) \in D_k} s\left(c_{l,k}, c_{r,k}\right) \quad , \tag{20}$$

where σ is a positive, empirically determined constant that depends on the resolution and the contact area one wants to consider. The best scoring cluster across all grasp candidates g is used to create a grasp pose. We use a standard path planning algorithm to calculate a collision-free trajectory to this pose. If no kinematically feasible plan can be found,

Figure 13: The mesh that were use for the experiments were taken from the publically available dataset of Mahler et al. (2016).

we discard all clusters of the corresponding grasp candidate use the second best scoring cluster.

2.2.7 Experiments and Conclusion

We printed 13 objects (see Fig. 13) from a publicly available dataset (Mahler et al., 2016) and recorded our results to evaluate our pipeline. Each object was placed within a predefined volume using various positions and orientations. The volume consisted of 320 × 320 × 320 voxels of size 1mm. Rendering 32 depth images, predicting qualities and computing a sorted set of grasps took 4.3 sec on average on an embedded system (NVidia Jetson AGX Xavier) excluding the exploration and the path computation. Each object was scanned and grasped 30 times. A grasp was considered a success if the object could be picked up and placed at a predefined location. Table 1 summarizes the results of our experiments. The overall success rate over all experiments turned out to be 92.3%.
We observed that many failures cases occurred when the reconstructed edges were smeared along the line of sight. This was caused by perspective depth discontinuities where windowing effects smear the foreground and background depths together (Kadambi, Bhandari, & Raskar, 2014). This indicates that the implicit assumption of identically distributed

Table 1: Results of real grasping experiments

	success	failure	
screwdriver	29	1	96.7 %
vase	28	2	93.3 %
small cup	26	4	86.7 %
medium cup	27	3	90.0 %
apple	27	3	90.0 %
dwarf	27	3	90.0 %
bottle	29	1	96.7 %
lock	28	2	93.3 %
can	29	1	96.7 %
shoe	30	0	100.0 %
stapler	27	3	90.0 %
flashlight	26	4	86.7 %
guitar	27	3	90.0 %

sensor noise of the traditional TSDF algorithm is not applicable for our environment or sensor. A second observation is the strong influence of extraneous illumination on the reconstruction and thus the outcome of the grasping test series. In the beginning, the reconstruction of objects was often strongly distorted in some regions until we realized that one particular ceiling light lead to partially overexposed infrared images from one side. This overexposure occurred only in the infrared region, was invisible to the naked eye and led to strong noise effects. Despite our attempts to adjust gain, exposure time and laser power, we were unable to find a setting that could handle the entire illumination range and hence had to switch off the particular ceiling light after which the reconstructions were visibly cleaner. These two observations led to the idea that the varying measurement reliability must be taken into account when estimating the reliability of grasp proposals.

2.3 Common Limitations and Research Gap

During the research on the two previously discussed publications and various experiments with other state-of-the-art approaches, we gained insights into the respective solution methodology for their respective settings but also identified common limitations in dealing with the uncontrollable, unknown household context, which we want to discuss

in the following sections.
Most robotic grasping approaches can be described as the mapping of sensory input S to some form of grasp quality, $Q \longleftarrow f(S)$. When data-driven approaches are considered, the evaluation of Q is not trivial, since the mapping function f is strongly coupled to the context of the training data. Three types of problems occur in a practical application within an unknown environment.

2.3.1 Training Population

If the encountered sensory input S is not within the population of the training-data, the performance of the algorithm might suffer. We encountered an example of this behavior during our experiments in section 2.2, where we used the *GGCNN* of Morrison et al. (2018) to estimate the grasp quality in pixel-wise manner. We employed a sensor-fusion method in order to obtain an estimation of the objects surface and used a ray casting algorithm to generate multiple overlapping views of the object. This was done in order to soften up the restriction to 2.5-dimensional grasp estimates that is inherent for all image-based approaches. We noticed that the quality estimates that emerged from perspectives that are semi-perpendicular to the ground plane were plausible. However, the estimates became more and more questionable with increasing angle between the table-top and the image plane (see figure 2). Often impossible grasp candidates were given high quality values. The reason for this behavior is that the training data (Cornell dataset Lenz et al., 2015) exclusively shows objects in a top-down manner on a planar background.
We came across another example during our tests with the VGN network (Breyer, Chung, et al., 2022) which uses a discrete volume representation of the environment. Both the object and the table surface are depicted within the cube-shaped volume. We found that the performance of the pre-trained network was very sensitive towards the "correct" positioning of the representation relative to the table surface. The approximations were significantly worse if the volume included too much of the unknown region below the table surface or too little. The reason for this behavior is the constant height of the volume representation relative to the planar surface below in the training data.
Although both behaviors are understandable in retrospect and the obvious solution would be to expand the datasets by including more diverse perspectives or volume representa-

tions, the fundamental problem remains. Contrary to an industrial setting, the household-environment is somewhat difficult to predict and will most likely contain samples that are outside the domain of the training data no matter how much effort we put into diversifying the training setup.
Will the object we are looking for be on a table, in a shelf or perhaps standing on a stool? Will it be isolated or in clutter? Which camera perspectives are plausible? What kind of objects will we encounter? How good is the illumination?
Creating a high-dimensional, labeled dataset that encompasses all potentially arising situations is a mammoth task that larger organizations than ours have to face.

2.3.2 Domain Gap

A common cause of performance drops in real-world applications is the domain gap problem. In the world of robotics, this problem is often encountered when algorithms that were trained with simulated data are transferred to the real-world. This is known as the sim-to-real gap. For example, the work of Kadian et al. (2020) is concerned with the quantification of this gap. The authors employ a public framework (Savva et al., 2019) to train a variety of agents for the task of visual navigation. They compare the simulated and real results, compute corresponding *Sim-vs-Real Correlation Coefficients* and conclude that the simulated success rates can significantly exceed the real ones. Their findings are related to visual navigation and therefore not directly transferable to robotic object manipulation. However, reduced grasp success rates in real experiments compared to the simulated results are a widespread occurrence, (e.g. Breyer, Chung, et al., 2022; Kasaei & Kasaei, 2023; Khansari, Kappler, Luo, Bingham, & Kalakrishnan, 2020; Lobbezoo & Kwon, 2023) and their exact causes are often not precisely discussed. In our experiments we found that two causes have a major contribution:

Physical Domain Gap

Modeling a physical environment is a trade-off between (numerical) stability and performance and realistic physics. For example, collisions are usually restricted to rigid bodies due to very high computational cost and the significant amount of fine-tuning that is necessary to simulate the collision of two soft bodies sufficiently realistic. The behavior of soft fingers or flexible objects like cereal boxes and stuffed animals, is therefore difficult to represent accurately in the simulation.

The rigid body collision behavior is then commonly represented by a finite number of contact points and a many models have been proposed to approximate the following interaction of the objects. Whether shearing motion or sliding behavior are computed often has a major influence on the success of a robotic grasp. In their evaluation of available contact models as implemented in common physics engines, Lidec et al. (2024) conclude that there is no fully satisfactory approach at the moment, as all existing solutions compromise either accuracy, robustness, or efficiency.
Adjustable "top-level" parameters like mass, inertia, center of gravity and friction play a large role, but optimal values are difficult to obtain. Among other things, Si, Zhu, Agarwal, Anderson, and Yuan (2022) performed a noteworthy ablation study about the impact of some of these parameters on the sim-to-real gap in the context of robotic grasping. Unsurprisingly, they found that the sim-to-real accuracy quickly drops if inaccurate parameters are used. Since these parameters do not accurately reflect the real world but should instead be understood as model variables, finding optimal simulation parameters is more complex than measuring their physical counterparts. Approaches have been proposed that try to minimize the gap between simulated data and a real, recorded distribution by varying parameters in a brute force manner. For example, Si et al. (2022) vary the friction of objects until the simulated success/failure rate is most similar to real results with the same grasps. Collins, Brown, Leitner, and Howard (2021) use a similar approach and optimize a large number of simulation parameters in order to match simulated trajectories with real counterparts for eight different object manipulation tasks. They show that even the choice of physics engine has to be adapted to the scenario as their performance varied significantly across the different manipulation tasks. These optimization approaches are very task-dependent and tremendously time-consuming. These drawbacks might be a reason why optimal (object-) parameters for larger datasets are seldom available, even if the physical counterparts of the objects exist, e.g. the YCB-dataset of Çalli et al. (2015).
Adapting a simulation to match reality as closely as possible is a high dimensional, extremely complex and time-consuming task that requires broad multidisciplinary expertise and sustained software development commitment (Choi et al., 2020). Even defining real behavior and measuring the distance to the simulated one is non-trivial and sometimes a practical challenge. There is currently no generally approved and commonly used system and instead a variety of customized environments are used, which makes even the

comparison between different publication results challenging. We observed hundreds of simulated grasping experiments with PyBullet and the open dynamics engine (ODE) and occasionally observed interactions that appeared unnatural, despite our best efforts to fine-tune physical parameters of the system and objects. In practice, compromises have to be made that inevitably lead to different behavior and therefore to the inevitable sim-to-real gap.

Optical Domain Gap

Labeling grasping datasets is very time consuming and therefore real world, human labeled grasping data sets are often limited in scale or use a low dimensional grasp representation such as the image based rectangle representation used by the Cornell dataset (Lenz et al., 2015) and the Multi-Object grasp dataset (Chu, Xu, & Vela, 2023). These restrictions gave rise to synthetic datasets where object-meshes are utilized to generate a large number of sensor measurements from various perspectives (e.g. Depierre, Dellandréa, & Chen, 2018; Eppner, Mousavian, & Fox, 2021; Mahler et al., 2016). These methods employ rendering programs that conveniently generate a clean depth map, called z-buffer as a by-product for ray tracing and rasterization. While Eppner et al. (2021) use the unaltered depth maps, Mahler et al. (2016) and Depierre et al. (2018) modify them in order to emulate real sensor behavior. Mahler et al. (2016) applies Gaussian and Gamma noise to the perfect depth image and Depierre et al. (2018) overlays two color images with a projector pattern and uses a stereo algorithm to calculate the depth. These concepts are limited because real sensor behavior is composed of a variety of effects. Approaches that rely on empirically determined sensor noise models e.g. Mahler et al. (2019) may be sensitive to different noise distribution as it occurs in another environment Tung, Su, Cai, Wan, and Cheng (2022).

Active stereo-vision depth sensors are widely adopted. Some examples of their noise behavior that are subjectively easy to implement are:

- The precision of the depth data decreases with increasing range. In case of stereo cameras this dependency is usually modelled as a quadratic function.
- The strength of the sensor noise increases as the angle of incidence becomes flatter. "Standard" disparity errors have greater effect if the surface is tilted along the cameras y-axis. This error approaches infinity as the angle converges to 90°.
- Out-of-Range noise usually appears when an object is placed outside of the config-

ured or hardware-specific range. Either the disparity range needs to be configured for the use case or the sensor must be chosen accordingly.

- Quantization effects limit the possible precision of the sensor. Images are a discrete representation of the environment. A stereo match between two images is therefore not exact but describes a possible region. Errors of this type become larger with increasing distance.

In addition to these errors, there is a variety of environmental effects that are very hard to model or implement in a generalized manner. Some selected examples of this are:

- Material specific parameters change the reflection behavior of infrared light. A prominent example for this are specular surfaces (e.g. mirrors) where the surface fails to diffuse the infrared light. Non-specular surfaces hardly reflect any infrared light, either due to full absorption or transparency. This behavior is by no means binary and the amount of reflected infrared light depends, among other things, on the color. Hence, some surfaces are "more difficult" to measure correctly than others.
- Shadow noise can happen if the infrared-emitter or the camera is partially obstructed. This can be frequently seen at object boundaries and leads to smearing effects along the line of sight.
- Similar to conventional cameras the environmental illumination can severely impact the infrared image. Some surfaces might be over- or under-exposed which leads to low contrast. Low contrast leads to ambiguous disparity values and thus to more frequent mismatches and errors.

Theoretic models exist for these error components but is practice it is difficult to attribute the individual error components to the perceived one in practice. For reasons of dimensionality and feasibility, only a subset of the effects can be considered when collecting necessary data and therefore the empirical model only reflects the conditions of the environment in which it was created. If an application has different parameters, the distribution of perceived data is different from the modeled one. Hence, simulated sensor data is inevitably different to the real one. This problem is known as the *optical-domain-gap* and closing it remains a unresolved problem and ongoing research goal.
Approaches to aim to close this gap can be roughly divided into two categories: *Domain*

Randomization and *Domain Adaption*. Domain randomization approaches in the context of robotic manipulation (e.g. Dai et al., 2022; Tremblay et al., 2023) randomize the "perfectly" simulated images in the training dataset in a non-realistic manner. They assume that if this randomization generates sufficient diversity, the generalization ability of the network can be ensured because real sensor data appears to the network simply as one of the many learned variations. Vuong, Vikram, Su, Gao, and Christensen (2023) show that the distribution of data augmentation has significant influence on the model's transferability. Since the true distribution is unknown it its uncertain that the augmentation can be performed in an efficient manner (Ma, Qin, Shi, Gao, & Huang, 2024).
On the other hand, domain adaption aims to make the simulation more realistic by designing rules or mappings that align data between the simulation and the real domain. This is done either through learning based approaches (Bousmalis et al., 2018) or via the acquisition of the coefficients needed to physically model the environment (e.g. Zhang et al., 2023). Both examples show that grasping success rates can be significantly increased by domain adaption. However, due to the previously discussed complexity, these approaches provide an approximation for one specific setup and are therefore more tailored towards a setting that is known in advance. Where Bousmalis et al. (2018) only considers a semi-static bin-picking scenario, Zhang et al. (2023) assumes that the geometry of the environment and the objects are precisely known and need to recreate the real setup in the simulation with high precision. This approach that is certainly not feasible for the context of assistive robotics where the environment can neither be controlled nor known beforehand to that degree.

2.3.3 Quantification of Uncertainty

The robustness of grasping algorithms could potentially be increased by using a time-series of sensor data instead of relying on single measurement and estimation. This seems reasonable since the camera is often attached to the end effector anyway, a setup which is often called an *eye-in-hand* setting. Therefore, a continuous data stream with typical frame rate in the range of 30 Herz to 60 Herz is available and the perspective of the camera can be adjusted almost arbitrarily. The utilization of a time-series of sensor data makes it possible to derive an associated uncertainty for grasp-predictions and the possibility to change sensor position and orientation also provides a tool to lower that score. Despite

these advantages and the possible gains, uncertainty quantification of time-series data in a Bayesian manner is a remarkably underutilized in the field of robotic manipulation. Some notable examples for antipodal grasp estimation based on a series of depth sensor measurements are discussed in the following.

In their follow-up work Morrison, Corke, and Leitner (2019b) and Breyer, Ott, Siegwart, and Chung (2022) partially seize this opportunity. Morrison et al. (2018) employ an entropy based exploration strategy while closing in on the object, hoping that the additional information will help the network to make more informed decision. The grasp is then chosen via maximum evaluation of the mean qualities. Breyer, Ott, et al. (2022) follow a similar approach where the robot is controlled in order to explore unknown volume. A grasp candidate is determined as stable if its quality prediction is above a user-defined threshold multiple consecutive times. This threshold effectively serves as a trade-off between the success rate and the abortion rate.

Although the evaluation of the history of quality-predictions surely helps to reject outliers, these approaches are sensitive towards the domain of the input data (see section 2.3.1). We experienced that if the sensor data is outside the trained domain, the predicted quality can be a poor measure of the actual reliability of a grasp. Our observations are supported by Shi et al. (2021) who report overconfident estimates for this approach, especially when the input image is outside the training domain. Blum, Sarlin, Nieto, Siegwart, and Cadena (2021) additionally note that quantifying the uncertainty with the confidence values of networks suffers from limited robustness.

Shi et al. (2021) propose a solution for this problem. They employ multiple object pose estimation networks trained with different datasets and use their outputs to calculate a disagreement score. They determine the optimal camera perspective via grid search where the best pose is the one with the smallest disagreement score. They then show that the success rate of an (unspecified) grasping algorithm can be remarkably improved if their method is used.

However, memory and computational complexity are crucial limitations for mobile applications. Therefore, the use of multiple networks for the same purpose is questionable in the assistive context. In addition, this approach suffers from similar problems, since no general statement can be made whether the smallest disagreement score leads to the greatest probability of success.

Common among these approaches is that the uncertainty of grasp candidates is given by some function (e.g. entropy over predictions, mean quality over several frames, disagreement score). Determining the effect of a robot action on this function value is non-trivial, especially when multiple candidates are considered.
Hence, computing an optimal robot action to minimize the overall uncertainty is infeasible, and these approaches need to resort to heuristics in order to determine an action instead.

2.4 Inference and Solution Proposal

The insights we gained from the research in the context of the two previous publications and various experiments with state-of-the-art approaches led to the idea of linking estimation uncertainty of the scene representation to the predicted success probability of a grasp. We propose to approximate the uncertainty for a grasp directly from the estimation variance of the corresponding surface elements. This allows us to identify uncertain grasps without being biased by the setup shown in a training dataset. Furthermore, we are able to determine how potential robot movements affect the entropy of the scene representation. We can determine which movement gives us the most clarity about whether a grasp will succeed of fail and are able to decide if enough information is available to opt for execution at runtime. Hence, we do not need an arbitrary window size or frame limit, but can adapt to the presented scenario. If the scene is fairly simple with a benign, free-standing object, grasps can be determined relatively quickly. On the other hand, if the scene is complex, quite noisy or the initial perspective is non-benevolent, then the search for a promising grasp takes longer and does not have to be aborted by an arbitrary, predefined limitation threshold. This conceptualization of active, closed-loop grasping is shown schematically in figure 14, where sensor measurements d from poses x at time i are fused into a volumetric representation of the scene (top, center box). This representation is used to sample grasp candidates and approximate their probability of success (right) and, if no sufficiently reliable possibility was found, to compute motion goals x^* that maximize the weighted information gain (bottom box) based on estimation variances σ^2.
This conceptualization is divided into the three disciplines which will be discussed in the following chapters in the same order:

- Sensor modeling and fusion of measurements 3

- Active exploration of the relevant volume 4
- Probabilistic grasp synthesis 5

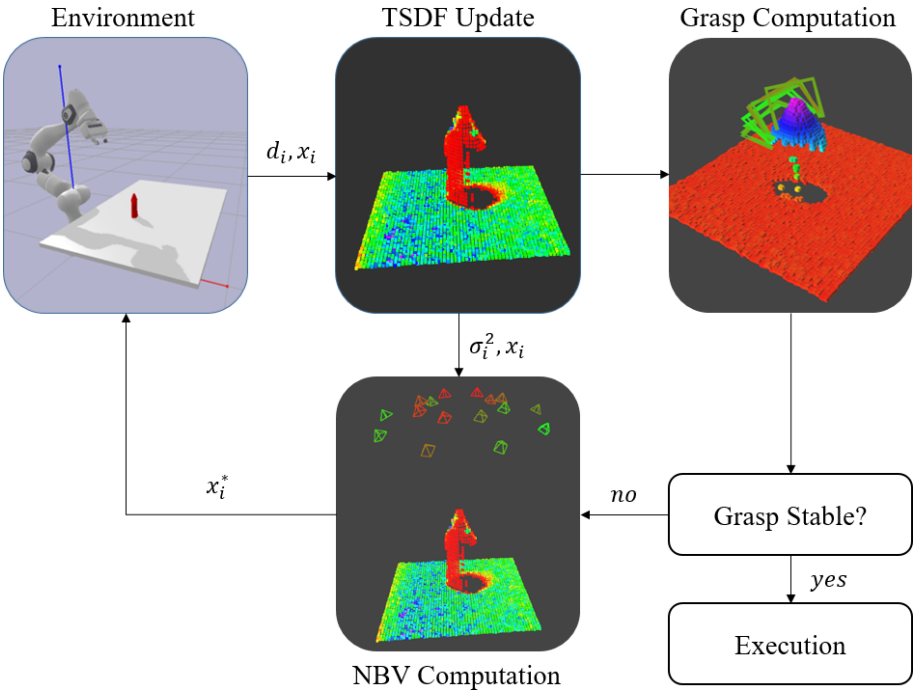

Figure 14: The proposed closed-loop pipeline.

3. Sensor Fusion for Robotic Grasping

3.1 Introduction

The chapter is largely based on the following publication.

> Schaub, Leuze, Hoh, and Schöttl (2023)
> *"Probabilistic Fusion of Depth Maps with Local Variance Estimation"*,
> in IEEE Sensors 2023

It can be regarded as an extension of the preceding publication.

> Schaub, Schöttl, and Hoh (2022)
> *"Probabilistic Fusion of Depth Maps With a Reliable Estimation of the Local Reconstruction Quality. "*,
> in IEEE Robotics and Automation Letters 2022

Generally, robotic grasping deals with the question of how to find the grasp with the highest probability of success given a representation of the scene. Many existing approaches represent the environment using a single point cloud combined with a statically mounted camera (e.g. Liang et al., 2019; Mousavian, Eppner, & Fox, 2019). This approach severely limits the flexibility and inherently relies on a benign first view. In addition, many authors, (e.g. Gualtieri, ten Pas, Saenko, & Platt, 2016; H.-Y. Lin, Liang, & Chen, 2021; Morrison et al., 2019b) found that considering additional perspectives opens up new, potentially better, grasping options and therefore improves the success rate of the overall system. This is especially true in the context of assistive robotics that we consider here. In the household context, it cannot be assumed that the initial perspective shows even a single promising, collision-free grasp. Neither can be assumed that the target object is placed isolated on the table. The environment can be cluttered, objects might only be partially visible and significant sensor noise might affect some sections of the scene. An intelligent assistive robot is therefore required to explore the scene at least partially and to fuse multiple measurements from different viewpoints. It is necessary to consider the uncertainty of estimations and the scene completeness before grasp execution to avoid

failures or collisions.
The fusion of data provides the following advantages in the context of the autonomous, robotic applications:

- *Increased Accuracy*:
  Since the data provided by depth sensors is subjected to some level of uncertainty and discrepancy, the fusion of multiple measurements via a Bayesian update scheme reduces the variance of the posterior distribution (Abdulhafiz & Khamis, 2013). This can be regarded as a decrease in the posterior uncertainty. Although Abdulhafiz and Khamis (2013) argue for the use of multiple sensors, this is certainly also true for the case of using the same sensor from difference perspectives.

- *Extended Spacial Coverage*:
  The three-dimensional space is reconstructed through the fusion of perspective measurements. This allows us to use the estimation confidence per discrete volume element to guide the robot's motion in order to maximize the information about the scene and to distinguish free from occupied volume, therefore avoiding collisions during the execution phase.

- *Linking estimation variance to grasp success probability* :
  If a suitable sensor model is used, the estimation variance of volume elements can be used to predict whether potential grasps will succeed or fail and whether enough measurements have been received to make this decision.

Two main characteristics make fusion of depth data different in the robotic grasping context than in the traditional context. First, a typical challenge for traditional sensor fusion application, e.g. self localization and mapping (SLAM), is the estimation of the sensor pose relative to an initially defined coordinate system and to prevent accumulating errors in the estimation of the traveled trajectory.
In an eye-in-hand setup, where the sensor is mounted on the end-effector of the robot arm, the sensor pose can assumed to be known at all times, with an accuracy similar to the accuracy of the forward kinematics. Although some authors (e.g. Žlajpah & Petrič, 2022), report positional errors of up 1.5 centimeters, this potential source of error is neglected in the following since the manufacturer claims an accuracy of the forward kinematics is in the sub-millimeter range according to the standards defined in 9283:1998(E) (2003).

Second, classic sensor fusion often for autonomous robots is mainly concerned with the accuracy and completeness of the reconstructed scene. Although these two metrics are certainly not unimportant in the context of robotic grasping, we found a third quantity also has great significance, namely the uncertainty of the estimates. We found that the solution space of grasps for traditional household objects is often significantly over-determined due to (semi-)axis or rotation symmetry. This means that many objects can be grasped in a variety of ways with many of those possibilities showing very similar quality metrics. By modeling and tracking the uncertainty, namely the estimation variance of corresponding surface estimations, the overall grasp success rates could be significantly improved. For example, if the currently best grasp candidate corresponds with highly uncertain estimates of surface segments, it might be beneficial to explore the object from a different viewpoint first before opting grasp execution in order to increase the probability of success.
The main contributions of this chapter is a novel, probabilistic algorithm that

- estimates the distance function of the surface under consideration by using a known empirical sensor model
- estimates the individual distribution of volume elements in parallel.

This enables us to specify the estimation variance also for surfaces where the measurements deviate from known sensor models for a variety of reasons and thus identify unreliably reconstructed sections with high probability. This algorithm is evaluated using a publicly available dataset (J. Yang, Gao, et al., 2021) and achieves better results than state-of-the-art algorithms.

3.1.1 The Truncated Signed Distance Function

Let S be a closed surface and $p \in \mathbb{R}^3$ and arbitrary point, then the signed, Euclidean distance function is defined as

$$f(p) = \underset{s \in S}{\operatorname{argmin}} \, | \, p - s \, |_2 \, \Psi(p) \;\; , \tag{21}$$

where the $\Psi(p)$ is given by

$$\Psi(p) = \begin{cases} -1, & \text{if } p \text{ inside} \\ 1, & \text{if } p \text{ outside} \end{cases} , \tag{22}$$

For most practical applications, whether the surface S, nor the signed distance function f are known in advance and the distinction of inside and outside is difficult to make. Hence, both the function and the surface must be estimated by a series of measurements. This can be interpreted as a discrete sampling of $f(p)$ in order to gain an approximation $\hat{f}$. For this purpose, the considered space is divided into equally sized cubic elements i.e. voxels, Their number as well as their edge lengths are predefined constants.
Every time-step i, the set of all voxel centroids P represented in the global coordinate is transformed into the sensor's coordinate systems $T_i \in SE3$ and then projected onto the image plane of the sensor. Let $p \in P \cap \mathbb{R}^3$, then

$$z_p \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = \lceil K\, T_i\, p \rfloor \ , \tag{23}$$

where similar to section 2.1.4 K represents the intrinsic camera matrix and $\lceil \cdot \rfloor$ denotes the rounding to the nearest integer image coordinates (u_p, v_p). If (u_p, v_p) falls on image coordinates outside the image boundaries or p is behind the image plane, i.e. $z_p < 0$, it is discarded and not further computations are performed. Otherwise, the normalized truncated distance t_p to the closest surface is computed. Let $(z_{i,p})$ be the i-th depth measurement at (u_p, v_p) and $\xi > 0$ the predefined truncation distance then

$$t_{p,i} = \Phi\left(z_{i,p} - z_p\right) \ , \tag{24}$$

with

$$\Phi(\psi) := \begin{cases} \min\left(\frac{|\psi|}{\xi}, 1\right) \operatorname{sgn}(\psi) & \text{if } |\psi| \leq \xi \\ \mathit{null} & \text{else} \end{cases} , \tag{25}$$

where *null* represents another rejection condition to ensure that measurements that lie outside the truncation band are not integrated into the grid. Its worth noting that cutting of

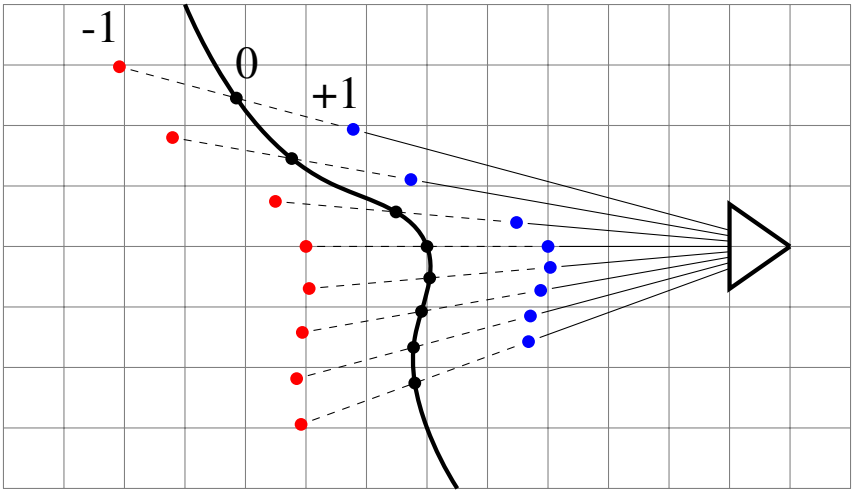

Figure 15: The triangle on the right represents the depth sensor. The perspective distance measurements of a surface are depicted in black whereas the truncation bands are colored.

positive measurements outside the truncation band is a deliberate design choice and the traditional algorithm fuses all measurements where $\psi \geq -\xi$. The asymmetric (with respect to the zero crossing) truncation of measurements introduces a bias, therefore leading to increased errors. We found that for the reconstruction (with known sensor poses) the information whether space outside the band is free or occupied has little significance and is therefore omitted in favor of greater accuracy.

A schematic 2D-representation of perspective distance measurements is shown in figure 15. The depth sensor is shown as a triangle on the right side. The distances of voxels to the black surface is evaluated along the line-of-sight rays. The distances are normalized to a predefined truncation distance ξ. Hence, values range from minus one to one. The area (actually the volume if $\mathbb{R}^3$ is considered) within the truncation band is colored red, if behind the surface, and colored blue if it is in front of the surface. Although the signed distance function f is defined using the Euclidean distance, for the approximation $\hat{f}$ the depth measurement (along the horizontal axis of the figure) of the corresponding ray is used. This leads to an error that can be seen in the width (perpendicular to the black surface) of the truncation band, where the truncation band is significantly smaller in the upper area. The error is determined via the angle of incidence of the ray and is zero either

when the voxel is directly on the surface or when the ray is perpendicular to the surface. The perspective error could be corrected by projecting the distance measurements onto the surface normal, but one would obviously need the "true" surface for this correction. Another approach would be the estimation of the surface normal via the corresponding image gradients, but we found that this approach is very sensible towards sensor noise. Newcombe et al. (2011) suggests that it is usually negligible if the surface is measured multiple times from multiple viewpoints. In addition, for most applications only the zero crossing of $\hat{f}$ is relevant where the error is zero anyway.

3.1.2 Traditional Fusion of Measurements

Let $T = \{t_1, t_2, ..., t_i\}$ be the set of truncated, normalized distance measurements for a single voxel centroid p. Since all depth measurements are subject to sensor noise the fusion of T can be regarded as the estimation $\hat{\mu}$ of the true value μ. A common assumption is that the noise behavior of a depth sensor measurement $z_{i,p}$ can be represented by a Gaussian distribution and therefore the truncated, normalized distance measurements $t \in T$ follow a (truncated) Gaussian distribution as well.
An estimate $\hat{\mu}$ for the true scaled distance μ is typically obtained by maximizing the corresponding likelihood function. If i independent, identically distributed measurements are assumed, the maximum likelihood estimation for the signed distance μ of one voxel centroid is

$$\hat{\mu}_i = \frac{1}{i} \sum_{j=1}^{i} t_j \ . \tag{26}$$

The naive implementation of equation 26 would require to store the full history of measurements for all voxels in the considered volume which implies a steadily increasing storage requirement. Hence, in implementations an equivalent online version parameterized with $(\hat{\mu}_i, W_i)$ is preferred. Let W_i represent the sum of weights the estimation received until timestamp i and $w_i > 0$ represent an arbitrary weight of the current measurement t_i. The update scheme is given by

$$\hat{\mu}_i = \frac{W_{i-1} \quad \hat{\mu}_{i-1} + w_i \, t_i}{W_{i-1} + w_i} \ , \tag{27}$$

and

$$W_i = w_i + W_{i-1} \quad . \tag{28}$$

To prevent the estimation from "getting stuck" and to have the option of removing dynamic elements from the map again, W_i is sometimes capped in practice. In order to enhance the reliability of the estimations, most post-processing applications additionally discard all voxels where W_i is below a predefined threshold.

3.1.3 A Probabilistic Perspective on Sensor Fusion

If all measurements t_i are independent and follow the same distribution (i.i.d) then equal weighting, i.e. $w_i = 1$, is the optimal estimator. Update equations (27) and (28) result in the simple average and are equivalent to equation (26), which is the approach suggested by Newcombe et al. (2011). The sum of weights W_i is then equivalent to the number of measurements a voxel has received. Although some treat the number of measurements as a measure of certainty of the voxel estimate (e.g. F. Li et al., 2016), this approach only allows comparing the cumulative weight relative to other weights since no conclusion about the magnitude of the absolute error can be drawn. In addition, the empirical evidence, (e.g. Halmetschlager-Funek, Suchi, Kampel, & Vincze, 2019) suggests that the i.i.d. assumption is flawed since the precision of most sensors decreases with increasing distance among other things.

A common strategy, as used by Dietrich, Chen, Wurm, von Wichert, and Ennen (2016) and Tung et al. (2022), to take this fact into account is to use an empirically determined sensor noise model, where parameters of a suitable function are fitted to match empirical data. The interpretation of such a model requires a probabilistic perspective of the update steps mentioned in 3.1.2. A common approach is to model the measured distance z as a random variable where the true distance z_{true} is perturbed by additive white Gaussian noise,

$$z \sim z_{true} + \mathcal{N}\left(0, \sigma^2_{z,i}\right) \quad , \tag{29}$$

where $\sigma^2_{s,i}$ the measurement variance at time i according to the sensor noise model. If instead the measured TSDF distance t of a point on the same view-ray to the next obstacle along the ray is considered, it is only a scaled displacement of the same Gaussian in 29.

The fact that t is truncated to ξ before fusion is commonly neglected and t is modeled as

$$t \sim t_{true} + \mathcal{N}\left(0, \sigma_{t,i}^2\right) \quad \text{where} \quad \sigma_{t,i}^2 = \frac{\sigma_{s,i}^2}{\xi^2}. \tag{30}$$

Although this approximation introduces an error into the estimation, it plays a minor role in practice since most subsequent algorithms are exclusively interested in the zero-crossing of the volume, i.e. the estimated surface, where the error is negligible if the truncation distance ξ is chosen large enough relative to the measurement's standard deviation σ.
Let $p(\mu)$ be the prior state estimation

$$p(\mu) = \left(2\,\pi\,\sigma_0^2\,\right)^{-1/2} \exp\left(-\frac{1}{2}\frac{(\mu-\mu_0)^2}{\sigma_0^2}\right) \ , \tag{31}$$

and $T = \{t_1, t_2, ...t_i\}$ be a set of independent measurements drawn from a normal distribution with known, corresponding measurement variances $\Sigma = \{\sigma_{t,1}^2, \sigma_{t,2}^2, ...\sigma_{t,i}^2\}$. The posterior distribution is then determined using the Bayes theorem

$$p(\mu \mid T, \Sigma) = \eta \ \ p(\mu) \ \prod_{j=1}^{i} p(\, t_j \mid \mu,\, \sigma_{t,j}^2) \ , \tag{32}$$

where η is a normalization constant. Since the posterior $p(\mu \mid T, \Sigma)$ is a normal distribution as well, posterior and prior are conjugate distributions and a closed form expression for equation (32) is available, which is given by the two following parametric update equations

$$\hat{\mu}_i = \frac{\hat{\mu}_{i-1}\ \sigma_{t,i}^2 + t_i\ \sigma_{i-1}^2}{\sigma_{t,i}^2 + \sigma_{i-1}^2} \tag{33}$$

and

$$\sigma_i^2 = \frac{\sigma_{t,i}^2\ \sigma_{i-1}^2}{\sigma_{t,i}^2 + \sigma_{i-1}^2} \ . \tag{34}$$

These update formulas are similar to the equations (27) and (28) where the update weights w_i are chosen to be inverse proportional to the measurement variance $\sigma_{t,i}^2$. This procedure is often called inverse variance weighting. Under the assumption that measurement noise

is perfectly known, equations (33) and (34) are known to be the minimal variance estimator (Shahar, 2017).

3.2 Measurement Model and Noise Characterisation

3.2.1 Choice of sensor

In our work, we opt for active stereo sensors. This is related to the requirements for our application. The sensor shall be

- *Small*:
  A big sensor that is mounted on the end-effector can severely hinder the robot's ability to maneuver. Large dimensions of the sensor points represent an enormous restriction for path planning. In practice, this can lead to significantly longer planning times, questionable path solutions or the inability to move towards some target poses at all.
- *Lightweight*:
  Robotic arms that are approved for the assistive sector are surprisingly weak. The *Kinova Jaco* has as weight limit of one kg at maximum reach (*Specifications of Jaco assistive robotic arm* (2025)). A heavy sensor would therefore severely restrict the choice of objects to pick up.
- *Capable at Close Ranges*:
  Most depth sensors are designed with specific a specific distance range in mind, either through hardware restrictions (e.g. base length) or software (e.g. image filter parameters that can not always be modified by the user). Naturally we are mainly interested in objects with the reach of the robot, i.e. distances < 1.5 m.
- *Affordable*:
  Although this is not a hard restriction, one could question the rationale of the proposed approach if the price of the sensor is in the same order of magnitude as that of the robot arm.

