Robust Action Primitives and Visual Perception Pipelines for Automation in Surgical and Industrial Robotics Applications

Kishore Srinivas

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-103
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-103.html

May 14, 2024

Robust Action Primitives and Visual Perception Pipelines for Automation in Surgical and Industrial Robotics Applications

by

Kishore Srinivas

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair
Professor Pieter Abbeel

Spring 2024

Robust Action Primitives and Visual Perception Pipelines for Automation in Surgical and Industrial Robotics Applications

by Kishore Srinivas

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

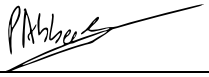
Committee:

Professor Ken Goldberg
Research Advisor

10 May 2024

(Date)

* * * * * * *



Professor Pieter Abbeel
Second Reader

10-May-2024

(Date)

Robust Action Primitives and Visual Perception Pipelines for Automation in Surgical and Industrial Robotics Applications

Abstract

Robust Action Primitives and Visual Perception Pipelines for Automation in Surgical and Industrial Robotics Applications

by

Kishore Srinivas

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

Many robotic manipulation tasks consist of the same few fundamental subtasks: perceiving the environment, building an informative state estimate, interacting with the environment to perform the desired manipulation, evaluating the success of the intended manipulation, and responding to any detected failures. Developing a reliable visual perception system and motion primitives provides the building blocks for automating the execution of these tasks. In this thesis, I present the frameworks we used to automate manipulation tasks in a variety of settings.

First, I consider the task of surgical suturing and introduce STITCH, a novel framework combining deep learning, analytical, and sampling-based approaches to perform 6D needle pose estimation for closed-loop control. We incorporate "interactive perception" for improving needle pose estimation and correction to increase robustness to uncertainty in perception, control, and physics. In experiments, we find that STITCH achieves an average of 4.47 successful sutures with human intervention. Next, I explore the task of tableware decluttering and present TIDY, a framework consisting of a classical vision pipeline for tableware detection and a set of action primitives leveraging multi-object grasping to efficiently clear tableware from a workspace. We developed two algorithms incorporating consolidation and multi-object grasps and find that this leads to a 1.8x improvement in the number of objects transported at once. Finally, I investigate the task of large scale 3D scene reconstruction and introduce Room-Scale LEGS, an online multi-camera 3DGS reconstruction system for large-scale scenes that constructs a hybrid 3D semantic representation using explicit 3D Gaussians for geometry and implicit scale-conditioned hash-grid for the semantics. We find that Room-Scale LEGS produces high quality Gaussian Splats in room-scale scenes with training times 3.5x faster than baselines.

This thesis presents the motivation, methods, and results for each of these frameworks, and briefly explores how they can be extended to other tasks in related domains.

to my family

Contents

List of Figures

List of Tables

Acknowledgments

Being part of the research community at BAIR has been an incredibly enriching experience. To my advisor Professor Ken Goldberg, I want to say thank you for bringing me into the AutoLAB community and for your continued mentorship and support over the last two and a half years.

None of the work presented in this thesis would have been possible without the hard work and dedication of all the collaborators I've been lucky to work with. In particular I'd like to thank Will Panitch, Karthik Dharmarajan, Hansoul Kim, Kush Hari, Justin Yu, Shreya Ganti, and Rishi Parikh. I've learned so much from each and every one of you, and you have grown to become some of my closest friends in the lab.

I want to thank my friends for sharing laughs, putting things in perspective, encouraging me to try new things, and overall making my Berkeley experience so incredibly fun.

Most importantly, I want to thank my family and loved ones who have supported me every step of the way and enabled me to do this work I've so thoroughly enjoyed. From listening to my long-winded explanations to asking challenging questions to encouraging me through trying times, they have always been there for me and pushed me to be the best version of myself, and for that I'm eternally grateful.

Chapter 1

Introduction

1.1 Overview

Many robotic manipulation tasks consist of the same few fundamental subtasks: perceiving the environment, building an informative state estimate, interacting with the environment to perform the desired manipulation, evaluating the success of the intended manipulation, and responding to any detected failures. Each task and each environment poses its own challenge, from cluttered visual fields to delicate objects requiring precise handling to limited environment observability, and requires an approach that is tolerant to the nuances of the environment while also generalizing beyond the immediate task at hand. Developing reliable visual perception systems and motion primitives that operate within the constraints of the environment while being easily extensible provides the building blocks for automating the execution of these tasks.

In this thesis, I present the robust action primitives and vision pipelines we built to tackle the following problems:

1. **Surgical Suturing:** Given a 3D wound phantom and a surgical needle with suturing thread, throw sutures along the length of the wound to stitch it up. We introduce action primitives for needle insertion, extraction, and handover, and a visual pipeline for 6D needle pose estimation, thread tracking, and action failure detection.

2. **Efficient Tableware Decluttering:** Given a workspace of scattered tableware and a collection bin a distance away from the workspace, declutter the table and transfer the tableware to the bin in an efficient and organized manner. We introduce action primitives for consolidating and grasping tableware, and a visual pipeline for identifying tableware and generating grasps.

Finally, continuing in the vein of these visual perception systems, I present recent yet-unpublished work on building rich 3D visual representations of large-scale scenes. We seek to incrementally build room-scale language embedded Gaussian Splats from a mobile robot base to enable localizing open-vocabulary object queries in previously unseen environments.

We hope to combine this hybrid 3D semantic representation with robust actions to build a generalized system for retrieving user-specified objects in a variety of environments.

1.2 Surgical Suturing

Surgical-assist robots such as Intuitive's da Vinci series are used by surgeons to perform over 2 million procedures annually to facilitate minimally-invasive ("keyhole") surgery to reduce pain, blood loss, scarring, complications, and recovery time. It is now the gold standard for procedures involving the appendix, colon, gallbladder, prostate, and many others. These robots are very sophisticated but every movement is currently controlled by human surgeons because surgery is extremely sensitive to errors – there are a vast number of rare but potentially dangerous edge conditions and the consequences of even a single failure can be fatal. So it may be a very long time before fully autonomous robots are sufficiently safe and reliable for the operating room. However, recent advances in AI are opening the door to augmenting surgeon skills when performing specific subtasks such as suturing, debridement, and resection. Goldberg [1] proposed the term "Augmented Dexterity" to describe systems where surgical subtasks are controlled by the robot under the close supervision of the human surgeon who is ready to take over at a moment's notice. Augmented Dexterity has potential to elevate good surgeons to the level of the best surgeons, making surgery safer, faster, and more reliable. In this paper we present new results on augmented dexterity for surgical suturing.

We introduce STITCH, a novel pipeline that performs **S**uture **T**hrows **I**ncluding **T**hread **C**oordination and **H**andoffs. STITCH achieves "level 2" task autonomy as defined by Yang, Cambias, Cleary, Daimler, Drake, Dupont, Hata, Kazanzides, Martel, Patel, *et al.* [2], using novel perception and control techniques. The STITCH perception system combines deep learning, analytical, and sampling-based approaches, and prior knowledge of needle geometry to perform 6D needle pose estimation for closed-loop control. We also incorporate "interactive perception" as proposed by Bohg, Hausman, Sankaran, Brock, Kragic, Schaal, and Sukhatme [3] for improving needle pose estimation and correction to increase robustness to uncertainty in perception, control, and physics [3], [4]. Furthermore, we introduce visual-based thread coordination motions such as sweeping to reduce the risk of the thread tangling with the needle or itself. STITCH also includes a new approach for automated recovery from motion failures using closed loop visual feedback where we retry needle extraction and handover motions if they are unsuccessful.

We present 3 main contributions:

- An augmented dexterity pipeline for robot-assisted surgery capable of suture throwing, thread coordination, and needle handoff.
- A novel visual 6D needle pose estimation framework.

- Experimental results showing an average of 4.47 successful sutures with human intervention.

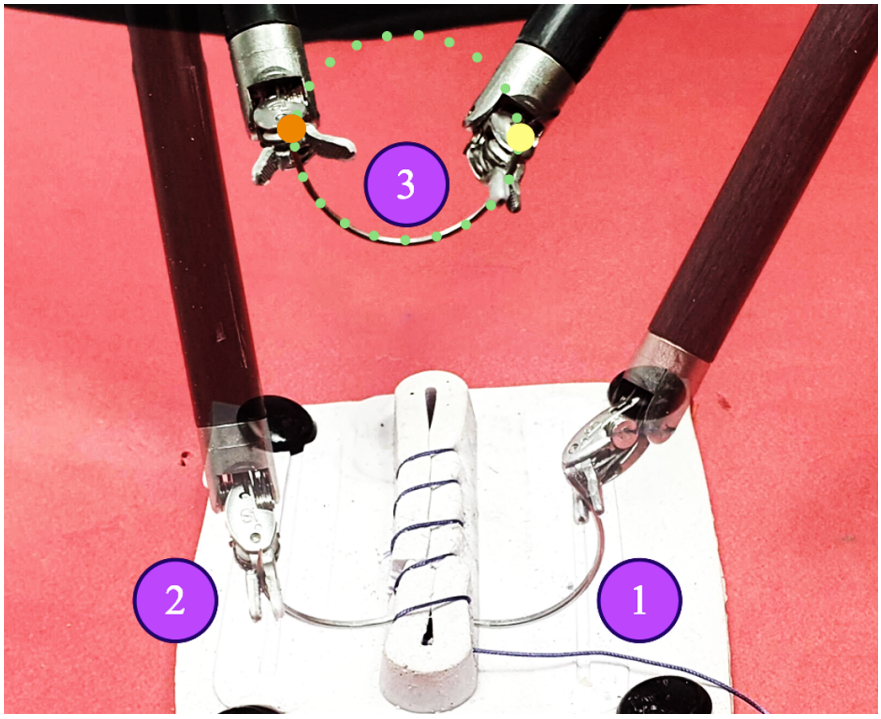


Figure 1.1: **6 sutures performed by STITCH.** Step 1 shows Surgical Needle Insertion, Step 2 shows Needle Extraction, and Step 3 shows Needle Handover with Pose Correction. Detections of the needle endpoints are shown in the photo with the needle tip point shown as the orange circle and the needle swage point shown as the yellow circle. The light green circle represents the estimated needle pose.

1.3 Efficient Tableware Decluttering

The post-meal task of clearing a dining table, commonly referred to as "bussing," requires moving cups, bowls, and utensils that are dispersed across the surface into a bin or tray to be cleaned in the kitchen. This is a common task that occurs after any event involving food service and dish collection, from daily household meals to casual picnics to formal cocktail parties and dinners. Automating this tedious and repetitive task could reduce fatigue and busy work for the skilled waiters who typically perform it.

We define the "Busboy Problem" as the efficient transfer of cups, bowls, and utensils (collectively called tableware) from the table into a designated collection bin while minimizing

the time required for completion. This is an interesting problem for automation because the tableware are of varying shape, requiring low-level planning to execute grasps and high-level planning to consolidate tableware for efficient transport. Even small inaccuracies can lead to toppling or dropping delicate and expensive tableware, so the system must be extremely reliable.

Previous work in multi-object grasping, object manipulation, and grasp candidate generation highlight the efficiency of grasping pre-stacked objects as well as objects manually oriented for multi-object grasps [5], [6]. Whereas these works explore situations with objects are already positioned for said grasps, our work investigates methods of stacking and clustering objects into these favorable positions for multi-object grasps.

In this paper, we present TIDY, a framework and algorithms for the Busboy Problem. We consider a scenario where multiple items are placed on a work surface (see Fig. 1.2), under an RGBD camera. We use the concept of multi-object grasping, which enables the robot to move multiple items simultaneously, thus reducing the number of pick-and-place actions needed.

We present 4 main contributions:

1. Formulation of the Busboy Problem.
2. Action primitives for rearranging and grasping cups, bowls, and utensils.
3. Two algorithms that leverage consolidation and multi-object grasps.
4. Experimental results indicating a 1.8x improvement in OpT.

1.4 Room-Scale Language Embedded Gaussian Splats (LEGS)

For robots to complete natural language search requests such as *"can you bring my spray bottle?"* or *"where is the stapler?"*, the robots must parse such queries, localize the object, and navigate to it. A large body of recent work uses large vision-language models by distilling their outputs into 3D representations like point clouds or NeRFs [7]. These semantic representations have been applied to both manipulation [8]–[10] and large-scale scene understanding [11]–[13], showing promise of using large models zero-shot for open-vocabulary task specification.

One key challenge for scaling these methods to large environments is the underlying 3D representation, which should be flexible to a variety of scales, able to update with new observations, and fast. Although NeRFs are commonly used as the 3D representation for distilling 2D semantic features [14]–[16], scaling NeRFs to large scenes can be cumbersome because they typically rely on a fixed spatial resolution [17]–[19], and are difficult to modify and slow to render. A frequently used alternative is pointclouds [11]–[13], [20], which work

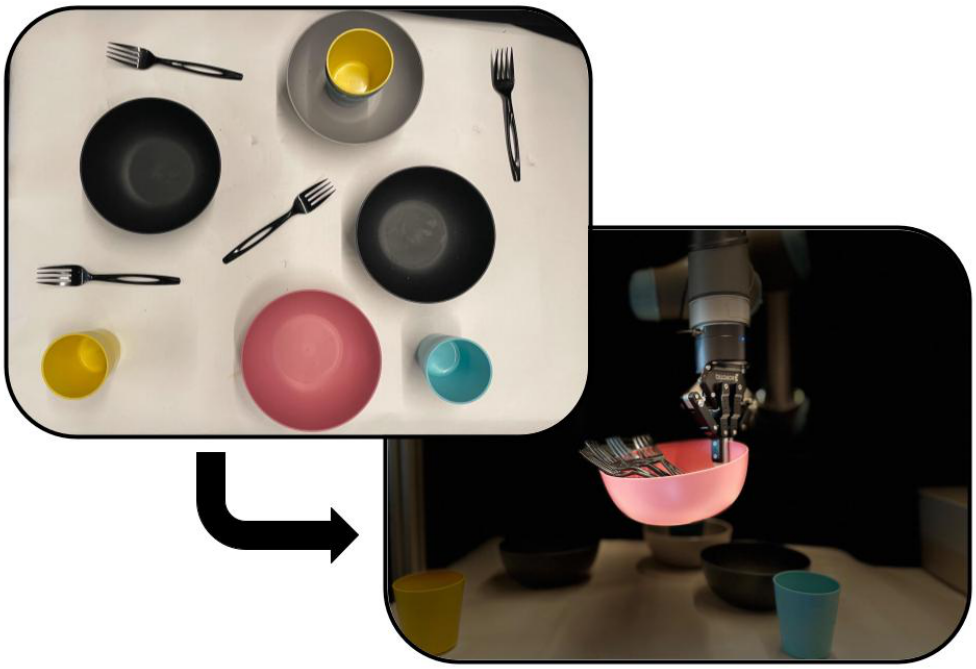


Figure 1.2: **The Busboy Problem.** We present a combination of the robust action primitives to declutter a workspace of cups, bowls, and utensils.

seamlessly with many SLAM algorithms. However, a given point in a pointcloud is assigned a single color and semantic feature, whereas a multi-scale understanding of the world may be required to simultaneously reason about objects and their parts, similar to how LERF-TOGO [10] leverages multi-scale semantics in LERF [16].

