Grid Inference and Partial Scan Registration for Intelligent Collaborative Robot Systems

by

Valerie K. Chen

S.B. Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2022

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Valerie K. Chen
Department of Electrical Engineering and Computer Science
May 12, 2023

Certified by: Julie A. Shah
H.N. Slater Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by: Joshua Gruenstein
CEO, Tutor Intelligence
Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Grid Inference and Partial Scan Registration for Intelligent Collaborative Robot Systems

by

Valerie K. Chen

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2023, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis proposes advancement of the collaborative and intelligent abilities of Tutor Intelligence robot systems through leveraging the geometry of array structures to perform online inference of object locations and registering partial in-hand scans to automatically orient objects. This research will automate portions of the data annotation process required for the robots' deep intelligence, enabling the collaborative robot systems to more efficiently and effectively perform pick-and-place tasks. Evaluation is conducted through an exploratory pilot study, and further design recommendations are given.

Thesis Supervisor: Julie A. Shah
Title: H.N. Slater Professor of Aeronautics and Astronautics

Acknowledgments

I would like to thank all of my colleagues at Tutor Intelligence, who I have had the honor to work alongside and learn from this past year. Special thanks go to my manager, Josh Fishman, for your help and jokes, and to the founders, Alon Kosowsky-Sachs and Josh Gruenstein, for sharing your vision and supporting pursuit of my Master's degree.

Thank you to my adviser, Prof. Julie Shah, for your guidance and inspiration. Another thank you goes to my labmates at the Interactive Robotics Group for your advice and friendship.

Lastly, I would like to thank my family for your constant and unconditional support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robots are increasingly being adopted beyond their traditional factory cages, working directly alongside human workers in factory lines to automate the most tedious and dangerous tasks [1, 2]. As such, there has been increased need for robots to robustly perform pick-and-place tasks; notably, the Amazon Picking Challenge has driven a body of work on the task [3]. In the United States, policy has also encouraged the adoption of robotics in manufacturing in order to increase the ability of firms to compete in a global market; for example, the Massachusetts Manufacturing Accelerate Program (MMAP) [4] provides capital and business connections for small to medium-sized facilities to purchase automation solutions. Adoption of robots into production lines where human workers are stationed enables automation of the most menial and monotonous tasks, providing the additional benefit of allowing facilities to subsequently upskill workers to supervise robots. Thus, not only can overall productivity can be increased, but also workers can be promoted into higher-skilled, higher-paying positions.

In situations where collaborative robots, termed cobots, work directly on lines with human workers, regularity of tasks is often not guaranteed; a worker may brush a box of product as they walk past, knocking the box out of place, or they may place product before the robot with the labels in various orientations. In order to handle the variations of irregular, out-of-cage work—for example, picking objects of different sizes or packed loosely into boxes—robotic intelligence beyond hand-designed, fixed

robot motions is required.

To this end, Tutor Intelligence has developed intelligent collaborative robots (B-1) that leverage deep learning for pick-and-place tasks [5]. Productivity gains can be achieved in factory settings by using the robots to perform tedious, repetitive motions and instead allowing the human worker to serve a more ubiquitous and less physically taxing role annotating data generated by the robot. These annotations are then utilized by the robots to improve performance.

Many factory settings involve products that are spaced at regular intervals or packed into geometric arrays in pallets or boxes. Moreover, there may be requirements for the products to move through production lines in certain orientations in order to read codes, affix labels, etc. Robotic intelligence can be used to augment the data labelling task in these cases, removing workload from the human labeller.

To automate label generation of the robot data, I have created a grid inference target labelling scheme that leverages geometric regularity of boxes or pallets of objects. This method infers the overall geometry of object placements based on the identification of individual objects within the structure. To allow the robot to automatically orient objects in specified orientations, I have developed an algorithm to register a partial scan against a reference object scan and calculate a correctional rotation to uniformly orient objects. Evaluation of the effectiveness of grid inference and automatic orientation in automating the data annotation process is performed through an exploratory pilot study.

The main contributions of this thesis work are:

- Online inference of geometric grid structure using Fourier Transform techniques,
- Orientation-agnostic scan registration to calculate correctional rotations for consistent object placement, and
- An exploratory pilot study investigating the effects of the above tooling on human labeller productivity and workload.

Chapter 2

Relevant Work

2.1 Section 1: Automation of the Data Labelling Process

As the relevance of data-driven deep learning approaches for robotics has grown, so has the need for ways to address the large amounts of data traditionally required to train these deep models. One approach to address this problem is to automate the data labelling process directly. The authors of [6] and [7] both bring the human into the training loop through a streamlined verification step during the active learning process for 3D segmentation and object detection, respectively. [6] propagates human annotations over the dynamic network, while [7] requires no additional initial annotation beyond image-level labels. This work seeks to additionally automate the data labelling process beyond the integration of a human into the training loop.

Researchers have also investigated methods for a robot to semi-automatically collect labelled data, for example where objects are placed in front of the robot during a "training phase," and the robot automatically collects RGBD scans, segments out the object, and uses the segmented object for training [8]. In [9], the authors explored the use of a physical virtual-reality marker to semi-automatically label robot-collected scans. However, the training phase required for these methods is not practical for real-world industrial applications where a robot going offline for training for each new

object would mean a production line constantly taken down and large penalties to revenue. Moreover, this training phase would require a human worker to supervise the robot and place objects in front of it as required, furthering the cost of such training.

2.2 Section 2: Image processing for Grid Inference

Development of the scalable, efficient Fast Fourier Transform algorithm [10, 11] enabled a large body of work exploring frequency domain analysis of various signals. Frequency domain analysis has been extended to RGB-D inputs for robotics applications as well, including post-processing the RGB-D images to fill in holes where depth signal was lost [12], and using various frequency bands to segment objects by textures [13]. However, these methods apply directly to the scene, whereas this work considers human inputs labelling portions of a scene.

To the best of the author's knowledge, no previous work has been done to explore inference of geometric structures made of arbitrary objects with human-given priors. For a task that considers boxes only or consistently-placed objects in a grid, there exist techniques for edge crease detection that would allow for grid reconstruction [14]. However, products are often not rectangular or regular (eg. bags of chips, stuffed animals, etc.) and can be packed without regard to orientation, resulting in a high variance of contours and rendering these approaches ineffective. The authors in [15] showed that the underlying symmetries in a scene can still be identified through overlaying noise, which this work will similarly seek to show.

2.3 Section 3: Registration

The Iterative Closest Point (ICP) algorithm has held the place of both industry and academic standard for registration of two roughly aligned point clouds since its introduction [16], with many variants developed afterwards by the research community [17]. However, depth-based registration is insufficient for orienting geometrically-symmetric objects with differing visual markings. Further work has been conducted

to extend ICP methods to incorporate color [18] and increase robustness to lighting noise [19], but ICP is known to be a time-intensive algorithm and have a convergence success highly sensitive to initialization.

Deep learning has proven to be effective for solving various challenges in perception in recent years, and researchers have explored deep learning-based methods for oriented object detection [20, 21], though orientation is limited to the axis of the 2-dimensional image with these methods. Deep-learning-based approaches have also been extended to registration of point clouds, both colorless [22] and colored [23, 24]. Though some existing methods, such as those presented in [24] for learning both visual and geometric visual features from RGB-D video, may present a potential solution for quick online registration and orientation of a grasped object, the requirement for training data prohibits their use for Tutor Intelligence's robots; it is impractical in a manufacturing setting to allocate time and resources to collecting training data for all the various objects the robots may pick.