We opt for the *Intel Realsense D435* sensor as it fulfills all these requirements. The requirement for high precision at close range was a main contributor for this decision. The

D435 is a stereo camera which often show less noise at short distances as the noise level of stereo sensors is a quadratic function of the distance while the noise of e.g. time-of-flight sensors is constant with respect to the distance.
This could explain why in applications that need to handle great distances, such as autonomous driving, LIDAR technology is widely used and the field of robotics, where the focus is on much shorter distances, is subjectively dominated by stereo technology, (e.g. D. Kim et al., 2020; C.-J. Lin, Chang-Chien, & Chen, 2023; Ye, Cui, Wang, Xie, & Ni, 2023). For example, Lourenço and Araújo (2021) show that the *Intel D415* sensor has higher accuracy than the *Intel L515* time-of-flight sensor, but only for distances smaller than roughly 1.2 meters. These findings align with the results of Halmetschlager-Funek et al. (2019) who show that the *Intel Realsene D435* is significantly more accurate than the time-of-flight *Micorsoft Kinect v2* in close ranges.
Finally, our choice of sensor is supported by the work of Vit and Shani (2018). They compare a number of sensors that we would consider to be consumer-level hardware for their applicability in a close range setting. They conclude that for ranges between 0.2 and 1.5 meters, the *Intel RealSense D435* produced the best results in terms of accuracy and exposure control. Like any other depth measurement technology, stereo sensors have situation-dependent sources of error which we would like to discuss in the following.

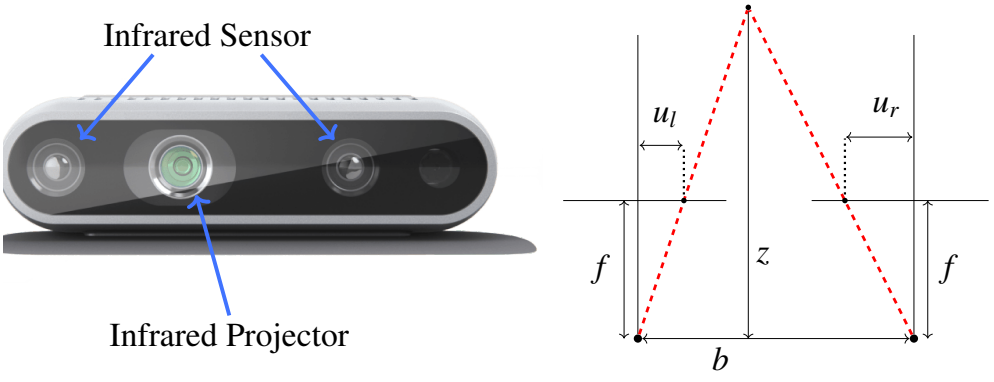

Figure 16: The Intel Realsense sensor and a stereo camera diagram.

3.2.2 Theoretical Error

The *Intel Realsense D435* depth sensor is an active stereo system and is shown in figure 16. It utilizes two infrared sensors and an infrared emitter to gain distance measurements. Although the two cameras are theoretically sufficient to determine depth values, the infrared emitter helps to find correspondences between the left and the right image by projecting a dot pattern which improves the local contrast. Given a correspondence between the left and the right image the disparity d is given by

$$d = u_l - u_r \ , \tag{35}$$

where u_l is the horizontal image coordinate in the left rectified infrared image and u_r the horizontal coordinate in the right rectified infrared image. The depth measurement z is calculated by

$$z = \frac{f\,b}{d} \ , \tag{36}$$

where f is the focal length and b the base length. A representation is shown in figure 16 on the right-hand side.

Assuming that focal length f and b the base length as well as the camera centers are perfectly known, then a disparity error Δd leads to a depth measurement error Δz,

$$\Delta z = \frac{\partial z}{\partial d}\Delta d = -\frac{f\,b}{d^2}\,\Delta d = -\frac{z^2}{f\,b}\,\Delta d \ . \tag{37}$$

The main takeaway is that the magnitude of depth measurement errors increases quadratically with increasing distance. If the sensor was sufficiently calibrated, the distribution of the measurements taken at a constant distance usually takes the form of a zero mean Gaussian bell curve (e.g. Haider & Hel-Or, 2022; Khoshelham & Elberink, 2012). Therefore, the noise behavior of stereo is often modeled as a normal distribution where the standard deviation is modeled as quadratic function of the depth where the parameters are found via a Least-Mean-Squares procedure with empirically obtained data (e.g. Ahn, Chae, Noh, Nam, & Hong, 2019; Chatterjee & Govindu, 2016).

3.2.3 Influence of Illumination

The search for corresponding pixels two images requires a distinct description of the pixels, for which their local neighborhood is evaluated. The distinctiveness is strongly diminished by low-contrast images with little texture or weakly illuminated environments and instead of clear one-to-one mappings, ambiguities arise. In order to enhance the contrast of the images, active stereo cameras rely on an additional optical projector. The projector overlays the visible scene with a semi-random pattern of dots in the infrared range of light. An example of this pattern can be seen in the top middle of figure 17. This pattern facilitates finding correspondences between the left and the right infrared images, particularly in the case of texture-less surfaces such as white walls.
As the intensity of the environmental illumination increases, the contrast decreases. This reduces the informative value of pixel descriptions. Hence, more and greater matching errors arise and the perceived depth noise increases. Grunnet-Jepsen, Sweetser, Winer, Takagi, and Rev (2017) evaluate the standard deviation for a distance of five meters and specify roughly 16 mm in complete darkness and roughly 58 mm for ambient lighting of 116 lux. Both values refer to the maximum power setting of the projector which represents the best setting within the scope of the experiment since lower power settings increased the standard deviation of the error significantly. Among other things, R. Chen, Xu, and Zhang (2022) showed that the standard deviation of the D435 measurements of a planar target at 1m distance increased from 4mm (dark room, 0 lux) to 8mm (strong environmental illumination, 2500 lux). Although Grunnet-Jepsen et al. (2017) and R. Chen et al. (2022) speak of environmental illumination, this phenomenon is not a global one and far from constant across the scene. Instead, the lack of contrast is local and heavily influenced by the measured material and the position of the surface relative to the source of lighting. The employed camera uses the infrared spectrum of light. Hence, the over-illumination is invisible to the user and the data of some sections of the scene just appears more noisy than others. A visual example for this phenomenon is given in figure 17 where the infrared image of a 3D-printed figure (left) is shown in the center. The scene appears to be well lit to the user, but the left side of the figure and segments of the table show little contrast in the infrared image. This leads to increased ambiguity for correspondence findings and heavily noisy regions in the depth image which is shown in the top right. Since the exact matching algorithm of the *Intel Realsense D435* (and most other depth sensors)

is private intellectual property of *Intel* and is not open-source, we implemented a custom stereo matching algorithm that uses the zero mean normalized cross correlation (NCC) to measure similarity,

$$NCC(u,v,d_{isp}) = \frac{\sum_{i \in K} \left(IR_l\left(u_i,v_i\right) - \mu_l\right) \left(IR_r\left(u_i - d_{isp,i}\, v\right) - \mu_r\right)}{\sigma_l \sigma_r} \tag{38}$$

where IR_l and IR_r are the two images of the stereo image pair. (u,v,d) are the image coordinates and the disparity value respectively. μ_l and μ_r are the standard deviations of all pixels in the 5×5 square window W in the left and right image, respectively. In case a pixel-to-pixel correspondence does not satisfy a custom confidence threshold, the depth data was invalidated and appears black in the computed depth image.

The bottom graph of figure 17 shows normalized cross correlation cost curve across the epipolar line within the valid disparity range for the center pixel of the red rectangle in the left infrared image. An ideal cost curve would show a single, distinct global minimum within the disparity range. In this case the multiple local minimal with similar cost make the localization of the correct correspondence hard and depending on the confidence threshold the corresponding depth measurements are either invalidated or show high levels of noise. This behavior is not constant across the scene but local. In this experiment, the emitter dot pattern is very distinct on the table surface and matches can be found with a high degree of reliability. The table surface therefore appears less noisy than the 3D-printed dwarf. Ironically, the white residue on the table and left side of the dwarf is a so-called scanning spray which is intended to reduce the reflective properties of the surface and was actually intended to improve the measurements. To reduce this illumination error for weakly textured surfaces, possible solutions include the recommendation of Grunnet-Jepsen et al. (2017) suggests the purchase of an infrared-band-pass filter and placing it in front of the sensor. The filter attenuates visible light and therefore increasing contrast and reducing the error magnitude. However, they also show that well textured surfaces strongly benefit from ambient illumination. Another suggestion of Grunnet-Jepsen et al. (2017) is the usage of multiple projectors that are positioned in such a way that the infrared pattern density is increased in the target area. Needless to say, that both strategies are very situation-dependent and represent by no means a holistic solution to the problem.

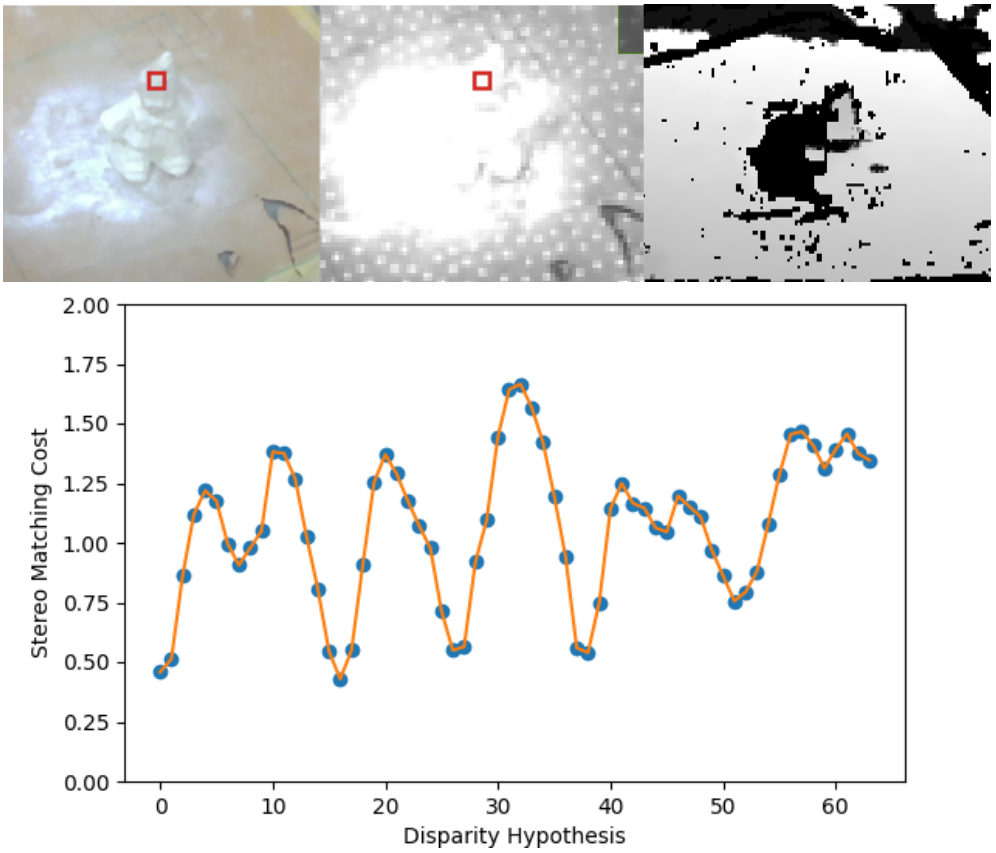

Figure 17: Color image of a 3D-printed figure (left), the rectified infrared image (middle) and depth image of a custom stereo algorithm (right). Some parts of the infrared image are clearly overexposed, which is not necessarily noticeable in the visible frequency range of light. The bottom graph shows the ambiguous cost curve for matching between the marked image position of the left infrared image and corresponding right one (not displayed).

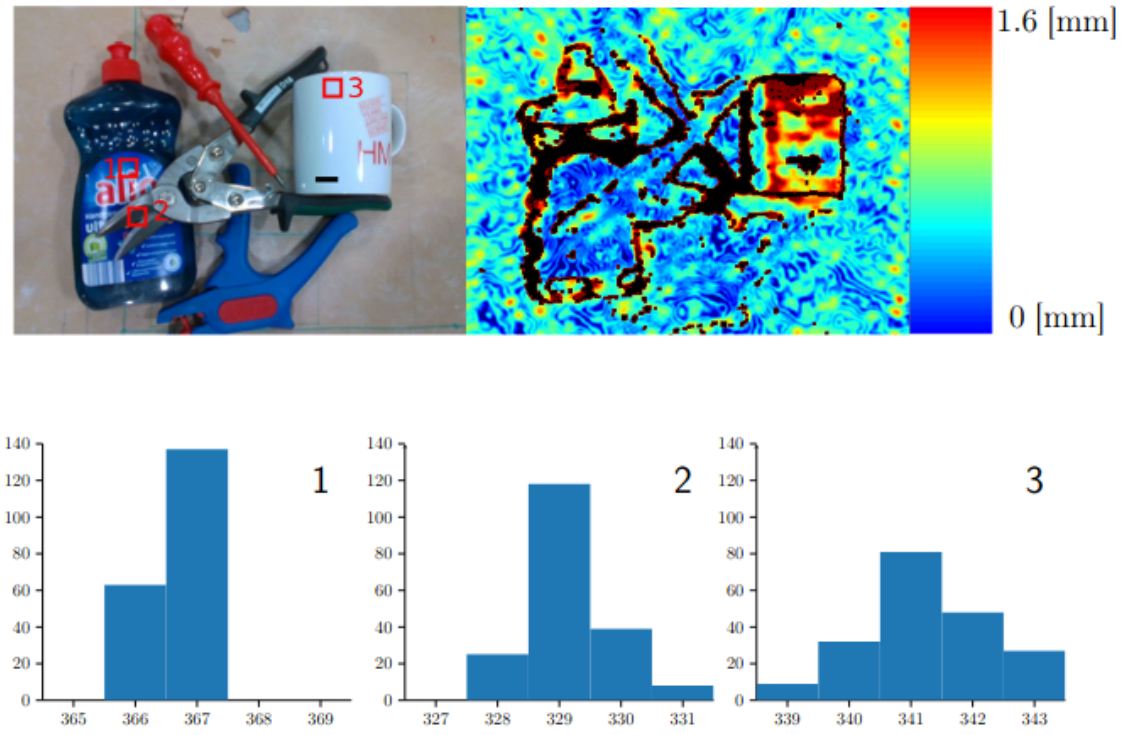

Figure 18: Three histograms of depth measurements taken in a static setup (top left). The image in the top right shows standard deviations of the scene.

3.2.4 Influence of Surface Material

Different materials reflect infrared radiation to varying degrees. Hence, surfaces can appear more or less distinctive in infrared images. Similar to the influence of illumination, some surface materials can therefore lead to more ambiguous stereo matches and therefore to more noise. A custom experiment to visualize this behavior is shown in figure 18. All objects on the table are at roughly the same distance. We gathered roughly 200 measurements of the scene and the empirical standard deviation of these measurements is shown as jet color map in the corresponding right image. For illustration purposes, all values $\geq$ 1.6 [mm] were colored in deep red and pixels that received less than ten measurements are colored in black. The lower tree graphs show the histograms of three manually selected positions which are highlighted in the upper left image. The histograms correspond only to a single image position, although the rectangles enclose a whole image

area for visibility reasons.
The first histogram of the well textured label shows very uniform readings and the small deviation is probably exclusively introduced by the quantization effect of reading the distances as integer millimeters. The distance measurements of the ceramic cup area however show a significantly wider spread and measurements can deviate from each other by up to five millimeters. Its worth noting that we spent a significant amount of time adjusting all available camera settings to get the depth measurements as noise-free as possible. Hence, we assume results are somewhat optimal in a practical sense for this environment, perspective and distance. Of course, this adjustment procedure would have to be repeated for a new setting.
The influence of measured material is widely known and two examples that give concrete numerical values for the Intel Realsense D435 sensor are the works of Halmetschlager-Funek et al. (2019) and W. Kim et al. (2023) whose findings are summarized in the following table 2.

Table 2: Standard deviation of depth measurements at 1.5 meter distance, rounded to whole millimeters for different materials.

	W. Kim et al. (2023)			Halmetschlager-Funek et al. (2019)					
Materials	Cement	Form-board	Plastic	Textiles	Aluminum	Plastic (Black)	Foam	Paper	Plastic (Blue)
σ [mm]	8	9	15	2	12	3	2	3	11

W. Kim et al. (2023) follow the common assumption that the measurement noise can be approximated by a Gaussian distribution. In order to enhance the realism of a drone simulation environment they propose to model the standard deviation as a heuristic function of depth and an additional material-specific parameter that is found through extensive experiments for tree different materials, *cement*, *form-board* and *plastic*. Values of this of model function of σ_z for a depth of 1.5 m can be found in the first three columns of table

2 and a difference of roughly 7mm (*cement* vs. *plastic*) can be observed.
Among other things, Halmetschlager-Funek et al. (2019) tested the performance of various depth sensors under the influence of different materials. The corresponding results of the *Intel Realsense 435* camera at a distance of 1.5m are shown in the last six columns of table 2. The results of Halmetschlager-Funek et al. (2019) confirm common the understanding that reflective materials (e.g. *aluminum*) can significantly decrease the performance of active stereo sensors and also show an interesting, significant difference in precision between measurements of *blue* and *black plastic*.
It cannot be assumed that all materials that could be encountered during an application and their influence on the noise behavior are known in advance. Instead, the reliability of the estimates must be approximated on the fly. Furthermore, this illustrates the difficulty of accurately representing real sensor noise in the simulation.

3.2.5 Problem description

A common procedure for estimation using several measured values, is weighting the measurements with the corresponding estimated inverse variance. To determine this variance estimate, an empirically determined model function is usually utilized. If the model function models the situation perfectly and therefore the estimated variance is equal to the true variance, the inverse variance weighting is known to be the optimal estimator. The problem that arises is that the model function is inevitably tailored to the setup in which the data was recorded and may not be suitable for a different environment. This is reflected by the significantly varying values for the expected standard deviation then can be found in the literature. Figure 19 shows all the model functions for the standard deviation of the Intel Realsense D435 sensor that we were able to find for the distance range that we consider reasonable for an eye-in-hand grasping application. For example, at a distance of one meter, depending on the selected model function, the expected value for the measurement standard deviation ranges from roughly one millimeter up to 1.2 centimeter. These differences are very significant if measurements are to be merged into a unified surface representation.
Additionally, the approximation of the reliability of an estimate is a problem if the "wrong" sensor noise model is selected.Let'ss assume that the estimated noise $\mathcal{N}\left(0,\sigma_z^2\right)$ is constant, but for the surface segment under consideration there is an additional, constant noise

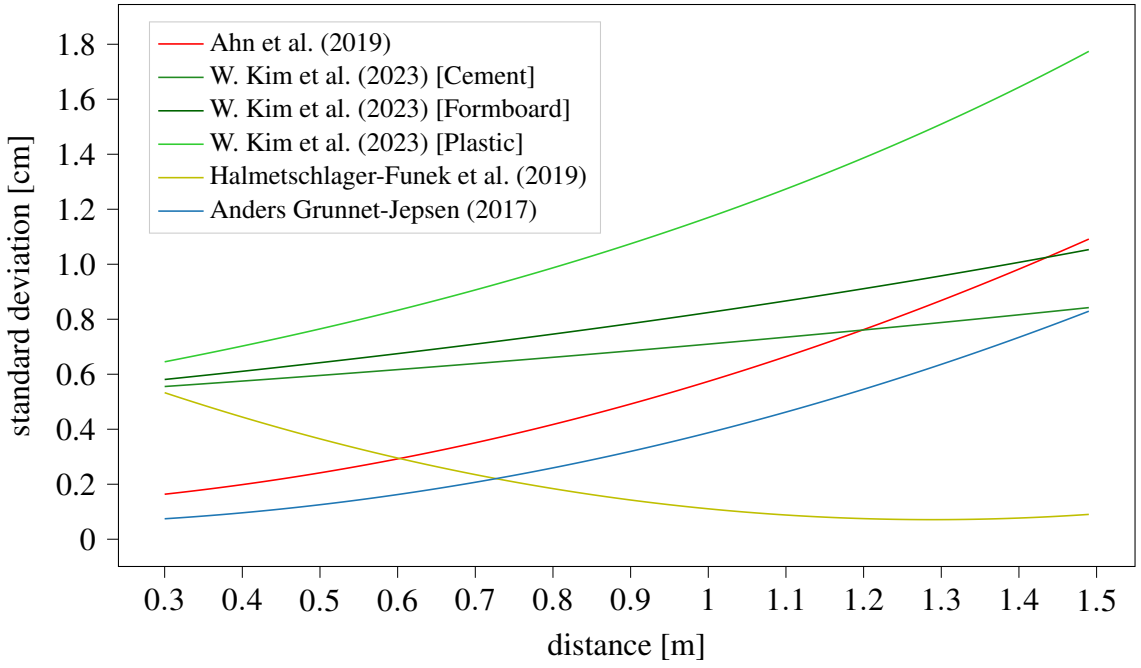

Figure 19: The standard deviation of depth measurements of the Intel Realsense D435 sensor as a function of the distance.

component $\mathcal{N}\left(0, \tau^2\right)$. A plausible example would be that we rely on the manufacturer's data (Anders Grunnet-Jepsen, 2017), as indicated by the blue curve in fig. 19, but the actual behavior is more in line with the data given by the works of Ahn et al. (2019) or W. Kim et al. (2023) that is represented by the red and the light green curve. In this case τ is given by the vertical distance of these graphs, which is almost constant across the distance range. The ratio of the estimated variance of the estimator in equation 33 after N measurements $\widetilde{\mathrm{Var}}\left(\hat{\mu}\right)$ to the correct variance $\mathrm{Var}\left(\hat{\mu}\right)$ is given by

$$\frac{\widetilde{\mathrm{Var}}(\hat{\mu}_N)}{\mathrm{Var}(\hat{\mu}_N)} = \frac{\frac{\sigma_c^2}{N}}{\frac{\sigma_z^2+\tau^2}{N}} = \frac{\sigma_z^2}{\sigma_z^2+\tau^2} \quad . \tag{39}$$

Note that the number of measurements N cancels out. Aside from the fact that the estimator is no longer optimal, if $\frac{1}{\widetilde{\mathrm{Var}}(\hat{\mu}_N)}$ is used as a measure of reliability and thus as a measure of when the estimate is "good enough", then this reliability is always overestimated by the ratio $\frac{\sigma_z^2+\tau^2}{\sigma_z^2}$. Hence, the collection of measurements could be stopped to early and a

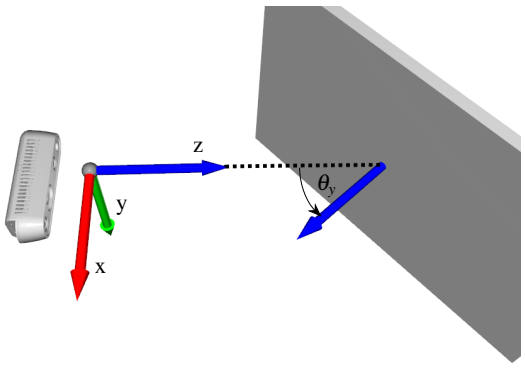

Figure 20: Schematic representation of the camera's coordinate system. θ_y represents the angle around the green y-axis to align the surface normal with the camera's y/z plane.

grasp might be chosen that is unlikely to succeed. Instead, τ must be considered in the estimation process if the estimated variance is to be used for the following application.

3.3 Proposed Sensor Model and Data Fusion

We propose a probabilistic truncated signed distance function (TSDF) approach. Each voxel contains a tuple $(\hat{\mu}, \hat{\tau}^2, v, W)$, where $\hat{\mu}$ corresponds to the estimated distance of the voxel's center towards the closest surface and $\hat{\tau}^2$ represents an estimate for the measurement scatter caused by the unknown surface properties or over-illumination of the closest surface element. v represents the current estimation variance of τ^2 and W the accumulated weights for $\hat{\mu}$. We model a single signed distance measurement t_i at time step i of one surface patch as

$$t_i \sim \mu + \mathcal{N}(0, \sigma_{t,i}^2 + \tau^2) \ , \tag{40}$$

where τ^2 describes the surface dependent error variance and μ is the real distance from the camera's focal point to the surface patch. The standard deviation $\sigma_{t,i}$ caused by the sensor can be empirically determined beforehand. We opt for the model of Ahn et al. (2019), as we found the angle of incidence of the view-ray has a non-negligible contribution to the magnitude of the error (see figure 20) and their work is, to our knowledge, the only one that models this dependency for the *D435*. Ahn et al. (2019) add a heuristic hyperbolic term to incorporate the significant loss of measurement precision of stereo sensors as the

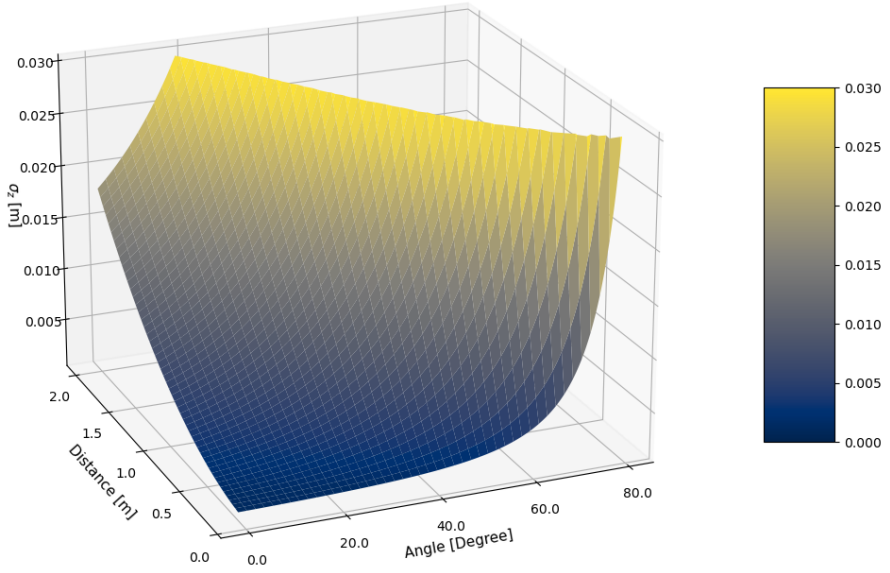

Figure 21: The standard deviation the sensor noise as a function of the distance and θ_y. The noise peaks toward infinity as the angle approaches 90°. For the sake of visibility the drawn surface is cut of at 3 cm.

angle of incidence approaches 90 degree. This behavior has been observed with other stereo sensors, such as the Kinect in the work of C. V. Nguyen, Izadi, and Lovell (2012). The standard deviation $\sigma_s(z, \theta_y)$ is given by

$$\sigma_z(z, \theta_y) = 0.001063 + 0.0007278\, z + 0.003949\, z^2 + 0.022^{\frac{3}{2}} \frac{\theta_y}{\left(\frac{\pi}{2} - \theta_y\right)^2} \quad , \tag{41}$$

where z is the measured depth and θ_y the angle of incidence. A visualization of this function is given in figure 21 where we cut off the graph for $\sigma_z > 3$ cm. The increase of the function towards infinity is clearly visible for angles approaching 90 degrees. In the following, we use this function to model the known sensor noise portion of signed distance measurements $\mathcal{N}\left(0, \sigma_t^2\right)$.

A typical approach to estimate μ as well as τ is via maximum likelihood estimation, where estimations are found via derivation of the corresponding likelihood function and iterating between estimations for μ and τ until parameter estimates do not change from one iteration to the next. This problem is comparable to the *Random effect model* that is

used in the meta analysis to aggregate the results of studies. The *Random effect model* assumes that the variability in effect size estimates drawn from a set of studies can be decomposed into two parts: heterogeneity due to random population effects (analog to our τ) and sampling variance (analog to our $\sigma_{t,i}$). However, the estimators used in the meta-analysis, (e.g. Dersimonian, Dersimonian, Laird, & Laird, 2001; J. E. Hunter & Schmidt, 2006) suffer from the same limitation as traditional maximum likelihood estimation.
They require the availability of all measurement data at the time of estimation, which implies the need to store the full measurement history $\left((t_i, \sigma_{t,i}^2), ..., (t_1, \sigma_{t,1}^2)\right)$ for every voxel. Even though, the memory requirement could be manageable if a moving window approach and a reasonably chosen voxel resolution were considered, we found that the corresponding update step of such an approach severely limits the real-time capabilities of the whole system even if a GPU implementation is used.
Instead, we propose to recursively refine the initial estimates for μ and τ in a Bayesian manner. Similar to equation 26, for known ρ, the least squares estimator is given by

$$\hat{\mu} = \frac{\sum_{j=1}^{i} t_j/\rho_j^2}{\sum_{j=1}^{i} 1/\rho_j^2} \ , \tag{42}$$

which can be expressed as the following update equation

$$\begin{aligned} \hat{\mu}_i &= \alpha_i x_i + (1-\alpha_i)\, \hat{\mu}_{i-1} \ , \\ \alpha &= \frac{1/\rho_i^2}{\sum_{j=1}^{i} 1/\rho_j^2} = \frac{1/\rho_i^2}{W_i} \ . \end{aligned} \tag{43}$$

If τ were known beforehand, then for $\rho_i^2 = \sigma_{t,i}^2 + \tau^2$ equation (42) and (43) would be minimum variance i.e. optimal estimator of μ. However, the additional standard deviation of a surface element τ, that can be introduced by a variety of reason is initially unknown. Hence, we propose to estimate τ^2 in parallel with a similar linear update scheme. Let ρ be

$$\rho_i^2 = \sigma_{t,i}^2 + \hat{\tau}_i^2 \ , \tag{44}$$

where $\hat{\tau}_i^2$ is the estimation of τ^2 at time step i. Let $\overset{*}{t}_i$ be the difference from the measurement to $\hat{\mu}_i$, i.e. $\overset{*}{t}_i := t_i - \hat{\mu}_i$. We propose to approximate $\rho^2 \approx \overset{*}{t}_i^2$, which leads to

$$\hat{\tau}^2 \approx \overset{*}{t}_i^2 - \sigma_{t,i}^2 \ . \tag{45}$$

The recursive update equation for $\hat{\tau}_i^2$ is then given by

$$\hat{\tau}_i^2 = \beta \ (\overset{*}{t}_i^2 - \sigma_{t,i}^2) + \gamma \ \hat{\tau}_{i-1}^2 \ , \tag{46}$$

where $\gamma = 1 - \beta$ in order for $\hat{\tau}^2$ to be unbiased. The variance $Var(\hat{\tau}_i^2) = v_k$ of this estimator is subsequently given by

$$v_i = \beta^2 \ Var(\overset{*}{t}_i^2 - \sigma_i^2) + (1-\beta)^2 \ Var(\hat{\tau}_{i-1}) \ , \tag{47}$$

where, taking into account that $\sigma_{t,i}^2$ is deterministic, the first variance term can be simplified to

$$\begin{aligned} &Var\left(\overset{*}{t}_k^2 - \sigma_{t,i}^2\right) = Var\left(\overset{*}{t}_i^2\right) \\ &= E\left(\overset{*}{t}_i^4\right) - E^2\left(E\left(\overset{*}{t}_i^2\right)\right) \ . \end{aligned} \tag{48}$$

Since our estimator $\hat{\mu}$ is unbiased, then $\overset{*}{t}_k \sim N\left(0, \sigma_{t,i}^2 + \tau^2\right)$ holds and $E\left(\overset{*}{t}_k^4\right)$ is equal to the fourth central moment of the Gaussian distribution,

$$E\left(\overset{*}{t}_k^4\right) = 3\rho_k^4 = 3\left(\sigma_{t,i}^2 - \tau^2\right)^2 \ , \tag{49}$$

and equation (48) simplifies to

$$\begin{aligned} Var\left(\overset{*}{t}_i^2\right) &= 3\left(\sigma_{t,i}^2 - \tau^2\right)^2 - \left(\sigma_{t,i}^2 - \tau^2\right)^2 \\ &= 2\left(\sigma_{t,i}^2 - \tau^2\right)^2 \end{aligned} \tag{50}$$

Substituting (50) into equation (47) leads to

$$v_i = \beta^2 \ 2(\tau^2 + \sigma_i^2)^2 + (1-\beta)^2 v_{i-1} \ . \tag{51}$$

Algorithm 2 Iterative update of one voxel

$\tau^2 = \tau_0^2$
$v = v_0$
$W = 0$
$\mu = 0$
For t, σ *in* (T, Σ)
  $\rho = 1 \,/\, (\sigma^2 + \tau^2)$
  $W = W + \rho$
  $\alpha = \rho \,/\, W$
  $\mu = \alpha\, t + (1-\alpha)\, \mu$
  $\overset{*}{t} = t - \mu$
  $\beta = v \,/\, (2(\tau^2 + \sigma^2)^2 + v)$
  $\tau^2 = \beta\, (\overset{*}{t}^2 - \sigma^2) + (1-\beta)\, \tau^2$
  $v = 2\, \beta^2\, (\tau^2 + \sigma^2)^2 + (1-\beta)^2\, v$

The optimal update scaling factor β yields the minimal variance. Hence, the optimal β can be obtained by differentiating (51) with respect to β, using the chain rule

$$\begin{aligned}\frac{\partial v_i}{\partial \beta} = 0 &= 2\,\beta\; 2\left(\tau^2 + \sigma_i^2\right)^2 + 2\,(\beta - 1)\, v_{i-1} \\ &= 2\,\beta\left(2\left(\tau^2 + \sigma_i^2\right)^2 + v_{i-1}\right) - 2\, v_{i-1}\end{aligned}$$

which after rearranging the equation, yields

$$\beta = \frac{v_{i-1}}{2\left(\tau^2 + \sigma_i^2\right)^2 + v_{i-1}} \quad \approx \quad \frac{v_{i-1}}{2\left(\hat{\tau}_i^2 + \sigma_i^2\right)^2 + v_{i-1}}\;, \tag{52}$$

where the real τ^2 is replaced by the most recent estimation $\hat{\tau}_i^2$. As outlined by algorithm 2, the state of one voxel can be defined by $(\hat{\mu}_i, \hat{\tau}_i, v_i, W_i)$ and updated using equations (43 - 52). By estimating τ parallel to μ, the estimation variance $\hat{\sigma}_i^2 = 1/W_i$ of the current estimate can be specified voxel-wise and converges with different speed, depending on the local noise behavior of the past measurements.

An example for this behavior is illustrated in figure (22) where the sensor was aimed

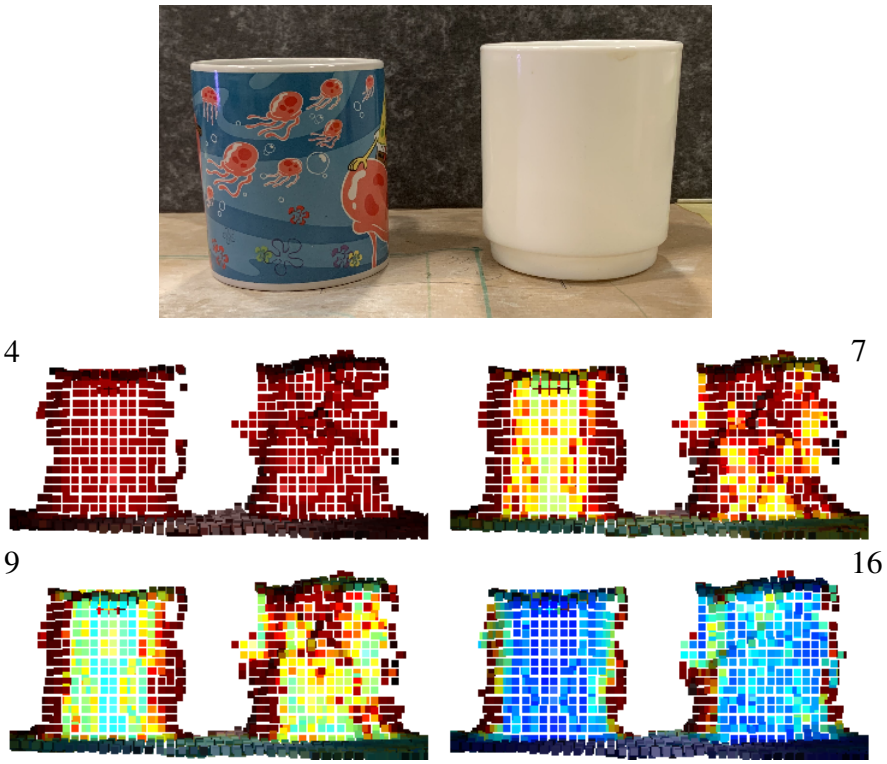

Figure 22: The convergence of $1/W_i$ for noisy surface segments (right) vs. less noisy surfaces (left) after 4, 7, 9 and 16 views. The points color indicates the corresponding estimation variance $1/W_i$, where red signals a high variance and dark blue represents values close to zero.

head-on at the two cups shown in the top image and remained static during the recording. Due to the less textured surface of the right cup, disparity errors happen more frequently and manifest themselves as strong noise. Compared to this, the left, distinctly textured cup shows a less noisy behavior. The vertices of the surface reconstruction are depicted after four, seven, nine and sixteen measurements. The coloring indicates the estimation variance $\hat{\sigma}_i^2 = 1/W_i$ of the closest voxel to the surface vertices, where red areas indicate high values of $\hat{\sigma}_i^2$ and estimation variances close to zero are colored in dark blue.
Its worth noting that in order for $\hat{\tau}^2$ to be an unbiased estimator we must allow for negative values. However, when determining the update weights in equations 44 and 52 we clip $\hat{\tau}^2$ to zero.