3D Gaussian Splatting (3DGS) [21] models the 3D scene as a large set of 3D Gaussians. This provides an appealing solution, as its explicit 3D representation is more conducive to robotics applications and may scale better to unbounded scenes. Recent works [22], [23] successfully assign semantic features to every Gaussian in the scene. However, all existing techniques combining semantic features and 3D Gaussian Splatting (3DGS) scene reconstruction require offline computation of keyframe transforms and 3D Gaussian initialization points.

In this paper, we focus on linking language understanding to Gaussian Splats in large-scale scenes, while incrementally training on a stream of RGBD images of the scene from a mobile robot. This incremental training method offers substantial benefits, notably enabling the robot to autonomously determine its position within the environment and subsequently use the map data for enhanced operational efficiency.

LEGS combines geometry and appearance information from 3DGS with semantic knowl-

edge from CLIP by grounding language embeddings into the 3DGS similar to the method described in [22]. LEGS incrementally registers images and simultaneously optimizes both 3D Gaussians and dense language fields. This allows robots to build maps that contain rich representations of their surroundings that can be queried with natural language.

We present 3 main contributions:

- An online multi-camera 3DGS reconstruction system for large-scale scenes. The system takes as input three image streams from a mobile robot, and incrementally builds the 3D scene.
- Language-Embedded Gaussian Splatting (LEGS), a hybrid 3D semantic representation that uses explicit 3D Gaussians for geometry and implicit scale-conditioned hashgrid [24] for the semantics.
- Results from physical experiments suggesting LEGS can produce high quality Gaussian Splats in room-scale scenes with training time 3.5x faster than the LERF baseline [16].

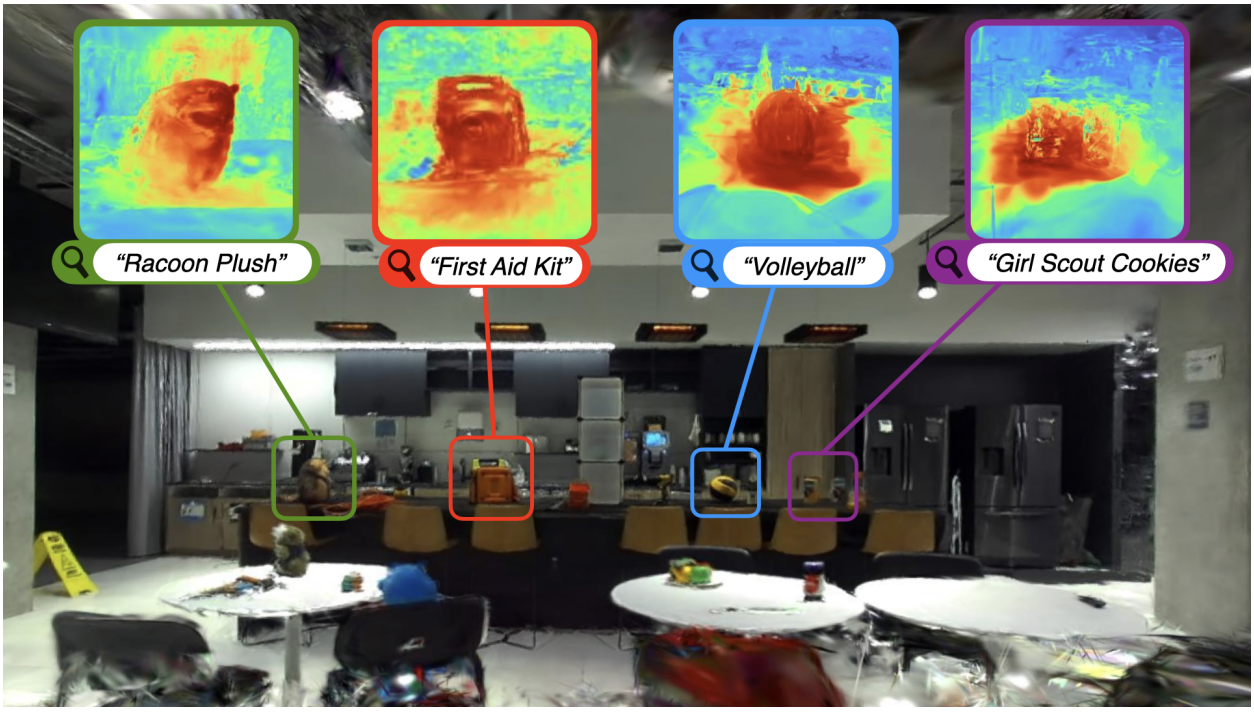


Figure 1.3: **Large-scale language-embedded Gaussian splatting setup.** The Gaussian splat 3D reconstruction was used to render a novel view of a large-scale environment. Given open-vocabulary queries, LEGS can localize the desired objects as seen with the heatmap activations.

Chapter 2

Surgical Suturing: Background

2.1 Related Work

Fully automated robot systems have been approved for hair restoration and external beam radiation [25], [26]. However, all surgical procedures are performed 100% under human surgeon teleoperation [27]. Some research efforts have focused on autonomously performing specific sub-tasks such as debridement [28], vascular shunt insertion [29], and brain tumor resection [30]. In this work, we focus on the suturing task.

Autonomous Suturing

Automating surgical suturing has been shown to be feasible in-vivo using IR markers, and industrial robot arms (like the KUKA LBR Med) both for the open-surgery setting [31] and the minimally invasive surgery setting [27] to perform in-vivo suturing on live pigs. More recent research efforts have been directed at automating surgical suturing using the da Vinci Research Kit (dVRK) [32]. While performing the whole suturing task autonomously remains an open problem [33], many researchers have explored the automation of sub-tasks such as suture placement [34], needle handover [35]–[37], needle extraction [38], needle pick-up [39], and knot tying [40]. Though each of these methods individually have shown high success rates (90%), performing these methods sequentially often has much lower success rate due to the product probability intersection of all of these events. In attempts to automate the complete suturing task, some hardware simplifications have been used in the past. Schwaner, Iturrate, Andersen, Jensen, and Savarimuthu [41] demonstrate impressive success rates with only a needle and no thread, avoiding the risk of the robot getting tangled in the thread. Sen, Garg, Gealy, McKinley, Jen, and Goldberg [33] use colored needles and a special mount for the gripper which help with needle detection and orientation. In this work we use unmodified surgical needles and suture thread.

Visual Servoing in Surgical Automation

Visual servoing has been explored for surgical automation, with needle [42], [43] and gripper [44] pose estimation using learned keypoint tracking models like DeepLabCut, a method that uses transfer learning to perform keypoint annotation for pose estimation by Mathis, Mamidanna, Cury, Abe, Murthy, Mathis, and Bethge [45]. Another line of research focuses on learning end-to-end visual servoing policies which implicitly model the object state [46], [47]. Wilcox, Kerr, Thananjeyan, Ichnowski, Hwang, Paradis, Fer, and Goldberg [35] and Paradis, Hwang, Thananjeyan, Ichnowski, Seita, Fer, Low, Gonzalez, and Goldberg [48] have proposed intermittent visual servoing in the context of peg transfer and needle handover, respectively, both of which use a visual servoing policy trained with imitation learning in lieu of classical trajectory optimization where high precision is required. In this work we propose a novel visual needle pose estimation approach using learned image segmentation models as well as known visual features and system dynamics. The estimated object poses are used to automate the surgical suturing sub-tasks of needle insertion, extraction, and handover using analytic control methods.

2.2 Problem Statement

Given a sequence of 3D suture entry and exit points overlaid on a wound phantom, perform as many sutures as possible along this sequence of points.

Assumptions

We assume known needle shape and diameter, predetermined 3D points for needle insertion and extraction, a calibrated stereo camera pair, and access to the transformation between the robot and camera coordinate frames. We also assume the wound is raised and orthogonal to the camera as shown in Fig. 1.1.

Objectives and Evaluation Metrics

We consider two evaluation metrics: number of successful consecutive sutures and completion time. We consider a suture throw to be successful if the robot is able to pass the needle through the wound, perform suture thread cinching (tightening), and return the needle to a neutral position outside the phantom with no tangles of the suture thread.

Chapter 3

STITCH

We present STITCH: an augmented dexterity pipeline that performs Suture Throws Including Thread Coordination and Handoffs. We introduce a novel visual 6D needle pose estimation framework using a stereo camera pair (Fig. 3.1) and new robust suturing motion primitives. STITCH iteratively performs needle insertion, thread sweeping, needle extraction, suture cinching, needle handover, and needle pose correction with failure recovery policies (Fig. 3.2).

3.1 6D Needle Pose Estimation

The Allied Vision Prosilica GC 1290 cameras capture a stereo pair of images with a resolution of 1280x960 at up to 33fps. With these images, we create a pointcloud of the scene using disparity images from RAFT-Stereo [49], a deep architecture for rectified two-view stereo that has been empirically shown to be more robust than standard stereopsis techniques and generalizes well to unseen real-world data.

To isolate the points relevant to the needle, we train a U-Net [50], an image-based segmentation model, to detect the needle in image space. The training data is labeled using the *Labels from UV* technique [51]. The segmentation mask produced by the U-Net is used to isolate the needle from the pointcloud generated by RAFT-Stereo.

Since we also need to accurately estimate the positions of the two endpoints on the needle and the orientation of the needle itself, we propose a novel 6D needle pose estimation module as seen in Figure 3.1. Based on prior work for needle handover (HOUSTON) [35], we propose an algorithm that determines the best-fitting 3D circle. A key distinction between this work and HOUSTON is that HOUSTON obtained a pointcloud from stereo point matching while this work uses RAFT-Stereo, a roboust stereopsis neural network. STITCH begins by using RANSAC to estimate a 3D plane equation that fits the needle pointcloud. Then, we calculate the normal vector from the estimated 3D plane to ascertain the orientation of the needle. All inlier needle points are projected onto that plane. To achieve higher accuracy, we then project all inlier points on the plane to the xy-plane where we use RANSAC to estimate

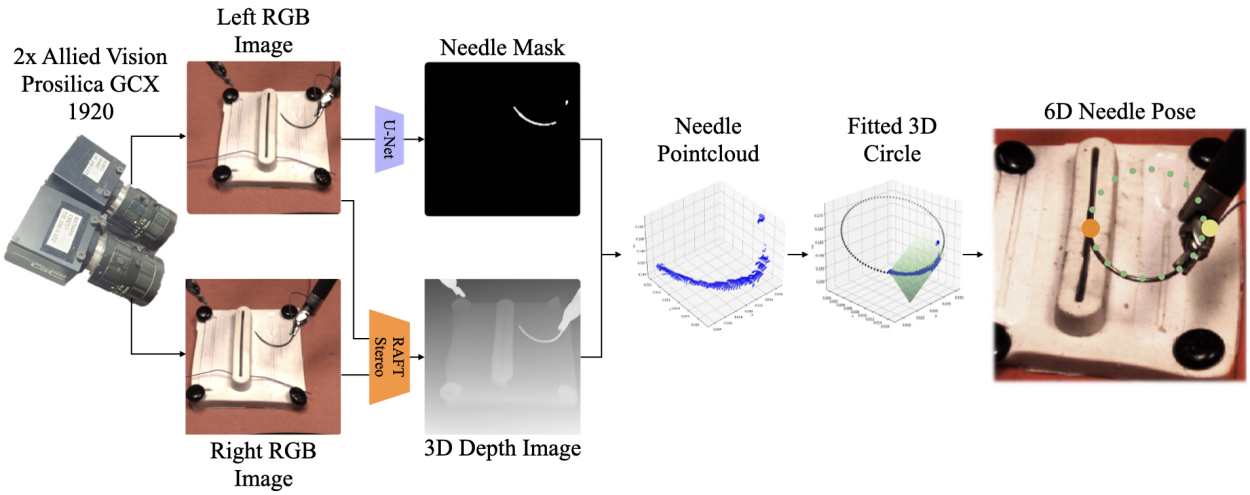


Figure 3.1: **6D Needle Pose Estimation.** The needle pose estimation pipeline takes in stereo left and right images, generates a disparity image using RAFT-Stereo, and generates a needle mask using a U-Net. The needle mask is applied to the disparity image to create a needle pointcloud, which is fitted to a 3D plane then a 2D circle using RANSAC, and finally the two farthest pointcloud points are selected as the needle endpoints.

a 2D circle equation. Next, we derive the positions of the two endpoints of the needle on the 2D circle by finding the 2 most distant needle inliers in the pointcloud. After that, we determine the 3D circle and endpoint positions by back-projecting onto the aforementioned 3D plane estimate.

3.2 Suture Motion Planning

The surgical suturing task is composed of several distinct motions. The motion controller directs the robot motions in every state, as well as the state transitions and when they ought to be performed. Each suture is composed of the sequence of needle insertion, a thread sweeping motion to clear excess thread from the suture site, needle extraction with suture cinching, needle handover, needle pose correction, and failure recovery, as shown in Fig. 3.2. The needle motion inside the tissue is designed to reduce tissue damage while remaining withing the kinematic bounds of the gripper.

Needle Insertion

The robot inserts the needle into the tissue phantom with a circular twisting motion at the specified insertion point such that the needle tip exits the tissue at the specified extraction point.

The needle is inserted in the tissue phantom in two steps to minimize strain on the tissue. Both steps are performed open loop as a large part of the needle is occluded by the tissue

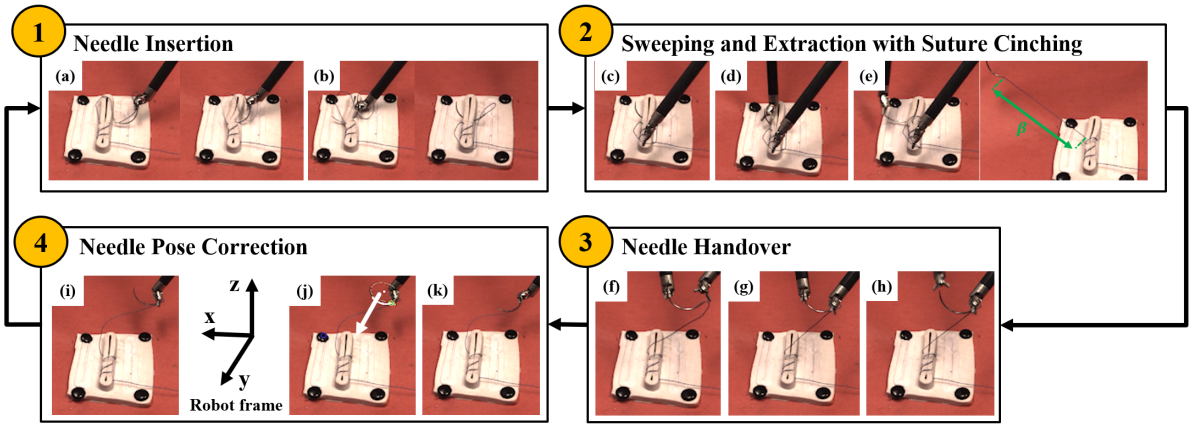


Figure 3.2: **STITCH Motion Controller** The motion controller combines the motion primitives into a state machine with 4 states: 1. Needle Insertion, 2. Sweeping and Extraction with Suture Cinching, 3. Needle Handover, and 4. Needle Pose Correction, which are continually looped until failure.

and the needle driver during insertion. First, the tip of the needle is pushed into the tissue phantom corresponding to the vector between the given insertion and extraction points (Fig. 3.2(a)). This allows for the straight tip of the needle to penetrate the tissue and exit at the needle extraction point. Second, the needle is rotated into the tissue by performing a 45° rotation around the estimated circle normal vector (Fig. 3.2(b)). This motion follows the curvature of the needle as it passes through the tissue, so that it passes along the hole made by the needle tip and does not stretch or tear the tissue.