Chapter 3

Task Definition

3.1 Robotic System

Tutor Intelligence robots leverage deep intelligence in order to perform a variety of pick-and-place tasks on differing types of objects. Because the deep intelligence requires labelled data to learn execution of various tasks, the role of human data-labellers in annotating pick and place locations in top-down scans of the product workspace saved by the robot, such as in figure B-5a, is integral to the process. Performance of the deep-learning-based system depends on both accuracy and correctness of labelled data.

The faster robot scan data can be accurately labelled, the greater the number of tasks Tutor Intelligence robots can perform. Without external aid, a single human worker is limited in the rate at which they can produce accurate and correct data labels. Thus, this work aims to increase the rate of data label generation by automating portions of the process and decreasing the human worker's workload, supporting the application of the robotic system to a wide variety of tasks.

3.2 Robot Tasks

Tutor Intelligence robots are deployed in collaborative production environments, where humans work directly on the same lines as, and sometimes interact directly

with, the robots (B-1). Multiple human workers complete the various tasks necessary for a production line, and the robots perform their own pick-and-place tasks on the same line next to the workers. The workers collaborate with the robots by loading products for the robots to pack / unpack and retrieving / placing empty containers for the robot to use. Workers are tasked with setting up the robot and its jobs, ensuring that it runs smoothly throughout the day.

Specific tasks Tutor Intelligence robots perform include picking products out of boxes and placing them onto a conveyor, picking products off a conveyor and onto another, and picking product from one bin into a box. Many of these scenarios may have elements of regularity that can be leveraged for increased labelling efficiency.

One such theme of regularity is the geometric placement of products in boxes or pallets. Robots may either be tasked with picking objects out of or placing them into these regular geometric arrays. Given that all of the products are identical, identification of a subset of the objects would allow for inference of a larger set of the objects.

Many production lines require products to enter parts of the line facing the same direction (eg. for accurate labelling, boxing, etc.). However, the robot may receive these objects rotated in various ways and must place them in a specified orientation. The data labellers' job of annotating the rotations by hand is time-consuming and even impossible in situations where there are no identifiable visual indicators of orientation in the images saved by the robot, for example when scanning pallets of canned goods from a top view. Moreover, a mistake in placement orientation often results in a product that must be thrown out, for example if it receives a label further down the line in an incorrect location. Thus, automating this process of deriving the correct placement rotation for each object will both improve the efficiency of data-labelling and allow the robot to perform jobs with a larger variety of products that require placement in specific orientations.

In particular, this thesis work will consider two classes of setups involving elements of regularity:

1. Products placed in regular geometric arrays, such as bars of soap packed neatly

into a shipping box or a pallet of cereal boxes, where the geometry of the product can be used to infer object labels, and

2. Oriented products, such as a box of soup cans or sauce bottles with labels, that are required to be placed facing the same direction.

Chapter 4

Approach

4.1 Grid-Based Inference

As products often are arranged into orderly geometric arrays, these geometries can be exploited to automate identification of all objects in the arrays for pick and place operations. The problem is reduced initially to a two-dimensional layer of objects to reduce parameter tuning for different objects. Additionally, the formulation is restricted to include only arrays with orthogonal axes, referred to as *grids*. The following presents an online, stateless approach to grid inference.

4.1.1 Specification of a Grid

Though grid inference in entirety does not depend on internal state, a representation is still necessary to internally refine and update the best understanding of geometric structure. A grid can be specified with the following information:

- p: grid corner
- $\vec{u}, \vec{v}$: unit basis vectors defining grid axes
- $dim1, dim2$: dimensions of the grid along axes $\vec{u}, \vec{v}$
- $step1, step2$: step sizes along axes $\vec{u}, \vec{v}$

To fully specify the grid without a 45 degree ambiguity of orientation, an assumption is made that the first two given points align along one axis of the grid; in other words, the first two points either belong to the same row or column of the grid. This requirement is given as part of the instructions for grid inference usage.

4.1.2 Planar Conversion

Points corresponding to objects in the camera frame must be first projected from the camera frame to 3-D world points to correct for camera distortions. However, since inference is being performed on a 2-D layer of objects, the dimensionality must then be reduced. To reduce dimensionality, points are first projected to the plane that lies on the pick face of the layer of objects. This plane is defined by the two largest components found through Principal Component Analysis (PCA).

Because of the noise of real-world data and to mitigate complications that arise when PCA is run with insufficient points, a collection of points is sampled from the point cloud surrounding each of the labeller-given points, which is then used to fit the plane. The process is as follows: the sampling kernel size in meters is first determined by computing $min(\epsilon * w, 0.02)$, where w is the width of the object found through a preliminary object scan step described in section 4.2.1 and ϵ is a tolerance scaling factor empirically determined to be 0.7. The width in meters is converted to pixel space by first projecting a vector of length w meters xyz coordinates into pixel space according to the projection equation

$$i_u, i_v = \left(\frac{K \cdot p}{p_z} \right)_{xy} \tag{4.1}$$

where i_u, i_v are the coordinates of the gripper in the image frame, p is the point to be projected from xyz coordinates, and the subscripts x, y, and z refer to the x, y, and z components of the position, respectively. The linearity of the equation allows for direct projection of the width w into pixel space, where the projected width w_p is $w_p = | < i_u, i_v > |$. For computational efficiency, a maximum of 2500 points were sampled with uniform stride a maximum of $\frac{w_p}{2}$ away from each labelled pixel.

4.1.3 Grid Parameter Inference

With the assumption that the first two given points a and b are axis-aligned, $\vec{u}$ is first defined as $\vec{u} = \frac{b-a}{|b-a|}$. Determination of the second basis vector, $\vec{v}$, requires definition of a parameter, ϵ, thresholding the allowable error in each row or column the grid. A subsequent point, c, given by the labeller is found to define the second dimension of the grid when

$$\frac{|\vec{u} \times (c-p)|}{|c-p|} > \epsilon \tag{4.2}$$

, where $|\vec{u} \times (c-p)|$ is the perpendicular distance between point c and the unit basis vector $\vec{u}$, and p is the corner of the grid. Once a point c, determined to lie in a different row/column from preceding points, is found, $\vec{v}$ can be taken to be the perpendicular vector to $\vec{u}$ with a positive dot product with the vector from p to c according to

$$\begin{aligned} \vec{w} &= (c-p) - \Big(\vec{u} \cdot (c-p)\Big)\vec{u} \\ \vec{v} &= \frac{\vec{w}}{|\vec{w}|} \end{aligned} \tag{4.3}$$

The corner p can be set arbitrarily and updated as the point with the most negative sum of dot products with the unit basis vectors $\vec{u}$ and $\vec{v}$. Grid dimensions $dim1, dim2$ are determined by projecting all labelled points along the basis vectors and taking the maximum distances for each axis.

4.1.4 Grid Step Size

Frequency response

The human-given inputs for grid cells will be prone to noise due to fatigue, human inaccuracy, low-resolution input images, etc. Moreover, any error in step sizes $step1, step2$ will be compounded and result in poor estimates as the number of objects increases. The periodicity of geometric grids can be leveraged to address this noise and mitigate intensive parameter-tuning for different target objects by using a transform-based analysis of step sizes.

Specifically, labelled points are first projected along the two basis vectors and represented as an unsigned distance from the corner. The vector of projected distances $\vec{d}$ is discretized into a signal with number of samples

$$N = max\left(0, \frac{round(max(\vec{d}))}{dx} + 2\right)$$

. dx is the predefined step size used in generating the signal, which was determined to be .002 meters. This signal is initialized with zeros and incremented at the corresponding index for each distance.

Frequencies of the signals are analyzed using the one-dimensional Fast Fourier Transform implemented in the NumPy FFT package [25, 10].

