The Thesis Committee for Adam David Allevato
Certifies that this is the approved version of the following thesis:

An Object Recognition and Pose Estimation Library for Intelligent Industrial Automation

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor: _____
Sheldon Landsberger

Co-Supervisor: _____
Mitch Pryor

An Object Recognition and Pose Estimation Library for Intelligent Industrial Automation

by

Adam David Allevato, B.S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2016

Dedication

To my parents, David and Mary Lou.

Acknowledgements

I would like to thank Mitch Pryor for his unending support of my research, from offering me a position late in the term, all the way to reading this thesis. A heartfelt thanks goes out to Sheldon Landsberger for his diligence in helping to make our laboratory a world-class institution. And, to my fellow lab researchers: thank you for all the laughs, thoughtful discussion, and research assistance.

I would also like to thank the Lord, as well as my entire family, for support and guidance. They have been with me every step of the way. And finally, thanks to all the teachers, whether professors or not, who have helped me throughout my academic pursuits.

This material is based upon work supported under an Integrated University Program Graduate Fellowship.

Abstract

An Object Recognition and Pose Estimation Library for Intelligent Industrial Automation

Adam David Allevato, M.S.E.

The University of Texas at Austin, 2016

Supervisor: Sheldon Landsberger

Co-Supervisor: Mitch Pryor

The nuclear-industrial complex is a field characterized by hazardous environments and stringent worker health regulations. Automation is one of the best ways to improve worker health, but many of the work-intensive tasks in the nuclear industry are difficult to automate using rigid industrial manipulators, which are often treated as glorified assembly lines. This thesis presents the idea of intelligent industrial automation, or IIA, as a way to implement automation in diverse and uncertain environments, and shows that robust computer vision is a key technology in achieving deployable IIA. Furthermore, with recent advances in the field of computer vision, including machine-learning based techniques, the time is better than ever for groups such as the Department of Energy (DOE) to implement computer vision and IIA in their processes. A modular software framework for object recognition and pose estimation (ORP) is developed and incorporated into three laboratory demonstrations, each of which represents a different capability relevant to DOE. By using well-proven computer vision techniques and libraries, ORP enables robust task completion

in domains that would have previously been impossible without human supervision or custom mechanical designs (such as task-specific fixtures). A vision-enabled manipulation system is shown to reliably pick and place small weapon detonator components 98% of the time, making it an ideal candidate for machine tending. A remote inspection and inventory system shows the ability to visually detect the position of nuclear material storage canisters with a standard deviation under 1 mm, allowing it to detect cans that have been moved or tampered with. Finally, using vision, an automated glovebox mixed-waste sorting system is able to sort small objects, which begin in a random configuration, into three containers based on their color (a surrogate for radiation signature) with 94.6% accuracy. All three demonstrations proceed autonomously, suggesting that implementing IIA can result in significant improvements in worker safety and productivity at DOE complex sites.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

As science has uncovered the dangers of radioactivity, safety regulations have swept over the nuclear-industrial complex, restricting allowable radiation doses at the expense of productivity. Government regulation has become especially restrictive, with safeguards in place for everything from beryllium dust exposure to repetitive stress injuries. While these safety rules are effective, they hinder manufacturing and technology adoption at DOE laboratories. According to Sig Hecker, the director of DOE's Los Alamos National Laboratory (LANL), the restrictive oversight of DOE "has diminished [LANL's] scientific quality and productivity" [1]. Occupational radiation exposure (ORE) limits are a prime example of restrictive regulation—LANL's regulations limit a worker's ORE to just 5 rem per year [2].

According to LANL engineers, deploying automation systems is one justifiable way to reduce ORE [3]. Automation solutions can help increase productivity while also meeting regulatory safety requirements, but these systems face their own challenges, and one of the largest of these challenges is the need for task-specific fixtures and tooling that rigidly constrain tasks and make them feasible for unintelligent manipulators to complete. With today's accessible and modular software, computer vision-based techniques are available for widespread deployment, allowing costly fixtures to be replaced by sensing and system intelligence. However, DOE facilities have not yet deployed these solutions due to lack of system verification and incomplete understanding of the full capabilities of computer vision. This chapter explores why the DOE should use flexible automation technology, and more specifically computer vision, to supplant task-specific fixtures and automation, as well as the benefits that will come about after adopting such technologies.