Thread Sweeping

The thread sweeping motion is designed to prevent failure during the needle extraction step (Fig. 3.2(c)). Failures occurring during the needle extraction process can be broadly classified into two cases: (1) the thread passes in front of the needle, occluding it and leading to detection errors; (2) when both the needle and the thread are accidentally grasped together during the re-grasping process, despite an accurate needle pose estimate. Pushing the thread out of the re-grasping site through a sweeping motion before the extraction step can prevent the failures described above.

We use the thread modelling method from Schorp, Panitch, Shivakumar, Viswanath, Kerr, Avigal, Fer, Ott, and Goldberg [52] to track the thread for the sweeping motion. In this work, a U-Net is trained to output a segmentation mask of the thread. Running an analytic tracer on that segmentation mask for the left and right images, a 3D NURBS spline is fitted for the thread.

The sweeping motion involves opening the gripper wide and passing it over the wound towards the camera so that the thread is caught and pushed ahead of the needle. With the

thread in front of the needle, entanglement risks during extraction are significantly reduced.

Needle Extraction

The purpose of the needle extraction motion is to remove the needle from the tissue phantom and to pull the thread taut to close the wound. To successfully perform multiple sutures, it is necessary to pull the thread to an appropriate length prior to the next insertion motion. The needle extraction motion is performed in a closed-loop fashion using the needle pose estimator. The frequency of needle pose estimator for feedback was measured as 0.67 Hz. The re-grasp point and the axis around which to rotate and remove the needle are also determined through visual estimation.

The needle extraction motion is composed of two main components. At the start of the routine, the left needle driver is positioned at the upper left edge of the workspace, above the tissue phantom. Between the two needle endpoints, determined as described in 3.1, the point closer to the left needle driver is defined as the re-grasp point. The left needle driver is moved to a point offset by 1 cm horizontally from the defined re-grasp point (Fig. 3.2(d). Then, the left needle driver is moved in the direction of the re-grasp point by 1.5 cm, and the gripper jaws are closed.

Once the needle has been grasped by the left needle driver, it is rotated by 80° about the estimated needle axis to extract the needle and to minimize tissue damage (similar to the "rotate in") motion in the needle insertion routine. At the end of this action, only a small bit of the needle remains inside the tissue so that the needle can be fully extracted by a linear motion (Fig. 3.2(e)).

Suture Cinching

To ensure each suture is properly tensioned, suture cinching (tightening) is performed after needle extraction. The length of the thread that needs to be pulled for suture cinching in the extraction motion can be defined as $\beta = l_{des}$ - $(i$ - $1)\times l_{each}$ depending on the number of sutures (Fig. 3.2(e)), where l_{des} represents the desired thread length for the final suture, i represents the number of sutures, and l_{each} is the length of thread used for a single suture.

Needle Handover

The goal of the needle handover motion is to transfer the extracted needle from the left needle driver to the right needle driver in preparation for the next insertion. Once the pose of the needle is estimated, we define the re-grasp point in handover as the endpoint further from the left needle driver, similar to the process in 3.2. The right needle driver opens its jaws and moves to a point offset by 1 cm horizontally from the defined re-grasp point (Fig. 3.2(f)). After moving by the offset in the direction of the re-grasp point, the right needle driver closes to grasp the needle, (Fig. 3.2(g)) and the left needle driver opens to release it (Fig. 3.2(h)). The entire sequence is done in a closed-loop fashion with visual feedback

tracking the pose of the needle. If the right needle driver moves into position to grasp the needle and the detected orientation of the needle remains unchanged, we pull the driver back, add a small (< 0.5cm) random horizontal offset, and reattempt the grasp, up to 5 additional times before declaring the handover a failure.

Needle Pose Correction

Since the pose of needle right before insertion significantly affects the insertion, we use interactive perception to improve needle tracking so it can be actuated to the ideal pose for the subsequent insertion. We adjust the needle for the insertion using a needle pose correction algorithm, which consists of three steps. First, the needle is moved to an optimal detection location at the lower, back, right corner of the workspace for RAFT-Stereo reconstruction as shown in Fig. 3.2(i). The lower back right corner was chosen empirically for consistent needle pose estimates. An ablation study confirmed this choice, showing 90% success rate in this corner for left gripper-held needles and 70% for right gripper-held needles, compared to 50% success in random positions. Identifying both endpoints is crucial for insertion, so obtaining pose estimates in the chosen corner maximized successful suture throws. In the next step, we sample 10 measurements of the needle normal vector, and rotate the gripper such that the normal vector of the needle is identical to the positive y axis in robot frame (Fig. 3.2(j)). This step constrains the discrepancy in orientation to be about the positive y-axis, which makes it more repeatable. Because the needle configuration during handover is relatively similar for each throw, the final step is a 90 degree rotation about the y-axis as seen in Fig. 3.2k. At this point, the STITCH pipeline is ready for the next suture throw insertion.

Motion Failure Recovery

The STITCH algorithm includes recovery mechanisms for both extraction motion failures and handover motion failures. For the extraction motion, the algorithm compares the positions of the needle endpoint before and after extraction. If the difference is below 2 cm, the extraction motion is retried up to 5 times. For the handover motion, the algorithm compares the normal vector of the needle before and after moving the right needle driver when both needle drivers have grasped the needle. If there is a nonzero change of the normal vector, then the handover motion is retried up to 5 times.

Chapter 4

STITCH Results

4.1 Experimental Setup

The experimental setup consists of a bi-manual dVRK robot [32], a soft tissue phantom consisting of a single wound from a 3-Dmed directional suture pad featuring parallel linear wounds, and a fixed RGB stereo camera pair. For our experiments, the stereo cameras are a pair of Allied Vision Prosilica GC 1290 industrial cameras, which each output images of size 1280x960 at 33 frames per second. The Edmund Optics lenses mounted on the sensors allow for precise adjustments to the focus and aperture based on our workspace depth and illumination. The cameras are angled at the phantom such that the full workspace of the robot is captured in the field of view. We define the workspace using a Cartesian (x, y, z) coordinate system. We use violet Polysorb^{TM} surgical suture thread from Covidien. The threads are of variable length between 10 and 40 cm, with 2-0 USP Size (0.35-0.399 mm in diameter) and are attached to a GS-21 half-circular surgical needle with a radius of 12 mm.

Baselines

We evaluate the full STITCH algorithm and two baselines, multi-throw suturing without thread manipulation or needle pose correction ("Sensing Only") and multi-throw suturing thread management without needle pose correction ("Thread Management"). The sensing-only baseline performs the needle insertion, extraction, and handover steps, omitting the suture cinching, thread sweeping, and needle pose correction motions. The thread management baseline performs insertion, extraction (with thread sweeping), cinching, and handover while omitting the needle pose correction step. We also evaluate a human-supervised setting ("STITCH + Human"), in which the robot can request intervention up to 2 times from a supervising surgeon. At these points, the human supervisor performs a single recovery motion to return the workspace to a safe configuration, and then immediately returns control to the robot.

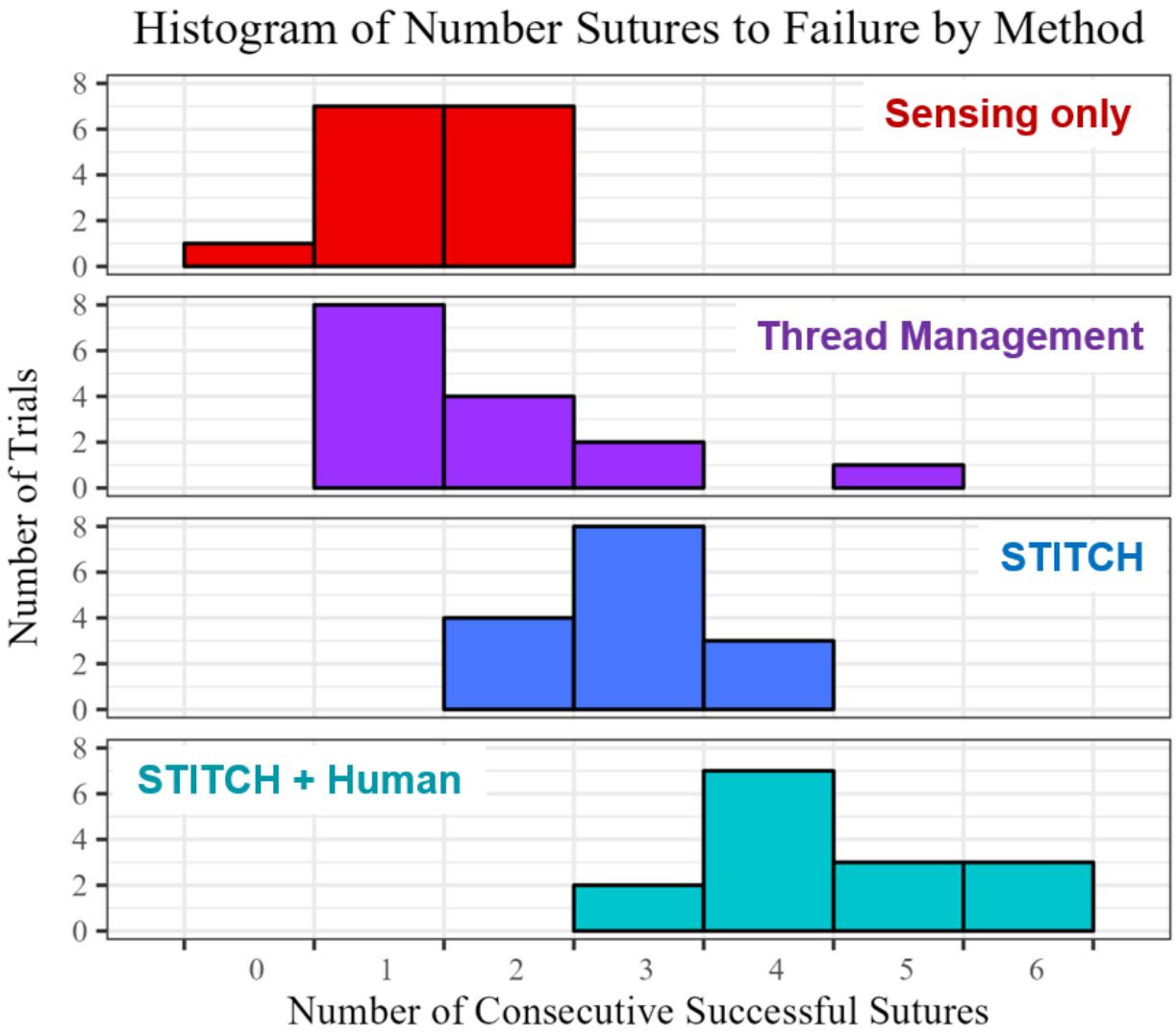


Figure 4.1: **Histogram of the Number of Sutures to Failure by Method.** The results in the histogram shown above is for 15 trials per each of the four methods.

Trial Specifications and Error Classification

Each trial produces n successive running suture throws. A suture throw is considered complete when the needle has successfully been inserted and extracted through the raised edges of the wound and the thread has been sufficiently tensioned to close the wound between the insertion and extraction points. A trial ends when the robot either enters an unrecoverable state, such as a dropped needle or tangled thread preventing further sutures, or successfully closes the wound by throwing 6 consecutive sutures. For each experimental baseline, we perform 15 trials of 1–6 throws each, and report the following metrics in Table 4.1:

- *Mean sutures to failure*: the average number of successfully completed suture throws before the first unrecoverable error is encountered.
- *Single-suture success rate*: the percentage of successful suture throws out of the total number of suture throw attempts.
- *Three throw success rate*: the percentage of trials which terminate after at least three successful suture throws.
- *Full wound success rate*: the percentage of trials which terminate in wound closure.
- *Mean time per suture*: the average time elapsed per suture throw.
- *Error Types*: the number of insertion (I), extraction (E), handover (H), and thread management (T) errors encountered.
- *Mean sutures to intervention*: The average number of autonomously completed suture throws between human intervention requests.

We report trial-ending errors according to the portion of the pipeline during which the failure occurred. We use the following schema:

- *Insertion error*s occur when the robot fails to insert the needle through the raised edge of the wound, or the needle enters a non-wound region of the phantom, or the needle exits the wound through the top or bottom of the phantom.
- *Extraction error*s occur when the needle remains in the wound after extraction.
- *Handover error*s occur when the robot drops the needle during handover or enters an unrecoverable configuration during the handover process.
- *Thread management error*s occur when the robot fails to properly cinch the suture thread to close the wound or becomes dangerously entangled in the thread.

Table 4.1: STITCH Success Metric Comparison across Ablations for 15 Trials.

	Mean Sutures to Failure	**Single-Suture Success Rate**	**Three Throw Success Rate**	**Full Wound Success Rate**	**Mean Time per Suture**	**Error types** I	E	H	T	**Mean Sutures to Intervention**
Sensing Only	1.40	51.6%	0.0%	0.0%	106.8 sec	9	6	0	0	–
Thread Handling	1.80	55.9%	20.0%	0.0%	117.9 sec	6	5	2	2	–
STITCH	2.93	69.4%	73.3%	0.0%	159.3 sec	8	5	0	2	–
STITCH + Human	4.47	83.3%	100.0%	20.0%	141.9 sec	16	10	0	2	2.25

4.2 Results

Table 4.1 and Fig. 4.1 report experimental results. Over 15 trials, STITCH achieves an average single-suture success rate of 69.39% and a mean sutures-to-failure of 2.93. When allowing the robot to request human intervention, the robot achieves a single-suture success rate of 83.33%, and a mean sutures-to-failure of 4.47. The perception algorithm proposed in this study was confirmed to accurately track the 6D pose of the surgical needle 42% of the time throughout the experimental trials.

4.3 Limitations & Future Work

There are 2 primary limitations of the STITCH pipeline:

1. The most common failure case is encountered when correcting the needle orientation so it is optimal for needle insertion for the subsequent insertion. This stems from high variance needle pointclouds from RAFT-Stereo. RAFT-Stereo struggles in high disparity areas, thus we move the needle to lower, back, right corner of the workspace as seen in Fig. 3.2(i). At this position, the perception algorithm can reliably detect the normal vector of the needle, allowing for the first rotation correction step to align the normal vector with the positive y axis of the robot frame as seen in Fig. 3.2(j). However, the needle endpoint detection at this position has a higher variance than desired. This means we cannot perform the final rotation step based on needle endpoint feedback. In the future, we will investigate alternate stereo methods better tuned for small, reflective objects. With a more reliable stereopsis method and improved needle endpoint tracking, we hope to mitigate the insertion failure case by using the needle endpoint estimates to servo the needle to the optimal insertion orientation.