Filtering

An additional high pass filtering step on the resulting frequency response is performed before calculation of step size in order to avoid calculating a step size that is a smaller factor of the true value. With knowledge of the object's width w and length l from the object scan, the minimum allowable frequency is found according to the following:

$$f_{min} = \frac{1}{max(w, l) * \alpha} \tag{4.4}$$

α is a value greater than 1 that accounts for human inaccuracy in the given labels by giving a more conservative estimate of the lowest acceptable frequency, empirically determined to be 1.25. Frequencies less than f_{min} are ignored.

Calculating step size

Step size can be calculated from the fundamental frequency of the geometric array by first finding the fundamental frequency $freq$:

$$freq = \frac{peak\_index}{dx * N} \tag{4.5}$$

$peak\_index$ is determined to be the lowest index of the detected peaks of the

Fourier Transform with a frequency value above the minimum frequency discussed in the Filtering step. Peaks are found using the SciPy Spatial `find_peaks` function with distance parameter 3 [26]. Consequently, step size is found as the inverse of the fundamental frequency:

$$step = \frac{1}{freq} \tag{4.6}$$

in meters.

4.1.5 Grid Generation and Online Updates

With grid parameters fully defined, a grid can be generated upon request. Labels are generated in 3D world coordinates first and then converted into the camera frame. Orientation of labels is taken to be identical to that of the first labeled data point given by the human annotater.

All grid parameters except the basis vectors are updated in an online manner for each new data label provided. Thus, as more labels are given, a more refined estimate of the grid of objects can be generated.

4.1.6 User Interface

An interactive interface was created by another member of the team in order to modify and load the grid of suggestions. The interface provides the ability to send labels to the grid generation tool, modify the labels (vertical and horizontal translation, vertical and horizontal rotation, clockwise and counter-clockwise rotation about the center of the grid), and then accept or reject the labels (individually or as a group). Additionally, this interface allows the user to save a modified grid and load it onto a subsequent scan.

4.2 Automatic Object Orientation

For this task, which requires that all products are identical and only rotated about the z-axis, two additional human inputs beyond that of the typical workflow are required: 1) collocated workers are asked to specify the correct orientation of the product by taking an image of the product correctly oriented at the place location. 2) The human data labellers use this reference image to label the correct orientation of a placed product.

The robot automatically calculates the necessary intermediate steps such that the data labellers can otherwise label product without considering orientation. First, the robot scans the object with the lower RGBD camera and creates a model of a "canonical" object. Then, features are extracted from the RGB image. For each object that is required to be placed, the robot will pick the object, save a single image of the picked object, and from that image calculate the rotation necessary to place the object in the requested orientation.

4.2.1 Object Scan

In order to correctly orient objects, there must be an understanding of the correct orientation. In this approach, I have chosen to utilize a "canonical" object as a reference and specify the correct rotation in relation to this canonical object. Previous work at Tutor Intelligence enables the robot to take multiple scans of an object's sides and build a point cloud model of the object from theses scans. I extract canonical features using the RGBD scans and object point cloud generated by this process.

4.2.2 Partial Scan and Object Filtering

Following a successful pick of an object, the robot moves to a predetermined joint configuration where the product is in view of the bottom RGBD camera and saves a single scan. The rigid transformations of the base camera in the world frame ${}^{c}_{w}T$ and gripper in the world frame ${}^{g}_{w}T$ are known and can be composed to find the location

of the gripper in the camera frame according to

$$ {}^{g}_{c}T = inv({}^{c}_{w}T) \cdot {}^{g}_{w}T \tag{4.7} $$

. The euclidean norm, $d_{gripper}$, of the translational component of ${}^{g}_{c}T$ is used to estimate the valid depths that correspond to the object in the depth scan. The object is filtered from the background by thresholding for depth values d that are within $\alpha * max\_w$ of the distance between the camera and the gripper according to

$$ d_{gripper} - \alpha * max\_w \leq d \leq d_{gripper} + \alpha * max\_w \tag{4.8} $$

. Maximum object width max_w is determined by taking the maximum of width w and length l, where w and l are determined during the earlier object scan step detailed in section 4.2.1, and α was empirically determined to be 1.3.

The bitwise depth mask created by this filtering step is then additionally processed to filter out points above the object (corresponding to the gripper) and below the object that may be close to the object (such as additional product). The 3D coordinates of the gripper can be found in the camera image according to equation 4.1, where p in this case would be the position of the base of the gripper in the camera frame. As the predetermined scanning joints require the object to be held vertically (in other words, roll and pitch of the object expressed in the world frame are both zero), the gripper is segmented out of the image with a horizontal line at its location in the image. Similarly, points below the object are segmented out by finding the location of the bottom of the object in 3D world coordinates using object height from the object scan, projecting this point into the image according to Equation 4.1, with p in this instance referring to the position of the gripper tip in the camera frame. After segmenting all values below the horizontal line in the camera image that intersects this point, the result is an image with the grasped object segmented from the remainder of the scene.

4.2.3 Feature Extraction and Matching

Keypoints and descriptors in the segmented RGB image are detected and computed using the OpenCV implementation of SIFT feature extraction [27] with default parameters. Keypoints and descriptors for all scans in the preliminary full object scan step are computed and saved, and keypoints and descriptors for the single partial scan taken during the pick-and-place task are computed at run time.

SSC

The methods presented by Bailo et al. in [28] were explored as a method to increase performance by forcing use of keypoints that describe varying features on the object. However, downsampling through Suppression via Square Covering (SSC) actually decreased accuracy of final calculated correctional rotation. A possible explanation for this decreased performance is that while many applications involving keypoint detection are sensitive to redundant information (eg. clusters of keypoints), the following registration step utilizes a voting-based approach to align two point clouds. Thus, it is more effective to have large numbers of correct matches than to have spread out matches with lower confidence.

Feature Matching

Matches between the descriptors of the canonical object scan and the partial scan were calculated using OpenCV brute-force matcher with L1 norm and cross check. The n matches with the lowest distances were used in following calculations, with n empirically determined to as $n = 50$.

For each match, the 3D locations of the corresponding keypoints in the canonical scan and the partial scan were calculated by projecting pixel values u, v from the RGB-D images back to a point in world coordinates p according to

$$p = K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} * d \tag{4.9}$$

where K is the camera intrinsics matrix and d is the sensed depth at pixel u, v. To constrain the algorithm to only calculating a rotation about the z-axis of the gripper, the z coordinates of these match points were zeroed.

4.2.4 Registration

Once feature extraction has been performed for both the reference and partial scans, the partial scan can be registered to the canonical scan.

After the 3D locations of the keypoints have been found, the canonical and partial sets are aligned using Random Sample Consensus (RANSAC) to account for noise and fitting models through the Kabsch-Umeyama algorithm, which minimizes the overall distance of clouds with known correspondences as described in [29] and [30]. RANSAC is run with a maximum of 500 iterations and an error threshold of .001. This generates a rigid transformation comprised of position and rotation of the partial scan keypoint point cloud to the canonical ${}^{p}_{c}T$. Since we assume all objects are picked up from the "top," the yaw component of the rotation is extracted as the correctional rotation.

Due to the variations of placement of the object in the robot gripper (ie. object z axes are not guaranteed to be aligned with the gripper z axis) and the zeroing of keypoint z-coordinates resulting in some rotational ambiguity, the RANSAC alignment may produce a rotation with a roll offset by approximately $\pm\pi$ radians accompanied by a yaw offset by approximately $\pm\pi$ from the true values. To address this edge case, an additional check was implemented on the value of the calculated roll component r of the rotation: if $abs(abs(r) - \pi) \leq \epsilon$, where ϵ is 0.2, the calculated yaw to reorient the object was rotated by π radians.