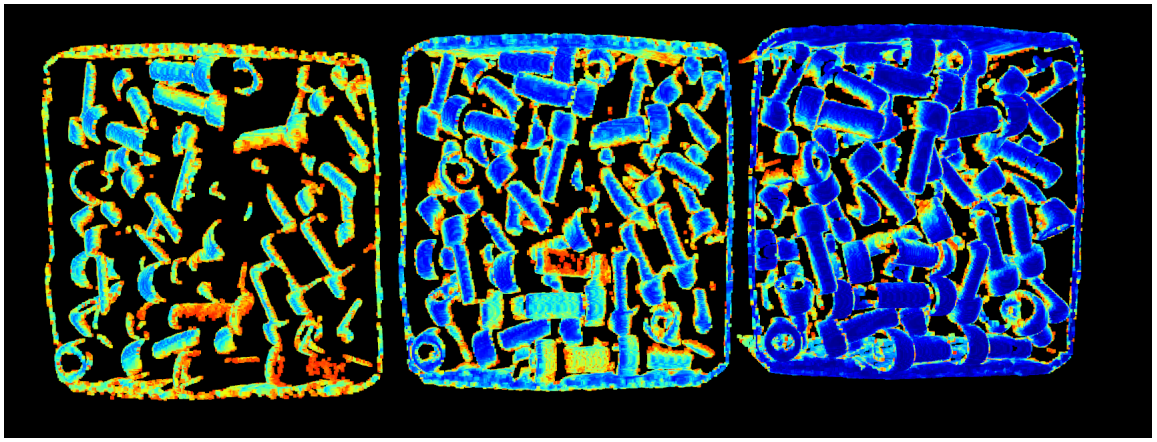

Figure 23: The evolution of σ_i^2 for the full-bin scenario with chrome screws after 3, 7 and 20 integrated frames. Red areas indicate high corresponding values of $\hat{\sigma}_i^2$ whereas estimation variance close to zero are colored in dark blue.

A second example of the progression of $\hat{\sigma}_i^2 = 1/W_i$ over time can be seen in figure 23. A bin filled with chrome screws was reconstructed after three, seven and twenty depth measurements taken from different points of view. The data and corresponding camera poses are taken from the ROBI dataset (J. Yang, Gao, et al., 2021). Similar to Fig. 22, the coloring indicates the estimation variance, where red areas indicate high values of $\hat{\sigma}^2$ whereas estimation variance close to zero are colored in dark blue. We extract the surface of the object via linear interpolation of two voxels at both sides of the zero-crossing. Let $(\mu_{1,i}, \mu_{2,i})$ be the estimated TSDF values of two adjacent voxels and $(\sigma_{1,i}^2, \sigma_{2,i}^2)$ their corresponding estimation variances. We determine a surface point between the voxels if the following conditions are met:

$$\begin{aligned} \mu_{1,i} \quad \mu_{2,i} < 0 \quad , \\ \sigma_{1,i} < \sigma_{thr} \quad \wedge \quad \sigma_{2,i} < \sigma_{thr} \quad , \end{aligned} \tag{53}$$

where $\sigma_{1,i} = \sqrt{1/W_{1,i}}$ with W_i being the accumulated weight of the voxel at time step i (see algorithm 2, where we omitted the timestamp i for the sake of readability). The empirical threshold parameter σ_{thr} roughly corresponds to three measurements. The application of σ_{thr} is a necessary operation to reject voxels that have not yet converged and

Figure 24: The seven different object types of the ROBI dataset (J. Yang, Gao, et al., 2021) and their respective full bin and low bin scenario.

overall improves the quality of the reconstructed surface. If the conditions were met, the extracted surface vertex position $\hat{p}_{v,i}$ is the linear interpolation between the voxel centroid positions p_1 and p_2 according to their corresponding signed distance estimations

$$\hat{p}_{v,i} = \frac{|\hat{\mu}_{1,i}|}{|\hat{\mu}_{1,i}| + |\hat{\mu}_{2,i}|} p_2 + \frac{|\hat{\mu}_{2,i}|}{|\hat{\mu}_{1,i}| + |\hat{\mu}_{2,i}|} p_1 \quad . \tag{54}$$

The conditions in equation (53) are applied to every pair of N6-neighboring voxels and the interpolation in equation (54) for every pair that meets the requirements. This process is highly parallelizable and computed on the GPU. The surface extraction algorithm (as well as the rest of the TSDF pipeline) was originally adopted from the work of Dong et al. (2022) who implemented a "traditional" TSDF fusion algorithm that is available in the *Open3d* library. However, it was highly modified in order to implement the proposed algorithm and fit the requirements of our application.

3.3.1 Quantitative Evaluation on Publicly Available Dataset

We evaluate the proposed approach on the ROBI dataset (J. Yang, Gao, et al., 2021) as it closely matches the intended use case of our work. The dataset consists of close-up depth images of six different object types that are stacked in a box and remain static during the scanning procedure. Each object type was scanned in two different scenarios. The Full Bin scenario, where the box was filled to the brim and the Low Bin scenario where only a few objects were used (see figure 24). All seven object types exhibit reflective behavior to varying degrees. For each scene the robot arm moved to a series of

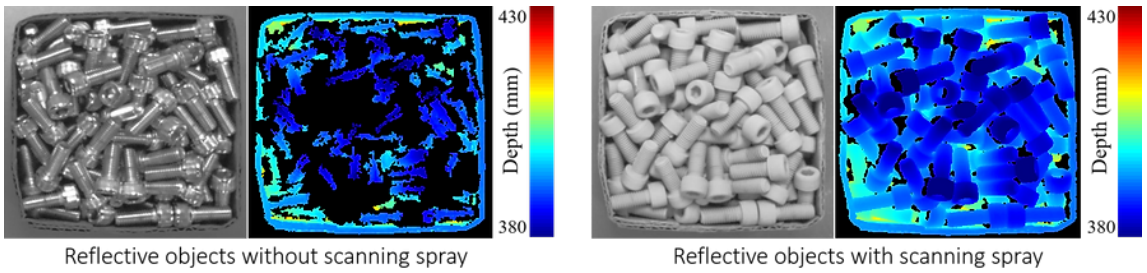

Figure 25: The Full Bin scenario of the chrome screw object. The left images (infrared, depth) show the "natural" recordings and the ones on the right show the same scene after the scanning spray was applied.

poses on a view sphere centered on the box, from approximately 45°to 90°of elevation with different distances to the box and captured depth images with a depth sensor that is mounted on its end effector. The target application of the high-cost depth sensor is an industrial one and in our experience its measurements are significantly more accurate than those of consumer-level sensors. The camera poses for each frame that are provided by the forward kinematic of the robot were additionally refined via an ICP-algorithm using calibration spheres and are therefore highly accurate. Several scenes were recorded a second time under identical conditions with the only difference being that a scanning spray was applied beforehand which greatly reduced the measurement noise. These scenes are treated as ground truth data. An example for the difference is given in figure 25. We only evaluate scenes for which these ground truth images are available. The full-bin category represents a large number of mostly metallic objects stacked in a box, where 106 depth images with corresponding sensor poses are provided. The low-bin category is similar, but with a considerably lower number of objects and only 68 depth images are provided. Since the scenes of this dataset contain a high number of very detailed objects, the traditional constant weighting along the whole length of the view ray can lead to degraded results. Similar to our previous publication (Schaub et al., 2022), we therefore implemented three weighting functions that were originally proposed by Bylow et al. (2013), the narrow exponential weight function w_{exp}, the narrow linear weight function w_{lin} and the narrow constant weighting function w_{const}. The weighting functions w_{exp} and w_{lin} are

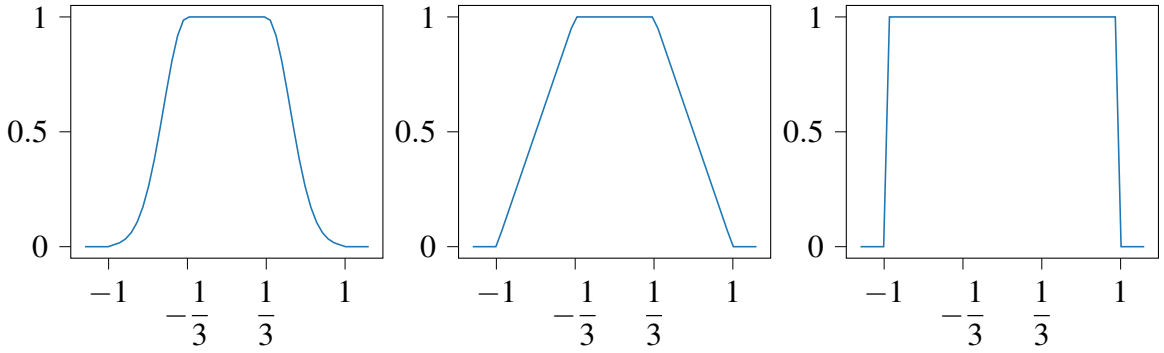

Figure 26: The three integrated weighting functions w_{exp}, w_{lin}, w_{const} proposed by Bylow et al. (2013) with respect to signed distance t on the x-axis. t is scaled to the truncation distance ξ. Hence, the relevant range is $[-1,1]$. All three functions weight measurements whose magnitude is greater than the truncation distance ξ with zero. A threshold of $1/3$ was selected for the transition to full weighting in case of w_{exp} and w_{lin}.

given by

$$w_{exp}(t) = \begin{cases} 0 & if \ \ |t| > 1 \\ 1 & if \ \ |t| < 1/3 \\ e^{-12\left(t - sgn(x)\frac{1}{3}\right)^2} & else \end{cases} \quad w_{lin}(t) = \begin{cases} 0 & if \ \ |t| > 1 \\ 1 & if \ \ |t| < 1/3 \\ \frac{|x|+1}{1-\frac{1}{3}} & else \end{cases} . \tag{55}$$

A visualization of all three weighting functions is depicted in figure 26.

We compare our results against thenon-probabilisticc TSDF implementation of Dong et al. (2022) and the results of J. Yang, Li, and Waslander (2021) and Schaub et al. (2022). The results of J. Yang, Li, and Waslander (2021) are taken from the corresponding publication. Its worth noting that although they reference the ROBI dataset, the reported number of views per scene does not match with the number of currently available views. This might imply that only a subset of the dataset was available at the time of publication, making a comparison unrepresentative. All other reported results are the product of our own experiments with the available open source implementations or self-written implementations, as in the case of the three weighting functions w_{exp}, w_{lin} and w_{const}. For each algorithm, the scene was reconstructed once with the noise-free ground truth data to

Table 3: Parameters used for the evaluation.

Inlier threshold		2 mm
Initial τ^2 estimation	$\hat{\tau}_0^2$	0.8
Initial v value	v_0	0.8
Object distance threshold		1 cm
Voxel size		0.5 mm
Truncation distance	ξ	1.5 mm

obtain the point cloud S and once with the noisy data to obtain the point cloud $\hat{S}$. The results of each algorithm are compared to their respective ground truth reconstruction to avoid bias. The relevant settings we used to evaluate all results can be seen in table 3.

The voxel size, the truncation distance as well as the inlier-threshold were adopted from J. Yang, Li, and Waslander (2021) and remain constant for all scenes. For the sake of a fair comparison, we utilize a standard strategy proposed by Bylow et al. (2013) and ignore all voxels that received less than W_{thr} measurements during the surface extraction of Dong et al. (2022) and the three weighting functions, where W_{thr} is set to 3 as suggested by J. Yang, Li, and Waslander (2021).

For the estimation of the current sensor noise of the *Ensenso N35* in equation (40), we use the model function of Halmetschlager-Funek et al. (2019). To the best of our knowledge this is currently the only empirical noise model function for the *Ensenso N35* sensor in the literature. Halmetschlager-Funek et al. (2019) represent the measurement variance solely as a function of the distance. We found that the angle of incidence plays a non-negligible role in the noise behavior of stereo sensors. Hence, we supplement the function with an additional angular component that was originally determined by Ahn et al. (2019) for the *Intel Realsense D435*. Although this is a heuristic addition and is not based on a corresponding series of measurements, we were able to achieve better results than without the addition of the angle component.

Each point $\hat{p} \in \hat{S}$ is compared to the closest point $p \in S$ and the following metrics were used to evaluate the results

- *Mean Point-to-Point Distance*: We build correspondences from $\hat{S}$ to S. The mean point-to-point distance is computed over all inliers and measures the reconstruction accuracy. Similar to J. Yang, Li, and Waslander (2021), we define inliers as

correspondences where the Euclidean distance is smaller 2 millimeters.

- *Outlier fraction o*: Given correspondences from $\hat{S}$ to S, o is defined as the fraction of outliers over the total number of ground truth surface points $|S|$.
- *Scene Completeness sc*: Given correspondences from S to $\hat{S}$, the scene completeness is computed as the fraction of the number of inliers over $|S|$.
- *F-Score f*: Given the scene completeness sc and the fraction of outliers o the F-Score is calculated as

$$f = 2\,\frac{sc\;\; r}{sc + r} \quad \text{, with} \quad r = 1 - o\,. \tag{56}$$

Similar to J. Yang, Li, and Waslander (2021) we only evaluate correspondences that are close to the object, and we reject correspondences on the comparatively easy to reconstruct table surface or the box itself. For this purpose, we use the ground truth poses of the objects and associated mesh files provided by J. Yang, Gao, et al. (2021) and discard every point in $\hat{S}$ and S that has a distance to the mesh that is greater than one centimeter. Table 4 shows the summarized results, where for the sake of readability the fractions o and sc were reported as percentages. The results of J. Yang, Li, and Waslander (2021) were adopted from the corresponding publication.

Table 4: Results using the ROBI Dataset

Bin	Object Category	Point Distance (mm)							Outliers (%)							Completeness (%)							F-Score (%)						
		*1**	*2**	*3**	*4**	*5**	*6**	*7**	*1**	*2**	*3**	*4**	*5**	*6**	*7**	*1**	*2**	*3**	*4**	*5**	*6**	*7**	*1**	*2**	*3**	*4**	*5**	*6**	*7**
Full	Large Size	0.19	0.18	0.19	0.29	0.25	0.19	0.14	2.8	2.9	3.6	4.5	4.1	3.6	1.6	99.3	99.3	99.2	99.0	99.3	99.2	99.6	98.2	98.2	97.8	97.2	97.6	98.8	99.0
	Complex Shape	0.29	0.28	0.29	0.35	0.39	0.28	0.21	1.1	1.0	1.2	1.7	1.9	0.8	0.3	90.6	90.7	90.7	91.2	96.8	91.5	92.0	94.6	94.6	94.6	94.6	97.5	95.1	95.7
	High Gloss	0.32	0.30	0.31	0.37	0.41	0.29	0.22	0.9	0.9	1.0	1.5	1.5	0.6	0.2	91.3	91.4	91.5	91.9	91.5	92.1	92.7	95.0	95.0	95.0	95.0	94.9	95.5	96.1
Low	Large Size	0.28	0.27	0.28	0.40	0.13	0.27	0.20	5.7	5.8	7.3	9.9	0.4	7.6	2.2	96.4	96.3	96.5	94.7	99.7	97.2	96.7	95.3	95.2	94.5	92.3	99.6	95.0	97.3
	Complex Shape	0.34	0.33	0.34	0.42	0.30	0.29	0.22	1.4	1.4	1.7	2.9	0.4	1.1	0.4	90.0	89.8	90.0	89.7	79.9	90.3	91.4	94.1	94.0	93.9	93.2	88.7	94.3	95.3
	High Gloss	0.33	0.32	0.33	0.40	0.24	0.28	0.20	0.9	0.9	1.0	1.8	0.2	0.6	0.2	89.9	89.6	89.4	90.4	77.0	89.8	95.0	94.2	94.1	93.8	94.0	87.0	94.3	97.3
Total		0.31	0.30	0.31	0.38	0.34	0.27	0.21	1.5	1.5	1.8	2.6	1.6	1.4	0.5	91.6	91.5	91.5	91.8	91.1	92.0	93.9	94.8	94.8	94.6	94.4	94.6	95.0	96.6

In order to keep the above table within the page boundaries, the authors of the respective algorithms were abbreviated as follows:

- *1** Bylow et al. (2013) w_{exp}
- *2** Bylow et al. (2013) w_{lin}
- *3** Bylow et al. (2013) w_{const}
- *4** Dong et al. (2022)
- *5** J. Yang, Li, and Waslander (2021) [a]
- *6** Schaub et al. (2022)
- *7** Proposed algorithm

[a] Although J. Yang, Li, and Waslander (2021) references the ROBI dataset, we noticed that the reported number of views per scene does not match with the currently available dataset. Only a subset of the data might have been available at the time of publication, making a comparison unrepresentative.

Compared to the baseline methods, we (columns *7*) were able to achieve a significantly reduced fraction of outliers, slightly lower point-to-point errors, as well as a higher scene completion. Noteworthy is the difference between the implementation of Dong et al. (2022) (*4*) vs the constant weighting function (*3*). The only difference between these two approaches is that Dong et al. (2022) integrates all signed distance measurements greater than the truncation distance ξ, whereas the constant weighting function weights all positive measurements that are greater than ξ with zero, effectively ignoring them. It can be seen that the implementation of Dong et al. (2022) shows greater point-to-point distances as well as a greater fraction of outliers. The reason for this is obvious in retrospect. If the noisy measurements are unbiased, then the asymmetric rejection (relative to the true value) of measurements introduces a bias which in our case leads to higher errors. However, by ignoring measurements outside the truncation band, the information whether the space is unknown or known and free or is lost. This can also be seen in the experiments of Bylow et al. (2013) where they show that ignoring measurements of free space leads to significantly higher tracking errors regardless which weighting function is chosen. When the sensor-pose of each frame is known via a high-precision robot arm, however, the reconstruction of the surface becomes worse.

For our implementation we mark all free voxels that were seen outside the truncation band with $\mu = NaN$. These voxels are ignored during the surface extraction step and keep this value until they experience a "valid" measurement where $|t \leq \xi$. This way keep the information whether the space is known and free or unknown but do not sacrifice the accuracy of $\hat{\mu}$. Figure 27 shows a typical example of the reconstruction of our algorithm (center) and the reconstruction of Dong et al. (2022) (right). It can be seen that large point-to-point distances are concentrated in a few scattered outlier, butt the areas are generally estimated more accurately.

Where the proposed algorithm stands out is the significantly reduced fraction of outliers. This emphasizes the superiority of thresholding using the estimated variance $\hat{\sigma}^2$ by the proposed algorithm over the weight based methods that were used for the experiments with the open source implementation of Dong et al. (2022) as well as for the three weighting functions. The thresholding operation is a commonly used operation to remove unreliable points from the point cloud but its impact differs depending on the predictive power of $\hat{\sigma}$.

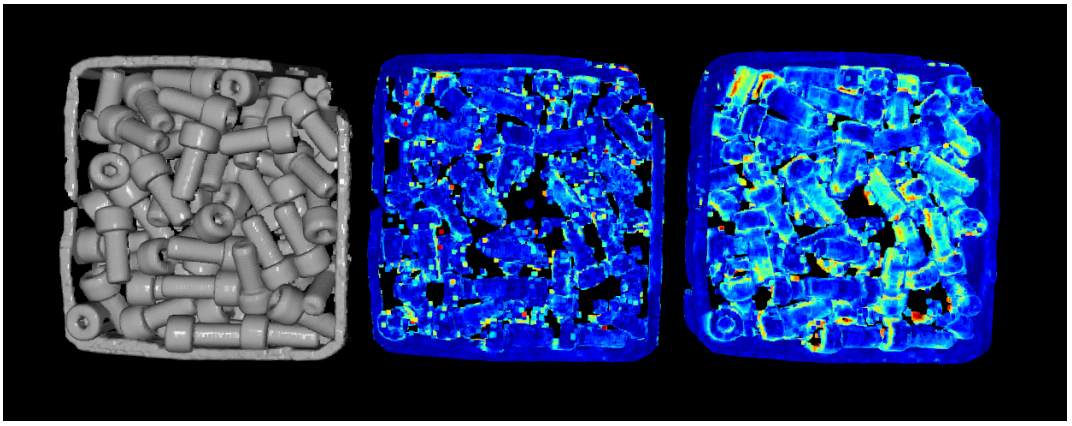

Figure 27: The ground truth reconstruction of a chrome screw full-bin scenario is shown at the left. The right point cloud represents the reconstruction using the algorithm of Dong et al. (2022) and center point cloud is the result of the proposed approach. The color indicates the error level. Points further away than 2mm from the ground truth reconstruction are depicted in dark red.

In order to evaluate the correlation of the estimated standard deviation of the point-to-point error $\hat{\sigma}_p$ and the actual error e, we calculate the non-thresholded (i.e. $\sigma_{thr} = \infty$ and $W_{thr=0}$) mean error $\overline{ME}$ of points within a distance limit of one centimeter to the ground truth object meshes. Additionally, we compute the weighted average error $\overline{WE}$ for the proposed algorithm, the work of Dong et al. (2022) and the three weighting functions. Each point-to-point distance error $|\hat{p} - p|_2$ is weighted according to the corresponding estimated inverse standard deviation $\hat{\sigma}_p$, where we define $\hat{\sigma}_p^2 = \frac{1}{\min(W_1, W_2)}$ where W_1 and W_2 represent the accumulated weight of both voxels that were involved in the 3D-point estimation. The weighted average error $\overline{WE}$ and the mean error $\overline{ME}$ are then given by

$$\overline{ME} = \frac{\sum_i^N |\hat{p}_i - p_i|_2}{N} \qquad \text{and} \qquad \overline{WE} = \frac{\sum_i^N (|\hat{p}_i - p_i|_2 \,/\, \hat{\sigma}_{p,i})}{\sum_i^N (1 \,/\, \hat{\sigma}_{p,i})}.$$

Setting $\hat{\sigma}_p^2 = \frac{1}{min(W_1, W_2)}$ is related to the standard threshold holding operation (see eq. 53) and can be regarded as equating the estimation variance of the 3D-point with the greater one of both involved voxels.

Table 5: Mean errors, corresponding weighted average and their ratio.

	$\overline{ME}$	$\overline{WE}$	$\overline{WE}/\overline{ME}$
w_{exp}	0.848	0.435	0.513
w_{lin}	0.817	0.424	0.519
w_{const}	0.792	0.438	0.553
Dong et al. (2022)	0.877	0.496	0.566
proposed alg.	0.781	0.176	0.226

We want to evaluate the ratio of $\overline{WE}/\overline{ME}$ for each algorithm. The basic idea is that, if σ_p is a reasonable predictor for the true error, then large errors $|\hat{p}-p|_2$ are "canceled" out by corresponding large σ_p and $\overline{WE}/\overline{ME}$ will be significantly smaller than 1. Conversely, nonsensically chosen weights would lead to ratios greater than 1. Table (5) shows the results.

It can be seen that the mean error of our algorithm is slightly smaller than the mean error of the baseline implementations. However, the weighted error shows a significant difference and the ratio $\overline{WE}/\overline{ME}$ of the proposed algorithm is better by a factor of more than two. This implies that our $\hat{\sigma}$ is a much more reliable estimation for the true error which is crucial for our subsequent algorithms. It implies that $\hat{\sigma}$ be used to distinguish well reconstructed areas from questionable ones, derive success probabilities for grasp options based on estimations for the local reconstruction quality and last but not least, it gives us the option to wait or explore the scene until additional measurements improve said success probabilities.

Figure 28 depicts the reasons for this ratio difference. We plot the inverse standard deviation $1/\hat{\sigma}_p = \sqrt{\min(W_1, W_2)}$ on the x-axis vs the actual error for our algorithm as well as for the constant weighting implementation. For scaling reasons, we excluded the smallest and largest five percent of $\hat{\sigma}_p$ values and divide resulting range in 16 equally spaced ranges. For each range, we show the median error in red, the inner 25% error range in brown and the inner 50% i.e. the interquartile range (IQR) in blue. After an initial settling phase, the proposed algorithm (left side) shows an approximately linear decreasing relationship between the error and $1/\hat{\sigma}_p$ whereas for the constant weighting method (right side) no such pattern is visible. On the contrary, the median error slightly rises in the second half. The other two weighting methods w_{lin} and w_{exp} display a similar behavior.

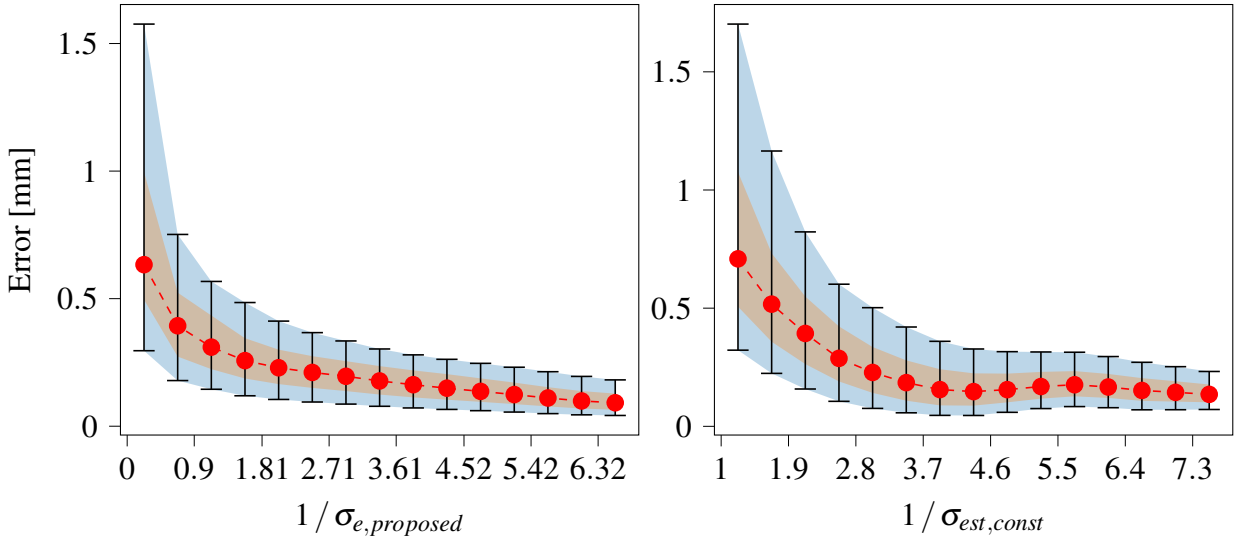

Figure 28: The inverse estimated standard deviation versus the actual error. The results of our algorithm are shown on the left and those using the constant weighting function on the right. The median error is depicted in red, the inner 25% error range in brown and the inner 50% (i.e. interquartile range IQR) error range in blue.

As a side note, the two signed distance estimations of neighboring voxels that lead to the point estimation between the voxels could be interpreted as two independent but shifted (by one voxel size) measurements of the same quantity (see eq. 53 and 54). This interpretation leads to $\hat{\sigma}_p^2 = 1/(W_1 + W_2)$. Another interpretation is that of the weighted average of two random variables where $\hat{\sigma}_p^2 = r^2 \frac{1}{W_1} + (1-r)^2 \frac{1}{W_2}$ with $r = |\mu_2| / (|\mu_2| + |\mu_2|)$. However, both interpretations lead to consistently greater weighted errors for all considered algorithms. This could imply that the implicit independence assumption of these two interpretations is flawed because neighboring voxels are often updated by the same measurement along a single view-ray. The definition in the previous paragraph with $\hat{\sigma}_p^2 = 1/\min(W_1, W_2)$ could be interpreted as a worst case evaluation and is consistent with the common practice of thresholding the individual estimations instead of the interpolation when extracting the surface.

3.3.2 Adaption to the Intended Application

The *Ensenso N45* depth sensor that was used in the ROBI dataset is unsuitable for the assistive eye-in-hand setup we are aiming for. The force of assistive robot arms is restricted for the sake of safety. For example, the *Kinova Jaco* arm is limited to a load of one kg at maximum reach (*Specifications of Jaco assistive robotic arm*, 2025). The weight of the *Ensenso N45* sensor (550 g) would significantly restrict the choice of objects to grasp.
Additionally, its physical dimensions severely restrict the arm's freedom of movement (more on that on the next chapter) and its price is considerable. In the following work we focus on the *Intel Realsense D435* sensor which is widely used in robotics and fulfills the criteria of our application: small minimal range, low price, small size and low weight. However, compared to the *Ensenso N45*, these advantages come at the price of significantly reduced accuracy.
In the proposed algorithm (see alg. 2) the estimation $\hat{\tau}^2$ is updated using the quadratic difference of measurement x_i from the latest $\hat{\mu}_i$ where the update factor β_i contains the fourth central moment of the current estimation of the underlying distribution $N\left(0, \sigma_{s,k}^2 + \tau^2\right)$. This leads to a fourth-order system of update-equations with a circular dependency of estimated values. The performance of the proposed recursive update scheme is therefore highly dependent on reasonably chosen initial guesses for $\hat{\tau}_0$ and v_0. The variance $v_0 = Var(\hat{\tau}_0^2)$ can be seen as a measure of confidence in the initial guess $\hat{\tau}_0$. v_0 determines the weighting of new τ^2 measurements $\left(\overset{*}{t}_k^2 - \sigma_{s,k}^2\right)$ and therefore the (initial) reaction speed of $\hat{\tau}^2$. If it is chosen too large, initial erroneous squared differences can bring the estimate significantly out of the range of values that are expected in reality and thus significantly slow down convergence of $\hat{\tau}^2$. If, on the other hand, it is chosen too small, then new measurements have a negligible influence and the initial guess $\hat{\tau}_0^2$ dominates, i.e. $v_0 = 0$ completely prevents an update and $\hat{\tau}_i^2 = \hat{\tau}_0^2$.
While we were able to find suitable parameters for the prior parameters $(\hat{\tau}_0^2, v_0)$ the previously discussed experiments with the ROBI-dataset (J. Yang, Gao, et al., 2021) in an empirical manner, finding appropriate parameters for the intended eye-in-hand eye in hand setup with the *Realsense D435* is not as trivial since we have no reliable ground truth values to compare the results against. We need to cover a large spectrum of situations as well as possible, instead of fine-tuning the parameters to the one specific situation of a data set.

We consult the literature to obtain an estimate of the magnitude of the expected noise in different scenarios and opt for a grid search in a Monte Carlo simulation in order to find a tuple $(\hat{\tau}_0^2, v_0)$ that operates reliably throughout.

- **Geometric considerations**:
  Since we use an eye-in-hand setup and the intended application is grasping, we are mainly interested in the space within the reach of the robot arm. The maximum reach (base to end effector) of the *Kinova Jaco* robot arm is 0.9 m (*Specifications of Jaco assistive robotic arm*, 2025) and therefore a maximum measuring distance of 1.2m appears to be a reasonable assumption. The minimum measuring distance of the *Intel Realsense D435* is specified as 0.2 m (*Intel ® RealSense TM D400 Series Product Family*, 2017). Hence, we model the distance measurements d within this range to be equally likely,

  $$d \sim \mathscr{U}\left(0.2\text{ m}, 1.2\text{ m}\right) \quad . \tag{57}$$

  C. V. Nguyen et al. (2012) proposes to reject measurements where the angle of incidence θ_y is greater than 70°, which we found to be too restrictive. We opt for a less restrictive threshold of 85° in order to be able to consider less reliable measurements but still avoid the dramatic increase in error size for angles close to 90 degrees (see figure 21). Similar to the measured depth, we model the angle of incidence θ to be uniformly distributed,

  $$\theta_y \sim \mathscr{U}\left(0.0^\circ, 85^\circ\right) \quad . \tag{58}$$

- **Magnitude of the noise**:
  The work of W. Kim et al. (2023) suggests a material-dependent difference of the standard deviation of roughly seven millimeters in the distance range that we consider. The empirical evaluation of R. Chen et al. (2022) suggests a difference of four millimeters for our sensor depending on the illumination of the scene. Hence, we opt for a conservative estimate for the maximum expected additional standard deviation τ_{max} of 1.5cm. We expect the proposed algorithm to produce satisfactory performance in the entire range $[0, \tau_{max}]$. Hence, we partition $[0, \tau_{max}]$ into a set of

Table 6: Parameters of the Monte Carlo grid search.

depth measurements	z	$\sim$	$\mathscr{U}(0.2, 1.2)$	[m]
angle of incidence	θ_y	$\sim$	$\mathscr{U}(0, 85)$	[°]
number of measurements	t_{max}	$=$	60	
surface variance	τ^2	$\in$	$\left\{\frac{k}{4}\tau^2_{max} \mid k \in \{0,..,4\} := K\right\}$	[m^2]
Initial τ^2 estimation	$\hat{\tau}^2_0$	$\in$	$\left\{\frac{l}{8}\hat{\tau}^2_{0,max} \mid l \in \{0,..,8\}\right\}$	[m^2]
Variance of τ^2_0	v_0	$\in$	$\left\{\frac{m}{8}v_{0,max} \mid m \in \{0,..,8\}\right\}$	[m^4]
number of samples	N	$=$	10000	

five equally spaced intervals.

For the grid search for suitable tuples $(\hat{\tau}^2_{0,opt}, v_{0,opt})$, the parameter domain $[0, \hat{\tau}_{0,max}] \times [0, \hat{v}_{0,max}]$ was also evenly partitioned into nine intervals, where $\hat{\tau}_{0,max} = 3\,\text{cm}^2$ and $\hat{v}_{0,max} = 5^{-8}\,\text{m}^4$ were determined empirically. An overview of the employed parameters of the Monte Carlo Grid search is given in table 6. For each grid cell in the $5 \times 9 \times 9$ grid, we conduct 100000 randomized trails where the unbiased measurements $t \sim \mathscr{N}\left(0, \sigma^2_z(d, \theta_y) + \tau^2_k\right)$ are treated according to the proposed algorithm 2. The values $\sigma^2_z(d, \theta_y)$ are obtained using the sensor noise model in equation 40.
For the sake of readability the considered combinations of initial estimates $(\hat{\tau}^2_{0,l}, v_{0,m})$ will be abbreviated using their corresponding grid indices $(l, m) \in [0, ..., 8] \times [0, ..., 8]$. Similarly, τ^2 values are abbreviated using the index k. We use the metrics known from the previous chapter to evaluate the results of the Monte Carlo grid search:

- *Mean Error*: Let $\hat{\mu}_n(k, l, m)$ be the one of N estimations after 60 measurements. The mean error of (l, m) is then calculated by

$$\overline{ME}(l, m) = \frac{\sum_k^{|K|} \sum_{n=1}^{N} \hat{\mu}_{klmn}}{N \ |K|} \tag{59}$$

- *Weighted Error*: Let R be the accumulated weight after 60 measurements corre-

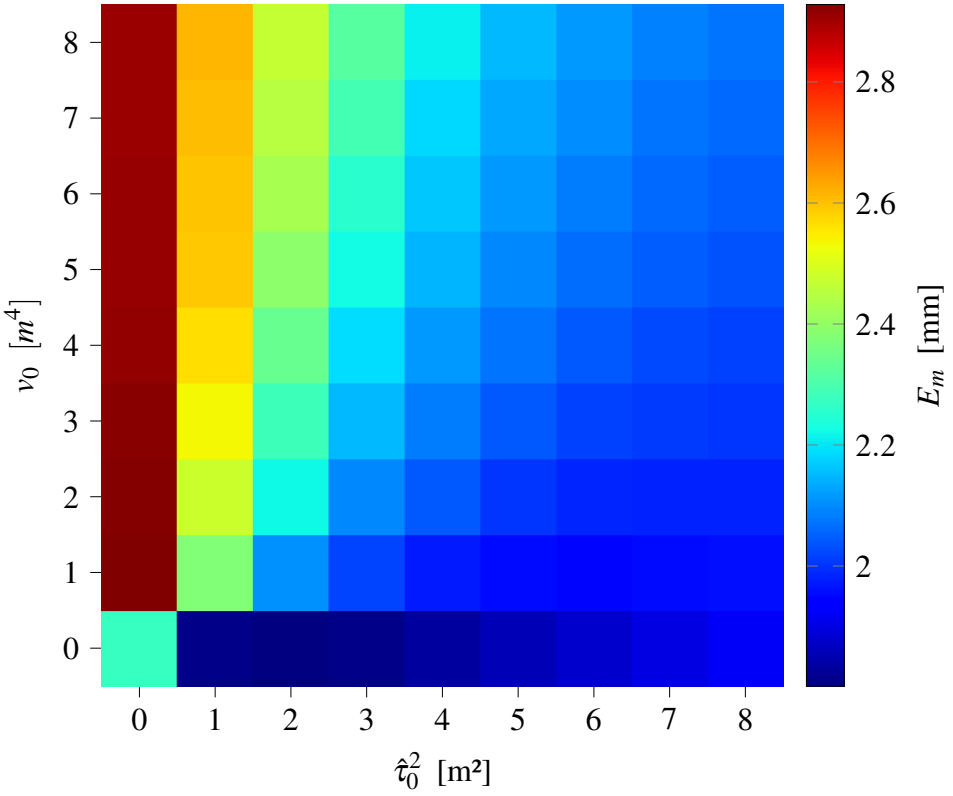

Figure 29: The mean error over all N trials and $|K|$ considered τ_k after 60 measurements for each combination of initial estimation parameters. A *jetmap* color scale was used and low errors are colored in dark blue and high errors in red.

sponding to $\hat{\mu}$. The weighted error of grid cell (l,m) is then calculated by:

$$\overline{WE}\,(l,m) = \frac{\sum_k^{|K|}\sum_{n=1}^{N}\hat{\mu}_{klmn}\,R_{klmn}^{0.5}}{\sum_k^{|K|}\sum_{n=1}^{N}R_{klmi}^{0.5}} \qquad \text{where} \qquad R_{klmn} = \sum_i^{60}\frac{1}{\sigma_{z,i}^2 + \hat{\tau}_{ki}^2}\ . \tag{60}$$

Compared to the results with the ROBI dataset in the previous chapter, the weighted error has a slightly different implication here. All estimations received the same number of measurements with similar, although randomized noise. Hence, the weighted error compared to the mean error can be seen as a more direct evaluation of the parallel τ estimation with regard to $Var\,(\hat{\mu}) = 1\ /\ R$.