2. Another common failure case involves thread tensioning issues. Even with the thread-sweeping move, two potential thread failures are still present:

    a) The sweeping move sometimes misses the thread and the extraction move will grab the thread with the endpoint causing system failure.

    b) The later sutures run out of thread and fail because not enough thread was pulled during the initial suture throw extraction.

In future work, we plan to introduce a suture cinching before handover where the left gripper holds the needle at the handover position such that a portion of the thread would be vertical. The other gripper would then move horizontally to push the vertical thread component to pull out additional thread and clear it from the grasping workspace. To perform suturing in in-vivo experiments, challenges like tissue deformation, visual changes due to blood, and viewing angle variations must be addressed.

Chapter 5

Efficient Tableware Decluttering: Background

5.1 Related work

Multi Object Grasping

Prior work on multi-object grasping includes different grasping techniques to facilitate multi-object grasps [53], detecting the number of objects in a grasp [54], decluttering surfaces [55], and multi-object grasping to place objects in virtual reality [56]. Yamada et al. considered the simplified multi-object grasping problem, where the objects are already in a configuration where they can be grasped at once [6]. Agboh et. al. [57] showed that friction can increase picks per hour for convex polygonal objects.

Some prior work has focused on the design of grippers for multi-object grasping. Jiang et. al. [58] proposed a vaccum gripper with multiple suction cups, while Nguyen et. al. [59] proposed a soft gripper based on elastic wires for multi-object grasping.

Object stacking [5], [60], [61] has the potential to improve the number of objects per trip. We take inspiration from these works to include a stacking primitive.

Pulling

Prior work by Berretty, Goldberg, Overmars, and Stappen [62] has examined the use of inside-out pulling to orient convex polygonal parts. We utilize a similar technique for circular cups and bowls. Furthermore, a planner for ensuring convergence to the final pose of pulling trajectories is proposed by Huang, Bhatia, Boots, and Mason [63], where they examine the motion of planar objects undergoing quasi-static movement.

Grasp Candidates

Satish, et al. [64] discuss using a synthetic data sampling distribution that combines grasps sampled from the policy action set with guiding samples from a robust grasping supervisor to construct grasp candidates. Additionally, Mahler, Pokorny, Niyaz, and Goldberg [65] discuss the use of energy-bounded caging to evaluate grasp candidates. They efficiently compute candidate rigid configurations of obstacles that form energy-bounded cages of an object, where the generated push-grasps are robust to perturbations. Mousavian, et al. [66] describe the process of using a variational autoencoder to generate grasps by mapping the partial point cloud of an observed object to a diverse set of grasps for the object. Because of the relative simplicity of our setup, we found that an analytical approach to constructing grasp candidates is sufficient. In the case of bowls and cups, we sample a random point uniformly on the rim and then orient the gripper perpendicular to the tangent of the circle at that point. In the case of utensils, we identify the axis of the utensil, and pick the highest depth point along that line, with the gripper perpendicular to the axis.

Object Manipulation in Cluttered Environments

Efficiently finding object manipulation plans in high-dimensional environments with a large number of objects is a challenging problem. Hasan et al. [67] addressed this problem by identifying high-level manipulation plans in humans, and transferring these skills to robot planners. Other work by Tirumala et al. [68] used tactile sensing to singulate layers of cloth from a stack. Different from these works, our goal in the cluttered environment is to bring objects together, or stack them, to enable multi-object grasps.

5.2 Problem Statement

The Busboy Problem involves the task of decluttering a workspace containing cups, bowls, and utensils, with the objective of minimizing both the time and number of trips required for completion.

Assumptions

In the initial configuration, a planar workspace is defined in a cartesian grid (x, y) and has n_c cups, n_b bowls, and n_u utensils scattered across its surface. All items are assumed to be face up, visible by camera, and within a workspace defined by the constraints of the robot arm. These items may be initially stacked on top of one another or resting individually on the surface, and we assume that the initial state meets the following criteria:

- All items are of known dimensions, and cups and bowls are circular when viewed from top-down. Cups have radius 4.5cm, bowls have radius 8.5cm, and utensils are at most 17cm $\times$ 1.8cm.

- Cups and bowls are upright, and utensils are laid flat on the surface.
- Any stacks that exist are stable, such that $r_0 \geq r_1 \geq ... \geq r_s$, where r_0 represents the radius of the vertically lowest item, and r_s the highest one.
- Initially, no two items are touching (items are singulated).

State

We use cups, bowls, and utensils (forks and spoons) as the tableware set - collectively called "tableware" - in this work. Each cup and bowl has a position $[x, y]$, and each utensil has a position $[x, y]$ and orientation θ.

Chapter 6

TIDY

We present TIDY, a framework of action primitives and policies to solve the Busboy Problem. We incorporate consolidation and multi-object grasping to improve efficiency, showing that these action primitives are robust under a variety of conditions and that the vision system allows the policies to be fault-tolerant.

6.1 Action primitives

We propose to use a combination of manipulation primitives to solve the Busboy Problem. We specifically propose to use single object grasps, multi-object grasps, pull-grasps, and stack-grasps to efficiently clear a work surface of items (Figure 6.3).

Grasp

We use both single and multi-object grasps in this work. Let $\mathbf{u}_G$ be the grasp to pickup objects — single or multiple. We represent this action as:

$$\mathbf{u}_G = [\mathbf{p}_G, \theta_G] \tag{6.1}$$

where $\mathbf{p}_G = [x_G, y_G, z_G]$ is the center point of the grasp, and θ_G is the grasp orientation.

Pull-Grasp

A pull-grasp action involves two steps: a pull of one object to another, then a multi-object grasp of both objects. We represent a pull action as:

$$\mathbf{u}_P = [\mathbf{p}_S, \theta_S, \mathbf{p}_E, \theta_E] \tag{6.2}$$

where $\mathbf{p}_S = [x_S, y_S, z_S]$ is the pull start point, θ_s is the gripper orientation at $\mathbf{p}_S$, $\mathbf{p}_E = [x_E, y_E, z_E]$ is the pull end point, and θ_E is the gripper orientation at $\mathbf{p}_E$. For circular objects such as bowls and cups, the gripper pulls outwards from the center of the dish using

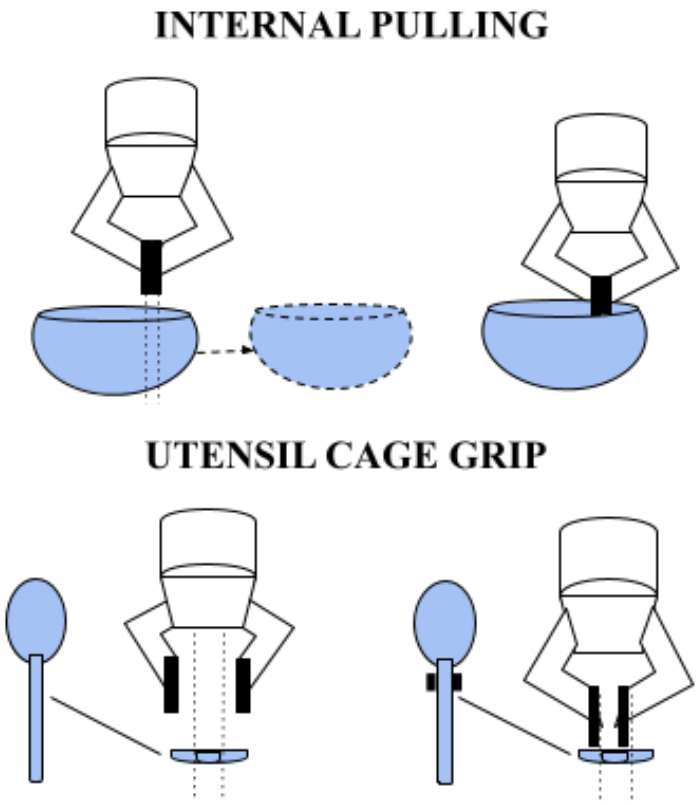


Figure 6.1: Bowls and cups are pulled with an internal pull, and utensils are grasped with a cage grip.

an internal pull, and for utensils, the gripper cages the utensil around its center point while moving it, as shown in Figure 6.1. Then, we denote a pull-grasp action as:

$$\mathbf{u_{PG}} = [\mathbf{u}_P, \mathbf{u}_G] \tag{6.3}$$

Stack-Grasp

A stack-grasp action involves two steps: a stack of one object onto another, then a multi-object grasp of both objects. We represent a stack action as:

$$\mathbf{u_S} = [\mathbf{u}_{G_i}, \mathbf{p}_L, \theta_L] \tag{6.4}$$

where $\mathbf{u}_{G_i}$ is a grasp on the lifted object, and $\mathbf{p}_L = [x_L, y_L, z_L]$ is the placement point on the stationary object, and θ_L is the gripper orientation at $\mathbf{p}_L$. Then, we denote a stack-grasp

action as:

$$\mathbf{u_{SG}} = [\mathbf{u}_S, \mathbf{u}_G] \tag{6.5}$$

6.2 Determining allowable actions

Grasp

A single-object grasp is always allowable. We can safely assume this since any dish or stack of items is already top-down graspable. When no other actions are allowed, the single-object grasp action is used as a default to clear the workspace.

A multi-object grasp is allowable when the grasp heights of both items are similar (within an adjustable threshold value) and if the lateral distance between the grasp points of both items is less than the width of the gripper. If the grasp heights of the items are significantly different, the gripper will have to either collide with the taller dish while attempting to grasp the shorter dish or grasp only the taller dish to avoid the collision, and either case results in a failure of grasping multiple items at once. Similarly, if the items are separated by more than the maximum inside width of the grippers, an attempt to grasp both at the same time will fail.

Pull

A pull of two items is allowable if a multi-object grasp can be executed on those items and if no other objects lie between the two items on the workspace. We disallow pull actions of items for which a multi-object grasp cannot be executed, since the pull becomes a wasted action. We also disallow pull actions of items with other objects between them to ensure that the intermediate objects are not displaced in a non-deterministic manner.

Stack

A stack of dish d_a with radius r_a onto dish d_b with radius r_b is allowable if $r_a \leq r_b$. This means that a cup can be stacked onto a bowl, but not vice versa, and that a utensil can be stacked onto any other dish, including another utensil. This is to ensure that the stack stability assumption present at the initial state remains valid after each action.

6.3 Robustness of action primitives

Johnson, et al. [69] propose a series of divergence metrics that quantifies how robust an action is to uncertainty due to an undersensed system based on contraction analysis. To quantify the robustness of the pull and pull-grasp actions, we use the expected divergence

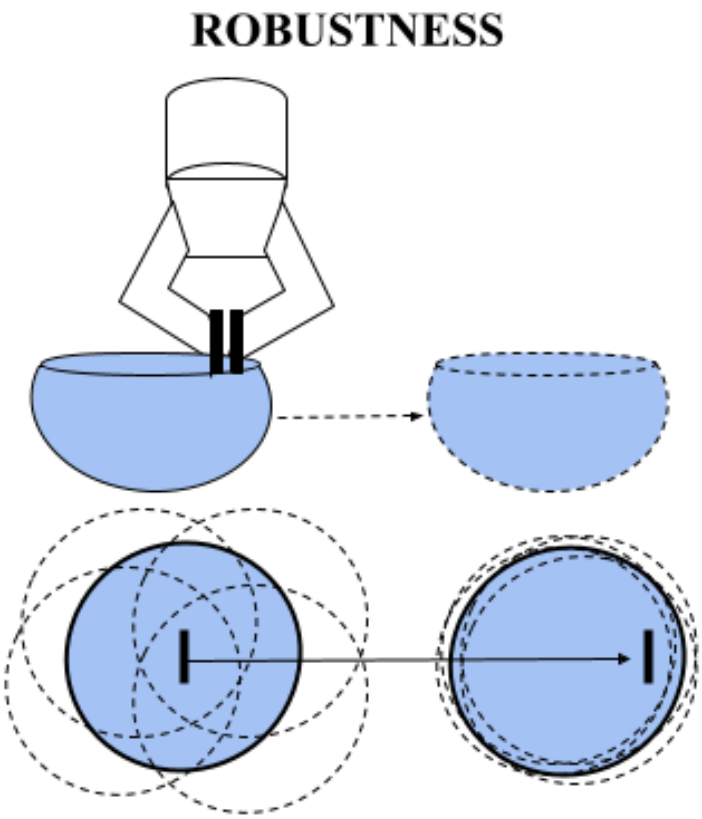


Figure 6.2: Uncertainty in a cup or bowl's location decreases once the arm has interacted with the dish. The position of the dish relative to the gripper's location is known with greater certainty after the action.

metric which is approximated by

$$\frac{\sum_{i=1}^{n}(||c_i^f - c_0^f||)}{\sum_{i=1}^{n}(||c_i^s - c_0^s||)}$$

where c^s and c^f are the starting and final center points, and i corresponds to the trail, where $i = 0$ corresponds to the desired pull action. An expected divergence of less than 1 means that the action reduces uncertainty in expectation, with a lower expected divergence corresponding to a greater reduction in uncertainty. We present three primitives to robustly execute the above actions.

Grasp

When executing a grasp at location x, y, z, the robot will open its grippers centered around x, y, and then move down to the appropriate height, as measured by the depth sensor,

before closing the gripper to grasp the object. The affordances granted by max gripper opening, gripper height, and gripper width ensure that an off-center grasp point x, y, z will still successfully complete the single-object or multi-object grasp of the object (Figure 6.2).

Pull

For cups and bowls, the gripper pulls outwards from the center of the dish, contacting the inner surface of the dish (Figure 6.1). This action is successful as both r_b and r_c are larger than the width of the gripper when closed. If the gripper is anywhere within the opening of the object, it will be able to move the target object to a specified location. For utensils, the gripper cages the utensil around its center point while moving it, preventing unwanted rotation and moving the utensil to its specified location.

Stack

For bowls and cups, the top lip radius is larger than the radius of the base, giving the sides a taper. Because a dish d_a is only stacked onto another dish d_b of equal or larger size, the base radius of d_a is guaranteed to be smaller than the top radius of d_b, allowing the tapered sides of the items to funnel d_a into place even if there is slight error in the placement of the dish. Placing a utensil onto a bowl is extremely robust to error because of the relative radii of the items, and placing a utensil onto another utensil is robust due to the curvature of the utensils themselves which slide a misplaced utensil into place, making them naturally conducive to stacking.

6.4 Policies

TIDY composes these action primitives into two main policies to declutter the workspace: a Pull policy and a Stack policy.

Pull Policy

The pull policy combines Pull-Grasp and Grasp actions. From the initial scene, it checks if any multi-object grasps can be executed right away, and executes those first. Then, it runs the Pull-Grasp action for all remaining items, pulling together items that don't cause collisions and executing multi-object grasps to clear them from the workspace. If any items remain after all possible multi-object grasps are executed, those items are cleared with single-object Grasp actions. After each action, a new image of the workspace is taken and the state representation is updated to reflect the new state of the workspace, including any tableware that has been moved or left behind by the previous action. This policy is formalized in Algorithm 1.