4.2.5 Flattened Scan Representation

Since target objects can be rotationally symmetric from a top-down view, a different representation is necessary to allow data labellers to visually identify the correct rotation of object placement and annotate the correct placement. A flattened view

of the object's sides is generated by first finding the minimum bounding box that includes the entirety of all masked views of the object's sides from the object scan step (4.2.1) and cropping the images by this bounding box. These cropped images are subsequently horizontally concatenated and warped into a ringlike shape. Since all component scans are constrained to be the same size and scans are taken at consistent angle intervals, the scan angles that correspond to columns in the concatenated array can be linearly interpolated. The correspondences of image columns to scan angles are stored, allowing determination of the rotation annotated by the data labeller. At run time, this rotation from the canonical orientation to the requested orientation, ${}^{r}_{c}R$, is then composed with the correctional rotation from the scan orientation to the canonical ${}^{c}_{s}R$ to generate an orientation for the place action according to

$$ {}^{r}_{s}R = {}^{c}_{s}R \cdot {}^{r}_{c}R \tag{4.10} $$

.

Chapter 5

Results and Evaluation

To evaluate these effectiveness of grid inference and automatic orientation as tools to increase data labelling efficiency, an exploratory pilot study was conducted exploring usage of these tools.

5.1 Aims of the Experiment

The aims of both the grid inference and auto-orientation of product is to increase the rate of successful annotations. The rate of successful annotation r can be measured as the number of successful annotations per minute (APM), calculated as the total number of successful annotations a divided by total time scan data are present on the participant's screen in minutes t according to

$$r = \frac{a}{t}$$

An annotation is considered successful if it results in a successful pick-place motion. The following hypotheses are proposed concerning rate of annotations:

Hypothesis 1. *The average rate of annotations will be greater when grid suggestions are enabled.*

Hypothesis 2. *The average rate of annotations will be greater when auto-orientation is enabled.*

Moreover, since auto-orientation is intended to be used in situations in which an object's place orientation must be accurate, the following hypothesis is proposed.

Hypothesis 3. *The average rate of erroneous annotations will be lower when auto-orientation is enabled.*

A pick-place motion is defined as erroneous when it results in a place where the object is rotated more than ± 30 degrees from the requested placement.

Both tools aim to perform portions of the work the human labeller would otherwise have to perform. Thus, we formulate the following two hypotheses surrounding workload:

Hypothesis 4. *Data labellers will experience decreased workload when grid suggestions are enabled.*

Hypothesis 5. *Data labellers will experience decreased workload when auto-orientation is enabled.*

To evaluate these two hypotheses, participants were asked to complete a NASA Task Load Index (TLX) survey, detailed in table A.2, following each experimental condition. The NASA TLX is a well-studied method for evaluation of multiple facets of perceived workload: Mental Demand, Physical Demand, Temporal Demand, Performance (reverse scale), Effort, and Frustration [31]

Additionally, participants responded to subject questions measuring workload and human-machine team fluency according to a five-point Likert response format ("strongly disagree," "weakly disagree," neutral," "weakly agree," and "strongly agree") to gain a greater understanding of workload perception [32]. Three categories of questions were asked: robot teammate traits, working alliance for human-robot teams, and additional measures of team fluency. These questions are enumerated in A.3.

Following completion of all trials, participants were given an open response survey, detailed in table A.4, in order to gain further insight into the tools. Given the small sample size ($n = 4$) of this exploratory pilot study, these responses were especially important for understanding usage and effectiveness of the tools.

5.2 Experimental Design

In this study, participants were asked to label data generated by two robotic systems performing an unloading task, which involved the robot picking items out of a 3-by-4 grid and placing them onto a conveyor belt as shown in Figure B-2. Both systems utilized a UFACTORY xArm 6 robot arm on a Swivellink stand equipped with a proprietary vacuum gripper. One system was outfitted with two Luxonis RGB-depth cameras and was set up to unload 4-by-3 trays of 16.3 oz. Skippy Super Chunk Peanut Butter jars. The other system was outfitted with two Intel RealSense RGB-depth cameras and was set up to unload 4-by-3 trays of 19 oz. Progresso Reduced Sodium Savory Chicken & Wild Rice soup cans. The dual setup was used in order to increase the cognitive workload of the participants by introducing an element of context-switching, as all participants were familiar with the labelling tasks prior to the start of the study.

Products are shown in Figure B-3. Neither of these two products had identifying features on the top face that would allow a labeller to determine object orientation from a top-down robot scan, so tape was placed in the centers of the pick faces of products labelled with either a "P" or "C" (corresponding to "peanut butter" and "chicken soup"). As these tape labels are rotationally asymmetric and are aligned with the product labels, it is possible to determine object orientation from orienting the letters. Trays of both labelled objects are shown in Figures B-4a and B-4b. Soup cans were picked from the bottom due to hardware limitations surrounding the pull tab on the top. The study refers to all components of the software (including the grid inference tool) and hardware collectively as "the robot" for ease of notation.

Utilization of the grid inference tool and auto-orientation tool were modulated in differing trials. As referenced in section 5.1, annotations were evaluated for this study by being used directly as inputs for the two robotic systems. Successful annotations were tallied, and for the three out of five trials that required evaluation of accuracy of orientation, accurate annotations were tallied. An image of the object in the correct orientation was given to the participant to specify target placement orientation.

Five ten-minute trials were conducted with participants to test varying configurations of the two independent variables:

1. **Experiment 1: Baseline.** Participants were asked to label pick and place locations for robot-generated scan data. No specific place orientation was required during this trial.

2. **Experiment 2: Baseline with Orientation Constraint.** Participants were asked to label pick and place locations for robot-generated scan data. A specific place orientation was required during this trial.

3. **Experiment 3: Grid Inference.** Participants were asked to use the grid inference tool to automatically generate multiple picks. Ultimately, participants were able to decide whether to accept the grid suggestions or reject them and generate labels by hand. No specific place orientation was required during this trial.

4. **Experiment 4: Auto-Orientation.** The auto-orientation tool was enabled for this trial. Participants were asked to label pick locations and the place location on the flattened object scan. A specific place orientation was required during this trial.

5. **Experiment 5: Grid Inference + Auto-Orientation.** The auto-orientation tool was enabled for this trial, and participants were asked to additionally generate suggestions using the grid inference tool. Participants were allowed to accept or reject grid suggestions, while orientation was determined by the auto-orientation tool and the annotated flattened object scan. A specific place orientation was required during this trial.

To reduce bias from the participants learning how to effectively use the tools, a random number generator was used to determine the order in which each participant conducted the five trials.

Before their first trial with each tool, participants were given a demonstration on usage. Additionally, prior their first trial using the grid inference tool, participants

were given five minutes to familiarize themselves with the tool due to the complexity of usage.

5.3 Participants and Participant Biases

The exploratory pilot study was run with four participants, all employed at the company. Thus, all participants were familiar with the robotic system and tasks, and all had previously annotated data generated by the robots (though none generated with the two products used in the study). Participants had various amounts of experience with data annotation, and two had seen and briefly used the grid inference tool prior to the study. Responses to the questionnaires may have been biased by a consideration that surveys may be used to evaluate participants' performance at the company. Moreover, participants had an understanding that the grid inference and auto-orientation tools were intended to improve productivity and efficiency and may have been primed to look upon them favorably. As all participants were colleagues with the author, they may have been less inclined to respond negatively, and they may have additionally shared information about the study with each other prior to the conclusion of the study.