We do not know in advance which conditions will occur, i.e. which τ we will find. We

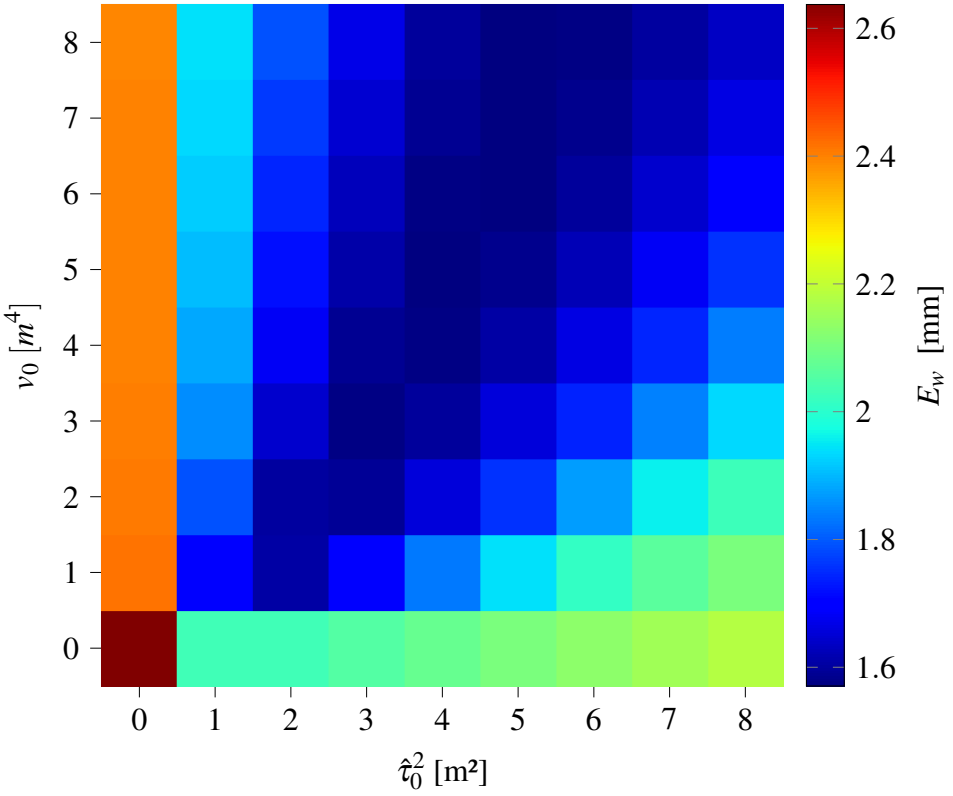

Figure 30: The weighted error over all N trials and $|K|$ considered τ_k after 60 measurements for each combination of initial estimation parameters. A *jetmap* color scale was used and low weighted errors are colored in dark blue and high weighted errors in red.

seek appropriate results for all conditions.Therefore, we sum over all $|K|$ considered surface variances τ^2 for both error metrics. This is equivalent to equally weighting all considered conditions which effectively reduces the 3D-error grid to two dimensions and the results are visible in figure 29 and 30.

It can be seen that the highest errors mean errors $\overline{ME}$ occur if the initial τ estimation $\hat{\tau}_0$ is set to zero. Unsurprisingly the smallest error can be found at indices $(2,0)$, where $\hat{\tau}_0^2$ is equal to the middle one of the five considered τ scenarios, i.e. equal to $\frac{2}{4}\tau_{max}^2$, and $\hat{v}_0 = 0$ which fully prevents updates of $\hat{\tau}$.

It is of great importance for our following algorithm to have a metric that reliably estimates the reconstruction quality, which is why we additionally evaluated the weighted

error $\overline{WE}$. Similarly, the graph shows the highest errors for $\hat{\tau}_o = 0$ and a maximum at $(0,0)$ which can be seen as ignoring the additional τ noise during the fusion of measurements. Interestingly, for the whole first horizontal axis with $v_0 = 0$, the weighted error $\overline{WE}$ is higher than $\overline{ME}$. Hence, $\overline{WE}/\overline{ME} > 1$ and therefore R is unsuitable for estimating the reliability of the corresponding $\hat{\mu}$. The estimation process for the comparatively benign index $(3,3)$ can be seen in figure 31 where the estimation progress over the 60 measurements on the x-axis of $\hat{\mu}$ is shown in the left column and $\hat{\tau}$ in the right column. The five different settings for the actual τ are displayed in ascending order top-to-bottom. The red curves show the median as well as the interquartile range. For comparison the optimal estimator (inverse variance weighting with known sensor and surface variance $(\sigma_{s,i}^2, \tau^2)$) is shown on the left side in green. One can see that the $\hat{\mu}$ estimation has the largest distance to the optimal estimator at $\tau^2 = 0$ (top-left). This indicates that under optimal scanning conditions i.e. $\tau = 0$ our estimator has a lower convergence rate compared to using only the sensor model since the initial high $\hat{\tau}$ leads to non-optimal weighting, and it takes some time before the influence of $\hat{\tau}^2$ can be neglected (see the top right graph). Its worth noting that if a typical stereo sensor with a frame-rate of 30 Hz is used, these graphs represent the first two seconds of the data fusion. The other four scenarios where $\tau^2 > 0$ show an estimation history that is somewhat close to the optimal estimator. The right column shows the median of the $\hat{\tau}^2$ estimation progress as well as the corresponding interquartile range in red and the true τ^2 value on in green. It can be seen that for all five scenarios our parallel $\hat{\tau}^2$ estimation converges towards the true value using these settings.

3.4 Discussion

We proposed a novel algorithm to reconstruct 3D geometries and corresponding spacial uncertainties probabilistic manner. Our approach relies on a pre-computed sensor model and is update estimated distribution parameters depending on the experienced measurement noise. This allows us to estimate the parameters of the underlying surface distribution and to obtain a more accurate estimation compared to several baseline approaches.
We showed that the local estimations of the surface reconstruction quality can be of great use for the intended, subsequent grasping algorithm and adapted the algorithm for the grasping use-case and the consumer-level sensor.

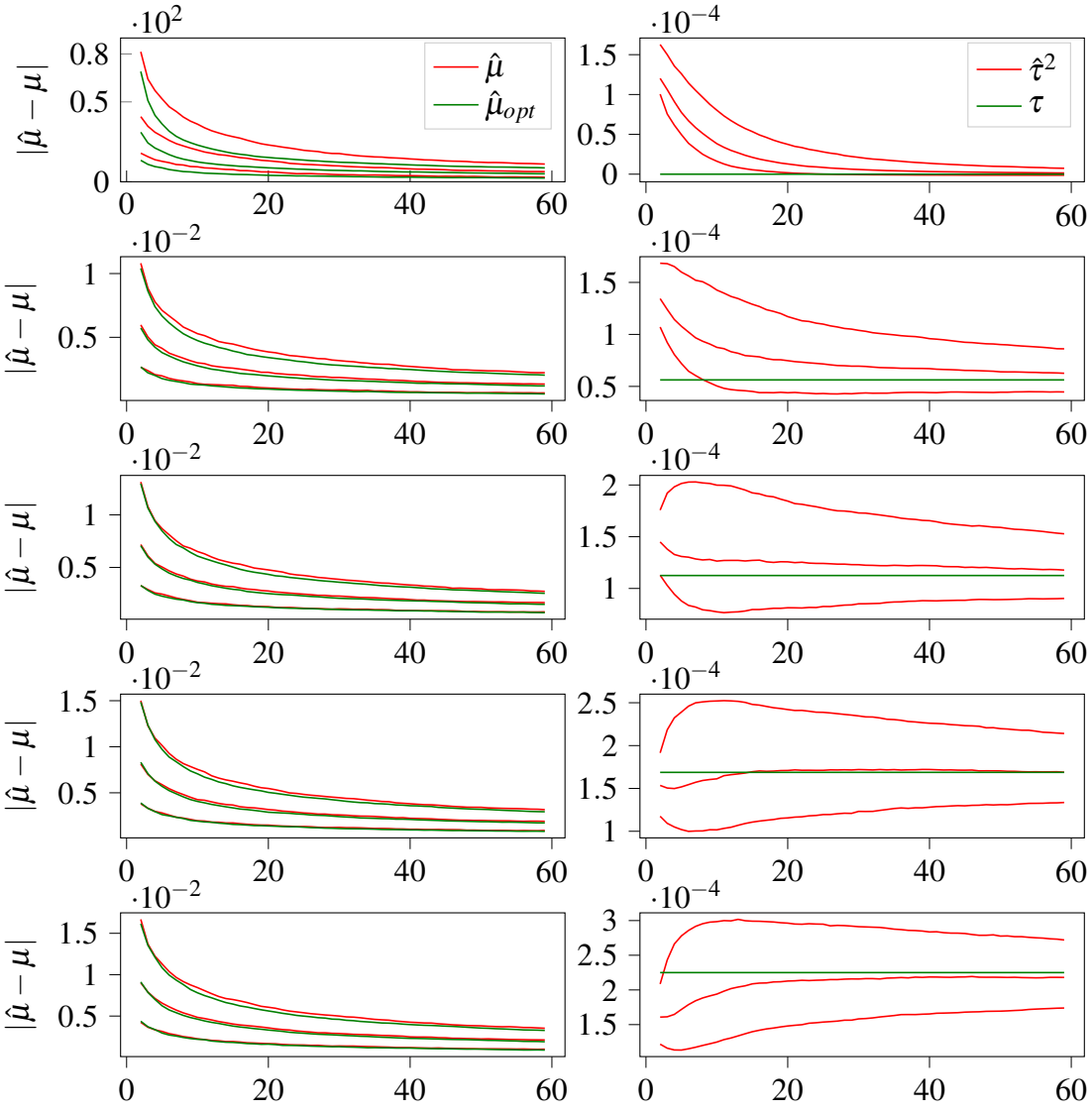

Figure 31: The median error of $\hat{\mu}$, as well as the interquartile range is shown on the left in red for all five scenarios. For the sake of comparison the same is shown for the optimal estimator. The right graphs show the history of $\hat{\tau}$ that is estimated in parallel (red) and the corresponding ground truth τ value (green).

4. Active Exploration for Robotic Manipulation

The task of grasping is a very natural task for humans. Even unknown objects can be handled with ease. An assistive robotic system faces the same problem in that it must be able to grasp unfamiliar items. In an unknown environment, this task requires a strategy to gather information about the object and surrounding scene instead of relying on existing assumptions e.g. the relative pose and shape of the objects. In the following chapter, this exploration aspect is examined and methods are provided on how the problem can be solved.

4.1 Introduction

Grasping in an uncontrolled, unstructured environment like the classical household setting faces very different challenges compared with for example grasping in an industrial setting. Whereas most grasping algorithms that are designed for industrial applications assume a benign first perspective, the same cannot be presumed for an assistive application. The object in question may be partially occluded and the first view might not even show a single, valid collision free grasp candidate.
Additionally due to quantisation effects, interfering light sources, reflective surfaces, depth discontinuities or unfavorable ray angles, the sensor data may be subject to errors.
Estimating a reliable representation of the object therefore requires the consideration of additional views and a method to move the camera towards them. Hence, it is no wonder that the eye-in-hand setup, where the sensor is mounted on the robots end-effector, is found in half of the publications on assistive robotic manipulators (Bengtson, Bak, Struijk, & Moeslund, 2019).
This can be also understood as an exploration algorithm where the system has to continually decide how to act in order to gain as much information as possible, while simultaneously taking its motion capabilities into account.
Our problem statement deviates in three major ways from traditional exploration algorithms, which are often designed for 2D-mobile robots or quadcopters. First, one of the main concerns of these algorithms is often the estimation of the sensor pose and its past

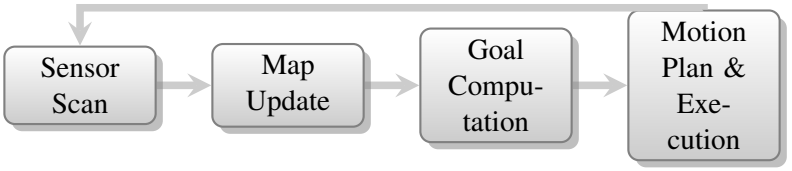

Figure 32: Typical workflow of next-best-view algorithms ((Zeng et al., 2020))

trajectory, where in our case this pose is directly calculable by the geometry of the robot that is known in advance and the current motor encoder values.
Second, the kinematics of a robot manipulator vastly differs from the kinematics of a mobile robot or quadcopter and introduces new challenges such as self collisions, singularities and non-trivial collision computation.
Finally, our goal is not to create a complete-as-possible reconstruction. Although a more complete object model is generally useful, we experienced that objects can often be grasped in numerous ways, where many of the possibilities are somewhat equivalent in quality. This implies that the goal should be more tailored towards the refinement of the scene representation than a binary distinction between known and unknown space.

4.2 Related Work and Problem Description

Most approaches for exploration with a manipulator fall into one of two categories. The first one being the sense-plan-act approach that is depicted in figure 32 which is according to Zeng et al. (2020) the typical pipeline of active vision algorithms. The second one is a continuous servoing approach.
Examples that follow the sense-plan-act strategy include the work of Y. Wang et al. (2019). They utilize a CNN to predict the best motion direction along the three axes in both directions based on the current observation in a classification-problem manner. They additionally use an occupancy grid and compute the entropy of view candidates in order to prevent the CNN from revisiting previous poses. They utilize MoveIt (Coleman et al., 2011) to plan paths in the most promising direction before repeating the cycle.
Monica and Aleotti (2017) use a TSDF to detect frontier contours between known and unknown space and sample a high number of views from which the contour is visible. Similar to Y. Wang et al. (2019), they employ the MoveIt framework (Coleman et al.,

2011) to find a feasible path to the goal and only integrate a single measurement at the next-best view configuration.
Menon, Zaenker, Dengler, and Bennewitz (2023) also use the TSDF representation and sample view candidates in the proximity of the current pose. They use a ray casting algorithm to determine the utility of each candidate via the ratio of rays that intersect unknown volume elements. The views are sorted according to their utility, serially forwarded to the MoveIt pipeline (Coleman et al., 2011) for collision-free motion planning and trajectory execution, until a successful execution can be performed.
Zhang et al. (2022) propose a network that predicts the next-best-view based on the current TSDF representation and also use MoveIt to steer the robot towards that pose and then fuse another measurement. This process is repeated until a promising grasp is found.
In order to reconstruct a high-quality 3D model, S. Pan et al. (2022) propose a learned approach that predicts the optimal sequence of sensor poses given a set of view candidates and the current occupancy map.
What these approaches have in common is that they integrate new measurements only once with each cycle. With modern hardware and reasonable sensor resolution, the standard implementations of the TSDF as well as the occupancy map are more than capable of fusing measurements with typical camera frame-rates of 30/60 Hz. Hence, the bottleneck of the workflow shown in figure 32 is the planning and execution phase, which poses a severe restriction. For example the algorithm of Monica and Aleotti (2017) spends roughly 75% of the time either planning for a goal or moving towards it. Kriegel, Rink, Bodenmüller, and Suppa (2013) report roughly 50%. Often the path-planning between views is not even constrained to look at the object during the motion and common success metrics include surface coverage and corresponding number of views. The fact that the down-times can be used (at least partially) to include several hundred more measurements and improve either the TSDF or the occupancy map is often ignored.
Servoing approaches mitigate this problem by performing the computation of robot motion commands in a separate thread. The typical workflow is depicted in figure 33. The planning and execution phase is replaced by a separate, continuously running control loop whose motion goal target is occasionally updated by the sensor fusion thread. This allows for the continuous fusion of measurements and both threads to run with a high frame-rate. Motion goals are only temporary and should be understood as an indication of direction as

they are usually not full reached and are overwritten as soon as new measurements arrive. A few examples for servoing approaches as depicted in figure 33 include the work of Morrison et al. (2019b), who utilize the low inference time of their generative grasping network to continuously steer the robot towards the highest entropy of past two-dimensional grasp predictions.

Breyer, Ott, et al. (2022) propose to use a ray casting algorithm with a TSDF and use a continuous velocity controller to move the robot in the direction of the view-candidate that currently "sees" the highest number of unknown voxels.

Cai et al. (2022) continuously predict a set of grasp candidates based on the latest TSDF and proposes the linear interpolation between the current pose and the best grasp candidate. This allows them to use the approach phase of grasping to refine the TSDF and thus increase the chance of success.

We opt for a probabilistic TSDF as described in section 3. Naturally, we want to fuse a high number of measurements in order to gain a precise representation of the object and its surroundings and therefore follow this servoing strategy.

The neglection of classical path planning obviously comes at the price of no (traditional) collision checks, no checks for reachability and generally only works under the assumption of a benign setup. These gaps must be closed by a strategy that does not compromise the requirement for high frequency.

We found that collisions with the environment can be avoided via an appropriate movement restriction of the robot and additionally experienced that the question of reachability can be largely solved by preceding inverse-kinematic checks. Apart from that, we encountered two major problems during our experiments with this approach.

The first one being self-collisions that occasionally happened during the scanning-phase

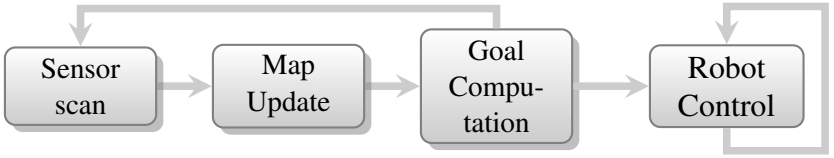

Figure 33: The typical division of servoing approaches. The first thread deals with the processing of sensor signals and the calculation of targets. The second thread continuously steers the robot towards the current goal.

of the object, especially when the robot repeatably scanned opposing sides of the object or when the object was too close to the robot base. The second problem were rapid accelerations when the robot came close to singularities, e.g. the goal was considered feasible during the inverse kinematic checks but only barely reachable. The following section describes the proposed algorithmic pipeline for the reconstruction phase of our grasping algorithm and the proposed approach for the workflow that is schematically depicted on the left side of figure 33 (first thread). Section 4.4 is concerned with the proposed real-time-solution for self-collisions, singularities and joint-limit avoidance at controller-level.

4.3 Exploration Strategy

Similar to our previous work (Schaub et al., 2021), we distribute a set of view candidates V over the hemisphere which is centered over the object in question. This placement of the view candidate placement is shown in Figure 34.

Compared to our previous work we opt for a smaller set of view candidates C in order to keep the computational effort low (compare to figure 34 to 2). We found that view candidates close to the table, i.e. polar angles θ close to 90°, are often superfluous, and their evaluation rarely has a significant advantage over candidates where θ is smaller. On the contrary, views close to the table surface are often not reachable anyway as they lead to collisions with the table itself or the objects placed on it. We therefore use a significantly smaller range for θ. If only a single, freestanding object is considered, four opposing candidates were usually sufficient in our experiments. However, when several objects are present and occlusions occur, then more candidates are required to utilize the gaps between the occluding objects. We therefore define the set of view candidates C to be

$$C = \Theta \times \Phi \times r = \{15^\circ, 30^\circ\} \times \{i\,45^\circ \mid i \in \{1,2,...,8\}\} \times r \ , \tag{61}$$

where the radius r is a constant that is chosen such that the distance between the object's bounding box and the spheres surface is greater than the cameras minimum distance (i.e. 25 cm). We found that this set covers all our experimental scenarios without compromising the real-time capability of our algorithm.

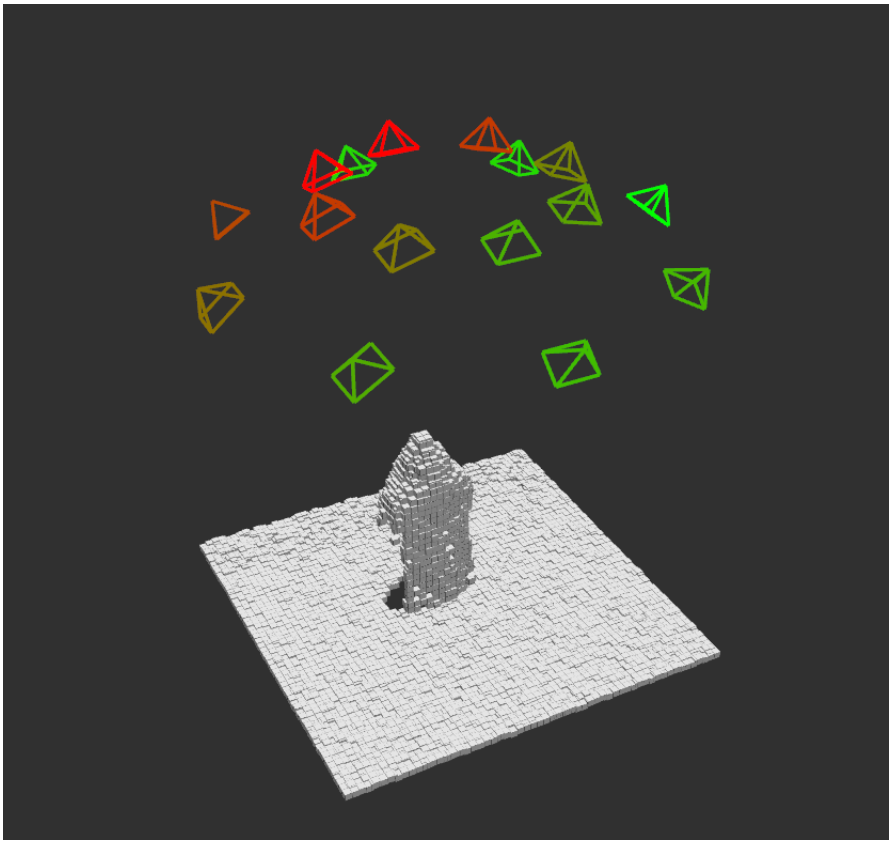

Figure 34: The set of view candidates P and their corresponding utility value (eq. 69) in the color channel.

4.3.1 Determination of Visible Voxels

We use a perspective sensor and hence use a perspective approach to find out which volume sections are "visible" from a given view. The rendering of an image, a given volumetric representation is traditionally realized via a ray casting algorithm, where for each image pixel a three-dimensional ray is computed. Starting from the focal point, the ray traverses though the volume until it hits an obstacle. This intersection is evaluated, and the pixel is filled with appropriate (color-)values.
In our context, two problems arise from this approach. First, the ray traverses multiple discrete volume-elements that contribute to the information gain and the informative value of a ray is not solely determined by the first obstacle. Second, many rays can traverse though

the same volume-elements and the evaluation over all pixel values therefore skews the information gain in favor of views that are closer to the object.
We propose a custom ray casting-algorithm that does not evaluate the volume in a pixel-wise manner but instead identifies the set of visible voxels V_{vis} for each view candidate. The algorithm is described in the following.
The Cartesian position of the focal point p_c of one view candidate $c = (\theta, \phi, r)$ with respect to the TSDF's base frame is given by

$$p_c = \left(r\cos\theta\cos\phi + \frac{s_x}{2}, r\cos\theta\ r\sin\phi + \frac{s_y}{2}, r\sin\theta \right)^T \quad , \tag{62}$$

where (s_x, s_y, s_z) denotes the size of the TSDF in the respective dimensions.
For each focal point we define the look-at direction of the corresponding frustum such that it points towards the center of the semi-sphere. We arbitrarily choose to align the frustums horizontal axis with the direction of rising ϕ and the vertical axis with rising θ. Hence, the rotation $R_c \in SO3$ of the frustum of view candidate c is given by

$$R_c = \left[\frac{u_\theta}{|u_\theta|}, \frac{u_\phi}{|u_\phi|}, -\frac{u_r}{|u_r|} \right] \quad , \qquad \text{with} \qquad u_\theta, u_\phi, u_r = \frac{\partial c_p}{\partial \theta}, \frac{\partial p_c}{\partial \phi}, \frac{\partial c_p}{\partial r} \tag{63}$$

Let (f_x, f_y) be the sensors focal lengths and (c_x, c_y) be the offset of the image center in the respective dimension and $r_d \in S^2$ be the normalized ray direction of an arbitrary of an arbitrary image position (u, v) on the sensor plane. We compute r_d, with

$$r_d = R_c \frac{a}{\|a\|} \quad , \qquad \text{with} \quad a = \frac{1}{f_x\, f_y} \begin{bmatrix} f_y & 0 & -c_x\, f_y \\ 0 & f_x & -c_y\, f_x \\ 0 & 0 & f_x\, f_y \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad , \tag{64}$$

where $R_c \in SO3$ denotes the orientation of the view candidate c with respect to the TSDF's frame. The ray r, also known as half-line is then given with respect to the TSDF's coordinate frame by

$$r \ = \ d\ r_d + p_c \ , \tag{65}$$

where the scalar $d > 0$ is the distance from the focal point p_c along the unit ray direction r_d. We are only interested in the line segment of r that is inside the TSDF volume and hence compute the first and second intersection with the TSDF at $d_{min}, d_{max} \in \mathbb{R}$ using

Algorithm 3 Compute both intersections of the ray $r \in S^2$ with the TSDF

Input: Ray direction $vec_3\ r_d$, Size of the TSDF $vec_3\ s$, Focal point $vec_3\ p_c$
Output: Binary mask of visible voxels M

$vec_3\ invRay \leftarrow (1/r_x, 1/r_y, 1/r_z)$
$vec_3\ bot \leftarrow (0,0,0)$
$vec_3\ top \leftarrow (s_x, s_x, s_z)$

//Compute the intersection with all six cube planes using the Hadamard product $\odot$
$vec_3\ bot \leftarrow invRay \odot (bot - p_c)$
$vec_3\ top \leftarrow invRay \odot (top - p_c)$

//Find the intersections closest and furthest to the sensor along the axes
$vec_3\ cls \leftarrow (\min(bot_x, top_x), \min(bot_y, top_y), \min(bot_z, top_z))$
$vec_3\ far \leftarrow (\max(bot_x, top_x), \max(bot_y, top_y), \max(bot_z, top_z))$

//Find the furthest close intersection and closest far intersection
$float\ d_{min} \leftarrow \max(\max(cls_x, cls_y), \max(cls_x, cls_z))$
$float\ d_{max} \leftarrow \min(\min(cls_x, cls_y), \min(cls_x, cls_z))$
return (d_{min}, d_{max})

algorithm 3.

The pose of the focal points (see equation 62 and 63), the size of the TSDF (s_x, s_y, s_z) as well as the camera parameters (f_x, f_y, c_x, c_y) remain static during the scanning application.

The computation of the ray directions per image position including the computation of the intersection distances, i.e. the ray's evaluation range $[d_{min}, d_{max}]$, only needs to be computed once. Hence, the equations (62 - 64), as well as algorithm 3 only need to be evaluated once during the initialization. The results can be stored in an appropriately sized static array in order to enhance the real-time performance of the proposed pipeline. During the runtime, only the ray evaluation needs be performed, which is described below.

We initialize a binary matrix M with *false* for each view candidate $c \in C$ where the dimensions of M are equal to those of the TSDF. We traverse all rays within their respective limits d_{min} and d_{max} and set M to *true* where corresponding voxels are visible, i.e. not occluded. The matrix M is used for masking operations on the TSDF and efficiently prevents redundant (ray-wise) evaluation of voxels, i.e. when multiple rays traverse through

Algorithm 4 Custom ray casting and determination of unique, relevant voxels.

Input: Rays directions $vec_3\ r_d \in R_{img}$, TSDF size $vec_3\ s$, Voxel size $float\ vs$, Focal point $vec_3\ p_c$
Output: Binary mask of visible voxels M

$float\ t, t_{prev} \leftarrow 0$
$M \leftarrow$ zeroInitialisation()
For $r \in R_{img}$
    $d_{min}, d_{max} \leftarrow$ computeIntersections$(s,\ vs\ , r)$
    $d \leftarrow d_{min}$
    While $d < d_{max}$
        $vec_3\ p \leftarrow r \cdot d + c_p$
        $int_3\ idx \leftarrow$ getIndex$(p,\ vs)$
        $t_{prev} \leftarrow t$
        $t \leftarrow TSDF(idx)$
        if $t \cdot t_{prev} < 0$
            break
        $bool\ b \leftarrow$ withinObjBoundingBox(idx)
        $M[idx] \leftarrow M[idx] \vee (b \wedge t < 1)$
        $d \leftarrow d + vs$
return M

the same voxel.

Let R_{img} be the set of all rays r for one view, c_p the position of the sensor's focal point and vs be the size of one voxel. The process is described by algorithm 4 and a graphic example is given in figure 35.

Although we utilize the full TSDF to detect occlusions, we only evaluate the voxels within the objects bounding-box. Each ray is traversed until either d_{max} or it hits an obstacle (defined by a change of sign in the signed distance values). We deliberately chose to ignore free voxels outside the truncation band, i.e. voxels where the signed distance is equal to one.

After the application of algorithm 4, we obtained the 3D binary matrix M which finally allows us to determine the set of visible voxels V_{vis} for each view via standard masking operation of the set of all voxels V

$$V_{vis} = \{ v_{ijk} \in V \mid m_{ijk} = true \} \qquad \text{where } m \in M \tag{66}$$

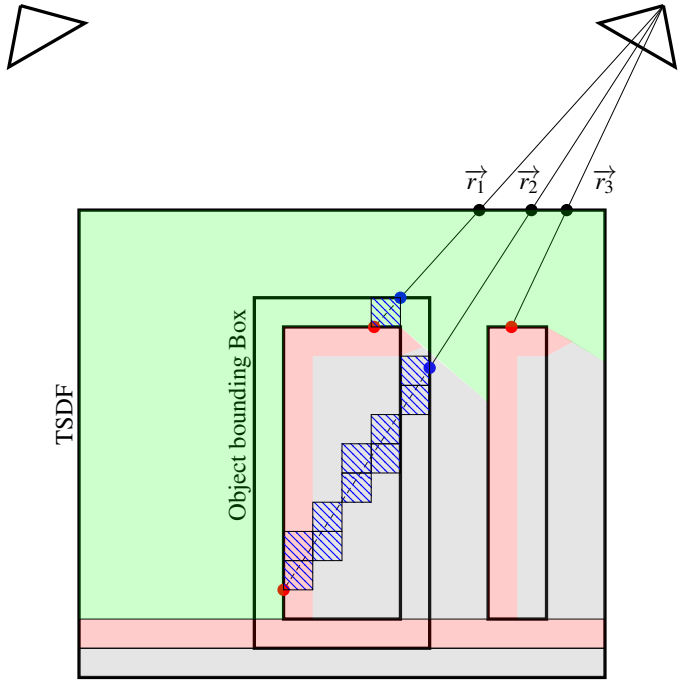

Figure 35: Visualisation of the proposed raycasting algorithm 4.

Figure 35 illustrates the situation where the box-shaped object was only seen from the top left perspective and the information gain of the view candidate at the top right is evaluated. The volume that is known to be free (i.e. positive) is shown is green, known occupied volume is shown in red and unknown volume is colored in gray. Three examples regarding the different evaluation of rays are denoted by r_1, r_2, and r_3. Voxels that are marked for entropy evaluation by these beams are colored in blue. Their first intersection of each ray with an obstacle is indicated by a red dot and implies the end of the ray traversal.

r_1 shows the typical case where the ray intersects the known object surface (i.e. $+/-$ sign change). We ignore all voxels outside the objects bounding box and only mark the voxels within the positive truncation band for further evaluation.

r_2 traverses though the unknown object volume and intersects with the known surface from within the object (i.e. $-/+$ sign change). All unknown voxels, as well as those in

the negative truncation band (i.e. red and gray) are marked for further evaluation.
r_3 hits an obstacle before entering the object bounding-box in the middle of the scene. No voxels are marked.
The ray casting from a total number of 16 viewpoints is computationally very expensive. We therefore follow the strategy of Breyer, Ott, et al. (2022) and reduce the resolution of the view candidates by a factor of ten compared to the actual sensor. Nonetheless, the calculation of the information gain can easily represent the bottleneck of the whole algorithm and is very dependent on an appropriate implementation.
In our experiments with various implementations (CPU with C++; GPU with CUDA) and parallelization over all ray-evaluations versus parallelization over views, we found the CPU/C++ and the parallelization over views to be the most efficient. This might be because we usually consider a small TSDF grid (i.e. $80 \times 80 \times 80$) and a relatively low ray casting-resolution where the advantages of GPU parallelization do not take effect yet. We used the Eigen3 library (Guennebaud & et al., 1996) for all vector computations and OpenMP (Chandra et al., 2019) for the parallelization of 16 view-candidate evaluations over eight CPU-cores (*Intel-i7*) and naturally used full compiler optimization.

4.3.2 Information Gain Formulation and Next-Best-View

To compute the information gain of a view candidate c, we choose one of the most promising metrics of Delmerico, Isler, Sabzevari, and Scaramuzza (2017), namely the *Average Entropy* $\overline{H}$, which was originally introduced by Kriegel et al. (2013) and is defined as

$$\overline{H}(c) = \frac{1}{|V_{vis}|} \sum_{v \in V_{vis}} H(v) \tag{67}$$

where V_{vis} refers to the set of visible voxels (see equation 66) that corresponds to c. In the proposed algorithm, the state of each voxel is given by $(\hat{\mu}, \hat{\tau}, v, W)$ (see algorithm 2). Due to the assumption of a Gaussian distribution, the entropy of one voxel is given by

$$H(v) = \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2} \ , \qquad \text{where} \qquad \sigma^2 = \mathrm{Var}(\hat{\mu}) = \frac{1}{W} \ , \tag{68}$$

and $\hat{\mu}, W$ refer to the state variables of one voxel.
Compared to binary metrics for *information gain*, e.g. the number of the visible, previ-

ously unobserved voxels or the number of frontier voxels, the *Average Entropy* permits the examination of the estimation quality and therefore aligns with our goal of computing local grasping success probabilities. Additionally, this metric allows the exploration to continue, even though all volume elements are "known" in a binary sense and no grasp is found yet. We found that this vastly reduces the number of cases where the exploration needs to be aborted.

The average entropy compared to the regular entropy (i.e. equation 67 without the fraction term) can also yield high information gain for known surfaces where the rays traverse less unknown voxels. Depending on the situation, this allows the robot to examine local areas first without always heading for the opposing side by default.