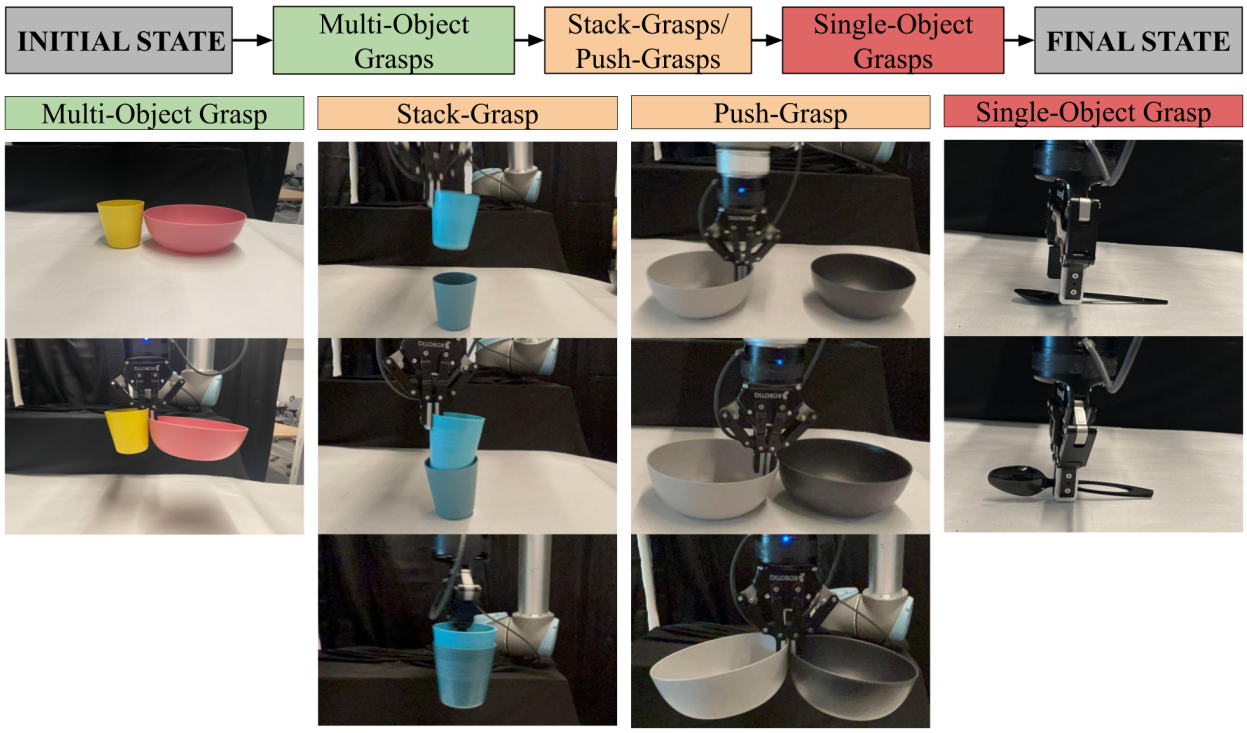


Figure 6.3: We employ the following robust action primitives to declutter a workspace through consolidation and multli-object grasps: single-object grasps, multi-object grasps, push-grasps, and stack-grasps.

Stack Policy

The stack policy combines Stack-Grasp and Grasp actions. It repeatedly executes the Stack-Grasp action to clear the workspace, and if there are any remaining items they are cleared with single-object Grasp actions. It prioritizes stacking utensils onto bowls and transporting them to the bin, and then tries to stack the remaining dishes. Stacking utensils first is an efficient way to improve the number of OpT for this policy. The policy is formalized in Algorithm 2.

After utensils are cleared, the stacks created by this policy are limited to be a combination of at most 2 existing stacks (i.e. once a Stack action is executed, the next action is necessarily a Grasp on the resulting stack, not another Stack action onto that stack). This is because we empirically found that when 4 or more bowls or cups are stacked, the height difference between the lip of the top dish and the lip of the bottom dish exceeds the height of the gripper jaws, causing many attempted grasps to fail. By limiting stacks to at most 2 existing stacks, we significantly reduce the chances of creating a stack with more than 3 dishes.

Algorithm 1 Pull Policy
Input: $D \leftarrow [d_1, d_2, ...d_n]$ - initial distribution of n items to be cleared.
Output: Executes an action sequence at each iteration.

while $D \neq \emptyset$ **do**
  if *MultiObjGraspPossible*(d_i, d_j) $\forall i, j \in$ *len(D)* **then**
    Grasp(d_i, d_j);
    D $\leftarrow$ UpdateStateFromImage();
    continue;
  if *PullGraspPossible*(d_i, d_j) $\forall i, j \in$ *len(D)* **then**
    Pull(d_i, d_j);
    Grasp(d_i, d_j);
    D $\leftarrow$ UpdateStateFromImage();
    continue;
  Grasp(d_i) $\forall i \in$ len(D);
  D $\leftarrow$ UpdateStateFromImage();

Algorithm 2 Stack Policy
Input: $D \leftarrow [d_1, d_2, ...d_n]$ - initial distribution of n items to be cleared.
Output: Executes an action sequence at each iteration.

while $D \neq \emptyset$ **do**
  if *UtensilsAndBowlsRemaining()* **then**
    Stack(u, b) $\forall u \in$ utensils;
    Grasp(b);
    D $\leftarrow$ UpdateStateFromImage();
    continue;
  if *StackGraspPossible*(d_i, d_j) $\forall i, j \in$ *len(D)* **then**
    Stack(d_i, d_j);
    Grasp(d_j);
    D $\leftarrow$ UpdateStateFromImage();
    continue;
  Grasp(d_i) $\forall i \in$ len(D);
  D $\leftarrow$ UpdateStateFromImage();

Chapter 7

TIDY Results

Through physical experiments, we demonstrate the robustness of the pulling action primitive and then evaluate TIDY on a real-world table clearing task.

7.1 Experimental Setup

We use a UR5 robot arm with a Robotiq 2F-85 gripper and Intel RealSense 455D RGBD camera mounted 83cm above the workspace. The workspace is a flat 78cm x 61cm surface with 4 cups, 4 bowls, and 4 utensils, $n_b = n_c = n_u = 4$. In our experimental setup, we calculated a max gripper opening of $w = 8.5cm$, gripper height of $h = 4.5cm$, bowl radius $r_b = 8.5cm$, cup radius $r_c = 4.5cm$ and utensil width $r_u = 1.8cm$.

TIDY identifies and locates tableware on the workspace with a vision pipeline. Since the surface of the workspace is white, we use darker colored tableware to be easily visible. To locate cups and bowls, we first use edge detection, contour forming, and HoughCircles to identify circular shapes on the workspace, then filter these circles based on the known image radius of cups and bowls. We cluster these circles by their centers and remove circles that overlap beyond a specified threshold, allowing an unambiguous detection of cups and bowls. To locate utensils, we use edge detection and contour forming, and then filter out the contours that are too "square", as determined by the aspect ratio of the identified contour. We draw an imaginary line through the lengthwise center of bounding rectangle of the contour, and sample depth values along that line; we use the highest depth point as the grasp point of the utensil to allow the gripper maximum clearance with the surface.

We define three tiers to evaluate the performance of our algorithm on scenes of increasing complexity.

- Tier 0: scenes contain 6 items, either all cups, all bowls, or all utensils, with no stacks in the initial state.
- Tier 1: scenes contain 4 items each of cups, bowls, and utensils, and have no stacks in the initial state.

- Tier 2: scenes contain 4 items each of cups, bowls, and utensils, but we allow stacks of at most 3 objects in the initial state.

For Tier 2, we limit initial stacks to at most 3 objects because of the dimensions of the gripper, as mentioned in Section 6.4. The number of objects in a stack, and not the actual dimensions of individual dishes, is the main limiting factor for the grasp, because we grasp dishes from the rim. The dishes could actually be much larger and still be graspable as long as the walls are thin enough to allow the gripper to slide over them, and the weight of the dish does not exceed the payload limitations of the gripper itself. We limit ourselves to a small set of known kitchenware objects for consistency in our experiments.

We evaluate the performance of TIDY's two policies against a baseline single-item policy, referred to as "Random" in Table 7.1. This policy picks a dish at random, and if the dish is a cup or bowl, it uniformly samples a point on the rim and grasps the dish at that point. If the dish is a utensil, it identifies the grasp point of the utensil as described above and grasps the utensil at that point. This policy is stack-agnostic, so even in Tier 2 when there are stacks present in the initial state, it treats each item in the stack as its own object, and clears the stack by transporting one item at a time.

Scene Generation

In order to evaluate our policies, we generate multiple scenes at each tier, and every policy is run once on each scene. To generate each scene, we use the dimensions of the workspace (78cm $\times$ 61cm), and r_b, r_c, r_u for the dimensions of the objects. We randomly sample x, y locations within the scene for each object. If an object intersects with another object, we create a stack of the two objects if the maximum number of intersections has not been exceeded, and resample a position for the object if it has. Tiers 0 and 1 allow no such intersections, whereas Tier 2 allows 4 intersections. For each trial we manually reset the scene to maintain consistency.

Evaluation

We evaluate TIDY on 9 scenes at Tier 0 (3 scenes per type of dish), 3 scenes at Tier 1, and 3 scenes at Tier 2. A trial is one execution of one policy on one scene, so we have a total of $(9 + 3 + 3) * 3 = 45$ trials. For each trial, we record the time in seconds to clear the table, the number of objects grasped per trip (OpT), and the number of failures. A failure occurs when the robot is unable to move all items to the collection bin, either because of a perception failure that leaves items behind on the workspace or a policy failure that drops a dish off the workspace.

To evaluate the performance of our policies in more realistic scenario, we present the theoretical improvement in execution time when the bin is placed further away from the

Tier	**Policy**	Time (sec)	Objects per Trip	Failures	Time Ratio	**OpT Ratio**
Tier 0 Cups	**Random**	78.2	0.8	0	-	-
	Stack	58.5	2.0	0	1.4	**2.6**
	Pull	48.8	1.6	2	1.6	**2.1**
Tier 0 Bowls	**Random**	63.3	1.0	0	-	-
	Stack	60.2	2.0	0	1.1	**2.0**
	Pull	41.3	1.8	0	1.5	**1.8**
Tier 0 Utensils	**Random**	64.1	1.0	0	-	-
	Stack	64.3	1.8	0	1.0	**1.8**
	Pull	55.3	1.8	1	1.2	**1.8**
Tier 1	**Random**	121.3	1.0	1	-	-
	Stack	111.3	2.0	2	1.1	**2.0**
	Pull	102.1	1.6	3	1.2	**1.6**
Tier 2	**Random**	93.5	1.4	0	-	-
	Stack	88.2	2.6	2	1.1	**1.8**
	Pull	84.2	2.3	3	1.1	**1.6**

Table 7.1: **Physical Experiments** We present the total time and number of trips to clear a table for each policy. We found that compared to the baseline policy, the **stack** policy makes at least a **1.8x** improvement in the number of objects grasped per trip (OpT) and the **pull** policy makes at least a **1.6x** improvement.

workspace, as might be seen in a home or professional kitchen. Given the physical limitation of the UR5 arm length, we simulated the lengthening distance by adding time delays of 3 and 5 seconds in both directions of motion (to and from the collection bin).

To show the robustness of the pulling action primitive, we evaluate the reduction in uncertainty of the position of the dish after pulling. We move the gripper to the same initial position and place the dish at various points around the gripper 10 times, with the constraint that the gripper remains inside the dish when placed. We record the average distance of the initial states from the nominal initial position, then execute predetermined pull actions of 20cm, 30cm, and 40cm, and compute the average distance of the final states from the nominal final position.

7.2 Results

Physical experiment results reported in Table 7.1 show that by using consolidation and multi-object grasps, TIDY clears the workspace efficiently, with the pull policy transporting

Tier	Policy	No delay Time Ratio	3 sec delay Time (sec)	3 sec delay Time Ratio	5 sec delay Time (sec)	5 sec delay Time Ratio
Tier 0 Cups	**Random**	-	124.2	-	154.8	-
	Stack	**1.4**	76.5	**1.6**	88.5	**1.7**
	Pull	**1.6**	70.8	**1.8**	85.4	**1.8**
Tier 0 Bowls	**Random**	-	99.3	-	123.3	-
	Stack	**1.1**	78.2	**1.3**	90.2	**1.4**
	Pull	**1.5**	61.3	**1.6**	74.6	**1.7**
Tier 0 Utensils	**Random**	-	100.1	-	124.1	-
	Stack	**1.0**	84.3	**1.2**	97.6	**1.3**
	Pull	**1.2**	75.3	**1.3**	88.7	**1.4**
Tier 1	**Random**	-	193.3	-	241.3	-
	Stack	**1.1**	147.3	**1.3**	171.3	**1.4**
	Pull	**1.2**	120.3	**1.3**	178.8	**1.3**
Tier 2	**Random**	-	143.5	-	176.9	-
	Stack	**1.1**	116.2	**1.2**	134.9	**1.3**
	Pull	**1.1**	116.2	**1.2**	137.5	**1.3**

Table 7.2: **Theoretical Delays** We present the theoretical improvement in the time ratio as delays are added to simulate the bin being further away from the workspace. We find that the time ratio improves as more delay is added, strengthening the performance of the stack and pull policies compared to the baseline policy.

at least 1.6x as many objects per trip, and the stack policy at least 1.8x.

For the experiments with theoretical time delays, we find that moving the bin further away causes the stack and pull policies to perform significantly better than the baseline policy because motions to and from the bin are penalized, making policies with fewer total actions perform better. We report these results in Table 7.2. The improvement in time to completion is less pronounced because of two main factors. Firstly, the collection bin is placed right next to the workspace, making it so that trips to the collection bin are not as expensive in time, allowing the random policy to be competitive with our policies. In a more realistic scenario, however, the collection bin may be in an entirely different room from where the tableware is, making trips more expensive and favoring policies that make fewer trips to the bin. Secondly, the motion libraries of the UR5 require us to add pauses between moving the arm and opening or closing the gripper, so as to avoid inadvertently pushing objects aside while actuating the gripper. This means our actions are punctuated by pauses while waiting for gripper actuation. This increases the execution time of especially our stack policy, which involves many gripper motions.

Trajectory Length (cm)	Avg. Initial Distance (cm)	Avg. Final Distance (cm)	**Expected Divergence**
Cups			
20	1.46	0.28	**0.19**
30	1.46	0.19	**0.13**
40	2.53	0.18	**0.07**
Bowls			
20	4.31	0.92	**0.21**
30	3.80	0.89	**0.24**
40	4.64	0.76	**0.16**

Table 7.3: **Robustness Physical Experiments** We present experiments to evaluate uncertainty before and after taking an action. Avg. Initial Distance is the average distance between the trajectory's initial location and nominal start location. Avg. Final Distance is the the average distance between the trajectory's end location and nominal goal location. The expected divergence being less than 1 for all instances shows that the push action reduces the uncertainty in the final location of the object.

For the experiments to show the robustness of the pulling action primitive, we report results in Table 7.3, and see that the internal pulling action has an expected divergence of at most 0.28 for cups and at most 0.24 for bowls, which are both less than 1. Additionally we find that divergence decreases the longer the trajectory is. This is likely because longer trajectories give the item more time to correct itself.