While statistically significant conclusions may not be drawn from such a small sample size, the data may be used to indicate general trends and as a window into the effectiveness of the grid inference and auto-orientation tools for increasing efficiency. A full user study would be required to corroborate or disprove the trends found through this preliminary study.

5.4 Experimental Results

5.4.1 Limiting Factors

Due to the nature of running the experiment in the company robot lab test setting, there were many factors that could have a confounding influence on the variables of interest in the study. The network was unstable across approximately three quarters

of all trials. Other employees in the office may have presented distractions moving through the lab space, having conversations, looking for tools, etc. The nature of the robotic system also means that scan data cannot always immediately be generated for the participants to annotate following completion of the current annotation task. As employees of the company, participants were also required to perform a number of complementary tasks during each trial that detracted from performance.

Additionally, a couple of hardware complications acted to reduce possible performance. The RGB-D cameras on the robotic system tasked to unload jars of peanut butter provided inconsistent depth information, affecting the success of the executed robot picks and grid inference. Additionally, due to the ridges on the soup cans and the lack of appropriate smaller grippers, the tolerance of the labelling task was very low for data generated with the cans. This limitation resulted in a constraint that annotations must produce a robot pick within approximately 2cm of the center of a can to be counted as successful. Participants generated approximately 30% more labels for the data generated with peanut butter jars than for the data generated with soup cans.

Data was not fully recorded for Trial 1 with Participant 2, and the trial was omitted from analysis. Participant 2 provided unnecessary manual orientation labels (as requested for Trial 2) during Trial 4, potentially biasing results for that trial.

5.4.2 Performance

Participants' average successful annotations per minute (APM), accurate APM, and inaccurate APM are given in table A.1. On average, participants' successful annotation rate was much higher for the baseline trial than all others at 17.69 APM, almost doubling the second quickest APM. Trial 3: Grid Inference produced the slowest successful annotation rate at 5.96 APM, and the other trial involving the grid inference tool (Trial 5) produced the second slowest successful annotation rate at 7.86 APM. Trials 2: Baseline with Orientation Constraint and 4: Auto-Orientation had similar successful annotation rates, but rate of accurate annotations was higher for Trial 4 (8.88 APM vs. 7.29 APM). Trial 2 also resulted in about 2.24 inaccurate

APM when evaluated on the robot system, while Trials 4 and 5, which utilized the auto-orientation tool, resulted in about 0.35 and 0.42 inaccurate APM, respectively.

5.4.3 Post-Trial Questionnaires

Responses for the NASA TLX survey measuring workload were tallied and averaged. Averages rounded to the nearest integer value are given in Figure B-7. Responses for the post-trial questionnaires were also tallied and averaged according to the mapping 1: strongly disagree; 2: weakly disagree; 3: neutral; 4: weakly agree; 5: strongly agree, except for question 4, which was evaluated on a reverse scale. Average responses across participants are given in figure B-6.

Trials where the grid inference tool were used (Trials 3 and 5) were found to be the most frustrating and require the greatest mental demand. Moreover, these two trials required the most effort to complete, closely followed by the trial requiring manual labelling of product orientations (Trial 2). These three trials were also perceived as more rushed, while Temporal Demand was perceived as lowest during Trial 4. Performance was perceived to be lowest during Trial 3.

5.4.4 Open-Response

Three out of four participants indicated that they preferred to label picks manually over accepting grid inference suggestions. Two of these participants cited irregularity of object grids as a contributing factor. Both products (peanut butter jars and soup cans) were placed in cardboard trays at the pick location, resulting in relatively regular product placement. However, the low error tolerance resulting from the geometry of the cans likely caused minute changes in product placement to have a disproportionate effect on success of annotations. Moreover, participants preferred the certainty of annotating pick locations by hand rather than relying on an uncertain automated aide.

Interestingly, the two participants with less experience labelling data were more open to using the grid inference tool than the two participants with more experience

labelling data. A likely explanation would be that the two with more experience have had more practice labelling by hand and are thus more comfortable with the manual workflow. This familiarity would be especially relevant as the controls for using the grid inference tool are significantly more involved than those for manual labelling, resulting in interacting with the grid inference tool "taking up more time." Grid inference performance also suffers when depth information is incomplete, and one of the robotic systems provided scans with significant amounts of invalid depth. Two participants indicated that they were less inclined to use the grid inference tool for scans with poor depth information.

Participant 2's response to question 3 was unrelated to the prompt and did not respond to question 4, so their data for both questions were not included in analysis. All other participants showed significant preference for the auto-orientation tool, one responding that the tool "was extremely helpful with cylindrical object[s] that [have] little to no visual indicator of the orientation from the top view." Two participants noted that the time required for the robot to scan the object to orient it during the evaluation phase slowed down its performance.

5.5 Evaluation

The findings of the exploratory study contradict **Hypotheses 1** and **2** as the successful annotation per minute APM of the Baseline trial was much higher than that of the Grid Inference trial, and the APM of the Baseline with Orientation Constraint trial was also higher than that of the Auto-Orientation trial. However, the APM of Trial 2 was only around 3 percent faster than that of Trial 4, and the rate of accurate APM was greater for both Trials 4 and 5 compared to the baseline of Trial 2. The error rate of annotations was greater than five times higher for Trial 2 than Trials 4 and 5, supporting **Hypothesis 3**. These findings suggest that, even with the tradeoff of the robot requiring additional time to perform the partial scan for auto-orientation during robot runtime, the auto-orientation tool is effective for increasing the rate of accurate annotations.

The significant increase in rate of the baseline trial compared to all other trials can be partially attributed to participants' familiarity with generating annotations through this workflow and unfamiliarity with all other workflows. Moreover, current engineering efforts have targeted the baseline workflow, whereas both grid inference and auto-orientation tools are experimental, potentially resulting in additional bias. Nonetheless, the resulting APM metrics suggest that both the grid inference and auto-orientation tools slow the labelling process. Further testing is necessary to confirm the factors that contribute to the lowered rate.

The post-trial NASA TLX survey ratings contradict **Hypothesis 4** and **Hypothesis 5**, as participants experienced increased workload when the grid inference tool and/or the auto-orientation tool were turned on. Moreover, the results suggest that the increase in participant workload was significant for the grid inference tool. 20 controls exist for interacting with the tool, likely resulting in a large learning curve that was not entirely addressed by the five minutes participants received before the first grid inference trial. Moreover, a combination of imperfect performance and lack of ability to easily modify individual labels (ie. deleting a label could only be achieved through cycling through all labels until the desired label was selected or deleting the entire set of labels) likely contributed to increased workload.

Participants' higher average workload rating for the auto-orientation tool when compared to the baseline trial likely is influenced at least in part by unfamiliarity, as comfort and understanding of the robot (Questions 4-5) were rated poorer in trials utilizing the two tools as opposed to the baseline. This rating likely is also affected by the previous experience of the participants in performing data labelling tasks. However, the lower rating of Temporal Demand for the auto-orientation tool trial compared to the baseline suggests that the tool may be helpful in increasing APM rate once labellers become comfortable with the tool.

5.6 Recommendations

Through the conduction of the exploratory pilot study, a number of recommendations to the company have been identified.

5.6.1 Grid Inference

Participants' main concerns about grid inference centered around flexibility and user agency. For example, several participants noted that more flexible controls in interacting with the grid generation feature would have improved the usage experience and annotation speed. The following are core recommendations to improve the grid generation user experience:

- Automatically save individual labels inputted to the grid inference tools. Currently, labellers can choose to directly label a scan or provide labels to the grid inference tool. This creates a cost for using the grid generation tool if the suggested grid labels are incorrect; the time spent annotating labels for the grid inference tool will have produced no viable labels. Instead, a mode to annotate as usual while sending the same annotations to the grid tool would incur no additional cost while generating a potentially time-saving grid of labels.