An undesired behavior can occur when the robot is between two promising view candidates. The robot moves towards one of them, explores the perspective a little, before the information gain of the other prevails and the target is changed abruptly. Apart from the fact that the jerky change of direction is certainly not beneficial for the hardware, this behavior is very inefficient as the same volume is repeatedly traversed in an attempt to view the opposing side. We mitigate this problem via the integration of the utility function proposed by Delmerico et al. (2017),

$$U(c, p_{cam}) = (1-\gamma)\frac{\overline{H}(c)}{\sum_{i=1}^{|C|}\overline{H}(c_i)} - \gamma\frac{g(c, p_{cam})}{\sum_{i=1}^{|C|} g(c_i, p_{i,cam})} \ , \tag{69}$$

where $g(c, p_{cam})$ is an arbitrary cost function of the view candidate c and current camera position p_{cam}. $\gamma \in [0,1]$ is a user defined cost weight. We find the next best view c^* by maximizing eq. (69),

$$c^* = \arg\max_{C_{reach}} U(c, p_{cam}) \ , \tag{70}$$

where C_{reach} denotes the set of reachable view candidates. We use the open source library Trac-IK (Beeson & Ames, 2015) to filter the generated views C and combine all candidates c where a corresponding, valid inverse kinematic solution was found into the reachable set C_{reach}.

In order to avoid potential collisions with the objects or the table, we move the end effector exclusively on the semi-sphere's surface. Therefore, we model the cost of a view to be proportional to the distance to reach it. This distance is given by the great circle distance

between the current polar position of the camera $p_{cam} = (\theta cam, \phi_{cam}, r_{cam})$ and the view candidate $c = (\theta, \phi, r)$.

$$g(c, p_{cam}) = \arccos\left(\sin\phi_{cam}\sin\phi + \cos\phi_{cam}\cos\phi\cos\Delta\theta\right) \quad , \tag{71}$$

where $\Delta\theta$ is the absolute distance between both longitude angles (λ_c, λ_x). Since in equation (69) the costs are evaluated relative to each other, the radius can be neglected.
It is worth noting that in the initial frames the camera is often outside the semi-spheres surface and the neglection of the radius is a deliberate choice. Due to this neglection, candidates with a small angular distance to the current position gain disproportionately low costs during the initial approach phase. In practice, this can often lead to the exploration of neighboring candidates during the approach phase instead of moving to the opposing side by default, which allows partial occlusions to be dealt with more efficiently.
Figure 36 shows examples of trajectories that our information gain formulation and associated utility function produce. Both scenarios were simulated using the open source simulation platform PyBullet Coumans (2019), where the red object is supposed to be grasped (and hence scanned) and the black objects are obstacles that cause occlusions. The initial perspective of both scenarios is shown on the top-right.
The robot explores the objects until it finds a valid grasp (visualized with the green U-shape). The traveled trajectory is shown in blue and the corresponding frustum is displayed every ten steps.
If one side of the object is fully visible at the start, there is often a dominating view candidate on the opposing side that the robot steers towards. In case of the left image, this candidate is top left (from the perspective of the initial image). As soon as enough information has been gathered, a grasp is usually found and the exploration stops.
The second example shows a scenario where the object is semi-occluded from the initial perspective. In this case the utility function, combined with the average entropy metric leads to a serpentine approach phase. Our ray casting algorithm 4 detects the occlusion and the robot initially steers to the right to look past the obstacle. As soon as the utility of the left side prevails, the robot steers to the left and finds a grasp for the cup shortly afterward.
In both scenarios, the robot was clearly outside the hemisphere on which the view candidates are located, which we consider to be the standard case as the radius is defined by the

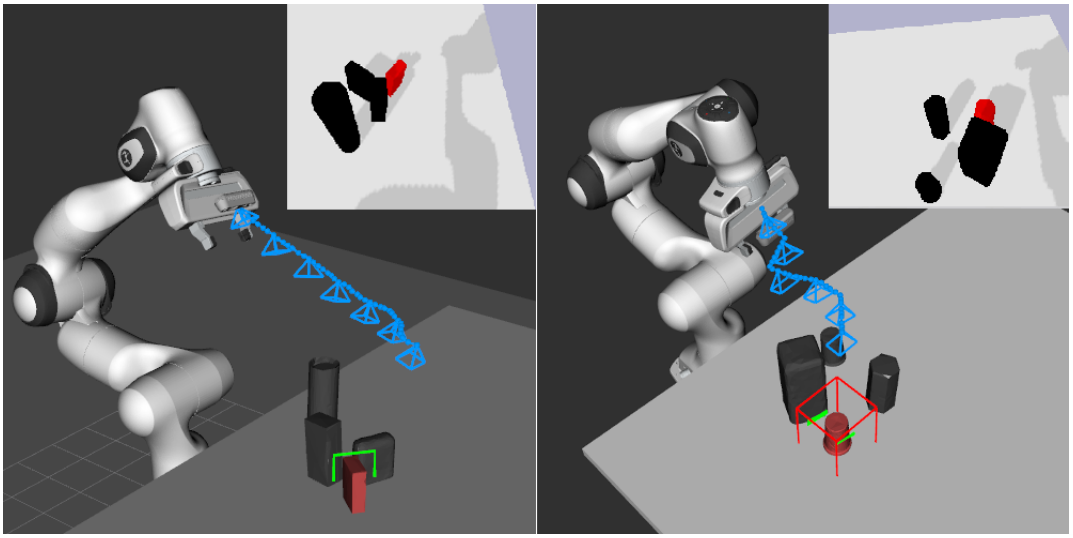

Figure 36: Two common grasp-exploration scenarios.

object size and the camera's minimum distance, which is 25 cm. Since we only consider the polar-angles of candidates in the cost function (and not the radius), neighboring view candidates can get relatively high utility values in the approach phase, especially if initial occlusions occur. This allows for the zigzag movement on the right side of figure 36 and leads to the fact that (in somewhat benign settings) our exploration phase often largely overlaps with the approach phase to the object, which we consider the desired behavior as it is time-effective.

Due to the implementation of the information gain computation as a C-library and the optimization steps described in section 4.3.1, our information gain computation is very efficient. This is reflected in the significantly reduced mean duration shown in table 7, compared to the implementation of Breyer, Ott, et al. (2022). This is despite the fact that our entropy computations is computationally more expensive than their counting of unknown voxels. This advantage enables our algorithm to process typical camera clock rates of 30 Hz without having to drop information.

The results in table 7 show the mean durations and corresponding standard deviations of roughly 2000 ray casting operations over all 16 view candidates. We used the same randomized scenarios, the same number of view-candidates and hardware for both series of measurements. Its worth noting that we excluded outlier-times in order to prevent the oc-

casional operating-system-hiccups from distorting the mean results. Similar to the default boxplot-visualization setting in the *Matplotlib* (J. D. Hunter, 2007), we defined outliers as measurements m that exceed the upper quartile by more than 1.5 times the interquartile range (IQR), i.e.

$$m < Q_{25} - 1.5\,IQR \quad \vee \quad m > Q_{75} + 1.5\,IQR \ , \quad \text{with} \quad IQR = Q_{75} - Q_{25} \ , \tag{72}$$

where Q_{75} and Q_{25} denote the 75% and 25% percentile.

Table 7: Mean duration and std. deviation of the Information Gain Evaluation

proposed algorithm	0.024 ± 0.019 [s]
Breyer, Ott, et al. (2022)	0.095 ± 0.048 [s]

4.4 Constrained robot control

How the robot is controlled in order to enable these Cartesian trajectories is discussed in the following section.

4.4.1 Background

A kinematic chain consists of a series of rigid-bodies which are connected by joints. Each joint connects two links. Hence, the pose of a link l can be represented as a function of the preceding joint coordinates $\{q_j,\ j \in [0,l)\}$. We are mainly interested on the pose of the sensor. Its pose ${}^0T_S$ with respect to the first link 0 is given by

$${}^0T_S = {}^0T_1(q_0) \cdot {}^1T_2(q_1) \cdot \ldots \cdot {}^{S-1}T_S(q_{S-1}) \in SE(3) \quad . \tag{73}$$

The mapping from joint coordinates, or robot configuration, to the end-effector pose (or any other link in the chain), is called forward kinematics (Waldron and Schmiedeler

(2016), p. 16). The configuration of a manipulator is a vector $q \in \mathbb{R}^n$

$$q = (q_1, \ \ldots \ q_n)^T \subset \mathbb{R}^n \ , \tag{74}$$

where n represents the number of joints in the manipulator system. All joints are usually bounded by joint limits $q_i \in [q_{i,min}, q_{i,max}]$. The derivative of ${}^0T_S$ with respect to the time (Corke (2017), p. 171 - 176) is given by

$${}^0\dot{T}_S S = \begin{bmatrix} {}^0\dot{R}_S & {}^0\dot{t}_S \\ 0_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \left[{}^0\omega_S\right]_\times {}^0R_S & {}^0\dot{t}_S \\ 0_{1\times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \ , \tag{75}$$

where

$$\left[{}^0\omega_S\right]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \ , \tag{76}$$

denotes the skew-symmetric matrix form of the angular velocity ${}^0\omega_S = (\omega_x, \omega_y, \omega_z)$. The spacial velocity of the end-effector is usually represented as a combination of the translational and the angular vector

$${}^0\nu_S = \left({}^0\dot{t}_S, {}^0\omega_S\right) = (\dot{t}_x, \dot{t}_y, \dot{t}_z, \omega_x, \omega_y, \omega_z) \in \mathbb{R}^6 \ . \tag{77}$$

We intend to control the robot in the velocity domain, i.e. we control the spatial movement of the end-effector via corresponding joint velocity commands. Since we are exclusively interested in the movement of the sensor, we neglect the reference frame ${}^0\cdot_S$ indices from now on. The partial derivative of the sensors pose with respect q is denoted by the geometric Jacobian matrix J,

$$\nu = J\,\dot{q} \ . \tag{78}$$

For many applications the motion command is defined in the Cartesian space and the user is therefore interested in the inversion which is given by

$$\dot{q} = J^{\#}(q)\,\nu \tag{79}$$

where $J^{\#}(q)$ denotes the pseudoinverse of $J(q)$ which provides a solution that minimizes $\|J(q)\,\dot{q} - \nu\|$ which is the error between the actual spacial velocity $J(q)\,\dot{q}$ and the desired one ν. Many robots are over-actuated or redundant. These are robots where the number of joints is greater than the number of considered Cartesian degrees of freedom.
In our application, we are mainly interested in aligning the look-at axis of the sensor with the direction towards the object and the rotation around that axis is of minor interest to us. This boils down to ignoring ω_z as well as neglecting the sixth row of J. Hence, equation 79 is overdetermined by two dimensions (five Cartesian degrees of freedom vs. seven joints) and allows for an infinite number of solutions.
Let I denote the identity matrix, then the pseudo inversion of equation 78 $J^{\#}(q)$ leads to

$$\dot{q} = J^{\#}(q)\;\nu + N(q)\;\dot{q}_a \;\;, \qquad \text{with} \qquad N(q) = I - J^{\#}(q)\;J^{T}(q) \;\;, \tag{80}$$

where the first term is, of all infinite number of solutions, the one that leads to the smallest $\|\dot{q}\|$ (Corke, 2017, p. 183). The over-determination allows us to include a secondary task (i.e. the arbitrary joint velocity $\dot{q}_a$) in the equation which does not affect the primary task (i.e. the Cartesian motion ν).
N is often called the nullspace projector as it projects the arbitrary joint velocity $\dot{q}_a$ into the nullspace of $J(q)$ such that it creates zero end-effector motion (in our case sensor motion). A common use of $\dot{q}_a$ is the application of a joint velocity that aims to center the robots joints between their respective limits.

4.5 Attractive Velocity and Frequent Issues

Each algorithm cycle, we want to steer the camera towards the currently most promising view candidate c^* (according to equation 70) in a velocity control manner, i.e. similar to equation 80, while avoiding invalid joint states.
Comparable to potential field methods, we interpret this as task as two components: An attractive velocity $\dot{q}_{attr}$ that guides the robot towards the current goal and a repulsive joint velocity that pushes it away from invalid joint states.
The attractive joint velocity $\dot{q}_{attr}$ is the product of our consideration in the previous section 4.3.2 and is constructed as follows. Let p_c be the Cartesian position of c^* with respect to the camera frame and $l_{vd} = 0.05$ [m/s] be the desired linear speed of the robot. The linear

Table 8: Variables used in this section and value in the case of a constant.

${}^0T_S$	Pose of the sensor with respect to the first link
v	Spacial velocity in the direction of the goal
$\dot{t}$	Positional vector component of v
ω	Rotational vector component of v
$\dot{q_{attr}}$	Joint velocity that corresponds to v_{attr}
$H(q)$	The cost of a joint state q
$\nabla H(q)$	Partial derivative of the cost w.r. to q
$J(q)$	Jacobian matrix (of the sensor link)
$J^{\#}(q)$	Pseudoinverse of J
N	Nullspace projector.
d_c	L_2 distance to the nearest collision in joint space
d_{thr}	L_2 Distance threshold for Nullspace projection
$A(d_c)$	Activation function
$m(q)$	Manipulability of a joint state
α	Weighting constant for the repulsive velocity
β	Weighting constant for the NS-projection of the repulsive velocity

velocity component of v_{attr} is then determined by

$$(\dot{t}_x, \dot{t}_y, \dot{t}_z) = \text{proj}_S \left(l_{vd} \, \frac{p_c}{|p_c|_2} \right) \ , \tag{81}$$

where $\text{proj}_S(\,.\,)$ denotes the projection of the velocity vector onto the view-candidate semi-sphere if the motion would lead to the sphere radius being crossed and leaves the velocity vector as is otherwise.

During the motion, we want to keep the cameras look-at direction $r_{la} = (0,0,1)^T$ pointing towards the scan target. Let r_{se} be the current normalized vector from the scan-target with respect to the camera system. The rotational component (i.e. the angular velocity) of v is then determined by

$$(\omega_x, \omega_y, \omega_z) = \angle(r_{la}, r_{se}) \, (r_{la} \times r_{se}) \ , \tag{82}$$

where $\angle(r_{la}, r_{se})$ denotes the angle between both vectors. The attractive joint velocity $\dot{q}_{attr}$ is then determined via

$$\dot{q}_{attr} = J^{\#}_{red}(q) \ (\dot{t}_x, \dot{t}_y, \dot{t}_z, \omega_x, \omega_y) \ , \tag{83}$$

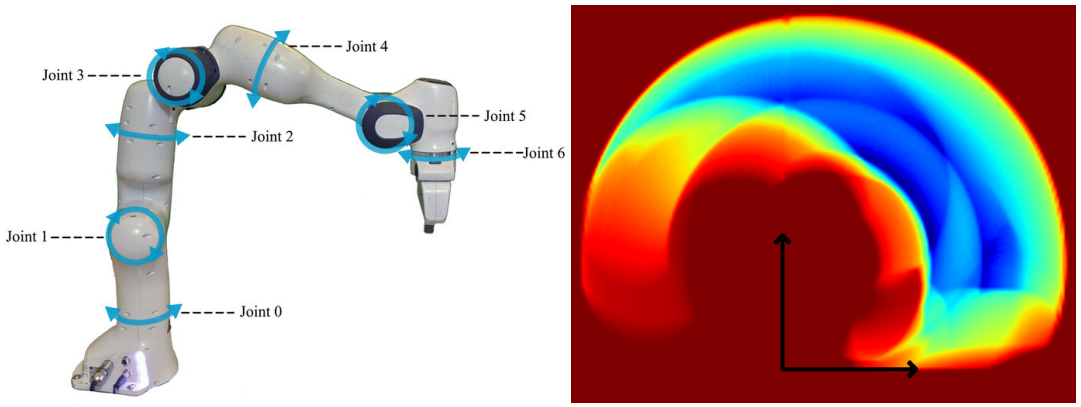

Figure 37: The *Franka Panda* (graphic from Reed et al. (2024)) and a custom-made slice through the X/Z-plane of the reachable space.

where we neglect the rotational component around the sensor z-axis ω_z and $J^{\#}_{red}$ denotes the appropriately reduced pseudo inverse of the Jacobian.

Relying solely on the attractive velocity can produce a variety of problems during the motion and generally requires a benign setup to perform without failure. This is because the collision-free, linear interpolation in Cartesian space is non-linear in joint space and neither necessarily collision-free or nor even in the valid value range.

A slice along the X/Z-plane of the reachable Cartesian space can be seen on the right-hand side of figure 37 and the left side shows the (up-scaled) *Franka Panda* and its joint axes for the sake of clearness. Joints 0, 2 and 4 were set to fixed values such that the arm only moves in the image plane, parallel to the X/Y plane. The remaining joint space was discretized and corresponding (image) coordinates were calculated for each valid configuration. The coloring of figure 37 depicts the relative frequency of valid configurations, i.e. dark red positions could not be reached at all and blue positions imply a high number of valid joint configurations.

In our experimental setup, collisions with the environment did not pose a problem due to the enforced distance via sphere projection in equation 81. However, three challenges had an impact on error-free exploration:

- Singularities

Singularities are configurations where $det\,(J\,(q))) = 0$ and therefore every non-zero $\dot{v}$ leads to an infinite $\dot{q}$. The most prominent example is when the robot arm is at full reach and cannot move any further in outwards direction. This can be seen at the spherical radius in figure 37, where the coloring quickly changes from green to yellow to red in outwards direction. Closing in on singularities leads to rapidly-increasing joint speeds as the gradient elements in the Jacobian become smaller and smaller. From a practical perspective this can lead to an emergency shutdown of the robot (if lucky) or rapid, uncontrolled behavior. Hence, areas close to singularities need to be avoided.

- Self collisions
  In our experiments, self-collisions could sometimes occur when the object was placed too close to the robot. For example, the blue area ins figure 37 shows the "comfort zone" to place objects. If the object is placed closer, the unconstrained velocity could lead to self collisions with its own base (the dark red center region) especially when repeatedly steering to opposing sides of the object.

- Joint Limitations
  Although the joint limits were not a major problem in our experiments if the starting joint-configuration was chosen sensibly, prolonged exploration could lead to exceeding the joint limitations if left unregulated.

4.5.1 Creation and Runtime Evaluation of Collision Map

Inspired by similar techniques for (2D) path planning and control of mobile robots, we want to represent the known invalid configuration space as a discrete cost grid, which can be evaluated at runtime with high efficiency.
We propose to register all invalid joint configurations in a "collision" grid H and compute gradients ∇H at runtime. ∇H then acts as repulsive velocity that "pushes the robot away" from invalid configurations.
Due to the high dimensionality of the grid (i.e. the number of joints), the axis-resolution is critical. A high resolution demands immense memory requirements and great computation runtime. For example, a discretization of the joint-space of the *Franka Panda* in steps of 2.5°, results in $1.62e14$ configurations that need to be checked for validity. If the

resolution is too low on the other hand, then appropriate scaling of cost gradients becomes non-trivial and the distinction between independent collision points and two neighboring points on the same collision surface is difficult.
We opt for a step-size of 7.5° between the respective axis limits, which results in an evaluation grid that consists of $45 \times 27 \times 45 \times 21 \times 45 \times 29$ joint configurations. We drop the last dimension (joint 6, see figure 37 as we found the orientation of the end effector has negligible influence on self-collisions and singularities and its limits might as well be evaluated at runtime.
We create the set of all invalid joint configurations Q_{inv}, where all elements q_{inv} need to meet at least one of the three following conditions:

1. q is close to an axis limit.
2. q results in a self-collision. [5]
3. the manipulability of q to is too low, i.e. $m(q) < m_{thr}$

We consider joint states where one of those conditions is met as a collision state and in the following we speak of invalid joint states and collisions interchangeably.
Regarding condition 3, we employ the manipulability definition proposed by Yoshikawa (1987). It quantifies the ability to arbitrarily change position and orientation of the link in question (usually the end-effector) and is defined as

$$m(q) = \sqrt{det(J(q)\ \ J^T(q))}\ \ . \tag{84}$$

It is worth noting that in their later works Yoshikawa (1985) introduced the *Dynamic Manipulability Measure*, which additionally takes the dynamics of the arm into account by including the inertia matrix of the joint state. However, since we are only interested in the position of invalid states, we rely on the static definition in equation 84.
The choice of threshold influences the reachable range of the arm. For example, Burgess-Limerick, Lehnert, Leitner, and Corke (2023) choose a threshold $m_{thr} = 0.08$ [m] in order to decide whether an object is in the manipulation range of the robot. This threshold roughly corresponds to a range of 0.75 [m]. E.g. consider the reachable space in figure 37 a sphere. In this visualization, m_{thr} primarily reduces the radius of the sphere and can

[5] We use FCL (Coleman et al., 2011) to check for collisions

be imagined as thresholding the outer color values, i.e. green is reachable and yellow is not.
Our goal is the binary categorization of joint space into a valid and an invalid subspace. Hence, we choose a less restrictive, empirically determined threshold of $m_{thr} = 0.02$ which roughly corresponds to a range of 0.82 m.
We are only interested in the surface of the invalid joint-subspace. Except for faulty simulated setups, a robot arm is never "stuck" within a collision object and the distance to a collision is always ≥ 0. This opens up the option to significantly reduce the size of the grid without loss of information.
We prune the set of invalid configurations Q_{inv} of all elements that are not in direct L1-neighborhood (within the 6 dimensional evaluation grid) with at least one valid configuration element, to gain the reduced set Q^*_{inv}.

Q^*_{inv} contains roughly $1e8$ invalid configurations. Inspired by similar problems in computer vision, where a computationally expensive part is often the nearest neighbor search for high-dimensional features, we utilize a KD-tree to organize Q^*_{inv}. KD-trees split the (in our case six-dimensional) space repeatedly into two sub-spaces. Both sub-spaces are called nodes and store the parameters of the splitting hyperplane as well as references to their respective children. This splitting process is repeated until the number of data points in the respective subspace falls below a predefined number. These nodes do not reference further children and are called leaves.
Our aim is to find and react to nearby collisions in Q^*_{inv} very quickly at runtime. This task translates to a nearest neighbor search based on the current configuration q.
When performing nearest neighbor search for a given point, the algorithm traverses down the tree structure to drastically reduce the search space. This reduction ends at the leaves where linear search of neighbors (one-by-one) and corresponding distance computations begins. Hence, setting a small maximum leaf size leads to greater computational effort when building the tree but usually decreases the search time for data queries. This is only valid up to some threshold since the effort of traversing the tree can dominate over the linear search at the leaf, once the tree reaches a certain size.
We use a static dataset and the tree only needs to be built once for each considered robot. Hence, the construction time of the tree is of little interest to us. It is worth noting that we considered the table as a static link and part of the robot for our experiments as well as

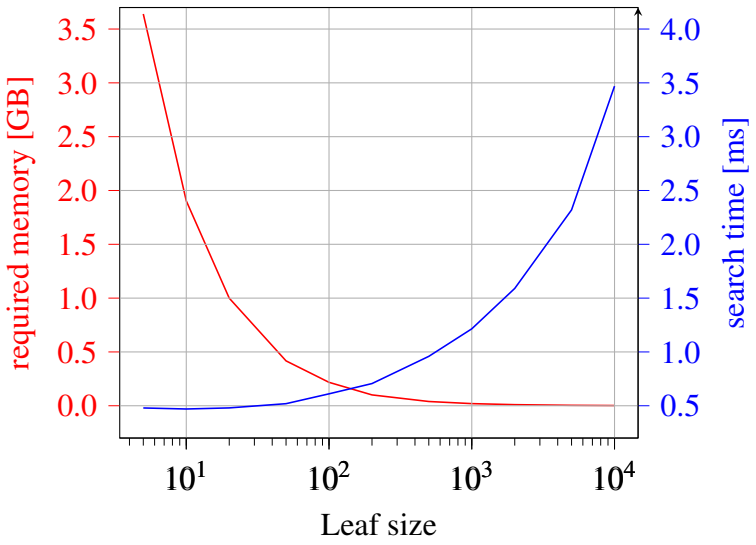

Figure 38: Benchmark tests of nearest neighbor searches with the KD-Tree.

figure 37. However, during the exploration phase we restrict the end effector's movement to the upper hemisphere of the volume and the presence of the table in the collision map rarely had any practical effect.

Since the intended application is a mobile one, the amount of allocated memory as well as the real-time capability need to be considered. Our use case is a somewhat unusual one in that both the number of dimensions (6) and the large number of data points ($\sim$1e8) does not fall into the scope of typical applications like 3D point cloud processing or feature vectors in computer vision.

We were not able to find an appropriate guideline of empirical data in the literature and hence conducted our own performance tests. The results for logarithmically increasing leaf sizes can be seen in figure 38.

Each blue data-point represents the average duration $[ms]$ of 100.000 nearest neighbor searches where for each trial the six DOF query point was randomly sampled within the joint limits and the 500 nearest neighbors were retrieved. The red line shows the memory that needs to be allocated by the tree structure. It can be seen that decreasing the leaf sizes below 50 has no noteworthy effect on the query duration, which seems to be lower bound by 0.5 $[ms]$ for our application. This lower limit might be related to the memory allocation and sampling of the random joint state. The red curve shows a dramatic increase in

required memory for leaf sizes below 10^3. Based on this data, we opted for an empirical trade-off value of 10^2 where the potential cycle frequency of the algorithm is still well above the 1000 Hz required by *Franka Panda* (Franka Emika GmbH, 2023) and it still leaves more than enough RAM for parallel applications on a mobile device.

4.5.2 Collision Avoidance and Determination of Repulsive Velocity

Let $q_c \in Q^*_{inv}(q)$ be one of k nearest, invalid neighbors for the current joint state q of the robot. $Q^*_{inv}(q)$ is created for every time step via classic k-nearest neighbor search utilizing the kd-tree that was addressed in the previous chapter. We define the cost of q_c to be inversely proportional to the current L2-distance, i.e.

$$H_c(q, q_c) = \frac{1}{\|q - q_c\|_2} \quad . \tag{85}$$

We model the "repulsive" joint direction using the gradient $\nabla H(q, q_c)$, which is given by

$$\nabla H_c(q, q_c) = \left[\frac{\partial H_c(q, q_c)}{\partial q_1}, ..., \frac{\partial H_c(q, q_c)}{\partial q_n} \right] \quad , \qquad \text{with} \qquad \frac{\partial H_c(q, q_c)}{\partial q_i} = \frac{q_{c,i} - q_i}{\|q - q_c\|_2^3} \quad . \tag{86}$$

We utilize the summed average of all k cost gradients to determine the combined repulsive direction of collisions for the current joint state q,

$$\nabla H(q) = \frac{1}{k} \sum_{i}^{k} \nabla H_c(q, q_{c,i}) \tag{87}$$

Instead of interpreting the cost gradient $\nabla H_c(q)$ directly as repulsive velocity, project it into the null space of the Jacobian J in order not to interfere with the Cartesian trajectory of the camera.

However, the cost gradient can seldom be fully projected into the nullspace and therefore the path must be adjusted if the distance to collision falls below a critical threshold d_{thr}. Inspired by Foresi et al. (2017), who use a comparable concept to avoid joint limitations, we propose the diagonal activation matrix A for a continuous transition between these two phases. Let d be the distance to the closest invalid configuration, then the diagonal

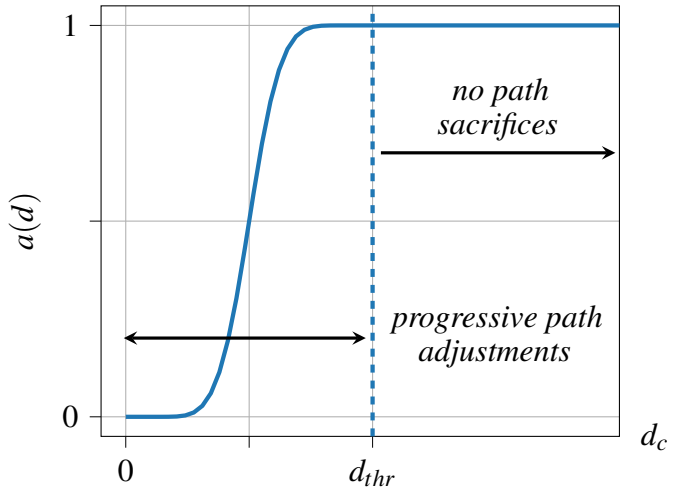

Figure 39: The transition function of the diagonal elements of $A(d_c)$

elements $a(d)$ of A are given by

$$a(d) = \begin{cases} t\left(\dfrac{d_{thr}-d}{d_{thr}}\right) & \text{for } d < d_{thr} \\ 1 & \text{else} \end{cases} , \tag{88}$$

$$\text{with} \qquad t(x) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1}{1-x} - \frac{1}{x}\right) , \; x \in [0,1] .$$

The constant d_{thr} acts as a threshold between the distance range where $\nabla H_c(q)$ is solely handled via in the null space projection and the range where path adjustments have be made in order to avoid collision. The transition function t prevents high accelerations and jerky movements between those ranges. The progression of the diagonal elements of the activation matrix A in relation to the collision distance can be seen in figure 39. Finally, the proposed control equation is given by

$$\dot{q} = \underbrace{J^{\#} v}_{\dot{q}_{attr}} + \underbrace{\alpha\left((I-A)+\beta A N\right) \nabla H}_{\dot{q}_{rep}} \qquad \text{with} \qquad N = I - J^{\#} J \tag{89}$$

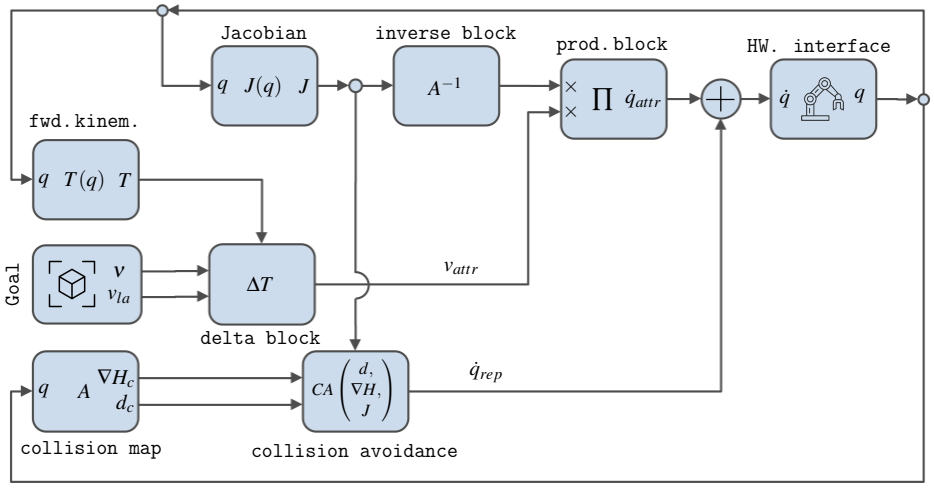

Figure 40: Flowchart of the proposed control algorithm.

where the constant $\alpha > 0$ is used to balance the repulsive velocity $\dot{q}_{rep}$ relative to $\dot{q}_{attr}$ and the constant $\beta > 0$ is used to compensate for the very small gradients ∇H at large distances and to achieve a "preemptive" evasion in the null space if possible. The block diagram that corresponds to equation 89 is shown in figure 40.

4.5.3 Simulated experiment

Figure 41 shows a simulated experiment which is designed to highlight the advantages of the proposed control scheme 89. The *Franka Panda* started in a configuration commonly referred to as the *"ready"* configuration $q = (0, -\pi/4, 0, -3\pi/4, 0, \pi/2, \pi/4)$, which is depicted in the left image. The green arrow represents the goal pose in Cartesian space. This target was chosen because the linear connection from start to target pose leads the end-effector through self-collisions in the center region. Figure 37 shows a simplified projection of this region where the lack of valid states is indicated by the deep red coloring. The green sphere represents the scanning target and must be the center of the camera's field of view (mounted on the end-effector) during the entire sequence. The right picture shows the trajectories which are generated by using the proposed equation 89.
For the sake of highlighting the benefits of the proposed control technique, we additionally implemented the "traditional" potential field method as it is often used to guide mobile

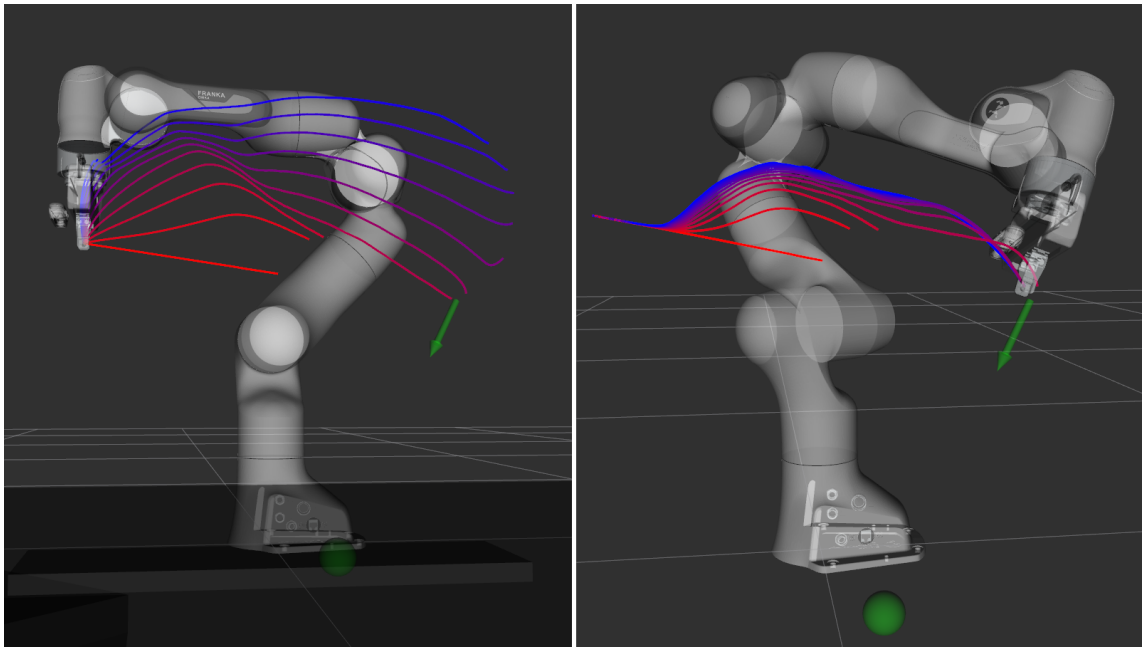

Figure 41: Trajectories using the potential field approach (left) and the proposed approach.

(two-dimensional) robots (e.g. Koren & Borenstein, 1991; Yin, Yin, & Lin, 2011). For this method, the combined velocity $\dot{q}_{tp}$ is given by

$$\dot{q}_{tp} = \underbrace{J^{\#}\,\dot{v}}_{\dot{q}_{tp,attr}} + \underbrace{\alpha \nabla H}_{\dot{q}_{tp,rep}} \quad . \tag{90}$$

Trajectories generated by this method can be seen on the left side of figure 41. For both methods the weighting factor $\alpha \in [0/50,\ 2/50,\ ...,\ 10/50]$ was gradually increased (deep red to dark blue, bottom to top) and in our case $\beta = 10$. In both images, the bottom line (deep red) corresponds to $\alpha = 0$, where the repulsive velocity is effectively ignored. Consequently, the linear interpolation from start to goal ends up in self collision in the center region.