7.3 Limitations and Future Work

In this work, we had only a single overhead RGBD camera, which limits the accuracy of our state estimation and can lead to failures due to occlusion and shadows. We also assume circular cups and bowls, which allows for a compact state representation and facilitates computing grasps. In future work, we will consider the more advanced grasp generation methods for general dishes and loosen the assumption of starting with singulated objects. We also hope to combine the pull and stack policies into a higher-level policy that can efficiently clear the workspace.

Chapter 8

Room-Scale LEGS: Background

8.1 Related work

Mobile Robot Mapping

Early robotic scene mapping research focused on the development of the core competencies in the metric [70]–[72] and topological [73], [74] knowledge spaces, extensively centered around the question of map and knowledge representation. For successful task execution, data-rich 3D scene representation and self-localization are critical, enabled by Simultaneous Localization and Mapping (SLAM) algorithms [75]–[77]. 3D spatial maps have traditionally been represented by voxel grids, points or surfels, and more recently, neural radiance fields [78], [79]. Each of these approaches comes with its own limitations. Occupancy and voxel grids are limited by their resolution. Points and surfels are discontinuous and difficult to optimize in a continuous manner. Neural radiance fields require computationally intensive ray-casting for rendering and offer only an implicit representation during training.

Natural Language for Robotics

Semantics in 3D has been a longstanding problem [80]. For mapping, semantic mapping has emerged as an effective scene understanding method that integrates semantic knowledge of the objects and surrounding environment into the mapped scene. The first known definition of semantic mapping for robotics is provided by Nüchter, *et al.* as a spatial map, 2D or 3D, augmented by information about entities, i.e., objects, functionalities, or events located in space [81]. An early work proposes concurrent object identification and localization using a supervised hierarchical neural network classifier on image color histogram feature vectors [82]. However, because these approaches rely on supervised datasets [83], [84], they work only on a closed set of vocabularies and do not generalize to new semantic queries.

More recent works have focused on using large vision-language models to support open-vocabulary queries. This includes both directions in 2D image [85]–[87] and 3D, such as VL-Maps [13] and CLIP-fields [12], which assigns a CLIP feature to every point in the

3D scene. This can be used for setting navigation goals with natural language queries. OpenScene [20] ensembles open-vocabulary 2D models and 3D point networks to form a per-point feature-vector allowing natural language querying. ConceptFusion [88] develops 3D open-set multimodal mapping by projecting CLIP [89] and pixel aligned features into 3D points, and additionally fuse other modalities such as audio into the scene representation. ConceptGraphs [90] model spatial relationships as well as the semantic objects in the scene to reason over spatial and semantic concepts.

Semantic fields have been applied not only to scene-level understanding for mobile robots but also to manipulation. In these settings, NeRFs [7] have been a popular 3D representation, following from Distilled Feature Fields [14], Neural Feature Fusion Fields [15], and Language Embedded Radiance Fields (LERFs) [16]. These works learn an semantic field in addition to the color field. LERF takes advantage of the implicit representation to support a scale-conditioned feature field, which takes in an extra scalar as input to facilitate multi-scale queries. For manipulation, the feature fields have been shown to facilitate learning from few-shot demonstrations [9], policy learning [8], zero-shot trajectory generation [91], and task-oriented grasping [10].

LERF-TOGO's [10] zero-shot task-oriented grasping performance is fully based on LERF, as LERF's multi-scale semantics allow for both object- and part-level understanding. This property is also valuable in scene-level settings, where a human may specify a collection of objects, e.g., utensils. However, the scale conditioning is simultaneously a blessing and a curse; natural language queries with LERF are bottlenecked by scale, as NeRF rendering is time intensive and query time increases linearly with the number of scale. LEGS maintains this multi-scale understanding, while speeding up the query time, by leverating Gaussian Splats [21] which have a significantly faster render time.

3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [21] is a recent work from 2023 that models a scene as an explicit collection of 3D Gaussians. Each Gaussian is described by its position vector μ, covariance matrix Σ, and an opacity parameter α, creating a representation that is both succinct and adaptable for static environments. The choice of 3D Gaussians over traditional point clouds is strategic; their inherent differentiability and the ease with which they can be rasterized into 2D splats enable accelerated α-blending during rendering. By avoiding volumetric ray casting employed by Neural Radiance Fields (NeRFs), Gaussian Splatting has a substantial speed advantage and can support real-time rendering capabilities.

Soon after its release, 3DGS has been applied to mapping [92], semantic mapping [93], navigation [94], and semantic fields [22], [23]. 3DGS's fast rendering time speeds up optimization, making it suitable for integrating visual SLAM and natural language queries for 3D semantic fields. They have also been demonstrated in both indoor datasets [95]–[97], and outdoor driving scenes with multiple cameras [98].

Learning semantic features for Gaussians have taken either one of two approaches: maintaining multi-dimensional features for each Gaussian, or calculating it on-the-fly by querying

a network. LangSplat [23] learns language features on a scene-specific latent space and uses SAM to learn hierarchical semantics for efficient and pixel-accurate querying. LangSplat shows drastic speedups over LERF, however it cannot be trained incrementally as it requires training a VAE over images of a scene before 3D optimization. FMGS [22] uses multi-resolution hash encodings [24] optimized with a render-time loss to combine CLIP features with a map of 3D Gaussians. LEGS similarly utilizes a hash encoding for its feature field, however it includes scale-conditioning as opposed to averaging CLIP across scales, retaining finer-grained language understanding.

8.2 Problem Statement

We consider a large indoor environment, specifically defined as a room encompassing approximately 1000 sq ft. The objective is to 3D reconstruct the a-priori unknown and unstructured environment and localize objects prompted by natural-language queries.

We make the following assumptions:

1. The environment and all objects within it are static.
2. Queried objects are seen at least once from the camera; it may not be currently visible to the robot system.

For each trial, the system is prompted by a natural-language query, and outputs the 3D coordinate of the most semantically relevant point in the scene. The trial is deemed successful if this point falls within the manually annotated bounding box for that query. The objective is to efficiently build a 3D representation that maximizes this success rate for large-scale scenes.

Chapter 9

Room-Scale LEGS

We use a Fetch mobile robot equipped with an RGB-D Realsense D455 camera and two side facing ZED 2 stereo cameras mounted with known relative poses. We build a map of the scene with a set of 3D Gaussians in an online fashion, registering new images as the robot drives around the environment. The overall pipeline for LEGS is outlined in Figure 9.1. There are three key components of the system:

1. **Multi-Camera Reconstruction**: To improve the quality and range of the optimized LEGS, we use multiple cameras pointing in different directions to provide more viewpoints of the environment.

2. **Incremental 3DGS Construction**: A significant challenge of large scene mapping is localization error because of its accumlative nature. To mitigate this error, we perform global Bundle Adjustment (BA) with DROID-SLAM to improve pose accuracy for all previously recorded poses in the scene. Global BA can be executed multiple times in a given traversal, and after each BA, the prior image-pose estimates are updated with the corresponding pose.

3. **Language-Embedded Gaussian Splatting**: We propose a variation of LERF [16] which uses the centers of gaussians to sample the underlying multi-scale language field, and rasterizes their features instead of volumetrically rendering.

9.1 Multi-Camera Reconstruction

Online image registration enables Gaussian Splatting on a mobile base with a comprehensive sensor suite including multiple camera views (left, right, center). During testing of vanilla Gaussian Splatting on our multi-camera setup, we found that offline Structure from Motion (SfM) pipelines frequently failed to find correspondences between images from different cameras, largely due to the lack of scene overlap from offset camera views. Because we peform online image registration with a visual SLAM algorithm for one camera and we know the

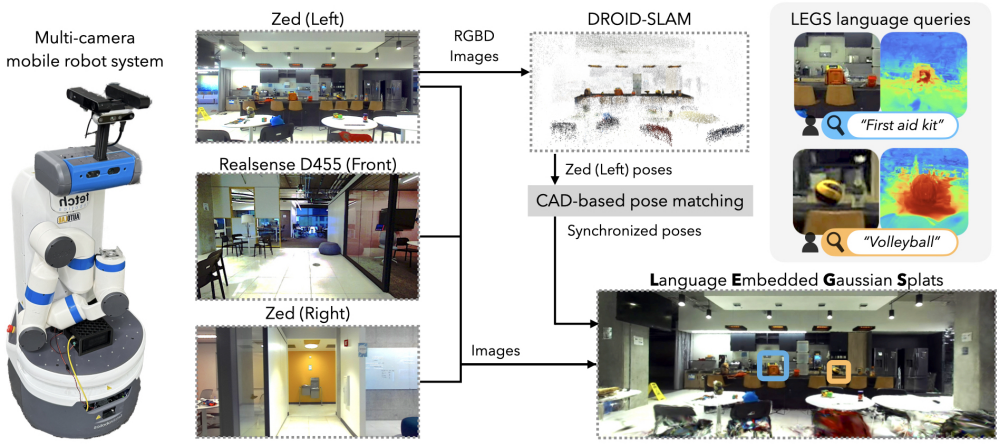


Figure 9.1: **LEGS System Integration** For LEGS, we use a Fetch robot with a custom multicamera configuration where a Realsense D455 is facing forward while 2 Zed cameras face the left and right sides respectively. The left Zed image stream is inputted into DROID-SLAM to compute pose estimates for the left camera, and the corresponding extrinsics are used to compute the pose estimates for the other Zed camera and D455. These image-poses are then used for concurrent Gaussian splat and CLIP training online. From there, the Gaussian splat can be queried for an object (ex. "First Aid Kit"), and the corresponding relevancy field will be computed to localize the desired object.

corresponding extrinsic transforms to the other cameras, we can use a ROS2 time synchronizer to align the different camera streams together and compute the corresponding pose estimate for each camera.

9.2 Incremental 3DGS Construction

Standard methods for radiance field optimization require image-pose pairs as input, and Gaussian Splatting greatly benefits from having a point cloud as a geometric prior for initialization. Poses and point clouds are typically provided through *offline* SfM techniques like COLMAP [99] which require all training images to be collected ahead of time. We build off of Nerfstudio's [100] Splatfacto implementation of Gaussian Splatting and modify it to operate on a stream of images and poses, and incorporate updates from global BA to refine poses.

Online Optimization

For online pose estimation we use DROID-SLAM [101], a monocular SLAM method that takes in monocular, stereo, or RGB-D ordered images and outputs per-keyframe pose esti-

mates and disparity maps. During operation, we feed DROID-SLAM input frames from one of the side-facing Zed cameras, and extrapolate the poses of other cameras via the camera mount CAD model. Registered RGBD keyframes from DROID-SLAM are incrementally added to Splatfacto's training set. We initialize new Gaussian means per-image by sampling 500 pixels from each depth image and deproject them into 3D using the corresponding metric depth measurement. We use a learned stereo depth model trained on simulated and real images [102], which can predict thin features and transparent objects with high accuracy.

Global Bundle Adjustment

Using images with pose drift from SLAM systems results in artifacts in 3DGS models like duplicated or fused objects, fuzzy geometry, and ghosting (Fig. 10.4). Though prior work has demonstrated that pose optimization inside a 3DGS can track camera pose [92], tracking iterations for the method with reported results takes up to a second to perform, making it difficult for online usage. Instead, to mitigate drift, we incorporate updates from global BA and update training camera poses in the 3DGS accordingly. This allows tracking of new camera frames at 30fps in tandem with continual 3DGS optimization for faster model convergence.

9.3 Language Embedded Gaussian Splats

Drawing inspiration from Language Embedded Radiance Fields (LERF), which optimizes a dense CLIP embedding field alongside the RGB field, we adopt LERF's language field representation and overlay a 3DGS over it as the geometry and RGB representation. In LERF, the language field is optimized by volumetrically rendering CLIP embeddings along rays during training. Instead of training on rays, LEGS samples the underlying language field at gaussian center-points, then rasterizes these features onto images for loss supervision. LEGS' language field is also trained with multi-scale crops for CLIP, which as shown in LERF is particularly important for semantic understanding in large scenes where object sizes may vary drastically. This is in contrast to prior work which averages CLIP embeddings as a tradeoff between speed and accuracy [22]. LEGS facilitates inference at approximately 50 Hz 1080p, and its hybrid explicit-implicit representation allows faster scene querying without volumetric rendering.

Given a natural language query, we query the language field to obtain a relevancy map as in LERF. To localize the query in the world frame, we find the relevancy of CLIP features from the hash encoding and take the argmax of this relevancy over 3D gaussian means. This method offers a significant speed increase over feature field methods distilled in NeRFs, where the volumetric representation must first render a dense pointcloud.

Chapter 10

Room-Scale LEGS Results

10.1 Physical Experiments

We evaluate LEGS through a series of open-vocabulary object recall tasks. These tasks are designed to measure the system's competency in capturing and organizing information based on both location and semantic meaning. We evaluate LEGS on three large-scale indoor environments, two kitchen scenes containing different objects and an office workspace as seen in Figure 10.1. The robot begins in a previously unseen environment and is manually moved around a pre-planned path (including straight lines, loops, figure-8, etc) while continuously registering new images until it finishes the path. The robot actuates its torso height to obtain multiple azimuthal perspectives from the same position on subsequent passes. Every 150 keyframes, we perform global bundle adjustment on all previous poses in DROID-SLAM and update accordingly in our 3D Gaussian map. Our system uses 2 NVIDIA 4090s, one for training LEGS, which takes 15 GB of memory and the other for DROID-SLAM, which can take up to 18 GB of memory.

The evaluation approach is as follows: We randomly sample images from our 3DGS training set and query Chat-GPT 4V with: "name an object in this scene in 3 words or fewer". This process is repeated until 15 unique queries are generated for each of the three scenes. The user then chooses a random novel view of the scene and manually annotates a 2D bounding box around each selected object of interest. Then, the user queries LEGS on each object and identifies the point of highest activation energy, and projects that point in the novel 2D view. If the projected point of highest activation is contained in the bounding box, we consider the query successful. This evaluation metric was adopted from previous 3D language mapping works [16], [103]. Additionally, we directly baseline our approach by running LERF [16] to compare the object recall capabilities for a large-scale scene in radiance field methods.

The results in Table 1 suggest that LERF and LEGS have similar language capabilities, recalling roughly the same number of objects per scene. However, to achieve the same visual quality, LERF takes an average time of 44 minutes to train while LEGS only takes 12

Figure 10.1: **3 Scene Environments.** The top 2 sets of images show the Kitchen 1 and Kitchen 2 environments respectively. Each kitchen environment has some unique objects and all objects are arranged in different positions between the two scenes. The bottom set of images shows an office work area.

Test Scene	LERF	LEGS
Kitchen 1	9/15	**10/15**
Kitchen 2	**11/15**	**11/15**
Office	**10/15**	9/15
Avg. Train Time	44min.	**12min.**

Table 10.1: **Object recall success rates.** Comparison between LERF and LEGS on large scenes where both receive the same SLAM poses. LEGS receives poses incrementally, while LERF receives the final poses. Average train time refers to the time until 20 average PSNR. For LEGS we consider the time after the final image is added to the train set.