- Improved interface for deletion of individual labels in the generated grid. Participants expressed frustration that deletion was only possible by cycling through generated labels until the desired one was reached or deleting all suggested labels at once.

- Ability to modify individual points or sections of labels in the grid structure. Due to a variety of factors, the target objects may not be presented in a perfect geometric array. Thus, a fast and efficient method of rectifying the regularly-arrayed grid labels to account for this imperfection would be to allow movement of groups of suggested labels independently of the rest of the suggestions.

- Opinions were divided on the effectiveness of the current mapping of keyboard controls to grid adjustment actions. Some participants indicated they would be

"more inclined to use gridding if the [keyboard] control / layout was improved," while others indicated they enjoyed using the current controls. Further user testing on the user experience of gridding may be necessary to develop an intuitive, effective interface.

- Invalid depths, as shown in figure B-5b, often caused poor grid suggestions. Another strategy for fitting a plane may be necessary to mitigate this effect (or alternatively, further investigation into camera hardware solutions). For example, it may be more effective to first segment out the product in the RGB image, and then employ a method such as RANSAC to fit a plane to the tops of the products using only valid depths.

In its current state, the use of the grid inference tool carries a high workload likely extending beyond a high workload generated by the learning process, especially because objects in the real world are often not presented in perfectly geometric arrays. This high workload is likely a driver of decreased rate and thus should be addressed before the tool is integrated into the annotation workflow. Future usage of the grid inference tool would benefit from improvements to robustness and ease of use.

5.6.2 Object Registration and Automatic Orientation

In general, the auto-orientation tool was found by study participants to be helpful and produce approximately the same amount of workload as the baseline. While usage of the tool did not increase the rate of successful APM, it increased the rate of accurate APM, generating fewer inaccurate annotations. However, robot performance was noted to be slower than that of the baseline during the validation execution step. The following are core recommendations to improve auto-orientation:

- Increasing speed of robot execution. Because the robot pauses to take a scan of the grasped object in a predetermined joint location, it is unable to pick and place objects as quickly as in the Baseline trial. While it may be reasonable to expect a decrease in pick-place rate when there is an orientation requirement, steps may be taken to speed up the motions of the robot.

  - Dynamic calculation of scan location. A scan joint location that results in little deviation from the planned path from pick to place joints can be calculated for each object pick / place pair. A straighter path should allow for an increased robot pick / place rate.
  - The pausing of the robot to save a partial scan of the object results in a reduced robot pick rate. The effect of auto-orientation on pick rate could be reduced by developing a method for the robot to take a scan while still moving toward the place location. Because it is necessary to mitigate blur on the scan in order to extract viable features, a scan protocol could be constructed where, for example, the robot slows down while taking the scan, then speeds back up after the scan has been taken.

- Larger flattened scan representation. Multiple participants mentioned that the flattened scan representation was "hard to see" at its current scale. A straightforward solution would be to increase the size of the image.

- Communication of the desired orientation of the product. Participants requested multiple confirmations of the desired product placement and time to view and understand product visuals. While both requests were possible to address in this controlled user study, in a practical application, labellers would not receive support beyond the singular image showing requested product orientation. One possible solution would be to show a scan of all sides of the object along with multiple images of the object placed in the correct orientation taken from varying angles in order to provide as much context as possible.

Overall, auto-orientation was found to be a useful tool. Notably, in this exploratory study, labels were given to visually distinguish orientation from a top-down view; this laboratory labelling step would not be available in real-world settings. Thus, this tool will be especially helpful in enabling robots to perform pick-and-place tasks for objects with little to no visual indicators from a top-down view. Even for tasks involving objects with rotationally-asymmetric identifying features on the top face, such as for printed boxes, the feature can help to increase accuracy and decrease

the temporal workload of labellers, potentially increasing productivity. The most hindering factor to utilization would be the slower speed at which it requires the robot to run. Once addressed, this feature would generate value integrated into the robotic systems.

Chapter 6

Discussion

6.1 Recommendations for System Designers

The exploratory pilot study uncovered the influence of the ease of user interaction, which appears to be equally important as reliability, on usage and efficacy of tooling. This suggest that when designing and developing features for robotic systems that incorporate an element of human interaction, system designers should thoroughly study and iterate upon the human experience in order to produce the most effective tooling. Moreover, familiarity with a particular workflow appears to confer a large advantage, the effects of which should not be discounted. A centralized training program may also be beneficial for tools with more involved usage.

6.2 Limitations

The assumptions made by the grid inference tool regarding dimensionality of the grid structure and reduction to problems involving arrays with orthogonal axes, while applicable to many factory settings, are limiting nontheless. Moreover, the internal grid representation does not take into account that the width and length of objects may be significantly different when calculating step size and could be modified to do so. The high-pass filtering step currently may result in a step size that is half the ground truth if an object has one dimension that is larger than twice the other

dimension.

Auto-orientation is limited by lighting conditions and scan quality; these methods are not robust to environmental changes such as sunlight hitting a workspace or dust blocking the cameras. These limitations could be mitigated by a controlling the environment of the robot systems.

The exploratory pilot study was affected by many external factors and was conducted with a small set of participants, so the data cannot be used for extensive analysis. Accuracy was evaluated by hand and is thus susceptible to human error.

6.3 Future Work

Future exploration of the grid inference tool includes extrapolating to a 3D array of objects, such as a full pallet of objects in a warehouse. Additionally, an auto-matching feature that employs computer vision methods such as template matching or deep segmentation in order to automatically match and move grid-generated labels to objects in a scan image could potentially result in large performance gains.

For auto-orientation, future work could include 3D automatic orientation, for example for bin picking tasks where objects are not guaranteed to be upright. Optimization for computational efficiency could also be performed.

6.3.1 Full User Study

The exploratory pilot study conducted as part of this work points towards promising methods for increasing productivity of human data labellers. As a major limitation of the exploratory pilot study was the small and biased set of participants, further evaluation conducted through a full user study would be valuable, allowing for more accurate evaluation of the grid inference and auto-orientation tools and their ability to increase efficiency and productivity of labelling. It would additionally be valuable to study the requirements and challenges to learning how to interact with and use the tools. Furthermore, this expanded study could explore more varied scenarios, including data generated by robot systems picking different geometric shapes, and

data generated by more than two systems. Analysis of the human labeller's workload and affected ability to context-switch could help to inform system design for further efficiency improvements.

6.4 Conclusions

The grid inference and auto-orientation tools show promise for aiding a human labeller in efficiently annotating data, especially in situations where the human worker may have insufficient information to correctly generate annotations. Further improvements are necessary for both to increase utility for the Tutor Intelligence robotic systems. A full user study is recommended to fully understand the usage and impacts of these tools.

Appendix A

Tables

Table A.1: Average annotations per minute (APM) across participants, where a successful annotation results in a pick and subsequent place (pick-place) of an object, an accurate annotation results in a pick-place of an object oriented within ±30 degrees of the requested orientation, and an inaccurate annotation results in a pick-place of an object oriented more than ±30 degrees differently compared to the requested orientation. Annotations per minute are rounded to two decimal places.

Trial	Successful APM	Accurate APM	Inaccurate APM
1: Baseline	17.69	N/A	N/A
2: Baseline with Orientation Constraint	9.54	7.29	2.24
3: Grid Inference	5.96	N/A	N/A
4: Auto-Orientation	9.24	8.88	0.35
5: Grid Inference + Auto-Orientation	7.86	7.44	0.42

Table A.2: Task Load Index (TLX) - Post-Trial Questionnaire. Evaluated on a 21-point scale from 0: Very Low, to 20: Very High.