It can be seen that for gradually increasing weight α (bottom to top) both methods show very different behavior. The trajectories of the traditional potential field method span increasingly wider arcs. By increasing the weighting factor α we increase the repulsive velocity of the highly collision-prone center area and effectively increase the radius of the

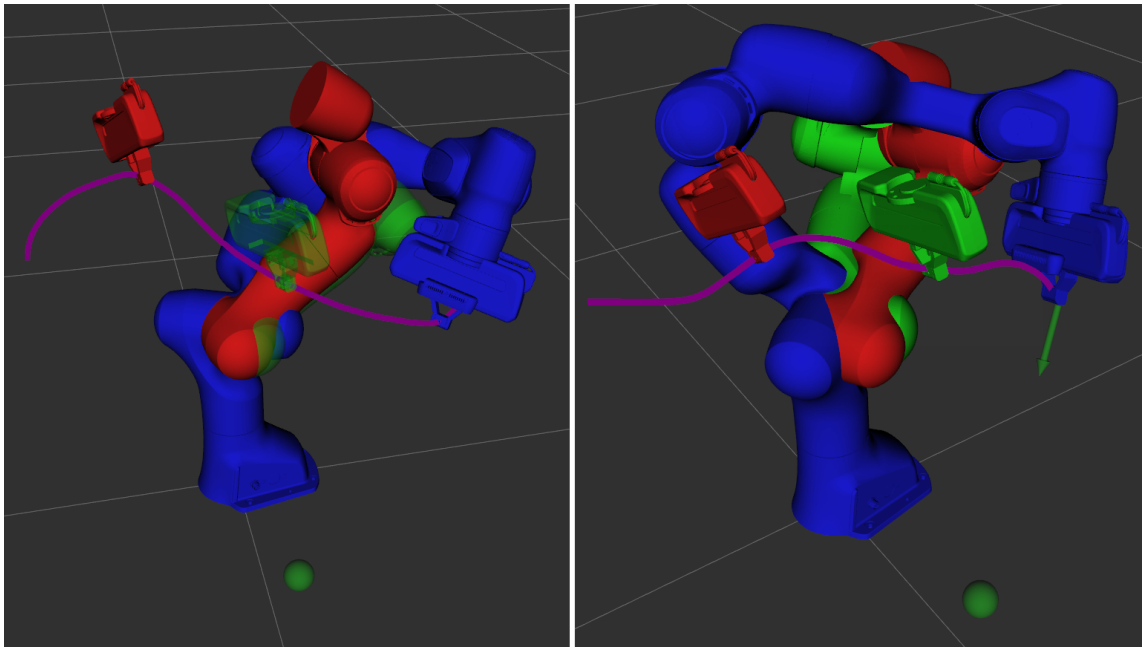

Figure 42: Comparison of the successful trajectories for $\alpha = 0.1$. For the sake of visibility several links are not visualized and only three sequential robot states are depicted (red $\rightarrow$ green $\rightarrow$ blue).

sphere-like center-region in figure 37.

In practice, this behavior requires a very delicate adjustment of α. Small α's do not prevent a collision (trajectories 0,1,2) while too large α's can make it impossible to reach a great number of target (trajectories 5-9).

The proposed control shows different behavior. Similar to the "traditional" potential field method the weighting factor for trajectories 0,1,2 is too small, and they also end in self-collision, but all other trajectories manage to reach the requested goal pose without collisions, low-manipulability or joint-limit-abortions. Due to the activation matrix in combination with the null space projector, all trajectories have a similar turning point where collision-gradients are no longer projected into the nullspace but result in path adjustments instead.

The comparatively small sensitivity to the weighting of the repulsive velocity originates in the start phase of the trajectories. Although the figure 41 suggests a linear interpolation up to the turning point, the behavior in the joint space is far from it. This difference is

illustrated in figure 42 where we show for $\alpha = 0.1$ three consecutive robot states (red $\rightarrow$ green $\rightarrow$ blue) for both approaches. For the sake of visibility only the relevant links are visualized, and the other ones were made transparent.
The "traditional" potential field method keeps the elbow in a relatively static position and hardly uses the first joint. The proposed method also employs the first joint (i.e. Joint 0 in figure 37) to preemptively rotate the arm in nullspace and ends up in a configuration with larger collision distance.
An inspection of the velocity curves of selected joints shown in figure 43 illustrates the different behavior. While joint 0 hardly experiences attractive nor repulsive force for the traditional potential field method (left side), it is heavily utilized by the proposed method (right side). This is because the relatively large weighting β of the nullspace projection of the cost.
The curves of joint 1 and 3 show another benign side effect of the preemptive nullspace projection as it often leads to a more benign initial situation for when the path needs to be adjusted to avoid a collision and the left side of the activation function in figure 39 comes into effect. The deadlock situation that is visible for joints 1 and 3 to the traditional potential field method (left side of 43), where the robot's joints slow down more and more, since repulsive and attractive velocities cancel each other out, can often be prevented.

4.5.4 Conclusion

A significant proportion of the repulsive velocity $\dot{q}_{rep}$ can often be preemptively handled in the nullspace. Nevertheless, path adjustments are often necessary to avoid the collision and both parts must be handled sensibly. Due to the employed activation function, some collisions can be avoided preemptively without the necessity to interfere with the 3D trajectory. Even if repulsive velocities cannot be completely projected into the null space, the proposed approach often leads to a more benign initial situation when the path needs to be adjusted.
Where other approaches rely on a benign setup, our approach can handle even more difficult cases where self-collisions, joint-limitations or singularities occur and the object-scanning task is translated into corresponding joint space movements without any difficulty.
Our information gain formulation combined with our probabilistic TSDF implementation

is able to reliably address complex objects or heavily occluded cases where other approaches fail due to a binary distinction into known and unknown volume. Our custom raycasting algorithm shows a significantly better computational performance than comparable state-of-the-art algorithms.

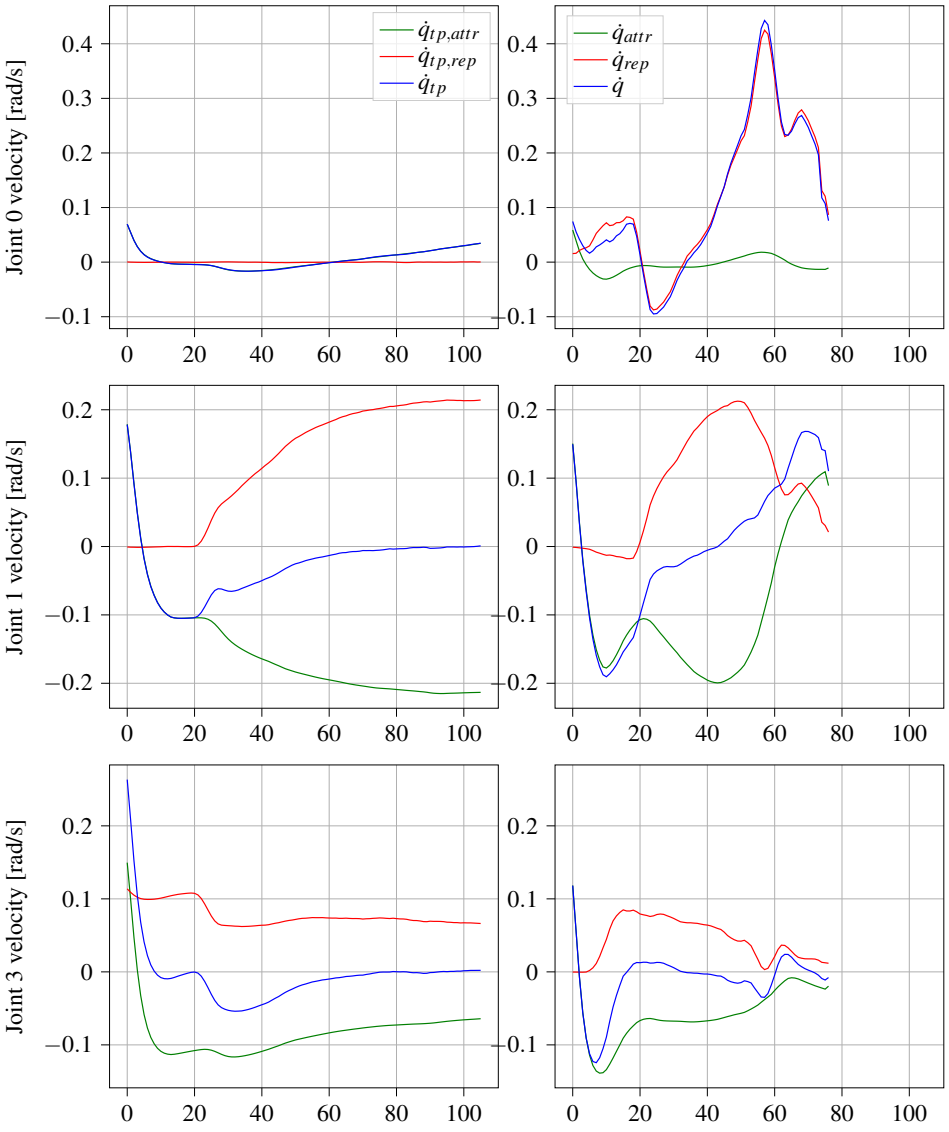

Figure 43: Comparison of the predominantly active joints for the exemplary trajectories shown in fig. 42. The figures show the attractive (green), the repulsive (red), and resulting (blue) joint velocities [rad/sec] for the potential field method (left) and the proposed method (right). It is noteworthy that the potential field method often got caught in deadlock scenarios where the attractive and repulsive speeds canceled each other out, resulting in a slow sliding along the collision surface. In this case the nullspace projection of ∇H lead to an avoidance of that stalemate situation by "preemptively" turning joint 0 (top right).

5. Grasp Candidate Sampling and Evaluation

The chapter is based on the following publication.

> Schaub, Wolff, Hoh, and Schöttl (2024)
> *"Probabilistic Closed-Loop Active Grasping"*,
> in IEEE Robotics and Automation Letters, vol 9, 2024

Manipulation of an object is the most essential task of robot arms and grasping the object, i.e. picking it up in the first place, is an integral part of this task. Grasping an object given some representation of the scene is an unsolved problem and ongoing research project.
Compared to other research areas, e.g. autonomous driving, image processing or speech recognition there is no standardized, widely used benchmark setting to evaluate grasp algorithms on. Similarly, there is no diverse, large-scale, real-world dataset available because data collection on real robots is difficult - "so difficult, in fact, that it takes a tremendous amount of engineering effort to even benchmark grasp planning methods ..." (Mahler, 2024, p. 164)).
The result is that grasping research (ours included) is usually only evaluated on a small set of hand-picked experimental setups. The setups differ significantly in the literature regarding the considered objects, the robot, collision scenery, and position and choice of the sensor.
This implies that many algorithms are to some degree tailored to the recorded data they were trained with or the setting they were developed for and perform significantly worse when they are used outside this context. This is illustrated by the dramatically different grasping success rates (i.e. the ratio of the number of times an object could be successfully lifted vs. the number of attempts) that are reported for (learned) approaches in the literature, e.g. Mahler et al. (2018) report a success rate of 58 - 98% for *DexNet 3.0* depending on the setting, whereas B. Yang, Singh, Grotz, Boots, and Smith (2007) report 24 - 84% for the same algorithm in their setups. Breyer, Chung, et al. (2022) report a success rate of 80% for their approach called *VGN*, whereas L. Wang et al. (2023) who

retrained the network report 22 - 41% depending on the setting. Morrison et al. (2018) developed *GGCNN* with success rates in the range of 87 - 100%, whereas *GGCNN* which only reached success rates of 46 - 57% in the experiments of Kasaei and Kasaei (2023). One reason for these differences is the difficulty to generalize to novel setups and objects, sometimes called the seen-to-unseen distribution gap (Cao et al., 2024). Another reason is the visual domain gap (H.-S. Fang, Gou, Wang, & Lu, 2023). We believe two aspects can significantly mitigate the impact of the visual domain gap:

- The consideration of variable sensor noise: Many authors apply white noise with an empirically determined standard deviation to their simulated training data in an effort to make it realistic. Under different conditions the sensor data might be locally more corrupted which in turn often leads to inaccurate grasp predictions. This reasoning is supported by the works of Zhang et al. (2023), who propose to precisely align simulated setup and real-world setup and to estimate all setup-specific material and lighting parameters. They then use the simulated replica of their experimental setup to generate realistic, noisy sensor data and report significant improvements if a grasping policy is trained with this data. Unfortunately, this approach is not suitable for the context of assistive robotics for obvious reasons.

- The use of a feasible measure of uncertainty: Using a measure of uncertainty enables the algorithm to simply wait with the execution until more measurements either confirm or deny the feasibility of a grasp candidate. For example, Shi et al. (2021) propose to employ several object pose-estimation-networks and to compute a disagreement score between their predictions. They observe the object from multiple pre-configured perspectives and choose the pose estimation from the perspective where the disagreement was the lowest. They then use that estimation to grasp the object.

  Breyer, Ott, et al. (2022) follows a similar concept. They register the estimated qualities of all grasp candidates frame by frame into a ring buffer and only opt for the execution of a candidate if the average quality of the buffer is above an empirically determined threshold.

  Both approaches somewhat acknowledge that the predictions are subject to many errors and deal with that similarly. The more predictions are in agreement, the smaller is the uncertainty of the combined estimate. Both show that the grasp success rate

can be significantly improved by this agreement score. Yet their approaches provide only a partial solution since the agreement of multiple predictions is a somewhat arbitrary metric as it cannot be mapped to a corresponding grasp success probability, especially not in a new environment.

In the previous chapter 3, we laid the foundation for the quantification of uncertainty regarding the surface reconstruction. In the following section 5.2, we want to describe the mapping from that measure of uncertainty to a corresponding grasp success probability. This mapping follows a probabilistic reasoning and does not suffer from out-of-domain errors.

Another common problem of many grasp estimation algorithms is the subsequent planning phase. Path planning is a computationally expensive, hence time-consuming task and many approaches use it to determine the feasibility of their grasp predictions which can result in long periods of downtime.

This problem occurs with methods that directly regress to a single grasp pose (e.g. Bicchi & Kumar, 2000; D. Yang, Tosun, Eisner, Isler, & Lee, 2021). If the pose is not reachable or in collision with the environment, the network's input needs to be altered, and the pipeline needs to be started again, hoping that the next output will be a valid one.

Methods that predict grasps in a dense manner, e.g. the work of Y. Li et al. (2022) and Breyer (2022), often suffer from a similar problem because a large set of proposals must be processed one by one, in the hope that a feasible proposal is among those with the highest estimated quality. Although some approaches mitigate this problem by applying non-maximum suppression to the grasp proposals and therefore guarantee that at least some distance is between the highest scoring grasp and the next best one, the general problem remains.

The proposals must be checked for feasibility before an attempt is made to determine the very same feasibility via computationally expensive path planning algorithms. This must happen quickly during the evaluation phase as the set of possibilities to grasp an object can be quite large.

This problem can of course be completely circumvented in a highly controlled environment in a laboratory or industrial setting by iterative adaption of the setup in such a manner that the predicted grasp poses can be reached in the vast majority of cases. However, the assistive household environment we are tackling does not permit such an option.

Our proposed solution that is presented in section 5.3 follows the same train of thought as in the previous chapter. If complete path planning for the whole set of grasp candidates is not possible within the timing constraint, then at least simplified checks can be performed at runtime in order to trim the set of considered grasp candidates from most of the infeasible ones.

5.1 Contact Point Estimations and Surface gradients

Let v be an arbitrary voxel, (x,y,z) its indices and $\hat{\mu}$ be the value of the corresponding TSDF estimation. The gradient $g \in \mathbb{R}^3$ is approximated by linearizing the N_6 TSDF-neighborhood of v. Since the estimator $\hat{\mu}$ in (43) is a linear combination of normal and independent random variables, $\hat{\mu}$ follows a normal distribution as well, and consequently $g \sim N(\mu_g, \Sigma_g)$ with

$$\hat{\mu}_g = \begin{bmatrix} \hat{\mu}(x_p,y,z) - \hat{\mu}(x_n,y,z) \\ \hat{\mu}(x,y_p,z) - \hat{\mu}(x,y_n,z) \\ \hat{\mu}(x,y,z_p) - \hat{\mu}(x,y,z_n) \end{bmatrix} \quad , \tag{91}$$

and

$$\hat{\Sigma}_g = I \cdot \begin{bmatrix} 1/W(x_p,y,z) + 1/W(x_n,y,z) \\ 1/W(x,y_p,z) + 1/W(x,y_n,z) \\ 1/W(x,y,z_p) + 1/W(x,y,z_n) \end{bmatrix} \quad , \tag{92}$$

where W is sum of weights the voxel received (see section 3.3) and I represents the identity matrix. The subscripts p and n denote the next neighbors in the respective dimension. If the sign of the expected values $(\hat{\mu}_1, \hat{\mu}_2)$ for a pair (v_1, v_2) of N_6 adjacent voxels is different, then a surface point is computed via linear interpolation between the corresponding voxel positions. The gradient g of this surface point is then determined similarly,

$$g \sim N\left(\beta\mu_{g,1} + (1-\beta)\mu_{g,2},\ \beta\,\Sigma_{g,1} + (1-\beta)\,\Sigma_{g,2}\right) \ , \tag{93}$$

where

$$\beta = -\hat{\mu}_1 / \left(\hat{\mu}_1 + \hat{\mu}_2\right) \ . \tag{94}$$

The gradient g is then the unnormalized vector perpendicular to the local surface.
The proposed algorithm evaluates all possible pairs of contact points.

A naive implementation would be of order $\mathcal{O}\left(n^2\right)$ and would create a $n \times n$ matrix of possible combinations and evaluate the matrix element by element. Since the ordering of points does not matter, there are $n\ (n-1)/2$ pairs to be evaluated. Still, the runtime of the proposed algorithm scales heavily with number of considered surface points and can easily become the bottleneck of the system. We therefore reduce the set of all points generated from the TSDF (see section 3.3) by checking the following three conditions:

- The points need to be within the objects bounding box. This ensures that only the desired object is grasped and not one of the surrounding objects or non-manipulable background i.e. the shelf on which the object stands.
- The points' surface normals need to point in a semi-horizontal direction. This is checked via

  $$\angle(\hat{z}, g) > \theta_{up} \tag{95}$$

  where $\hat{z}$ is the unit vector in upwards direction and $\theta_{up} = \pi/6$. This condition effectively excludes points on the table surface as well as the top-side of all objects. We found that most valid antipodal grasps in the household context are somewhat parallel to the table plane and antipodal grasps that rely on points where the surface normal points in an upwards direction are often in collision with the tables surface anyway. Although we limit the solution space and disregard some fringe cases where this reasoning fails, we found the significant reduction of the computational complexity to be well worth it.
- We reject points where the gradient significantly differs from its neighbors.

  $$\text{argmax}\left(\angle(g, g_c)\right) > \theta_n \quad g_c \in G_c \ , \tag{96}$$

  where $\theta_n = \pi/6$ and G_c is the set of all points that are connected to the point corresponding to g via at least one face (triangle). G_c is determined by the classical Marching Cubes algorithm (Lorensen & Cline, 1987).
  In our setting, the direct voxel neighborhood is roughly equal to half of the finger contact area. This condition has three benign effects on our algorithm. Firstly, sharp

edges are discarded e.g. our robot no longer tries to grasp boxes over the diagonals. Although such grasps are technically correct, they require highly accurate forward kinematics but the *Franka Panda* robot arm can have positioning errors greater than one centimeter (Žlajpah & Petri, 2023). Secondly, this check acts as a verification of the estimation with the local neighborhood and hence discards questionable outliers. Thirdly, we follow the common, idealized assumption that all contacts can be represented as point contacts (Bicchi & Kumar, 2000). However, in reality, the contact set is (usually) an area rather than a point and this condition softens up the point-contact assumption.

5.2 Grasp Success Probability

We rely on the force closure classification (I.-M. Chen & Burdick, 1993) to score grasp candidates. Assuming the Coulomb friction model and a constant friction coefficient f, force closure is a binary classifier that determines if any external forces and moments can be compensated by a positive linear combination of contact forces exerted by the fingers. Geometrically, $\theta_f = \arctan(f)$ can be interpreted as the maximum angle under which the object can be loaded without slipping, aka critical ramp angle. Let (p_1, p_2) be a pair of potential contact points with corresponding normal vectors (n_1, n_2). This pair is in force closure if

$$n_1 \cdot \frac{p_1 - p_2}{\|p_1 - p_2\|} > c_f \quad \wedge \quad c_f < n_2 \cdot \frac{p_2 - p_1}{\|p_2 - p_1\|} \ , \tag{97}$$

where $c_f = \cos(\arctan(f))$ and f is the friction coefficient.
We modify the FC-classifier to enable for a qualitative comparison of grasp candidates. Let p be the vector from p_1 to p_2. We are interested in the probability of the grasp defined by p, n_1 and n_2 being in force closure. The surface normals are calculated by normalizing the TSDF gradient (g_1, g_2). Assuming independence of both sides, this probability can be defined using (97),

$$P_{fc} = P\left(\frac{p \cdot g_1}{\|p\|\|g_1\|} > c_f\right) \cdot P\left(\frac{p \cdot g_2}{\|p\|\|g_2\|} > c_f\right) \ . \tag{98}$$

The normalized Gaussian vector $g/\|g\|$ is equivalent to its projection on the unit sphere, which is said to have a projected normal distribution $PN_3(\mu, \Sigma)$. The polar representation

of g is given by

$$u = \big(\sin\theta\cos\varphi,\ \sin\theta\sin\varphi,\ \cos\theta\big) \qquad \text{and} \qquad r = \|g\| \ . \tag{99}$$

Its probability density function (pdf) is as follows (Hernandez-Stumpfhauser, Breidt, & van der Woerd, 2017),

$$\text{pdf}(r,\theta,\varphi) = \left(\prod_i^3 2\pi\sigma_i^2\right)^{-0.5} \exp\left(-\sum_i^3 \frac{(r\,u_i-\mu_i)^2}{2\sigma_i^2}\right) r^2 \ . \tag{100}$$

Since the distribution is rotation invariant, we are able to choose the coordinate system such that the polar axis aligns with p. The pdf f of the angle $\theta = \angle(\frac{g}{\|g\|}, p)$ is then given by

$$f(\theta) = \int_0^\infty \int_0^{2\pi} g(r,\theta)\, dr\, d\varphi \ , \tag{101}$$

and the probability of the surface normal being within the friction cone is consequently given by

$$P\left(\frac{p\cdot g}{\|p\|\|g\|} > c_f\right) = \int_0^{\theta_f} \text{f}(\theta)\, d\theta \ . \tag{102}$$

The polar representation of (102) is shown schematically in figure (44). Although there exists a parameterized representation for the Projected Normal Distribution for the three-dimensional case (Hernandez-Stumpfhauser et al., 2017) and would allow for the evaluation of equation (102), we opt for a computationally more efficient look-up-table approach.

In order to reduce the dimensionality of the approach we perform the simplification

$$\Sigma_g \approx \sigma_m^2\, I_3 \tag{103}$$

where $\sigma_m = max(\sigma_1,\sigma_2,\sigma_3)$ in order to err on the side of caution and I is the identity matrix. Another observation helps to reduce dimensionality: the shape of the distribution $f(\theta) = g/\|g\|$ does not change if the random variable g is scaled for any scalar greater than zero.

The look-up-table can therefore be solely parameterized by the ratio of the expected length to the maximum standard deviation along all axes $E(\|g\|)/\sigma_m$ and the expected angle

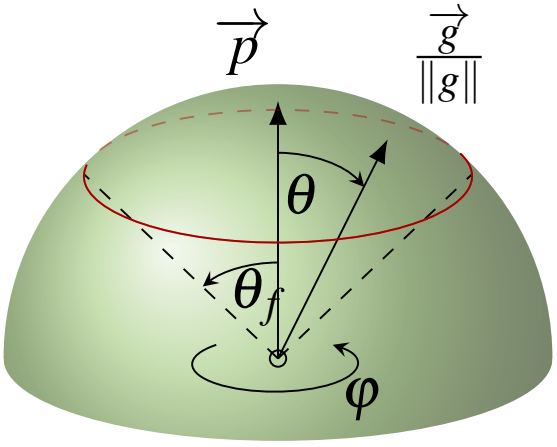

Figure 44: Polar representation of the vector p connecting the antipodal points and the friction cone with half apex angle θ_f depicted in red. Equation (102) evaluates the probability of $g/\|g\|$ being within the friction cone.

$E(\theta)$.

We empirically choose the range of the considered angles θ_{lut} to be

$$\theta_{lut} \in [0, \pi/4] \tag{104}$$

and the range of the considered fractions

$$\frac{\|g\|}{\sigma_m} \in (0, 10] \ . \tag{105}$$

We create a uniform grid within these ranges and use 100000 simulated trials for each grid index. The number 100000 was found empirically by gradually increasing the number of trials by a power of ten until the resulting graph was sufficiently continuous between discrete grid evaluations (see fig. 45) for our application. We consider the number of positive evaluations where

$$\frac{p \cdot g}{\|p\|\|g\|} > c_f \ , \tag{106}$$

as associated probability value, where we choose the friction angle $\theta_f = 25°$ which results in $c_f = \cos\theta_f \approx 0.9$. The grid is then stored and used as look-up table where equation

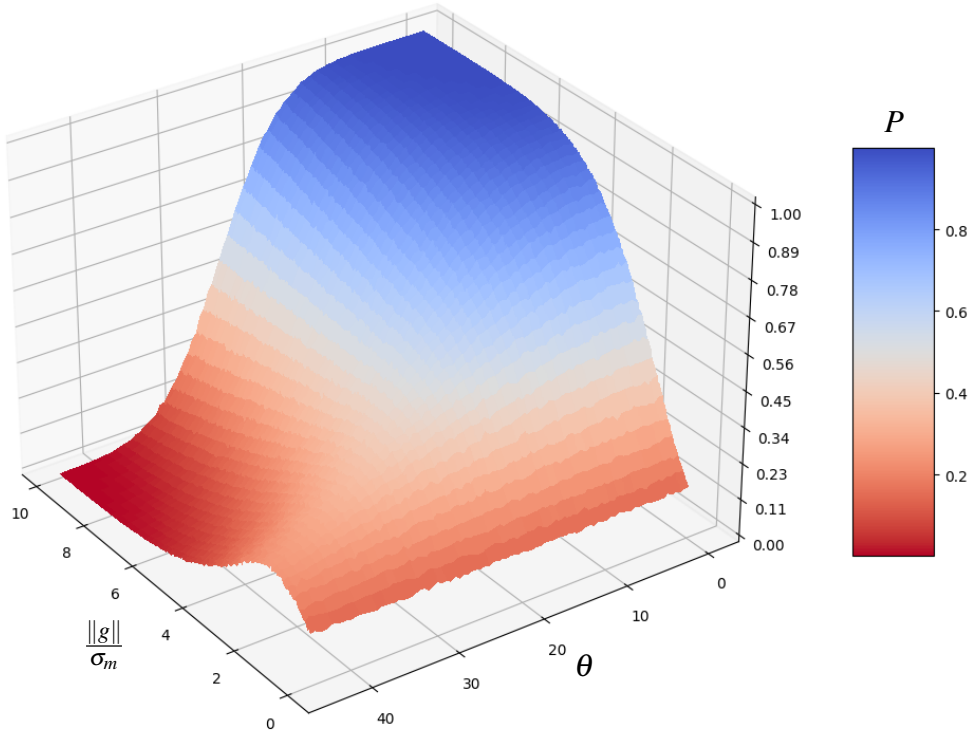

Figure 45: The approximated cumulative distribution function.

(102) can be conveniently evaluated at runtime with high performance. A user defined threshold P_{thr} is used to determine whether the probability of two point to be in force closure is sufficient. Hence, the pair is stored for further evaluations if

$$P_{fc} > P_{thr} \quad , \tag{107}$$

where P_{thr} acts as a trade-off between grasp safety and search time.

A surface plot of the cumulative distribution function that is approximated in this manner can be seen in figure 45 where we divided the ranges into a grid of size 90×200. We use this approximation at runtime to approximate the probability of one point lying in the friction cone of another one and thus to determine the probability of the pair of points to lead to a force-closure grasp. Points outside the value range of the grid are assigned a

probability of zero.

5.3 Redundancy Resolution and Feasibility Checks

If condition (107) is satisfied, the corresponding pair of antipodal points is likely to be in force closure. We therefore know where to place the fingers but do not know the corresponding grasp pose. The distance of the fingers to the end-effector is a constant and the grasp approach vector must be orthogonal to the vector that connects the contact pair. The set of possible end-effector poses that correspond to the pair of contact points is therefore positioned on a circle with radius equivalent to the distance from the fingertips to the end effector. Its center is the mid-point of both points.
Inspired by the work of Cai et al. (2022), we use the TSDF to resolve this under-determination and eliminate potential collisions with the environment at the same time. As the TSDF is a spatial representation of the environment, we can project the gripper into the volume and check corresponding signed distance values for collision. While Cai et al. (2022) use the mesh of the end effector, we uniformly discretize the volume of the end effector and the fingers in the set of points EE. This conveniently allows adapting the discretization of the projected gripper volume to the resolution to the TSDF. Compared to using the surface vertices as Cai et al. (2022), a volumetric representation leads to a more reliable collision evaluation since the occupied space is represented uniformly whereas a mesh can over- or underrepresent segments and even leave small collisions in low resolution areas undetected.
We uniformly sample a set of grasp poses $T \in T_{all}$ on the upper semicircle with angular distances of $\pi/18$. All poses that occupy voxels with negative SDF values are discarded to build the set of non-collision poses T_{nc}

$$T_{nc} = \{T \mid \underset{ee \in EE}{\operatorname{argmin}} \; TSDF(T \cdot ee) > 0\} \; . \tag{108}$$

This process is illustrated by the red boxes in figure 46, where several considered grasp directions collide with the edges of the box and hence are discarded. Of course, it works the same way if a grasp would collide with other obstacles in the scene.
Only the grasp pose $T_g \in T_{nc}$ with the greatest sum of truncated distances is considered

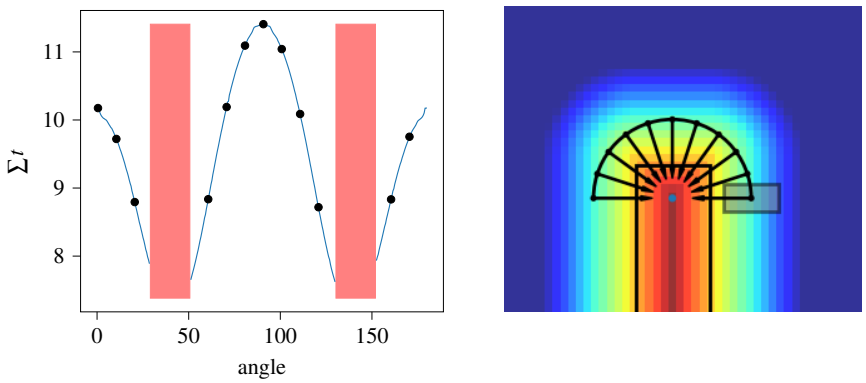

Figure 46: 2D example of the TSDF of a box and evaluation of all grasping directions (right side.) The graph on the right represents the quality function where the considered grasping directions are marked as black dots. The red boxes indicate infeasible angle ranges where the end-effector box collides with negative TSDF values.

for further evaluation

$$T_g = \underset{T \in T_{nc}}{\operatorname{argmax}}\, sd(T) \qquad \text{with} \qquad sd(T) = \sum_{ee \in EE} \text{TSDF}\,(T \cdot ee) \quad . \tag{109}$$

A more typical approach of dealing with collision-distances would be to use the shortest distance or local minima, instead of the sum or the average over all. With this in mind, we initially tried to maximize the shortest distance, but we found this approach too error-prone against estimation-outliers as sometimes faulty voxel estimations could cause poor grasp orientations. Instead, we use the summation in 109 which could be regarded as numeric integration of the required gripper volume or as a low-pass evaluation of the collision distance. In practice this summation leads to grasp approach directions that align with the average inverse TSDF gradient, i.e. most o the time the gripper approaches perpendicular to the surface, which we consider the desired behavior since a perpendicular grasping direction leaves the largest margin for errors during the approach phase of the robot.

A 2D-example of this evaluation is shown in figure 46. The right image shows the TSDF-values as color-scale of a box-shaped object in the middle. The object is supposed to be grasped at the blue contact point in the center of the image. We consider grasp positions

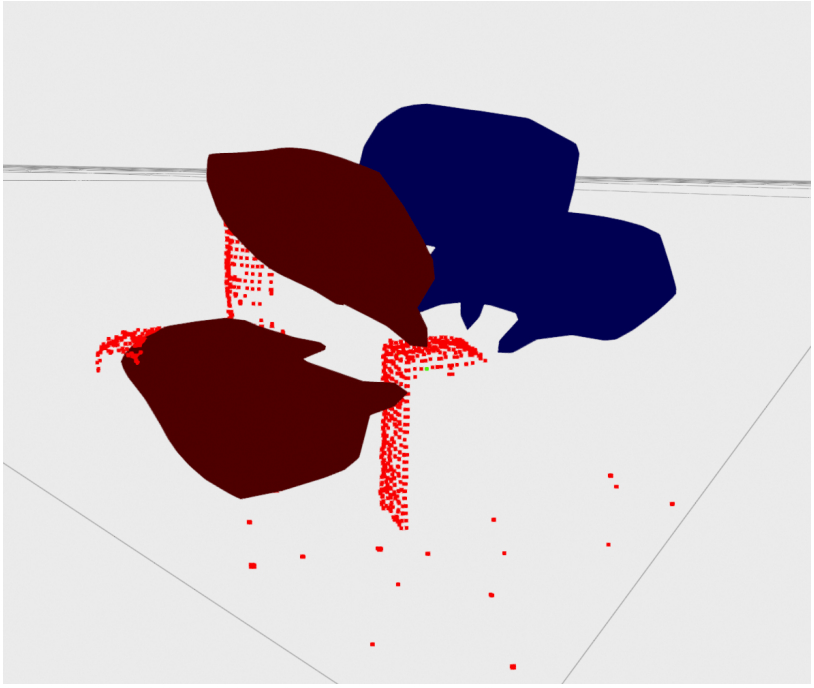

Figure 47: Evaluation of the grasping direction for the partially known box and collision evaluation.

on the upper semicircle around the axis that connects both contact points. These grasp approach candidates are indicated by the black arrows. The radius of the circle corresponds to the length of the robots fingers. For each position, we sum the values of all TSDF elements that would lie within the end-effector volume (shown as the gray rectangle).
The left graph shows the summed TSDF-values over the grasp directions where the black dots correspond to the actually considered grasp directions, i.e. the black arrows. In this example, the approach from the top, i.e. 90 degrees, shows the highest T_g according to equation 109, since it is the "furthest" away from collision.
Grasps that would collide with the object or the rest of the scene include negative elements (displayed in reddish colors) and are discarded (see equation 108). This behavior is indicated by the two red, invalid angle ranges in figure 46 where the end effector would collide with the object's corners.
Another simulated example of the evaluation of the grasp direction is shown in figure 47. The center box is supposed to be grasped (using the green center point) and currently only

the front and the top surface of the box is known. For the sake of visibility we used a low resolution mesh of the end effector and only four grasp directions. In this case the two left, red options would collide with the surrounding objects and are therefore discarded.
Each grasp candidate $T_g \in SE_3$ is then checked for a valid joint configuration where we use the open source library Trac-IK (Beeson & Ames, 2015). If a joint configuration exists we additionally check that configuration utilizing the collision look-up table we proposed in section 4.5.1.
With the grasp candidate checks we perform in this chapter we make sure that:

- all remaining grasp poses are not in self-collision
- all poses are reachable and not close to singularities.
- the end-effector volume of poses is not in collision with the environment

Its worth noting that we neither made sure the joint configuration is not in collision with the environment nor can we ensure that there is a feasible path to that configuration. Nonetheless, we are able to prune the set of grasp candidates of the vast majority of infeasible elements in a negligible time-frame before we opt for computationally expensive path-planning.

5.4 Simulated experiments

We evaluate the proposed approach in two different setups simulated with the physics engine *PyBullet*, which allows us to test our approach over a large number of randomly generated scenes. The first setup (*cluttered*) was adopted from Breyer, Ott, et al. (2022), were randomly chosen objects are iteratively placed at random positions on the table in front of a *Franka Panda* arm in an upright pose. Positions that end up in collision with already placed objects are rejected and resampled until either four objects are placed or a maximum number of attempts is reached. The objects are mostly cylindrical or box-like and large enough to lead to significant occlusions and potential collisions. A wire-frame visualization of most objects for the *cluttered* scenario can be seen in figure 48.
The second setup (*single*) has a similar configuration except for the type and number of objects. For each iteration, we only use a single object randomly chosen from a subset[6]

[6] *https://github.com/eleramp/pybullet-object-models*

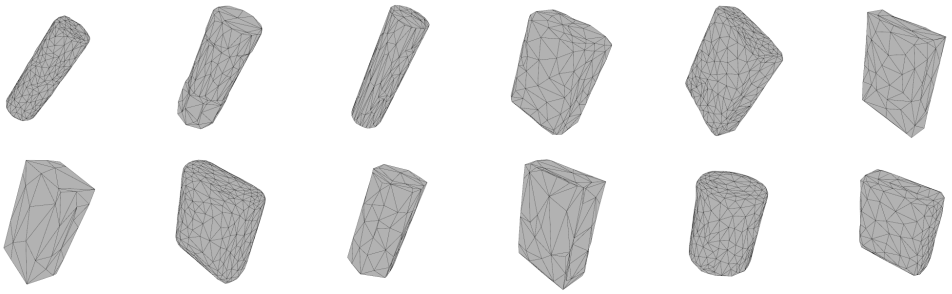

Figure 48: A subset of the objects that were used in the cluttered scenario.

of the YCB dataset (Çalli et al., 2015), which are more complex compared to the first setup. These objects are more diverse than in the *cluttered* setup, and they are depicted in figure 49. The dataset additionally provides simplified collision meshes and physics parameters (e.g. mass, friction parameters, center of mass) that are fine-tuned and tested for robotic manipulation tasks in *PyBullet*. We additionally computed the inertia tensors for each object using *Meshlab* (Cignoni et al., 2021).
Since grasps that require less friction are arguably better. In order to make both setups more discriminative we set the lateral friction coefficients of all objects to 0.2 compared to the setting of 1.0 in Breyer, Ott, et al. (2022) and the original range of the YCB subset [0.3, 0.8]. The lateral friction of the fingers was set to a constant value of 0.5 for all experiments. These friction settings dramatically increase the level of difficulty as only grasps where the fingers are well aligned with the object surface are successful and even

Figure 49: A subset of the objects that were used in the single scenario.

small angular deviations lead to the object slipping when lifted.
The depth images were imposed with Gaussian noise according the noise model of Ahn et al. (2019) for both setups. The bounding box of the target object is provided by the physics simulator.
Although each grasp was checked for inverse kinematic solutions and for collision-free placement of the end effector, occasionally *MoveIt* failed to find a plan. This can occur for two reasons. First, the check for the collision-free placement of the end-effector does not necessarily mean that there is a collision-free configuration for the rest of the arm. Second, we utilize the RRT* (rapidly exploring random trees) path-planner (Karaman & Frazzoli, 2011) that is available in the OMPL (Open Motion Planning Library, Şucan, Moll, and Kavraki (2012)). As the name suggests, this path planning algorithm utilizes random sampling of collision-free configurations and occasionally did not return a valid path within the pre-set duration limit of two seconds. These cases for which no valid path could be found, were removed from the results. All tests were performed using a *NVIDIA 3090 RTX* graphics card and an *Intel i7-9700K* CPU. Table 9 lists the used parameters of our approach.