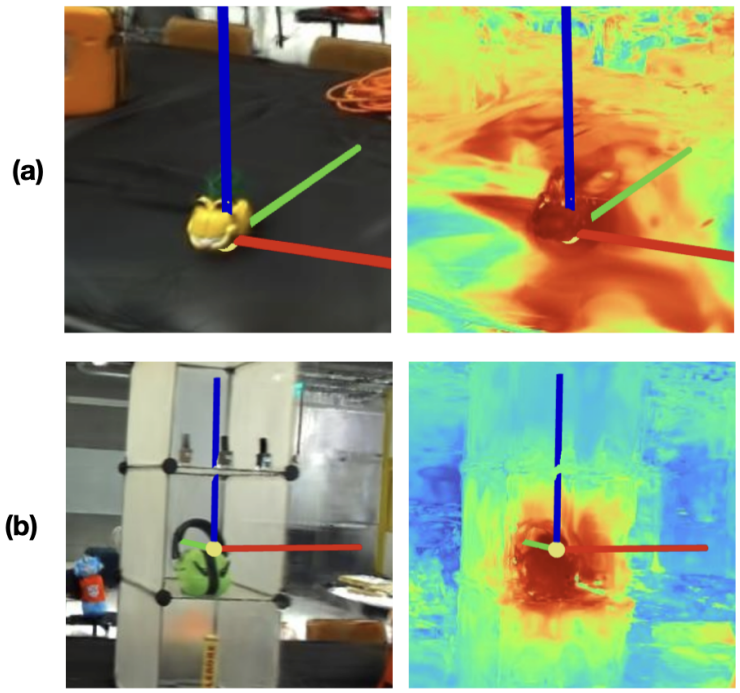


Figure 10.2: **Successful query localization results** on (a) "garfield," (b) "hearing protection." The argmax location for each query is visualized with axes.

minutes. Figure 10.2 shows examples of successful object localization queries. Localization may fail when objects are not seen well in the training views or have similar color to their background, as shown in Figure 10.3.

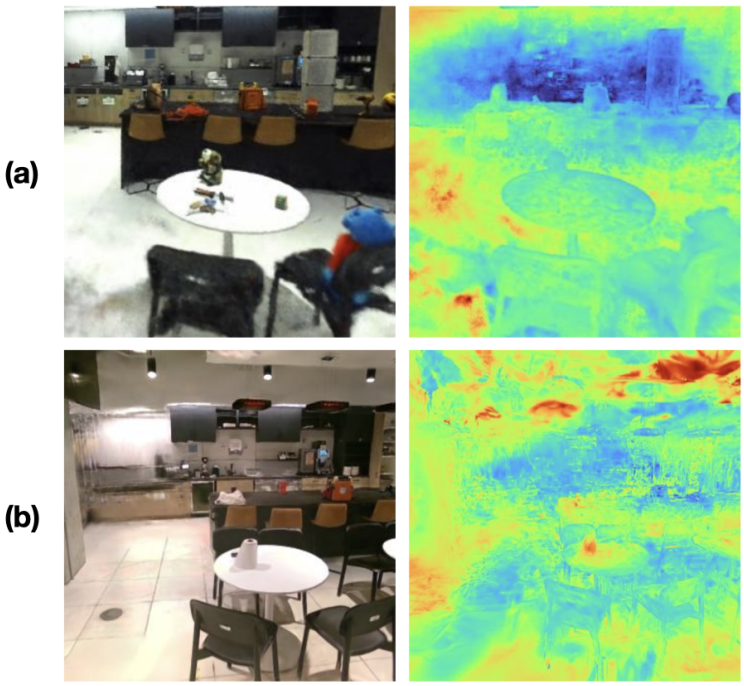


Figure 10.3: **Failed query localization** on (a) "scissors," (b) "paper roll." These showcase common failure conditions caused by: object of interest being too small in the training view, and lack of color feature in queried object and surrounding environment.

	PSNR	
Camera	**w/o BA**	**BA**
D435	18.6	**22.7**
D455	20.0	**23.5**
Zed 2	19.2	**23.8**

Table 10.2: **PSNR (Peak Signal-Noise Ratio) scores** across different cameras with and without bundle adjustment quantifying training-view reconstruction quality. Trained until 20k iterations after final image is added to the train set.

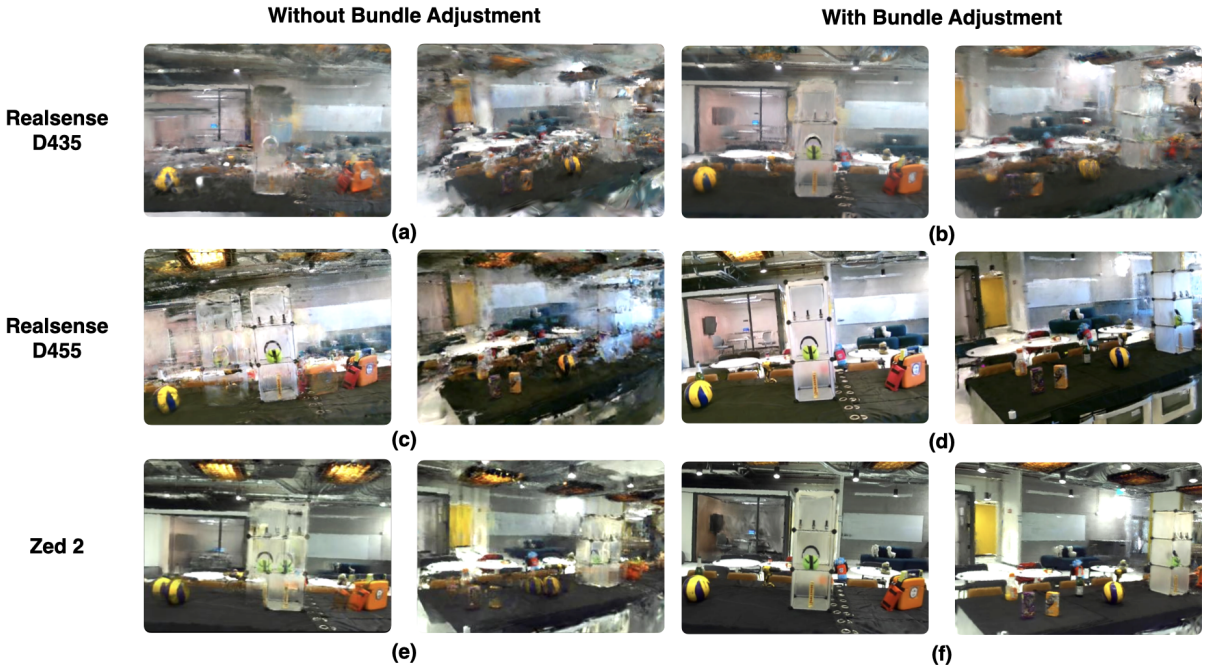


Figure 10.4: **Single Camera Reconstruction Comparison Results.** We compare the quality of Gaussian splats on an Intel Realsense D435, Intel Realsense D455, and Stereolabs Zed 2 with and without bundle adjustment. For each configuration we present two views: one of the Gaussian splat facing the kitchen island head-on and another view at an angle.

10.2 Reconstruction quality comparison

We study how camera configuration and bundle adjustment (BA) affect the quality of the LEGS Gaussian splat as summarized in Figure 10.4 and Table 2. With respect to the camera configuration ablation, we evaluated different depth and stereo cameras including the Realsense D435, Realsense D455, and Zed 2. For each camera configuration, we also run a BA ablation where we either run bundle adjustment at the end of the traversal or not at all.

Global Bundle Adjustment: Table 10.2 suggests bundle adjustment improves Gaussian splat quality for all camera configurations, removing ghostly duplicate artifacts. This is especially true for the Realsense D455 and Zed 2 cameras where the bundle adjustment configurations yielded near-photorealistic views of the scene. However, without bundle adjustment, both the Realsense D455 and Zed 2 camera have significantly more phantom Gaussians and the pose error leads to phantom objects (i.e. the left image in Figure 10.4 part (e) has two volleyballs). The Realsense D435 performs slightly better with bundle adjustment,

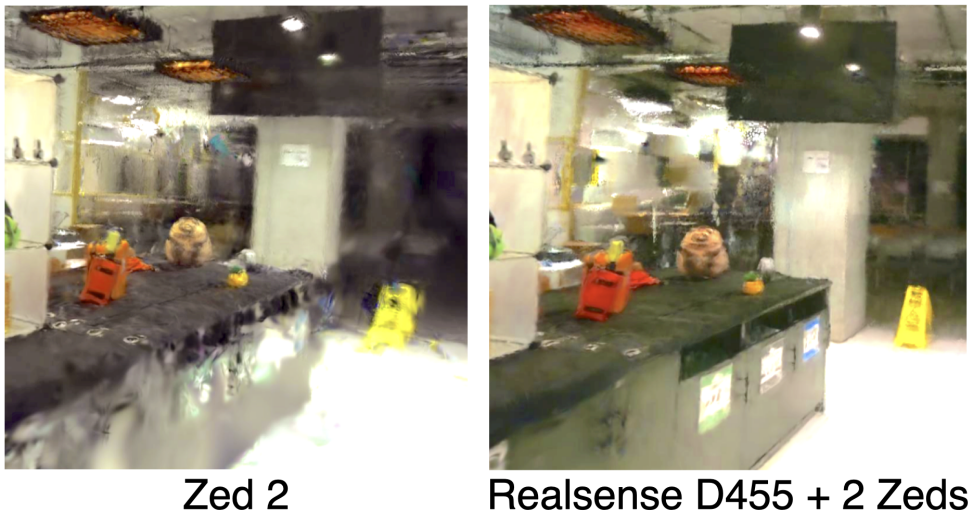


Figure 10.5: **Single Camera vs Multi Camera Reconstruction.** We compare the quality of Gaussian splats on a single Zed 2 against a multi-camera setup including a Realsense D455 and 2 Zed cameras. Overall, the rendering quality is similar for both camera setups, but the effective field of view is increased, elucidating more of the scene such as the wet floor sign and trash chutes on the side faces of the table.

but neither D435 configuration yields a high quality Gaussian splat. This is explained by the D435 having a $69° \times 42°$ horizontal and vertical FOV, while the D455 and Zed 2 have a $90° \times 65°$ and a $110° \times 70°$ FOV respectively. A higher field of view yields more scene information in a single image, and additionally reduced overall pose error in visual-SLAM localization.

We also compared a single Zed 2 camera to a multi-camera setup where the D455 Realsense is front-facing and 2 Zed cameras face the left and right side. Both gaussian splats perform well and properly render objects that were well viewed in the traversal ("raccoon toy" and "first aid kit") as seen in Figure 10.5. However, none of the cameras were pointing toward the ground leading to sparse views of objects near the floor. Because the multi-camera setup captures more views of the scene, it is able to construct a Gaussian splat that is better able to render these low-view objects such as the trash chutes and wet floor sign.

10.3 Limitations and Future Work

The motion smoothness of the Fetch mobile base can have a large affect on the LEGS reconstruction quality; the high stiction between the robot's caster wheels and the environment introduces jolts, causing camera pose inaccuracies and image blurs. For the experiments in

the paper, a human manually pushes the base along a predefined trajectory. In the future, we hope to correct this with a new mobile base.

Although autonomous navigation and obstacle avoidance solutions have been extensively studied [104], these obstacles pose a problem when it comes to the 3D Gaussian map as they are only briefly visible in a few of the ground truth images. 3D Gaussians are initialized at the deprojected points from these few images, but there are not enough views to refine and properly train these Gaussians; the result is oddly colored floaters that obstruct some parts of the static scene of interest.

When performing natural language queries, LEGS inherits the limitations of CLIP distillation into 3D described by similar works [16]. In our experimentation, we find that a large scene scale brings additional challenges in querying, particularly in 1) small or far objects in the perspective of a training view, 2) similar item-background color features, such as white objects on white. Language embedded Gaussian splats can also produce false-positives when querying an object that is not in the scene due to the presence of visually or semantically similar objects. These objects may get incorrectly classified as our object of interest.

We also assume a static environment where no object can move during traversal. This limits the scope of this work because many inventory monitoring applications involve dynamic scenes with moving objects. In future work, we will adapt our method to work for dynamic scenes.

Chapter 11

Conclusion

In this work, I have presented our approaches for automating solutions to two different problems: surgical suturing and efficient tableware decluttering. In both cases, we developed unique robust action primitives and visual perception pipelines, tying them together with related frameworks to tackle the robotic manipulation challenges in each task. For surgical suturing, we developed action primitives for needle insertion, extraction, and handover, and a visual pipeline for 6D needle pose estimation and thread tracking. For tableware decluttering, we built action primitives for consolidating and grasping tableware, and a visual pipeline for identifying tableware and generating grasps. In addition, I present recent work on understanding large scale 3D scenes both visually and semantically, enabling object identification given open vocabulary queries in room-scale scenes.

However, the original problems we set out to tackle are far from solved. Future work on these projects will focus on generalizing the systems we have built by incrementally removing constraints and assumptions we introduced for the sake of simplicity. As reliable as these systems might be, deploying them outside of a laboratory setting will require many more levels of fault tolerance, adaptability to a variety of environments, and the ability to perform the task for extended periods of time with minimal human intervention.

Meeting these requirements is of course easier said than done, as even in a laboratory setting, we faced many challenges along the way. Camera calibration is an extremely sensitive task, and we had several instances where the camera moved accidentally by less than a degree, causing the robot to miss grasps and collide with objects, and requiring extensive recalibration. Our vision systems were also highly sensitive to lighting conditions, especially when detecting the shiny steel surgical needle and using color thresholding to isolate tableware. Despite working in a brightly lit indoor lab, the ambient outdoor lighting from different weather conditions and different times of day often affected the accuracy of our visual perception methods. Even with perfect object detections, we sometimes encountered failures due to kinematic limitations of the robots themselves, requiring a full system reboot. We were not deterred by these hardships, and instead tackled each challenge as it presented itself to build systems that performed despite these difficulties.

Being part of the 5th year MS has been an incredible experience, and a great way to

close out my time at Berkeley. I've learned and grown so much over these past years, both in the AutoLAB and outside. I'm grateful for all the opportunities I've had to dive into the rich world of robotics and see it evolve around me, and I'm glad to leave my mark, however small, on this amazing field. The little kid tinkering with his Raspberry Pi and breadboard kit 15 years ago would hardly believe his silly pet projects have taken him this far. As I embark on my next adventure, I will forever cherish the experiences, lessons, and memories of this place, knowing that it has prepared me well for the exciting journey ahead.

Bibliography

[1] K. Goldberg, "Augmented dexterity: How robots can enhance surgeon dexterity," *Science Robotics*, Under Review (Summer 2023).

[2] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel, *et al.*, "Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy," *Science Robotics*, vol. 2, no. 4, eaam8638, 2017.

[3] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017. DOI: 10.1109/TRO.2017.2721939.

[4] T. Sadjadpour, J. Yu, A. O'Neill, M. Khfifi, L. Y. Chen, R. Cheng, A. Balakrishna, T. Kollar, and K. Goldberg, "Manip: A modular architecture for integrating interactive perception into long-horizon robot manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Under Review)*, 2024.