1. Mental Demand: How mentally demanding was the task? (Very Low: 0 - Very High: 20)
2. Physical Demand: How physically demanding was the task? (Very Low: 0 - Very High: 20)
3. Temporal Demand: How hurried or rushed was the pace of the task?(Very Low: 0 - Very High: 20)
4. Performance: How successful were you in accomplishing what you were asked to do? (Perfect: 0 - Failure: 20)
5. Effort: How hard did you have to work to accomplish your level of performance? (Very Low: 0 - Very High: 20)
6. Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you? (Very Low: 0 - Very High: 20)

Table A.3: Subjective Measures - Post-Trial Questionnaire. Questions 1-3 pertain to robot teammate traits, questions 4-5 pertain to the working alliance for human-robot teams, and questions 6-10 pertain to additional measures of team fluency.

Robot teammate traits	1. The robot was intelligent. 2. The robot was trustworthy. 3. The robot was committed to the task.
Working alliance	4. I feel uncomfortable with the robot (reverse scale). 5. The robot and I understand each other.
Additional measures ‘	6. I was satisfied by the productivity of the robot. 7. I would work with the robot the next time the tasks were to be completed. 8. The robot’s intelligence increased the productivity of the team. 9. The robot’s intelligence allows me to do less work. 10. The robot and I worked as quickly as possible.

Table A.4: Subjective Measures - Post-Study Open-Response

1. When the products were not required to be placed in a certain orientation, which of the scenarios did you prefer, and why?
2. When the products were required to be placed in a certain orientation, which of the scenarios did you prefer, and why?
3. If you were going to add a robotic assistant to a manufacturing team, which setup (baseline, grid inference, auto-orientation, grid inference + auto-orientation) would you use, and why?
4. What aspects of grid inference and auto-orientation made you more or less inclined to use each feature?
5. How useful was the circular display of the product for auto-orientation? How could the display be more helpful?

Appendix B

Figures

Figure B-1: Person moving collaborative Tutor Intelligence robot to the correct position.

Figure B-2: Exploratory pilot study test setup with robot system, product, and conveyor belt.

Figure B-3: Exploratory pilot study test products: soup can and peanut butter jar.

(a) 3x4 tray of peanut butter with labelled lids.

(b) 3x4 tray of upside-down soup cans with labelled bottom faces.

Figure B-4: 3x4 trays of products used for the conveyor unloading task in the exploratory pilot study from a top-down view. Tape labels are aligned to product labels; orientations of the products are varied.

(a) Top-down product scan, RGB image only

(b) Top-down product scan, RGB image masked by valid depths

Figure B-5: Two example scans of target pick products from a RGB-D camera, (a) showing the original RGB scan, and (b) showing the RGB scan masked by valid depths. Significant amounts of invalid depths are returned by the camera in this example of a "poor" scan.

Figure B-6: Likert Scale responses to post-trial surveys averaged across participants, grouped by question category. Questions are listed in table A.3

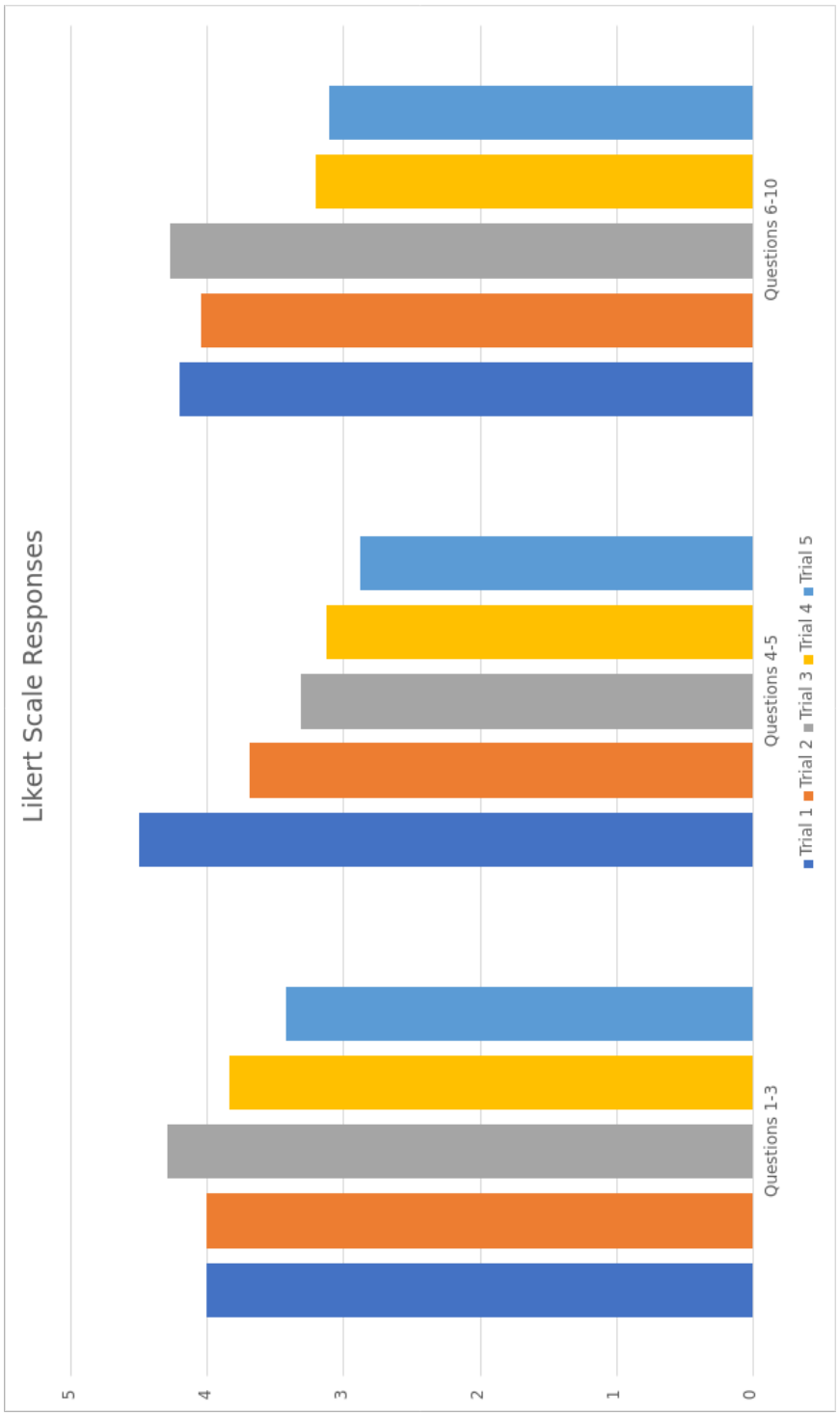
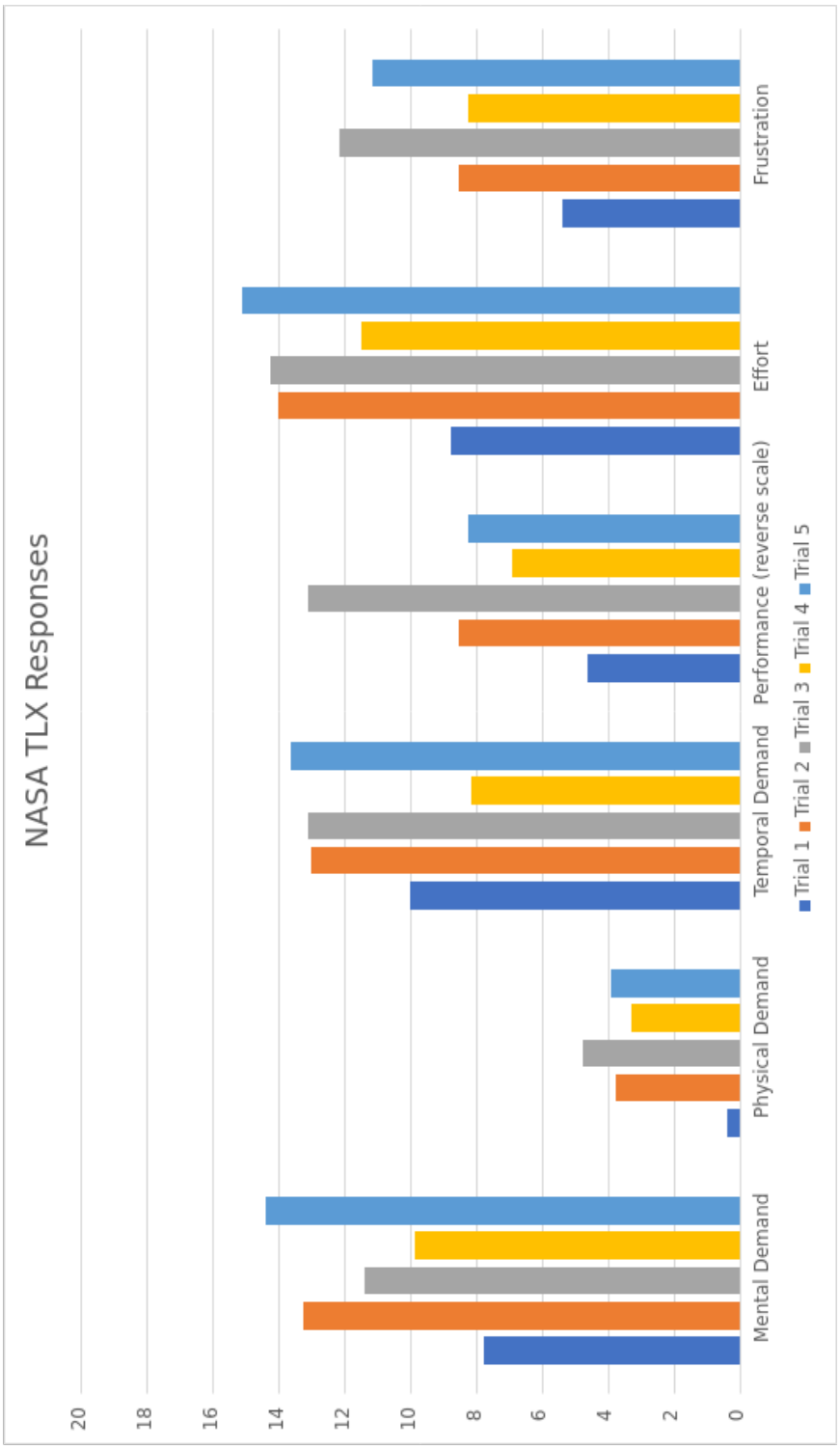