Table 9: Parameters used for the experiments.

TSDF size		0.3^3 m^3
Voxel count per side		80
Policy rate		5 Hz
Number of view candidates	$\lvert T_c \rvert$	16
Maximum number of views		100
Probability threshold	P_{thr}	85%
Linear velocity		5 cm/s
Gripper Force		10 N
Critical ramp angle	$\arctan(f)$	25°
Initial τ estimation	τ_0	0.9
Initial v value	v_0	0.0375
Utility weight	γ	0.05

A grasp was considered a success if the object could be lifted by 10 cm. We report the following metrics for performance evaluation:

- Success Rate (**SR**): Ratio of runs where the target was successfully grasped.
- Failure Rate (**FR**): Ratio of runs where a grasp was detected, but failed during execution.
- Abortion Rate (**AR**): Ratio of runs where no grasp on the target object was found.
- Search time (**ST**): Time elapsed between receiving a bounding box and returning a grasp configuration.
- Distance (**D**): Distance traveled by the end effector before opting for grasp execution.

We compare our results against those of Breyer, Ott, et al. (2022) (*vgn*) where we left the parameters of their algorithm at their default values except for the window size which was set to 25 since the authors report the highest success rate for this setting. Additionally, we compare our results against those of Cai et al. (2022) (*vcpd*). They propose a closed loop approach where a neural network is leveraged to predict the grasp-quality of pairs of contact points. Their control strategy commands linear movement towards the grasp pose with the highest estimated quality until a distance threshold is reached without any exploration strategy. Hence, if appropriate grasping possibilities are initially occluded, the algorithm might settle for insufficient ones. Therefore, we additionally implemented the network as grasping evaluator ($o.w.vcpd_n$) in our pipeline in order to measure the performance gain introduced by our exploratory method.
Cai et al. (2022) assume that a sufficient grasp must exist by the time they opt for execution and perform maximum-selection over all grasp qualities. Therefore, their approach lacks a built-in quality threshold that indicates whether a grasp candidate is "sufficient" or not. Despite our efforts, we did not succeed in determining a suitable threshold either, as we found that there were always grasp candidates whose quality is very close to the maximum value even when the scene (subjectively) did not show any valid possibility. As a workaround, we force the exploration of $o.w.vcpd_n$ trials for n views before permitting grasp execution.
In order to evaluate effectiveness of the employed information gain formulation we additionally implemented two other formulations in the proposed pipeline ($ours_{se}$, $ours_{rs}$).

Table 10: Results for both simulated setups.

Setup	Policy	SR	FR	AR	ST (s)	D (m)
clutter	*ours*	92 %	7 %	1 %	6.2	0.20
	$ours_{se}$	91 %	8 %	1 %	5.6	0.18
	$ours_{rs}$	86 %	7 %	7 %	6.4	0.21
	vgn	55 %	37 %	8 %	8.3	0.29
	vcpd	52 %	48 %	-	4.5	0.12
	$o.w.vcpd_{10}$	58 %	42 %	-	3.7	0.08
	$o.w.vcpd_{20}$	73 %	27 %	-	5.7	0.17
	$o.w.vcpd_{30}$	87 %	13 %	-	7.8	0.25
single	*ours*	90 %	8 %	2 %	7.1	0.22
	vgn	77 %	21 %	2 %	8.9	0.30
	vcpd	51 %	49 %	-	5.1	0.15
	$o.w.vcpd_{10}$	58 %	42 %	-	3.8	0.08
	$o.w.vcpd_{20}$	64 %	38 %	-	5.8	0.17
	$o.w.vcpd_{30}$	75 %	25 %	-	7.9	0.26

The first is the summed entropy, which is similar to eq. (67) but without averaging over the visible voxels. The second one is the rear-side formulation, as utilized by Breyer, Chung, et al. (2022), which counts the number of visible voxels with negative distance values.

The mean results of $\sim$240 trials in both setups (*single*, *clutter*) can be seen in table 10. Since *vcpd* performs maximum selection of estimated grasp qualities as well as collision scores its abortion rate is at a constant 0%. The success rate of trials where we implemented *vcpd* as an alternative grasping evaluator ($o.w.vcpd_n$) into our approach show a significant improvement compared to the original closed-loop approach *vcpd*. Our utility function (69) often computed trajectories similar to the one visible on the right in figure 36 which lead to more complete object representations and helped the network to make informed decisions, where in case of $n = 20$ the slightly increased distance and search time are negligible. In the cluttered setup the performance of our probabilistic force closure grasp evaluator is comparable to $o.w.vcpd_{30}$. However, our approach shows a significant advantage in the single setup where the objects are more complex. This might be related

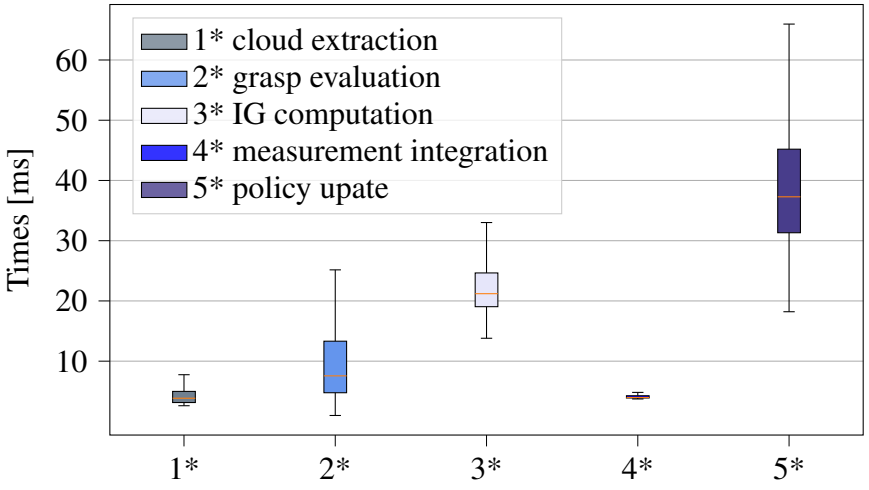

Figure 50: Boxplot of times for noteworthy algorithm components.

to the fact that the network of Cai et al. (2022) was trained using primitive-shaped objects which were also used in the cluttered setup and highlights the invariance of our approach to the used object category.

The beneficial effect of the entropy based information gain formulation on our approach is evident in the results table 10. The binary partitioning into known and unknown subsets of the rear-side formulation *(rs)* volume did not motivate our algorithm to improve the reconstruction and smearing effects of the TSDF had a greater impact on the performance, leading to a higher failure rate. Additionally, the robot could get stuck when the rear side has been fully explored and still no valid grasp was, leading to an increased number of aborted runs. Although the proposed IG formulation shows a slightly lower abortion rate than I_{ae} the results are not conclusive and further tests in more diverse setups might be necessary.

5.4.1 Computation times

Figure (50) shows the notable computation times for several sub-tasks of the proposed algorithm, as well as the duration of complete update steps of the policy. The plot was obtained from 30 simulated grasp trials in the *single* setting.

The duration for the grasp evaluation tends to increase with the number of reconstructed

points that need to be considered, i.e. the longer the exploration lasts the more the object is "known" and the more grasping possibilities need to be evaluated. Hence, durations in the upper end of this grasp-evaluation distribution happen mostly close to the execution stage of the algorithm. Its worth noting that due to the larger size of objects in the *cluttered* setting the median of the grasp duration is roughly 5 ms higher than in the *single* setting.
The median update duration of the proposed algorithm is 37 milliseconds which corresponds to a frequency of 27 Hz. This means that typical sensor frequencies of 30 Hz can be processed (with the occasional frame drop). This represents a significant improvement over the algorithm of Breyer, Ott, et al. (2022) where the median policy update takes 118 milliseconds even though their IG formulation is less complex, and they do not perform collision checks for grasp candidates.
The main bottlenecks of our algorithm are the computation of the information gain as well as the evaluation of grasp options. Hence, outsourcing these tasks into separate threads should be considered in future work as both tasks do not necessarily have to be processed with every incoming sensor frame.

5.5 Real-world experiments

The intended main benefit of our algorithm is the ability to handle severe surface dependent noise, which is difficult to reproduce in simulation. Figure (51) presents four scenarios that highlight the advantages of our approach and showcase the challenges due to partial occlusion and obstacles (scene a, c, d), transparency (b, c) and reflections (a) in the assistive-robot context. These objects were chosen deliberately to highlight the benefits of the proposed algorithm. Except for the lemon, recordings of all objects show strong sensor noise that varies across the surface.
The target object of scene (a) is the corned beef that is partially hidden behind the box. Although it is a seemingly simple box-shaped object, it has a reflective, almost textureless top side that can introduce strong noisy behavior. The reason for this is shown in the left image of figure 52. The white dots of the IR-emitter are not visible on the top of the object. However, they are visible from other angles but still far less distinct than those on the objects side or those on the table. This creates disparity errors which manifest as strong noisy behavior and smearing effects that are not present at the side of the can. In

scene (b) and (c) the target object is the isolated soft drink bottle and the detergent bottle in clutter respectively. Both scenes serve a similar purpose where parts of the object are semi-transparent and introduce strong noise while other parts, such as the label, show comparatively normal behavior.
A typical point cloud of a plastic bottle can be seen in the right image of figure 52. The non-transparent cap of the bottle and the label section in the center can usually be reconstructed cleanly. However, the semi-transparent neck of the bottle and the lower part are particularly noisy and pose a significant problem. The target of scene (d) is the partially occluded lemon. Its surface did not show any additional noise besides the regular one in our experiments. Thus, scene (d) represents a "trouble-free" setting where only collisions and occlusions pose a problem.
Again, we compare our approach against (*vgn*) and (*vcpd*) and report the results in table 11. Unsurprisingly, in scene (d) all policies were able to grasp the object with high accuracy. We assume this is due to the relatively benign surface properties of the lemon.

In the other scenes however, the performance of both networks show significant variations. This is especially apparent in scene (a) and (b). In scene (a), the distorted form of the TSDF (due to the reflective top surface) seemed to "confuse" the *vgn* network, and it tried to grasp the corned beef from the inside.
In scene (b) the noisy measurements in the transparent regions of the bottle led to faulty reconstruction-points. The algorithm of Breyer, Ott, et al. (2022) requires the estimated

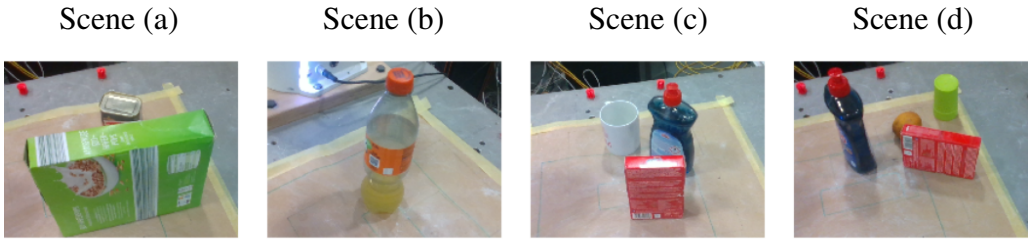

Figure 51: Images of the initial camera view for each tested setting. In scene (a) the target object is the corned beef can behind the cornflakes box, in scene (c) the dishwashing liquid and in scene (d) the lemon in the image center.

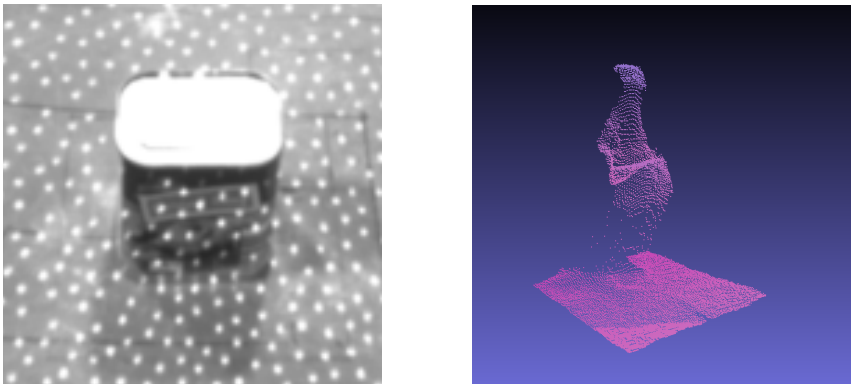

Figure 52: Infrared image of the corned beef can and typical point cloud of a plastic bottle.

quality of a grasp-candidate to be greater than a threshold for a number of consecutive frames. This prevented many failures and led to the bottle only being grasped at the head section. The algorithm of Cai et al. (2022) uses no such quality control technique and the grasp decision is made at the final frame where either the distance limit is reached *(vcpd)* or the frame-limit is exceeded $(o.w.vcpd_{60})$. As a result, it was often a matter of chance whether clearly faulty outlier-points were used as grasp candidates or subjectively well-reconstructed surface points.
Additionally, the benefit of the forced exploration becomes very clear in scene (b). The effect of noise and smearing effects of the transparent bottle becomes somewhat mitigated if the object is explored from multiple perspectives. This led to a significantly decreased ratio of outliers and hence a significantly lower failure rate of $(o.w.vcpd_{60})$ compared to *vcpd*.
Among other things, our algorithm utilizes the estimation variance in order to estimate the success probability of a grasp. This led to consistently high success rates across all scenes because challenging sections of the objects were consequently avoided, and the objects were always grasped at the relatively benign surface segments, i.e. the sides of the corned beef, the cap region or the labelled region of the bottle and the dish washing liquid.
Its worth noting that we experienced noise at some surface sections of the objects that can certainly not be modelled as an unbiased Gaussian distribution, e.g. the semi-transparent regions in scene (b) and (c) of the bottle, which are usually deformed inwards but never outwards or smearing effects at depth disparities that never smear towards the camera

Table 11: Results from real world experiments.

Scene	Policy	SR	FR	AR	ST (s)	D (m)
(a)	*ours*	9 / 10	0 / 10	1 / 10	14.5	0.52
	vgn	6 / 10	4 / 10	0 / 10	13.1	0.38
	vcpd	3 / 10	7 / 10	0 / 10	11.0	0.39
	$o.w.vcpd_{60}$	8 / 10	2 / 10	0 / 10	15.2	0.55
(b)	*ours*	10 / 10	0 / 10	0 / 10	18.6	0.67
	vgn	8 / 10	2 / 10	0 / 10	13.5	0.40
	vcpd	1 / 10	9 / 10	0 / 10	9.5	0.32
	$o.w.vcpd_{60}$	6 / 10	4 / 10	0 / 10	16.5	0.59
(c)	*ours*	10 / 10	0 / 10	0 / 10	16.7	0.57
	vgn	9 / 10	1 / 10	0 / 10	16.3	0.55
	vcpd	3 / 10	7 / 10	0 / 10	10.2	0.35
	$o.w.vcpd_{60}$	7 / 10	3 / 10	0 / 10	16.2	0.59
(d)	*ours*	9 / 10	1 / 10	0 / 10	14.4	0.52
	vgn	8 / 10	0 / 10	2 / 10	16.0	0.50
	vcpd	5 / 10	5 / 10	0 / 10	12.2	0.45
	$o.w.vcpd_{60}$	9 / 10	1 / 10	0 / 10	14.1	0.51

but always in the view direction. Nonetheless, these measurements show a very high variance and hence associated voxel estimations are reliably assigned relatively high estimation variances by our proposed reconstruction algorithm (see section 3.3). For this reason, benign surface sections are consistently considered first for grasping, long before the estimation variance of faulty reconstruction segments converges to acceptable values.

5.6 Conclusion

We presented a closed-loop grasp system based on a probabilistic, truncated signed distance function that achieves 6-dof grasping of unknown household objects with real-time performance. The system is able to deal with the various sources of error in assistive robotics, such as heavy, non-constant sensor noise and occlusions and does neither require prior object knowledge nor a dataset tailored to the use case.

The presented method consistently finds stable grasps in challenging situations where others perform worse. Simulated as well as real-world experiments show the effectiveness

and reliability of the approach.

6. Prototype

We implemented the algorithmic pipeline that was presented in chapter 3, 4 and 5 pipeline in a proof-of-concept hardware setup.

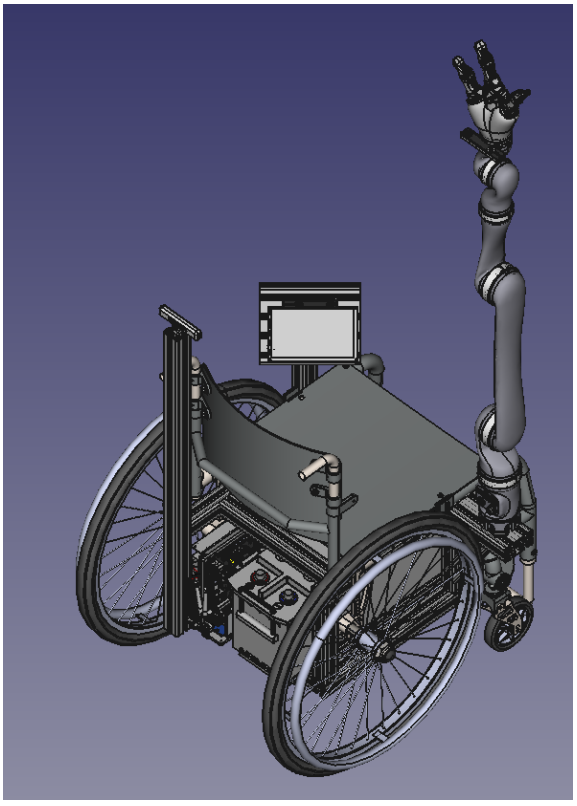

Figure 53: Early 3d-model of the prototype.

6.1 Hardware

Figure 53 shows our original 3D-conceptualization. It is the result of considerations about which hardware is required for our application and how the corresponding dimensions can be accommodated for on the wheelchair.

Figure 54: The current prototype.

We acquired a conventional manually operated wheelchair, sometimes called active wheelchair, and modified it using aluminum profiles in order to hold the required hardware:

- the robot arm on the front right,
- the touch display on the front left,
- a computer that is mounted beneath the seat.

The "final" prototype can be seen in figure 54. Our initial plans of mounting an additional camera that looks over the left shoulder of the user were abandoned as we found in our experiments that the adjustable, end-effector mounted camera can cover most scenarios. The camera mounted at the top left rarely had any added value, as the additional distance meant that the spatial resolution was significantly lower and the static nature of the mount meant that occlusions could not be resolved.
For the prototype we use the *Kinova Jaco*. The *Kinova Jaco* requires no additional controller box, weights 5.2 kg, has an average power consumption of 25 W and since it is a CE certified medical product and therefore can be covered by (german) health insur-

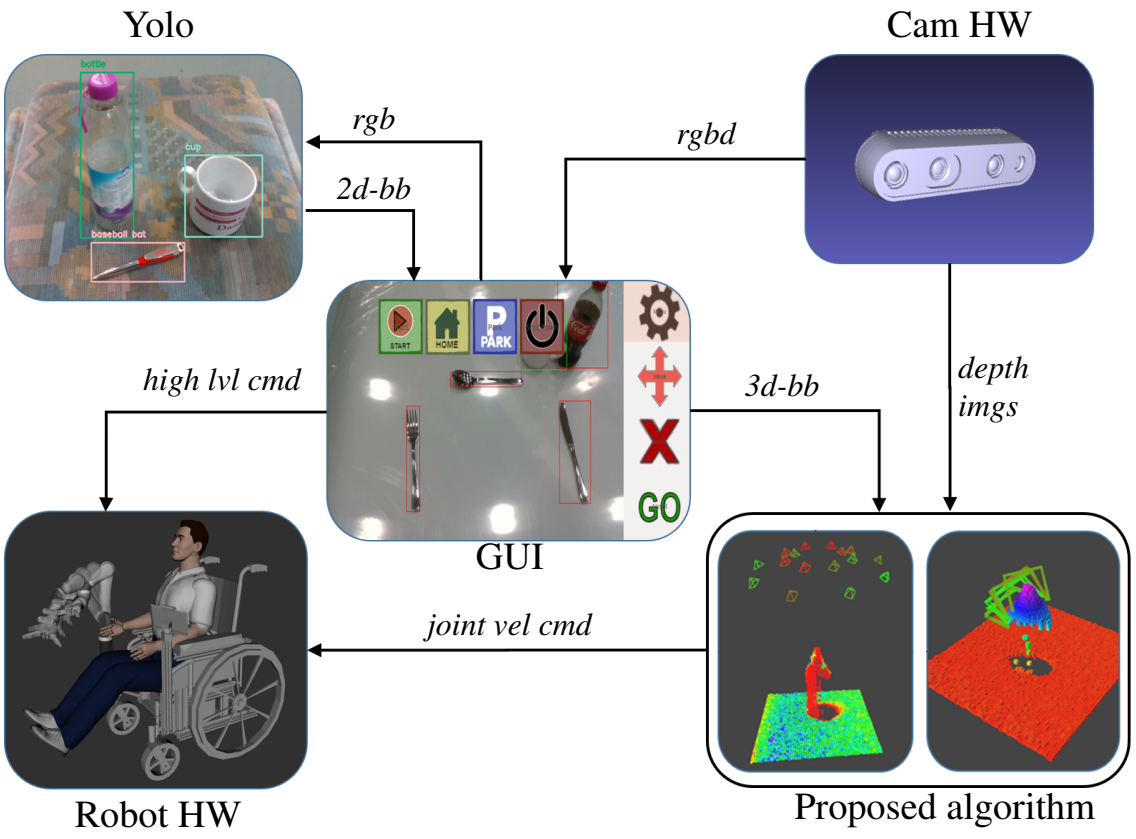

Figure 55: Schematic flowchart of the prototype.

ance, it is a "common" product within the robotic assistive market (*Specifications of Jaco assistive robotic arm*, 2025). The fact that we already had a such a robot arm available in the laboratory was certainly not an insignificant factor in the decision-making process, considering the considerable price of (assistive) robot arms.

The assistive device is aimed at people that live with Spinal muscular atrophy, Muscular dystrophy, Amyotrophic lateral sclerosis (ALS), Spinal cord injury, Cerebral palsy, Quadriplegia, Amputation, Stroke.

6.2 Software

Figure 55 shows the current algorithmic flowchart of the prototype, where the grasping flowchart from figure 14 is shown in the bottom right corner. The grasping algorithm experienced a lot of re-parametrization changes and was adapted to accommodate for the new robot arm e.g. the collision map from section 4.5.1 was rebuild using the collision model of the wheelchair, the finger-to-end-effector distance from section 5.3 was adapted, the controller-code from equation 89 was rewritten to match the available API of the *Kinova Jaco* and the smaller number of joints and so on. The algorithm presented so far works well with different objects and lighting settings. In the previous experiments in section 5.5 (and in the corresponding publication Schaub et al., 2024) we worked under the assumption that the position of the "grasping-object" is known, as the detection and pose estimation is outside the scope of this thesis. This gap is closed in our prototype by an off-the-shelf object detection network. In addition, we implemented a self-made graphical user interface, where the user is able to start and stop the autonomous grasping pipeline and is able to choose between objects in case several "manipulatable" objects are visible.

The object detection network is the YOLO network that is available in the OpenCV library (Bradski, 2019). We opt for the YOLOv4-tiny version of C.-Y. Wang, Bochkovskiy, and Liao (2021) with pre-trained weights as the average precision is comparable, but the computational effort is dramatically reduced which results in a significantly higher frame rate. The network was implemented as a ROS node and a custom service data type was developed where the request consists of an RGB-image and the response is a list of bounding boxes with corresponding label strings. The extensive list of detectable object classes was reduced to the object types that we consider "graspable" and detections of other classes are ignored.

The basic functionality of the graphical user interface (GUI) was written by the author with C++ using the QT library and iteratively extended by student projects in the context of the course "autonomous systems" with additional buttons and associated functionality. The GUI is another ROS node that subscribes to a camera topic where the depth information is combined with the color information via perspective alignment. For every frame the current color information is sent to the object detection node via service call and the returning bounding-boxes are then overlaid with the color image and shown on the dis-

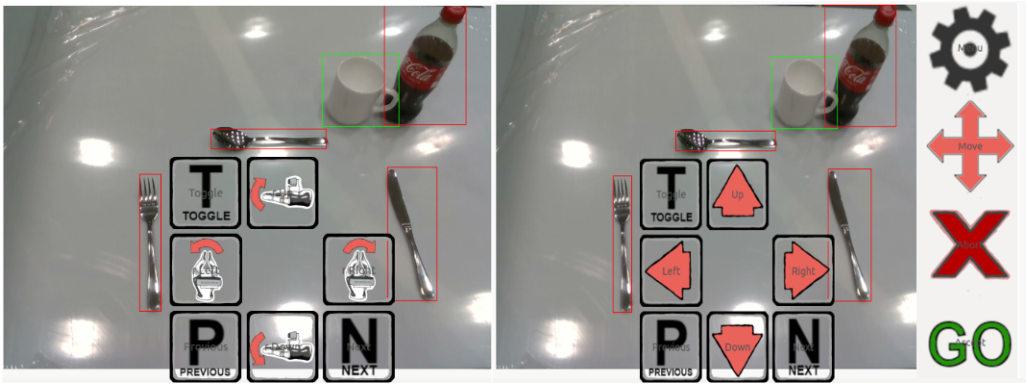

Figure 56: The developed graphical user interface

play. The user is then able to select the desired object by iterating though the current list. Since depth values are available for (almost) all pixels, we can calculate a rough estimate for the 3D bounding box that corresponds to the selected object. As soon as the user presses the green "GO" button, this 3D bounding box is slightly inflated in all directions to ensure that it completely contains the object and then given to the proposed grasping pipeline to find grasps within.

For convenience reasons, a simplified Cartesian control for the robot was also implemented in the graphical user interface, which can be seen in figure 56. It can be used if the current perspective does not show the desired object bounding box, either due to occlusions or because the corresponding object detection did not exceed the confidence threshold. In both cases a small change of perspective usually helps and that can be conveniently achieved via pressing the arrow symbols on the touch display.

6.3 Experiments

We implemented the algorithm that was presented in the previous chapter 5 into the prototype and carried out a series of qualitative tests for two reasons:

- to fine tune pre-configured parameters that were tailored to the laboratory setup.
- to ensure the functionality of the system in a lifelike setting.

The testing setups were all similar in that one or several objects were placed on a table with roughly the same height as the mounting position of the arm. This height selection was chosen because the presented algorithm currently only considers view candidates on the semi sphere **above** the object, i.e. if the object is placed too high there might be too few reachable view-candidates to scan the object from antipodal sides and hence the scanning phase could fail. Although this could be seen as a hard restriction, we argue that the user environment must be tailored to accommodate working with the robotic arm anyway. However, future prototype versions could include an additional vertical axis to enable grasping higher located objects. One characteristic example of the testing setups can be seen in figure 57.
No additional steps were taken to improve the scene conditions, e.g. no additional light source was used to enhance the illumination and no lights were shut off in an effort to avoid over-illumination.
We used the same objects that were presented in the real world experiments in section 5.5 as we consider those objects and their individual challenges to be representative for the household setting. We also experimented with a variety of typical household objects, such as soft drink bottles with varying filling levels, cups and drinking glasses, cornflakes-boxes and cutlery.
The results of experiments with the prototype are very promising, and the algorithm performs robust. Except for a few classification errors the objects were reliably detected and the corresponding 2D bounding-boxes presented on the display. The user could then iterate though the visible, manipulatable objects and start the scanning phase which typically took only a few seconds before a grasp was found, the chosen object was picked up, and then got presented to the user at chest height. This semi-automated process allowed the user to automatically carry out complex manipulation processes without any mentally taxing motion commands.
Compared to our experiments with the *Franka Panda* the grasps were even more robust. This is a consequence of the different gripper. The gripper of the *Mico* has additional finger joints that enable wrapping the fingers around the object. Moreover, its contact areas are coated in rubber with a significantly higher friction coefficient. This led to very reliable performance of the grasping pipeline and the number of erroneous grasp predictions was negligible. Overall, the performance of the algorithm was satisfactory.

Figure 57: Typical testing setup.

Nevertheless, the algorithm suffers from design limitations and currently can not handle two types of objects, i.e. objects with very *problematic* surfaces and relatively thin or tiny objects. The first category includes e.g. fully transparent objects, such as a drinking glass and semi-reflective, low contrast objects such as white cups. These objects cannot be addressed with our algorithm as they cannot be scanned with the precision our that the algorithm needs, regardless of whether a Lidar, a stereo camera or a structured-light sensor is used. Hence, the user would be required to either modify the objects e.g. by applying texture, or fall back to the still enabled manual control via joystick.
The second problematic category consists of filigree objects, such as cutlery. This is a consequence of using the TSDF as a scene representation. The predefined truncation distance and resolution in combination with the accuracy of the low-cost sensor leads to a blurring effect of high-frequency structures. This makes it difficult to accurately reconstruct thin objects such as forks and pens and impairs our quality prediction of grasp candidates for these kinds of objects. However, it might be argued that the level of precision (sensory as well as robot-motion) that is required to grasp and move e.g. a fork is beyond the capa-

bilities of the assistive hardware and therefore beyond the scope of this thesis.

6.4 Future Work

The current hardware is a proof-of-concept setup and as such there is still a long way to go before the final customer-ready product. Most of the still outstanding tasks fall within the scope of engineering work. One possible further development that would simultaneously deal with the tricky cases mentioned above, as well as improve the overall success rate is to consult the user before execution. Before opting for execution one could utilize a non-maxima suppression filter on all currently valid grasp-candidates and present a diverse set of possibilities to the user. The user could then select the subjectively best one via the same interface that was previously used to select the desired object.
This train of thought of involving the user more is in line with the findings of Latikka, Savela, Koivula, and Oksanen (2021). They show that the user acceptance is significantly higher for semi-autonomous, assistive robots than for fully autonomous ones. Involving the user at critical decision points would give him or her a sense of control instead of being at the mercy of the system. Although this semi-autonomous strategy has already been partially implemented via the existing buttons (motion-arrows, stop, home and object selection), it must also characterize the further development of the prototype. Future features should present the user with options at a high semantic level but hide the computational complexity of the underlying task to keep the required mental load low.
Instead of the adapted active wheelchair presented, an electric wheelchair must of course be used. Currently, no battery is implemented, and the setup is powered by cables. In the same way, the graphical user interface needs to be seen as proof-of-concept since in the current setup the user can only operate the display using his or her second arm (and by leaning slightly forward). As such it seemingly contradicts the use case we are aiming for. But due to the large number of disabilities covered by the product and their varying severity, the user interface must inevitably be tailored to the individual user. With this in mind, researching and integrating of typical assistive input hardware such as a chin- or lips-joystick or eye-tracking is a logical follow-up project.
Currently, our algorithmic pipeline ends when an object grasp pose is chosen and then handed to the user at chest height. Conceptually, the functionality does not have to end

here. Further development should include saving the pick-up position as well as the collision scene and then moving the object back there collision-free. Likewise, implementing a teaching mode to automate recurring processes (e.g. drinking and setting down) would certainly be useful quality-of-life feature. These functionalities should be mapped in the GUI in a meaningful manner which should then serve as a state-machine instead of the currently implemented one-way functionality i.e. object decision followed by the proposed pipeline. The platform presented offers plenty of room for useful follow-up development, both academic and didactic and is at the time of writing the subject of several master-theses.

7. Conclusion

This thesis presented a closed-loop algorithm for grasping in the assistive context. The unique challenges of the unknown, unstructured environment are not fully solved by the current state of the art.
The proposed pipeline is able to deal with various sources of error in assistive robotics, such as heavy, non-constant sensor noise and occlusions and does neither require prior object knowledge or a dataset tailored to the use case. Our system is lightweight in terms of the required memory and efficient enough to be real-time capable. We do not require an initial grasp proposal to guide the algorithm but created a pipeline that autonomously searches for and grasp with high success probability. By combining this with a reliable estimate of local surface elements, our algorithm can consistently find stable grasps where other methods perform worse.

7.1 Summary of Contributions

In chapter 2 we were able to lift a state-of-the-art algorithm from its inherent restriction of 2.5 dimensional grasps to the 6-DOF solution space that is required in the assistive context. We were able to identify challenges that are unique in the household context and are not fully covered by the state-of-the-art. We were able to solve some of these challenges using various heuristics. Our experiments in the context of the two publications revealed the need for a more holistic approach to assistive grasping.

Chapter 3 proposed a novel sensor fusion algorithm that combines a pre-computed sensor noise model with an adaptive estimation of local distribution parameters depending on the measurement noise. Our experiments on a publicly available dataset show that this approach is able to accurately reconstruct the scene. Moreover, we showed that the local estimation of the surface estimation error strongly correlates with the actual error. This allows subsequent grasping algorithms to disregard grasp proposals that correspond to "uncertain" surface estimations and therefore properly utilize the usually significantly overdetermined grasp solution space.

In chapter 4 we presented an algorithm that translates Cartesian movement commands into corresponding joint movements at controller level. This algorithm is able to handle various motion problems instantaneously that are normally solved via time-consuming path planning algorithms or prevented by choosing a benign initial setup. We show how self-collisions as well as singularities can be avoided and how the user-safety can be ensured at controller level. We utilized the volumetric representation of the previous chapter and computed the information gain of reachable view candidates by evaluating the visible entropy. This strategy allows us to "search" for grasps if the initial perspective does not show valid grasping options.

Chapter 5 used the scene representation from chapter 3 and proposed a novel method to sample grasping candidates. We approximated their success probability in an explainable manner by utilizing the estimation variance of the corresponding surface segments. We showed how a majority of colliding grasp candidates be detected with high efficiency during runtime where other algorithms stop, validate the solution via path-planning and then resume if necessary. The simulated and real experiments demonstrated the synthesis of the algorithms that were presented in this thesis. The results show that the presented approach performs more reliable than comparable methods in simulated as well as challenging, real-world scenarios.

Chapter 6 presented a functional, assistive grasping prototype. We extended the previous academic work by several user-centric components, implemented an object detector and adapted the existing algorithm to match the wheelchair setup. We wrote a custom graphical user interface that handles the communication between the algorithm and the user via a touch display. This prototype allows users to perform complex object manipulation tasks in a semi-automated manner without the need for mentally taxing motion commands.