[5] A. X. Lee, C. Devin, Y. Zhou, T. Lampe, K. Bousmalis, J. T. Springenberg, A. Byravan, A. Abdolmaleki, N. Gileadi, D. Khosid, C. Fantacci, J. E. Chen, A. Raju, R. Jeong, M. Neunert, A. Laurens, S. Saliceti, F. Casarini, M. A. Riedmiller, R. Hadsell, and F. Nori, "Beyond pick-and-place: Tackling robotic stacking of diverse shapes," *CoRR*, 2021. arXiv: 2110.06192.

[6] T. Yamada, S. Yamanaka, M. Yamada, Y. Funahashi, and H. Yamamoto, "Grasp stability analysis of multiple planar objects," in *IEEE International Conference on Robotics and Biomimetics*, 2009.

[7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[8] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang, "Gnfactor: Multi-task real robot learning with generalizable neural feature fields," in *Conference on Robot Learning*, PMLR, 2023, pp. 284–301.

[9] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *7th Annual Conference on Robot Learning*, 2023.

[10] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: `https://openreview.net/forum?id=k-Fg8JDQmc`.

[11] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. Tenenbaum, C. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba, "Conceptfusion: Open-set multimodal 3d mapping," *Robotics: Science and Systems (RSS)*, 2023.

[12] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, *Clip-fields: Weakly supervised semantic fields for robotic memory*, 2023. arXiv: `2210.05663 [cs.RO]`.

[13] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[14] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 311–23 330, 2022.

[15] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi, "Neural feature fusion fields: 3d distillation of self-supervised 2d image representations," in *2022 International Conference on 3D Vision (3DV)*, IEEE, 2022, pp. 443–453.

[16] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 729–19 739.

[17] A. Meuleman, Y.-L. Liu, C. Gao, J.-B. Huang, C. Kim, M. H. Kim, and J. Kopf, "Progressively optimized local radiance fields for robust view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 539–16 548.

[18] P. Wang, Y. Liu, Z. Chen, L. Liu, Z. Liu, T. Komura, C. Theobalt, and W. Wang, "F2-nerf: Fast neural radiance field training with free camera trajectories," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4150–4159.

[19] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.

[20] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, "Openscene: 3d scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 815–824.

[21] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.

[22] X. Zuo, P. Samangouei, Y. Zhou, Y. Di, and M. Li, *Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding*, 2024. arXiv: 2401.01970 [cs.CV].

[23] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, *Langsplat: 3d language gaussian splatting*, 2023. arXiv: 2312.16084 [cs.CV].

[24] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, 102:1–102:15, Jul. 2022. DOI: 10.1145/3528223.3530127. [Online]. Available: https://doi.org/10.1145/3528223.3530127.

[25] P. T. Rose and B. Nusbaum, "Robotic hair restoration," *Dermatologic Clinics*, vol. 32, no. 1, pp. 97–107, 2014, Advances in Cosmetic Dermatology, ISSN: 0733-8635. DOI: https://doi.org/10.1016/j.det.2013.09.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0733863513000946.

[26] W. Kilby, J. Dooley, G. Kuduvalli, S. Sayeh, and C. Maurer Jr, "The cyberknife® robotic radiosurgery system in 2010," *Technology in cancer research & treatment*, vol. 9, no. 5, pp. 433–452, 2010.

[27] H. Saeidi, J. D. Opfermann, M. Kam, S. Wei, S. Leonard, M. H. Hsieh, J. U. Kang, and A. Krieger, "Autonomous robotic laparoscopic surgery for intestinal anastomosis," *Science Robotics*, vol. 7, pp. 1–14, 62 2022, ISSN: 24709476. DOI: 10.1126/scirobotics.abj2908.

[28] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 1202–1209.

[29] K. Dharmarajan, W. Panitch, M. Jiang, K. Srinivas, B. Shi, Y. Avigal, H. Huang, T. Low, D. Fer, and K. Goldberg, "Automating vascular shunt insertion with the dvrk surgical robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 6781–6788. DOI: 10.1109/ICRA48891.2023.10160966.

[30] D. Hu, Y. Gong, E. J. Seibel, L. N. Sekhar, and B. Hannaford, "Semi-autonomous image-guided brain tumour resection using an integrated robotic system: A bench-top study," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 14, no. 1, e1872, 2018.

[31] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. W. Kim, "Supervised autonomous robotic soft tissue surgery," *Science Translational Medicine*, vol. 8, no. 337, 337ra64–337ra64, 2016.

[32] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2014, pp. 6434–6439.

[33] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, pp. 4178–4185.

[34] V. Kamat, V. Ramakrishnan, Y. Mohnot, H. Jalan, J. Isaac, V. Schorp, Y. Avigal, A. Adler, D. M. Fer, and K. Goldberg, "Automating 2d suture placement," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2023.

[35] A. Wilcox, J. Kerr, B. Thananjeyan, J. Ichnowski, M. Hwang, S. Paradis, D. Fer, and K. Goldberg, "Learning to localize, grasp, and hand over unmodified surgical needles," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 9637–9643.

[36] Z.-Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco, and M. C. Yip, "Bimanual regrasping for suture needles using reinforcement learning for rapid motion planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 7737–7743. DOI: 10.1109/ICRA48506.2021.9561673.

[37] V. M. Varier, D. K. Rajamani, N. Goldfarb, F. Tavakkolmoghaddam, A. Munawar, and G. S. Fischer, "Collaborative suturing: A reinforcement learning approach to automate hand-off task in suturing for surgical robots," in *2020 29th IEEE international conference on robot and human interactive communication (RO-MAN)*, IEEE, 2020, pp. 1380–1386.

[38] P. Sundaresan, B. Thananjeyan, J. Chiu, D. Fer, and K. Goldberg, "Automated extraction of surgical needles from tissue phantoms," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2019, pp. 170–177.

[39] C. D'Ettorre, G. Dwyer, X. Du, F. Chadebecq, F. Vasconcelos, E. De Momi, and D. Stoyanov, "Automated pick-up of suturing needles for robotic surgical assistance," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1370–1377.

[40] B. Thananjeyan, A. Balakrishna, U. Rosolia, F. Li, R. McAllister, J. E. Gonzalez, S. Levine, F. Borrelli, and K. Goldberg, "Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3612–3619, 2020.

[41] K. L. Schwaner, I. Iturrate, J. K. H. Andersen, P. T. Jensen, and T. R. Savarimuthu, "Autonomous bi-manual surgical suturing based on skills learned from demonstration," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4017–4024. DOI: 10.1109/IROS51168.2021.9636432.

[42] Z.-Y. Chiu, A. Z. Liao, F. Richter, B. Johnson, and M. C. Yip, "Markerless suture needle 6d pose tracking with robust uncertainty estimation for autonomous minimally invasive robotic surgery," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 5286–5292.

[43] Y. Jiang, H. Zhou, and G. S. Fischer, "Markerless suture needle tracking from a robotic endoscope based on deep learning," in *2023 International Symposium on Medical Robotics (ISMR)*, 2023, pp. 1–7. DOI: 10.1109/ISMR57123.2023.10130199.

[44] J. Lu, F. Richter, and M. C. Yip, "Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4622–4629, 2022.

[45] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning," *Nature neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.

[46] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[47] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.

[48] S. Paradis, M. Hwang, B. Thananjeyan, J. Ichnowski, D. Seita, D. Fer, T. Low, J. E. Gonzalez, and K. Goldberg, "Intermittent visual servoing: Efficiently learning policies robust to instrument changes for high-precision surgical manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 7166–7173.

[49] L. Lipson, Z. Teed, and J. Deng, "Raft-stereo: Multilevel recurrent field transforms for stereo matching," in *International Conference on 3D Vision (3DV)*, 2021.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[51] B. Thananjeyan, J. Kerr, H. Huang, J. E. Gonzalez, and K. Goldberg, "All you need is luv: Unsupervised collection of labeled images using uv-fluorescent markings," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 3241–3248.

[52] V. Schorp, W. Panitch, K. Shivakumar, V. Viswanath, J. Kerr, Y. Avigal, D. M. Fer, L. Ott, and K. Goldberg, "Self-supervised learning for interactive perception of surgical thread for autonomous suture tail-shortening," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2023.

[53] Y. Sun, E. Amatova, and T. Chen, "Multi-object grasping - types and taxonomy," in *IEEE ICRA*, 2022.

[54] T. Chen, A. Shenoy, A. Kolinko, S. Shah, and Y. Sun, "Multi-object grasping – estimating the number of objects in a robotic grasp," in *IEEE IROS*, 2021.

[55] W. C. Agboh, J. Ichnowski, K. Goldberg, and M. R. Dogar, "Multi-object grasping in the plane," in *ISRR*, 2022.

[56] U. J. Fernández, S. Elizondo, N. Iriarte, R. Morales, A. Ortiz, S. Marichal, O. Ardaiz, and A. Marzo, "A multi-object grasp technique for placement of objects in virtual reality," *Applied Sciences*, 2022. DOI: 10.3390/app12094193.

[57] W. C. Agboh, S. Sharma, K. Srinivas, M. Parulekar, G. Datta, T. Qiu, J. Ichnowski, E. Solowjow, M. Dogar, and K. Goldberg, "Learning to efficiently plan robust frictional multi-object grasps," *CoRR*, 2022.

[58] P. Jiang, J. Oaki, Y. Ishihara, and J. Ooga, *Multiple-object grasping using a multiple-suction-cup vacuum gripper in cluttered scenes*, 2023. arXiv: 2304.10693 [cs.RO].

[59] V. P. Nguyen and W. T. Chow, "Wiring-claw gripper for soft-stable picking up multiple objects," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3972–3979, 2023. DOI: 10.1109/LRA.2023.3273512.

[60] H. Huang, L. Fu, M. Danielczuk, C. M. Kim, Z. Tam, J. Ichnowski, A. Angelova, B. Ichter, and K. Goldberg, "Mechanical search on shelves with efficient stacking and destacking of objects," *ISRR*, 2022.

[61] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *IJRR*, vol. 31, no. 9, pp. 1021–1043, 2012.

[62] R.-P. Berretty, K. Goldberg, M. Overmars, and A. van der Stappen, "Orienting parts by inside-out pulling," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 2, 2001, 1053–1058 vol.2. DOI: 10.1109/ROBOT.2001.932733.

[63] E. Huang, A. Bhatia, B. Boots, and M. T. Mason, "Exact bounds on the contact driven motion of a sliding object, with applications to robotic pulling," in *Robotics: Science and Systems*, 2017.

[64] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, 2019.

[65] J. Mahler, F. T. Pokorny, S. Niyaz, and K. Goldberg, "Synthesis of energy-bounded planar caging grasps using persistent homology," *IEEE Transactions on Automation Science and Engineering*, 2018.

[66] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," *ICCV*, 2019.

[67] M. Hasan, M. Warburton, W. C. Agboh, M. R. Dogar, M. Leonetti, H. Wang, F. Mushtaq, M. Mon-Williams, and A. G. Cohn, "Human-like planning for reaching in cluttered environments," *ICRA*, 2020.

[68] S. Tirumala, T. Weng, D. Seita, O. Kroemer, Z. Temel, and D. Held, "Learning to singulate layers of cloth using tactile feedback," in *IEEE/RSJ IROS*, 2022, pp. 7773–7780. DOI: 10.1109/IROS47612.2022.9981341.

[69] W. L. a and b. Jean-Jacques E. Slotine a, "On contraction analysis for nonlinear planning," *IEEE Robotics and Automation Letters*, pp. 1044–1051, 2016. [Online]. Available: http://web.mit.edu/nsl/www/preprints/contraction.pdf.

[70] K. O. Arras, "Feature-based robot navigation in known and unknown environments," 2003.

[71] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 1985, pp. 138–145.

[72] G. Jiang, L. Yin, S. Jin, C. Tian, X. Ma, and Y. Ou, "A simultaneous localization and mapping (slam) framework for 2.5 d map building based on low-cost lidar and vision fusion," *Applied Sciences*, vol. 9, no. 10, p. 2105, 2019.

[73] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization," *IEEE Transactions on robotics and automation*, vol. 17, no. 2, pp. 125–137, 2001.

[74] A. Tapus, "Topological slam: Simultaneous localization and mapping with fingerprints of places," 2005.

[75] B. Alsadik and S. Karam, "The simultaneous localization and mapping (slam)-an overview," *Journal of Applied Science and Technology Trends*, vol. 2, no. 02, pp. 147–158, 2021.

[76] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE international symposium on safety, security, and rescue robotics*, IEEE, 2011, pp. 155–160.

[77] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, pp. 1271–1278.

[78] L. Huang, "Review on lidar-based slam techniques," in *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*, IEEE, 2021, pp. 163–168.

[79] M. T. Lázaro, R. Capobianco, and G. Grisetti, "Efficient long-term mapping in dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 153–160.

[80] L. Roldao, R. De Charette, and A. Verroust-Blondet, "3d semantic scene completion: A survey," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1978–2005, 2022.

[81] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.

[82] H. A. Kestler, S. Sablatnög, S. Simon, S. Enderle, A. Baune, G. K. Kraetzschmar, F. Schwenker, and G. Palm, "Concurrent object identification and localization for a mobile robot," *KI*, vol. 14, no. 4, pp. 23–29, 2000.

[83] K. Genova, X. Yin, A. Kundu, C. Pantofaru, F. Cole, A. Sud, B. Brewington, B. Shucker, and T. Funkhouser, "Learning 3d semantic segmentation with only 2d image supervision," in *2021 International Conference on 3D Vision (3DV)*, IEEE, 2021, pp. 361–372.

[84] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez, *et al.*, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 75–82.

[85] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*, PMLR, 2023, pp. 287–318.

[86] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[87] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*, PMLR, 2023, pp. 2165–2183.

[88] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *arXiv preprint arXiv:2302.07241*, 2023.

[89] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[90] Q. Gu, A. Kuwajerwala, S. Morin, K. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C. de Melo, J. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv*, 2023.

[91] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," 2023.

[92] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track & map 3d gaussians for dense rgb-d slam," *arXiv preprint arXiv:2312.02126*, 2023.

[93] M. Li, S. Liu, H. Zhou, G. Zhu, N. Cheng, and H. Wang, *Sgs-slam: Semantic gaussian splatting for neural dense slam*, 2024. arXiv: 2402.03246 [cs.CV].

[94] T. Chen, O. Shorinwa, W. Zeng, J. Bruno, P. Dames, and M. Schwager, *Splat-nav: Safe real-time robot navigation in gaussian splatting maps*, 2024. arXiv: 2403.02751 [cs.RO].

[95] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[96] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, *Scannet++: A high-fidelity dataset of 3d indoor scenes*, 2023. arXiv: 2308.11417 [cs.CV].

[97] T. Schöps, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[98] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, *Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes*, 2024. arXiv: 2312.07920 [cs.CV].

[99] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[100] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, *et al.*, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–12.

[101] Z. Teed and J. Deng, *Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras*, 2022. arXiv: 2108.10869 [cs.CV].

[102] K. Shankar, M. Tjersland, J. Ma, K. Stone, and M. Bajracharya, "A learned stereo depth system for robotic manipulation in homes," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2305–2312, 2022.

[103] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, "Language embedded 3d gaussians for open-vocabulary scene understanding," *arXiv preprint arXiv:2311.18482*, 2023.

[104] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00 022, 2017.