Figure B-7: Responses to post-trial NASA TLX workload surveys averaged across participants.



Bibliography

[1] Mikkel Knudsen and Jari Kaivo-oja. Collaborative robots: Frontiers of current literature. *Journal of Intelligent Systems: Theory and Applications*, 3:13–20, 06 2020.

[2] F. Sherwani, Muhammad Mujtaba Asad, and B.S.K.K. Ibrahim. Collaborative robots and industrial revolution 4.0 (ir 4.0). In *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pages 1–5, 2020.

[3] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R. Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, Nima Fazeli, Ferran Alet, Nikhil Chavan Dafle, Rachel Holladay, Isabella Morona, Prem Qu Nair, Druck Green, Ian Taylor, Weber Liu, Thomas Funkhouser, and Alberto Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7):690–705, 2022.

[4] Massachusetts Manufacturing Accelerate Program (MMAP). `https://cam.masstech.org/mmap`, 2023. Accessed on May 12, 2023.

[5] Tutor Intelligence. `https://www.tutorintelligence.com/`, 2023. Accessed on May 12, 2023.

[6] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 35(6), dec 2016.

[7] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. We don't need no bounding-boxes: Training object class detectors using only human verification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 854–863, 2016.

[8] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6D pose estimation in the Amazon picking challenge. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1386–1383, 2017.

[9] Daniele De Gregorio, Alessio Tonioni, Gianluca Palli, and Luigi Di Stefano. Semiautomatic labeling for deep learning in robotics. *IEEE Transactions on Automation Science and Engineering*, 17(2):611–620, 2020.

[10] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19:297–301, 1965.

[11] E. O. Brigham and R. E. Morrow. The fast Fourier transform. *IEEE Spectrum*, 4(12):63–70, 1967.

[12] Amir Atapour-Abarghouei, Gregoire Payen de La Garanderie, and Toby P. Breckon. Back to Butterworth - a Fourier basis for 3D surface relief hole filling within RGB-D imagery. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2813–2818, 2016.

[13] Andreas Richtsfeld, Thomas Mörwald, Johann Prankl, Michael Zillich, and Markus Vincze. Learning of perceptual grouping for object segmentation on RGB-D data. *J. Vis. Comun. Image Represent.*, 25(1):64–73, jan 2014.

[14] Massimo Zanetti and Lorenzo Bruzzone. Edge-crease detection and surface reconstruction from point clouds using a second-order variational model. volume 9244, 09 2014.

[15] Natesh Srinivasan, Luca Carlone, and Frank Dellaert. Structural symmetries from motion for scene reconstruction and understanding. pages 136.1–136.13, 01 2015.

[16] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[17] Fang Wang and Zijian Zhao. A survey of iterative closest point algorithm. In *2017 Chinese Automation Congress (CAC)*, pages 4395–4399, 2017.

[18] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 143–152, 2017.

[19] Siyu Ren, Xiaodong Chen, Huaiyu Cai, Yi Wang, Haitao Liang, and Haotian Li. Color point cloud registration algorithm based on hue. *Applied Sciences*, 11(12), 2021.

[20] Zhiming Chen, Kean Chen, Weiyao Lin, John See, Hui Yu, Yan Ke, and Cong Yang. Piou loss: Towards accurate oriented object detection in complex environments. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 195–211, Cham, 2020. Springer International Publishing.

[21] Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022.

[22] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3DNet: 3D object detection using hybrid geometric primitives. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 311–329, Cham, 2020. Springer International Publishing.

[23] Yu Zhang, Junle Yu, Xiaolin Huang, Wenhui Zhou, and Ji Hou. PCR-CG: Point cloud registration via deep explicit color and geometry. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, page 443–459, Berlin, Heidelberg, 2022. Springer-Verlag.

[24] Mohamed El Banani and Justin Johnson. Bootstrap your own correspondences. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6413–6422, 2021.

[25] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[26] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

[27] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, nov 2004.

[28] Oleksandr Bailo, Francois Rameau, Kyungdon Joo, Jinsun Park, Oleksandr Bogdan, and In So Kweon. Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution. *Pattern Recognition Letters*, 106:53–60, 2018.

[29] Jim Lawrence, Javier Bernal, and Christoph Witzgall. A purely algebraic justification of the Kabsch-Umeyama algorithm. *Journal of research of the National Institute of Standards and Technology*, 124:1–6, 2019.

[30] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981.

[31] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland, 1988.

[32] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):5–55, 1932.