References

9283:1998(E), I. (2003). *Manipulating industrial robots - performance criteria and related test methods* (Vol. 2003; Standard). International Organization for Standardization. doi: 10.3403/01859871u

Abdulhafiz, W., & Khamis, A. (2013, 01). Handling data uncertainty and inconsistency using multisensor data fusion. *Advances in Artificial Intelligence*, *2013*, 241260:1-241260:11. doi: 10.1155/2013/241260

Ahn, M. S., Chae, H., Noh, D., Nam, H., & Hong, D. W. (2019). Analysis and noise modeling of the intel realsense d435 for mobile robots. *2019 16th International Conference on Ubiquitous Robots (UR)*, 707-711. doi: 10.1109/urai.2019.8768489

Al-Halimi, R., & Moussa, M. A. (2017). Performing complex tasks by users with upper-extremity disabilities using a 6-dof robotic arm: A study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*, 686-693. doi: 10.1109/tnsre.2016.2603472

Anders Grunnet-Jepsen, J. W., John N. Sweetser. (2017). *Tuning depth cameras for best performance.* Institute of Electrical and Electronics Engineers (IEEE). Retrieved from `https://dev.intelrealsense.com/docs/tuning-depth-cameras-for-best-performance` doi: 10.1109/msp.2017.2669347

Avigal, Y., Paradis, S., & Zhang, H. (2015). 6-dof grasp planning using fast 3d reconstruction and grasp quality cnn. *IEEE Transactions on Robotics*, *abs/2009.08618*, 1393-1403. doi: 10.1109/tro.2015.2492863

Beeson, P., & Ames, B. (2015). Trac-ik: An open-source library for improved solving of generic inverse kinematics. *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 928-935. doi: 10.1109/humanoids.2015.7363472

Bengtson, S. H., Bak, T., Struijk, L. N. S. A., & Moeslund, T. B. (2019). A review of computer vision for semi-autonomous control of assistive robotic manipulators (arms). *Disability and Rehabilitation: Assistive Technology*, *15*, 731-745. doi: 10.1080/17483107.2019.1615998

Bernardini, F., Mittleman, J., Rushmeier, H. E., Silva, C. T., & Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualiza-*

tion and Computer Graphics, *5*, 349-359. doi: 10.1109/2945.817351

Bicchi, A., & Kumar, V. R. (2000). Robotic grasping and contact: a review. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, *1*, 348-353. doi: 10.1109/robot.2000.844081

Blum, H., Sarlin, P.-E., Nieto, J. I., Siegwart, R. Y., & Cadena, C. (2021). The fishyscapes benchmark: Measuring blind spots in semantic segmentation. *International Journal of Computer Vision*, *129*, 3119-3135. doi: 10.1007/s11263-021-01511-6

Botsch, M., & Kobbelt, L. P. (2003). High-quality point-based rendering on modern gpus. *11th Pacific Conference onComputer Graphics and Applications, 2003. Proceedings.*, 335-343. doi: 10.1109/pccga.2003.1238275

Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., … Vanhoucke, V. (2018). Using simulation and domain adaptation to improve efficiency of deep robotic grasping. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 4243-4250. doi: 10.1109/icra.2018.8460875

Bradski, G. (2019). The OpenCV Library. *CRAN: Contributed Packages*. doi: 10.32614/cran.package.opencv

Breyer, M. (2022). *Efficient robotic grasping in unstructured environments.* Doctoral dissertation, ETH Zürich. doi: 10.3929/ethz-b-000597371

Breyer, M., Chung, J. J., Ott, L., Siegwart, R. Y., & Nieto, J. I. (2022). Volumetric grasping network: Real-time 6 dof grasp detection in clutter. In *Conference on robot learning* (p. 6208). MDPI AG. doi: 10.3390/s22166208

Breyer, M., Ott, L., Siegwart, R. Y., & Chung, J. J. (2022). Closed-loop next-best-view planning for target-driven grasping. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1411-1416. doi: 10.1109/iros47612.2022.9981472

Burgess-Limerick, B., Lehnert, C. F., Leitner, J., & Corke, P. (2023). An architecture for reactive mobile manipulation on-the-move. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 1623-1629. doi: 10.1109/icra48891.2023.10161021

Bylow, E., Sturm, J., Kerl, C., Kahl, F., & Cremers, D. (2013). Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and*

systems. Robotics: Science and Systems Foundation. doi: 10.15607/rss.2013.ix.035

Cai, J., Su, J., Zhou, Z., Cheng, H., Chen, Q., & Wang, M. Y. (2022, 14–18 Dec). Volumetric-based contact point detection for 7-dof grasping. In K. Liu, D. Kulic, & J. Ichnowski (Eds.), *Proceedings of the 6th conference on robot learning* (Vol. 205, p. 1651-1655). IEEE. doi: 10.1109/robio55434.2022.10011999

Cao, R., Yang, B., Li, Y., Fu, C.-W., Heng, P.-A., & Liu, Y.-H. (2024). Uncertainty-aware suction grasping for cluttered scenes. *IEEE Robotics and Automation Letters*, *9*(6), 4934-4941. doi: 10.1109/lra.2024.3385609

Chandra, R., Dagum, L., Kohr, D., Menon, R., Maydan, D., & McDonald, J. (2019). *Parallel programming in openmp*. Chapman and Hall/CRC. doi: 10.1201/9780429326097-21

Chatterjee, A., & Govindu, V. M. (2016). Noise in structured-light stereo depth cameras: Modeling and its applications. *Time-of-Flight and Structured Light Depth Cameras*, *abs/1505.01936*, 43-79. doi: 10.1007/978-3-319-30973-6_2

Chen, I.-M., & Burdick, J. W. (1993). Finding antipodal point grasps on irregularly shaped objects. *IEEE Transactions on Robotics and Automation*, 507-512. doi: 10.1109/70.246063

Chen, R., Xu, J., & Zhang, S. (2022). Comparative study on 3d optical sensors for short range applications. *Optics and Lasers in Engineering*, 106763. doi: 10.1016/j.optlaseng.2021.106763

Chen, Y., & Medioni, G. G. (1991). Object modeling by registration of multiple range images. *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 2724-2729. doi: 10.1109/robot.1991.132043

Choi, H., Crump, C., Duriez, C., Elmquist, A., Hager, G. D., Han, D., ... Trinkle, J. C. (2020). On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences*, *118*. doi: 10.1073/pnas.1907856118

Chu, F.-J., Xu, R., & Vela, P. A. (2023). Real-world multi-object, multi-grasp detection. *Proceedings of the 2023 8th International Conference on Systems, Control and Communications*, 7-18. doi: 10.1145/3634865.3634869

Chung, C.-S., Wang, H., & Cooper, R. A. (2013). Functional assessment and performance

evaluation for assistive robotic manipulators: Literature review. *The Journal of Spinal Cord Medicine*, *36*, 273-289. doi: 10.1179/2045772313y.0000000132

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., & Ranzuglia, G. (2021). Meshlab: an open-source mesh processing tool. In *European interdisciplinary cybersecurity conference* (p. 2046-2069). Springer Science and Business Media LLC. doi: 10.3758/s13428-021-01717-z

Coleman, D., Sucan, I. A., Chitta, S., & Correll, N. (2011). Reducing the barrier to entry of complex robotic software: a moveit! case study. *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*, *abs/1404.3785*, 1-1. doi: 10 .1145/2016741.2016792

Collins, J. J., Brown, R., Leitner, J., & Howard, D. (2021). Traversing the reality gap via simulator tuning. *Journal of Robotics and Mechatronics*, *abs/2003.01369*, 665-675. doi: 10.20965/jrm.2021.p0665

Corke, P. (2017). *Robotics, vision and control: Fundamental algorithms in matlab* (2nd ed., Vol. 118). Cham: Springer. doi: 10.1007/978-3-319-54413-7

Coumans, E. (2019). *Pybullet: Python module for physics simulation.* `https://github.com/bulletphysics/bullet3`. Elsevier BV. (Accessed: 2026-02-20) doi: 10 .1016/j.softx.2019.100273

Dahl, V., Aanæs, H., & Bærentzen, J. (2010). Surfel based geometry reconstruction. *Theory and Practice of Computer Graphics 2010, TPCG 2010 - Eurographics UK Chapter Proceedings*. doi: 10.2312/LocalChapterEvents/TPCG/TPCG10/039-044

Dai, Q., Zhang, J., Li, Q., Wu, T., Dong, H., Liu, Z., ... Wang, H. (2022). Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects. In *European conference on computer vision* (p. 374-391). Springer Nature Switzerland. doi: 10.1007/978-3-031-19842-7_22

Delmerico, J. A., Isler, S., Sabzevari, R., & Scaramuzza, D. (2017). A comparison of volumetric information gain metrics for active 3d object reconstruction. *Autonomous Robots*, *42*, 197-208. doi: 10.1007/s10514-017-9634-0

Dengler, N., Pan, S., Kalagaturu, V., Menon, R., Dawood, M., & Bennewitz, M. (2023). Viewpoint push planning for mapping of unknown confined spaces. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1178-1184. doi: 10.1109/iros55552.2023.10341809

Depierre, A., Dellandréa, E., & Chen, L. (2018). Jacquard: A large scale dataset for robotic grasp detection. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3511-3516. doi: 10.1109/iros.2018.8593950

Dersimonian, R., Dersimonian, R., Laird, N. M., & Laird, N. M. (2001). Meta-analysis in clinical trials. *Applied Statistics in the Pharmaceutical Industry*, *7 3*, 397-424. doi: 10.1007/978-1-4757-3466-9_16

Dietrich, V., Chen, D., Wurm, K. M., von Wichert, G., & Ennen, P. (2016). Probabilistic multi-sensor fusion based on signed distance functions. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1873-1878. doi: 10.1109/icra .2016.7487333

Dong, W., Lao, Y., Kaess, M., & Koltun, V. (2022). Ash: A modern framework for parallel spatial hashing in 3d perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*, 1-18. doi: 10.1109/tpami.2022.3214347

Dong, W., Wang, Q., Wang, X., & Zha, H. (2018). Psdf fusion: Probabilistic signed distance function for on-the-fly 3d data fusion and scene reconstruction. *Lecture Notes in Computer Science*, *abs/1807.11034*, 714-730. doi: 10.1007/978-3-030 -01240-3_43

Driessen, B., Evers, H., & v Woerden, J. A. (2001). Manus—a wheelchair-mounted rehabilitation robot. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, *215*, 285-290. doi: 10.1243/ 0954411011535876

Eisoldt, M., Gaal, J., Wiemann, T., Flottmann, M., Rothmann, M., Tassemeier, M., & Porrmann, M. (2022). A fully integrated system for hardware-accelerated tsdf slam with lidar sensors (hatsdf slam). *Robotics and Autonomous Systems*, *156*, 104205. doi: 10.1016/j.robot.2022.104205

Eppner, C., Mousavian, A., & Fox, D. (2021). Acronym: A large-scale grasp dataset based on simulation. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 6222-6227. doi: 10.1109/icra48506.2021.9560844

Fang, H., Wang, C., Gou, M., & Lu, C. (2020). Graspnet-1billion: A large-scale benchmark for general object grasping. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11441-11450. doi: 10.1109/cvpr42600.2020 .01146

Fang, H.-S., Gou, M., Wang, C., & Lu, C. (2023, 08). Robust grasping across diverse sensor qualities: The graspnet-1billion dataset. *The International Journal of Robotics Research*, *42*, 1094-1103. doi: 10.1177/02783649231193710

Fehr, M., Furrer, F., Dryanovski, I., Sturm, J., Gilitschenski, I., Siegwart, R. Y., & Cadena, C. (2017). Tsdf-based change detection for consistent long-term dense reconstruction and dynamic object discovery. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5237-5244. doi: 10.1109/icra.2017.7989614

Ferri, G., Tesei, A., Stinco, P., & LePage, K. D. (2019). A bayesian occupancy grid mapping method for the control of passive sonar robotics surveillance networks. *OCEANS 2019 - Marseille*, 1-9. doi: 10.1109/oceanse.2019.8867152

Foresi, G., Freddi, A., Kyrki, V., Monteriù, A., Muthusamy, R., Ortenzi, D., & Pagnotta, D. P. (2017). An avoidance control strategy for joint-position limits of dual-arm robots. *IFAC-PapersOnLine*, *50*, 1056-1061. doi: 10.1016/j.ifacol.2017.08.217

Franka Emika GmbH. (2023). Franka emika panda – betriebsanleitung [Computer software manual]. München, Deutschland: IEEE. doi: 10.1109/cacee61121.2023.00018

Geforce rtx™ 4090 gaming x trio 24g. (2026). SAGE Publications: SAGE Business Cases Originals. Retrieved from `https://storage-asset.msi.com/datasheet/vga/global/GeForce-RTX-4090-GAMING-X-TRIO-24G.pdf` doi: 10.4135/9798348847326

Gregorio, D. D., Tombari, F., & di Stefano, L. (2016). Robotfusion: Grasping with a robotic manipulator via multi-view reconstruction. In *Eccv workshops* (p. 634-647). Springer International Publishing. doi: 10.1007/978-3-319-49409-8_54

Grinvald, M., Tombari, F., Siegwart, R. Y., & Nieto, J. I. (2021). Tsdf++: A multi-object formulation for dynamic object tracking and reconstruction. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 14192-14198. doi: 10.1109/icra48506.2021.9560923

Grunnet-Jepsen, A., Sweetser, J., Winer, P., Takagi, A., & Rev, J. W. (2017). Projectors for intel ® realsense ™ depth cameras d4xx. In (p. 1267-1276). IEEE. doi: 10.1109/cvprw.2017.167

Gualtieri, M., ten Pas, A., Saenko, K., & Platt, R. W. (2016). High precision grasp pose detection in dense clutter. *2016 IEEE/RSJ International Conference on Intelligent*

Robots and Systems (IROS), 598-605. doi: 10.1109/iros.2016.7759114

Guennebaud, G., & et al., B. J. (1996). *Eigen v3*. http://eigen.tuxfamily.org. Ghent University. doi: 10.21825/agora.v12i2.9532

Haider, A., & Hel-Or, H. (2022). What can we learn from depth camera sensor noise? *Sensors*, *22*, 5448. doi: 10.3390/s22145448

Halmetschlager-Funek, G., Suchi, M., Kampel, M., & Vincze, M. (2019). An empirical evaluation of ten depth cameras: Bias, precision, lateral noise, different lighting conditions and materials, and multiple sensor setups in indoor environments. *IEEE Robotics and Automation Magazine*, *26*, 67-77. doi: 10.1109/mra.2018.2852795

Herlant, L., Holladay, R., & Srinivasa, S. S. (2016). Assistive teleoperation of robot arms via automatic time-optimal mode switching. *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 35-42. doi: 10.1109/hri.2016.7451731

Hernandez-Stumpfhauser, D., Breidt, F. J., & van der Woerd, M. J. (2017). Supplementary material of the general projected normal distribution of arbitrary dimension: Modeling and bayesian inference. *Bayesian Analysis*. doi: 10.1214/15-ba989

Holz, D., & Behnke, S. (2013). Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In *Annual meeting of the ieee industry applications society* (p. 61-73). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-33932-5_7

Holz, D., Topalidou-Kyniazopoulou, A., Stückler, J., & Behnke, S. (2015). Real-time object detection, localization and verification for fast robotic depalletizing. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1459-1466. doi: 10.1109/iros.2015.7353560

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*. Retrieved from `https://octomap.github.io` (Software available at `https://octomap.github.io`) doi: 10.1007/s10514-012-9321-0

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science and Engineering*, *9*(3), 90-95. doi: 10.1109/mcse.2007.55

Hunter, J. E., & Schmidt, F. L. (2006). Methods of meta-analysis: Correcting error and bias in research findings. In (p. 236-237). Elsevier BV. doi: 10.1016/j.evalprogplan

.2006.06.002

Intel ® realsense tm d400 series product family. (2017). IEEE. Retrieved from `https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf` doi: 10.1109/cvprw.2017.167

Iturralde, K., Kinoshita, T., & Bock, T. (2019). Grasped element position recognition and robot pose adjustment during assembly. *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*. doi: 10.22260/isarc2019/0062

Jiang, Y., Moseson, S., & Saxena, A. (2011). Efficient grasping from rgbd images: Learning using a new rectangle representation. *2011 IEEE International Conference on Robotics and Automation*, 3304-3311. doi: 10.1109/icra.2011.5980145

Ka, H. W., Chung, C.-S., Ding, D., James, K. A., & Cooper, R. A. (2017). Performance evaluation of 3d vision-based semi-autonomous control method for assistive robotic manipulator. *Disability and Rehabilitation: Assistive Technology*, *13*, 140-145. doi: 10.1080/17483107.2017.1299804

Kadambi, A., Bhandari, A., & Raskar, R. (2014, 07). 3d depth cameras in vision: Benefits and limitations of the hardware. In (p. 3-26). doi: 10.1007/978-3-319-08651-4_1

Kadian, A., Truong, J., Gokaslan, A., Clegg, A., Wijmans, E., Lee, S., ... Batra, D. (2020). Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, *5*, 6670-6677. doi: 10.1109/lra.2020.3013848

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., ... Levine, S. (2025). Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *Machines*, *abs/1806.10293*, 451. doi: 10.3390/machines13060451

Karaman, S., & Frazzoli, E. (2011). Incremental sampling-based algorithms for optimal motion planning. *Robotics*, *abs/1005.0416*, 267-274. doi: 10.7551/mitpress/9123.003.0038

Kasaei, H., & Kasaei, M. M. (2023). Mvgrasp: Real-time multi-view 3d object grasping in highly cluttered environments. *Robotics and Autonomous Systems*, *abs/2103.10997*, 104313. doi: 10.1016/j.robot.2022.104313

Khansari, M., Kappler, D., Luo, J., Bingham, J., & Kalakrishnan, M. (2020). Action

image representation: Learning scalable deep grasping policies with zero real world data. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 3597-3603. doi: 10.1109/icra40945.2020.9197415

Khoshelham, K., & Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, *12*, 1437-1454. doi: 10.3390/s120201437

Kim, D., Carballo, D., Carlo, J. D., Katz, B., Bledt, G., Lim, B., & Kim, S. (2020). Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2464-2470. doi: 10.1109/icra40945.2020.9196777

Kim, W., Luong, T. A., Ha, Y. J., Doh, M., Yax, J. F. M., & Moon, H. (2023). High-fidelity drone simulation with depth camera noise and improved air drag force models. *Applied Sciences*, 10631. doi: 10.3390/app131910631

Kinova jaco assistive robot, user guide. (2021). Institute of Electrical and Electronics Engineers (IEEE). Retrieved from `https://assistive.kinovarobotics.com/uploads/EN-UG-007-Jaco-Assistive-robot-user-guide-r7.1.pdf` doi: 10.1109/mra.2020.3004149

Koren, Y., & Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1398-1404. doi: 10.1109/robot.1991.131810

Kriegel, S., Rink, C., Bodenmüller, T., & Suppa, M. (2013). Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, *10*, 611-631. doi: 10.1007/s11554-013-0386-6

Latikka, R., Savela, N., Koivula, A., & Oksanen, A. (2021). Attitudes toward robots as equipment and coworkers and the impact of robot autonomy level. *International Journal of Social Robotics*, *13*, 1747-1759. doi: 10.1007/s12369-020-00743-9

Lehnert, C. F., Sa, I., McCool, C., Upcroft, B., & Perez, T. (2016). Sweet pepper pose detection and grasping for automated crop harvesting. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2428-2434. doi: 10.1109/icra.2016.7487394

Lenz, I., Lee, H., & Saxena, A. (2015). Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, *34*, 705-724. doi: 10.1177/

0278364914549607

Li, F., Du, Y., & Liu, R. (2016). Truncated signed distance function volume integration based on voxel-level optimization for 3d reconstruction. In *3d image processing, measurement , and applications* (p. 3DIPM-398.1-3DIPM-398.6). Society for Imaging Science and Technology. doi: 10.2352/issn.2470-1173.2016.21.3dipm -398

Li, Y., Wei, W., Li, D., Wang, P., Li, W., & Zhong, J. (2022). Hgc-net: Deep anthropomorphic hand grasping in clutter. In *2022 international conference on robotics and automation (icra)* (p. 714-720). IEEE. doi: 10.1109/icra46639.2022.9811756

Liang, H., Ma, X., Li, S., Görner, M., Tang, S., Fang, B., ... Zhang, J. (2019). Pointnetgpd: Detecting grasp configurations from point sets. *2019 International Conference on Robotics and Automation (ICRA)*, 3629-3635. doi: 10.1109/icra.2019.8794435

Lidec, Q. L., Jallet, W., Montaut, L., Laptev, I., Schmid, C., & Carpentier, J. (2024). Contact models in robotics: a comparative analysis. *IEEE Transactions on Robotics*, *abs/2304.06372*, 3716-3733. doi: 10.1109/tro.2024.3434208

Lin, C.-J., Chang-Chien, Y.-S., & Chen, Y.-Q. (2023). Image-servo robotic manipulator using yolo model for intelligent fish farm. *2023 International Automatic Control Conference (CACS)*, 1-6. doi: 10.1109/cacs60074.2023.10326201

Lin, H.-Y., Liang, S.-C., & Chen, Y.-K. (2021). Robotic grasping with multi-view image acquisition and model-based pose estimation. *IEEE Sensors Journal*, *21*, 11870-11878. doi: 10.1109/jsen.2020.3030791

Lobbezoo, A., & Kwon, H.-J. (2023). Simulated and real robotic reach, grasp, and pick-and-place using combined reinforcement learning and traditional controls. *Robotics*, *12*, 12. doi: 10.3390/robotics12010012

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 163-169. doi: 10.1145/37401.37422

Lourenço, F., & Araújo, H. (2021). Intel realsense sr305, d415 and l515: Experimental evaluation and comparison of depth estimation. In *Visigrapp.* SCITEPRESS - Science and Technology Publications. doi: 10.5220/0010254203620369

Low, K.-L. (2021). Linear least-squares optimization for point-to-plane icp surface registration. In (p. 48-71). Elsevier BV. doi: 10.1016/j.neucom.2021.08.080

Ma, H., Qin, R., Shi, M., Gao, B., & Huang, D. (2024). Sim-to-real grasp detection with global-to-local rgb-d adaptation. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, *abs/2403.11511*, 13910-13917. doi: 10.1109/icra57147.2024.10611165

Mahler, J. (2024). Efficient policy learning for robust robot grasping. In (p. 1-6). IEEE. doi: 10.1109/iciea61579.2024.10665124

Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., ... Goldberg, K. (2017). Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *Robotics: Science and Systems XIII*, *abs/1703.09312*. doi: 10.15607/rss.2017.xiii.058

Mahler, J., Matl, M., Liu, X., Li, A., Gealy, D. V., & Goldberg, K. (2018). Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, *abs/1709.06670*, 5620-5627. doi: 10.1109/icra.2018.8460887

Mahler, J., Pokorny, F. T., Hou, B., Bauza, M., Herzog, A., Srinivasa, S. S., ... Goldberg, K. (2019). Learning ambidextrous robot grasping policies. *Science Robotics*, *4*(26). doi: 10.1126/scirobotics.aau4984

Mahler, J., Pokorny, F. T., Hou, B., Roderick, M., Laskey, M., Aubry, M., ... Goldberg, K. (2016). Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1957-1964. doi: 10.1109/icra.2016.7487342

Menon, R., Zaenker, T., Dengler, N., & Bennewitz, M. (2023). Nbv-sc: Next best view planning based on shape completion for fruit mapping and reconstruction. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4197-4203. doi: 10.1109/iros55552.2023.10341855

Monica, R., & Aleotti, J. (2017). Contour-based next-best view planning from point cloud segmentation of unknown objects. *Autonomous Robots*, *42*, 443-458. doi: 10.1007/s10514-017-9618-0

Monica, R., & Aleotti, J. (2018). Surfel-based next best view planning. *IEEE Robotics and Automation Letters*, *3*, 3324-3331. doi: 10.1109/lra.2018.2852778

Moravec, H. P., & Elfes, A. (1985). High resolution maps from wide angle sonar. *Proceed-*

ings. 1985 IEEE International Conference on Robotics and Automation, *2*, 116-121. doi: 10.1109/robot.1985.1087316

Morrison, D., Corke, P., & Leitner, J. (2018). Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *Robotics: Science and Systems XIV*, *abs/1804.05172*. doi: 10.15607/rss.2018.xiv.021

Morrison, D., Corke, P., & Leitner, J. (2019a). Learning robust, real-time, reactive robotic grasping. *The International Journal of Robotics Research*, *39*, 183-201. doi: 10 .1177/0278364919859066

Morrison, D., Corke, P., & Leitner, J. (2019b). Multi-view picking: Next-best-view reaching for improved grasping in clutter. *2019 International Conference on Robotics and Automation (ICRA)*, 8762-8768. doi: 10.1109/icra.2019.8793805

Mousavian, A., Eppner, C., & Fox, D. (2019). 6-dof graspnet: Variational grasp generation for object manipulation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2901-2910. doi: 10.1109/iccv.2019.00299

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., ... Fitzgibbon, A. W. (2011). Kinectfusion: Real-time dense surface mapping and tracking. *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 127-136. doi: 10.1109/ismar.2011.6092378

Nguyen, C. V., Izadi, S., & Lovell, D. R. (2012). Modeling kinect sensor noise for improved 3d reconstruction and tracking. *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 524-530. doi: 10.1109/3dimpvt.2012.84

Nguyen, V.-D. (1988). Constructing force-closure grasps. *The International Journal of Robotics Research*, *7*(3), 3-16. doi: 10.1177/027836498800700301

Niedermayr, D., & Wolfartsberger, J. (2022). Analyzing the potential of a time-of-flight depth sensor for assembly assistance. In *Ieee international symposium on multimedia* (p. 648-659). Elsevier BV. doi: 10.1016/j.procs.2022.01.263

Pan, J., Chitta, S., & Manocha, D. (2012). Fcl: A general purpose library for collision and proximity queries. *2012 IEEE International Conference on Robotics and Automation*, 3859-3866. doi: 10.1109/icra.2012.6225337

Pan, S., Hu, H., Wei, H., Dengler, N., Zaenker, T., & Bennewitz, M. (2022, 04). One-shot view planning for fast and complete unknown object reconstruction. *IEEE Robotics*

and Automation Letters, 1463-1470. doi: 10.1109/lra.2022.3140449

Pfister, H., Zwicker, M., van Baar, J., & Gross, M. H. (2007). Surfels: surface elements as rendering primitives. *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, 15-22. doi: 10.1145/1274871.1274874

Pinto, L., & Gupta, A. K. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 3406-3413. doi: 10.1109/icra.2016.7487517

Reed, A., Albin, D., Pasricha, A., Roncone, A., & Heckman, C. (2024, 02). Transformer-based learning models of dynamical systems for robotic state prediction. *PREPRINT*. doi: 10.21203/rs.3.rs-3919154/v1

Rogers, A., Eshaghi, K., Nejat, G., & Benhabib, B. (2023). Occupancy grid mapping via resource-constrained robotic swarms: A collaborative exploration strategy. *Robotics*, *12*, 70. doi: 10.3390/robotics12030070

Rusu, R. B., & Cousins, S. (2011, May 9-13). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)* (p. 1-4). Shanghai, China: IEEE. doi: 10.1109/icra.2011.5980567

Saulnier, K., Atanasov, N. A., Pappas, G. J., & Kumar, V. R. (2020). Information theoretic active exploration in signed distance fields. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 4080-4085. doi: 10.1109/icra40945.2020.9196882

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., ... Batra, D. (2019). Habitat: A platform for embodied ai research. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9338-9346. doi: 10.1109/iccv.2019.00943

Schaub, H., Leuze, N., Hoh, M., & Schöttl, A. (2023). Probabilistic fusion of depth maps with local variance estimation. *2023 IEEE SENSORS*, 1-4. doi: 10.1109/sensors56945.2023.10325039

Schaub, H., & Schöttl, A. (2020). 6-dof grasp detection for unknown objects. *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*, 400-403. doi: 10.1109/acit49673.2020.9208918

Schaub, H., Schöttl, A., & Hoh, M. (2021). 6-dof grasp detection for unknown objects using surface reconstruction. *2021 3rd International Congress on Human-*

Computer Interaction, Optimization and Robotic Applications (HORA), 1-6. doi: 10.1109/hora52670.2021.9461271

Schaub, H., Schöttl, A., & Hoh, M. (2022). Probabilistic fusion of depth maps with a reliable estimation of the local reconstruction quality. *IEEE Robotics and Automation Letters*, *7*, 11982-11989. doi: 10.1109/lra.2022.3208371

Schaub, H., Wolff, C., Hoh, M., & Schöttl, A. (2024). Probabilistic closed-loop active grasping. *IEEE Robotics and Automation Letters*, *9*(4), 3964-3971. doi: 10.1109/lra.2024.3371328

Shahar, D. J. (2017). Minimizing the variance of a weighted average. *Open Journal of Statistics*, *07*, 216-224. doi: 10.4236/ojs.2017.72017

Shi, G., Zhu, Y., Tremblay, J., Birchfield, S., Ramos, F. T., Anandkumar, A., & Zhu, Y. (2021). Fast uncertainty quantification for deep object pose estimation. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 5200-5207. doi: 10.1109/icra48506.2021.9561483

Si, Z., Zhu, Z., Agarwal, A., Anderson, S., & Yuan, W. (2022). Grasp stability prediction with sim-to-real transfer from tactile sensing. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7809-7816. doi: 10.1109/iros47612.2022.9981863

Specifications of jaco assistive robotic arm. (2025). Ovid Technologies (Wolters Kluwer Health). Retrieved from `https://assistive.kinovarobotics.com/product/jaco-robotic-arm#ProductSpecs` doi: 10.1097/01.bmsas.0001095736.16694.70

Stückler, J., & Behnke, S. (2012). Benchmarking mobile manipulation in everyday environments. *2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 1-6. doi: 10.1109/arso.2012.6213389

Şucan, I. A., Moll, M., & Kavraki, L. E. (2012, December). The Open Motion Planning Library. *IEEE Robotics and Automation Magazine*, *19*(4), 72–82. (`https://ompl.kavrakilab.org`) doi: 10.1109/MRA.2012.2205651

Sundermeyer, M., Mousavian, A., Triebel, R., & Fox, D. (2021). Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 13438-13444. doi: 10.1109/icra48506.2021.9561877

ten Pas, A., Gualtieri, M., Saenko, K., & Platt, R. W. (2017). Grasp pose detection in point clouds. *The International Journal of Robotics Research*, *36*, 1455-1473. doi: 10.1177/0278364917735594

Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., & Birchfield, S. (2023). Deep object pose estimation for semantic robotic grasping of household objects. *Communications in Computer and Information Science*, *abs/1809.10790*, 15-30. doi: 10.1007/978-3-031-31417-9_2

Tsai, R. Y., & Lenz, R. K. (1989). A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, *5*, 345-358. doi: 10.1109/70.34770

Tung, K., Su, J., Cai, J., Wan, Z., & Cheng, H. (2022). Uncertainty-based exploring strategy in densely cluttered scenes for vacuum cup grasping. *2022 International Conference on Robotics and Automation (ICRA)*, 3483-3489. doi: 10.1109/icra46639.2022.9811599

Varley, J., DeChant, C., Richardson, A., Ruales, J., & Allen, P. K. (2017). Shape completion enabled robotic grasping. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2442-2447. doi: 10.1109/iros.2017.8206060

Vit, A., & Shani, G. (2018). Comparing rgb-d sensors for close range outdoor agricultural phenotyping. *Sensors*, *18*, 4413. doi: 10.3390/s18124413

Vogiatzis, G., & Hernández, C. (2011). Video-based, real-time multi-view stereo. *Image and Vision Computing*, *29*, 434-441. doi: 10.1016/j.imavis.2011.01.006

Vuong, Q. H., Vikram, S., Su, H., Gao, S., & Christensen, H. I. (2023). How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies? *IEEE Access*, *abs/1903.11774*, 136809-136824. doi: 10.1109/access.2023.3339568

Waldron, K. J., & Schmiedeler, J. (2016). Kinematics. In *Springer handbook of robotics* (p. 11-36). Springer International Publishing. doi: 10.1007/978-3-319-32552-1_2

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2021). Scaled-yolov4: Scaling cross stage partial network. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13024-13033. doi: 10.1109/cvpr46437.2021.01283

Wang, L., Guo, R., Vuong, Q. H., Qin, Y., Su, H., & Christensen, H. I. (2023). A real2sim2real method for robust object grasping with neural surface reconstruction.

2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), *abs/2210.02685*, 1-8. doi: 10.1109/case56687.2023.10260554

Wang, Y., James, S., Stathopoulou, E. K., Beltrán-González, C., Konishi, Y., & Del Bue, A. (2019). Autonomous 3-d reconstruction, mapping, and exploration of indoor environments with a robotic arm. *IEEE Robotics and Automation Letters*, *4*(4), 3340-3347. doi: 10.1109/LRA.2019.2926676

Whelan, T., Leutenegger, S., Salas-Moreno, R. F., Glocker, B., & Davison, A. J. (2015). Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and systems.* Robotics: Science and Systems Foundation. doi: 10.15607/rss.2015.xi.001

Wijaya, S. F. A. E., Purnomo, D. S., Utomo, E. B., & Anandito, M. A. (2019). Research study of occupancy grid map mapping method on hector slam technique. *2019 International Electronics Symposium (IES)*, 238-241. doi: 10.1109/elecsym.2019.8901657

Yan, X., Hsu, J., Khansari, M., Bai, Y., Pathak, A., Gupta, A., … Lee, H. (2018). Learning 6-dof grasping interaction via deep geometry-aware 3d representations. In *Proceedings of the ieee international conference on robotics and automation (icra)* (pp. 3766–3773). IEEE. doi: 10.1109/ICRA.2018.8460609

Yang, B., Singh, S., Grotz, M., Boots, B., & Smith, J. R. (2007). Dynamo-grasp: Dynamics-aware optimization for grasp point detection in suction grippers. In *Conference on robot learning* (p. 18-33). Elsevier BV. doi: 10.1016/j.mechmachtheory.2006.02.007

Yang, D., Tosun, T., Eisner, B., Isler, V., & Lee, D. (2021). Robotic grasping through combined image-based grasp proposal and 3d reconstruction. In *2021 ieee international conference on robotics and automation (icra)* (p. 6350-6356). IEEE. doi: 10.1109/icra48506.2021.9562046

Yang, J., Gao, Y., Li, D., & Waslander, S. L. (2021). Robi: A multi-view dataset for reflective objects in robotic bin-picking. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9788-9795. doi: 10.1109/iros51168.2021.9635871

Yang, J., Li, D., & Waslander, S. L. (2021). Probabilistic multi-view fusion of active stereo depth maps for robotic bin-picking. *IEEE Robotics and Automation Letters*, *6*, 4472-4479. doi: 10.1109/lra.2021.3068706

Ye, X., Cui, H., Wang, L., Xie, S., & Ni, H. (2023). Vision-guided hierarchical control and autonomous positioning for aerial manipulator. *Applied Sciences*, 12172. doi: 10.3390/app132212172

Yin, L., Yin, Y., & Lin, C. (2011). A new potential field method for mobile robot path planning in the dynamic environments. *2011 Seventh International Conference on Natural Computation*, *11*, 1011-1014. doi: 10.1109/icnc.2011.6022190

Yoshikawa, T. (1985). Dynamic manipulability of robot manipulators. *Transactions of the Society of Instrument and Control Engineers*, *2*, 970-975. doi: 10.9746/sicetr1965.21.970

Yoshikawa, T. (1987). Analysis and control of robot manipulators with redundancy. *The International Journal of Robotics Research*, 32-42. doi: 10.1177/027836498700600103

Zeng, R., Wen, Y.-H., Zhao, W., & Liu, Y.-J. (2020). View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, *6*, 225-245. doi: 10.1007/s41095-020-0179-3

Zhang, X., Chen, R., Li, A., Xiang, F., Qin, Y., Gu, J., ... Su, H. (2023). Close the optical sensing domain gap by physics-grounded active stereo sensor simulation. *IEEE Transactions on Robotics*, *39*, 2429-2447. doi: 10.1109/tro.2023.3235591

Zhang, X., Wang, D., Han, S., Li, W., Zhao, B., Wang, Z., ... He, J. (2022). Affordance-driven next-best-view planning for robotic grasping. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, *abs/2309.09556*, 1411-1416. doi: 10.1109/iros47612.2022.9981472

Zhao, Z., Yu, H., Wu, H., & Zhang, X. (2024). 6-dof robotic grasping with transformer. *IEEE Robotics and Automation Letters*, *abs/2301.12476*, 7063-7069. doi: 10.1109/lra.2024.3416773

Žlajpah, L., & Petrič, T. (2022). Geometric identification of denavit-hartenberg parameters with optical measuring system. In A. Müller & M. Brandstötter (Eds.), *Advances in service and industrial robotics* (p. 3-10). Cham: Springer International Publishing. doi: 10.1007/978-3-031-04870-8_1

Çalli, B., Walsman, A., Singh, A., Srinivasa, S. S., Abbeel, P., & Dollar, A. M. (2015). Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *IEEE Robotics and Automation Magazine*, *abs/1502.03143*,

36-52. doi: 10.1109/mra.2015.2448951

Žlajpah, L., & Petri, T. (2023). Kinematic calibration for collaborative robots on a mobile platform using motion capture system. *Robotics and Computer-Integrated Manufacturing*, *79*, 102446. doi: 10.1016/j.rcim.2022.102446

Erklärung der Urheberschaft

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich aus der Literatur oder aus anderen Quellen übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Die vorgelegten Druckexemplare und die vorgelegte digitale Version sind identisch.

Ort, Datum, Unterschrift