1.1 MOTIVATION

Traditional approaches to automation in the nuclear complex have been restricted to the use of rigid robots and control architectures. Companies such as Yaskawa Motoman produce modern industrial robots, such as the SIA5 (see Figure 1-1), for performance and precision. This robot can achieve a position tolerance of +/- 0.05 mm [4], and in our lab, the University of Texas at Austin Nuclear Robotics Group (NRG), we have shown that the SIA5 can thread a needle repeatedly using teleoperation [5]. To achieve this level of precision, industrial manipulators must be rigid machines that do not have the ability to automatically adapt to physical environment changes.

Physical rigidity does not preclude safety. Newer industrial systems such as the Universal Robots UR5 provide precise position control while also having built-in torque and current monitoring to automatically detect anomalies and safely stop before causing damage. Eighty percent of Universal Robots' systems are deployed without safety cages, attesting to their safety architecture's effectiveness [6]. Industrial robots such as the UR5 are safer than ever before, but they still cannot automatically adapt to slight changes in the physical environment, such as calibration or alignment errors. One way to combat this issue is to follow the recent trend towards compliant systems, such as the Baxter collaborative robot [7]. By using series elastic actuators and back-drivable joints that can deviate from their commanded positions, robots like Baxter can safely operate around humans while also allowing for slight inaccuracies when interacting with the physical world. Unfortunately, compliant systems come at the cost of a large reduction in precision and strength, and somewhat ironically, the inherent variability of these systems makes them unsuited for use in DOE mission-critical applications. High-consequence procedures require maximum stability and repeatability, which uncovers a largely unaddressed dilemma: how can robots repeatedly perform precise actions in imprecise environments?

Figure 1-1: Examples of the two types of robot on the market today. Left: Universal Robots UR5, a 6-degree of freedom industrial manipulator [8]. Right: Baxter, a dual-arm collaborative robot platform that uses series elastic actuators [7].

Safety-critical tasks such as those usually found in nuclear materials handling require rigid industrial systems, but engineers often assume that a rigid system must have a correspondingly rigid control architecture. By constraining software to be as restrictive as the hardware, we throw out potential intelligence and adaptability that can make a system far more useful.

The Advanced Recovery and Integrated Extraction System (ARIES), created by Los Alamos National Laboratory (LANL), is a good example of task rigidity. ARIES is an automated glovebox designed for plutonium weapon component dismantlement. This overall goal consists of subtasks such as inspecting welds on nuclear material storage containers and transporting materials around a glovebox using a gantry system [9]. ARIES's different subcomponents include both custom and off-the-shelf rigid industrial systems like the SIA5, and therefore the system is well equipped to complete high-precision tasks. In a similar manner, the tasks that ARIES performs are extremely well-

defined, with written procedures and tight tolerances for critical metrics such as weld thickness. The samples in the glovebox can only move along certain axes and the system performs the same operations on each weapons component. ARIES is essentially a custom assembly line, which is a decidedly unintelligent system, and yet it is often used as a premier example of successful automation at LANL [3].

ARIES was first deployed 20 years ago, and development has continued since that time, suggesting that the DOE has already solved the glovebox automation problem. But the system by design only handles one type of component—plutonium weapons pits. In addition, it uses custom tooling and fixtures to strictly define the task being performed, positioning objects in the area to be exactly where the robot expects them. According to LANL engineers, "ARIES automation systems lack significant intelligence" [3].

Industrial systems and intelligent task completion are not mutually exclusive. Figure 1-2 shows that robots have a range of both hardware rigidity and software adaptability. Stakeholders often make the unintended (and incorrect) assumption that industrial robots must be repetitive, falling in the lower right quadrant of the figure, and that the only place for adaptive or intelligent control is in the sphere of compliant robots or other non-critical systems. But, in actuality, we need to deploy systems that use adaptive, intelligent control while still maintaining the benefits offered by industrial systems. Unfortunately, the nuclear-industrial complex contains almost none of this *Intelligent Industrial Automation*, or IIA.

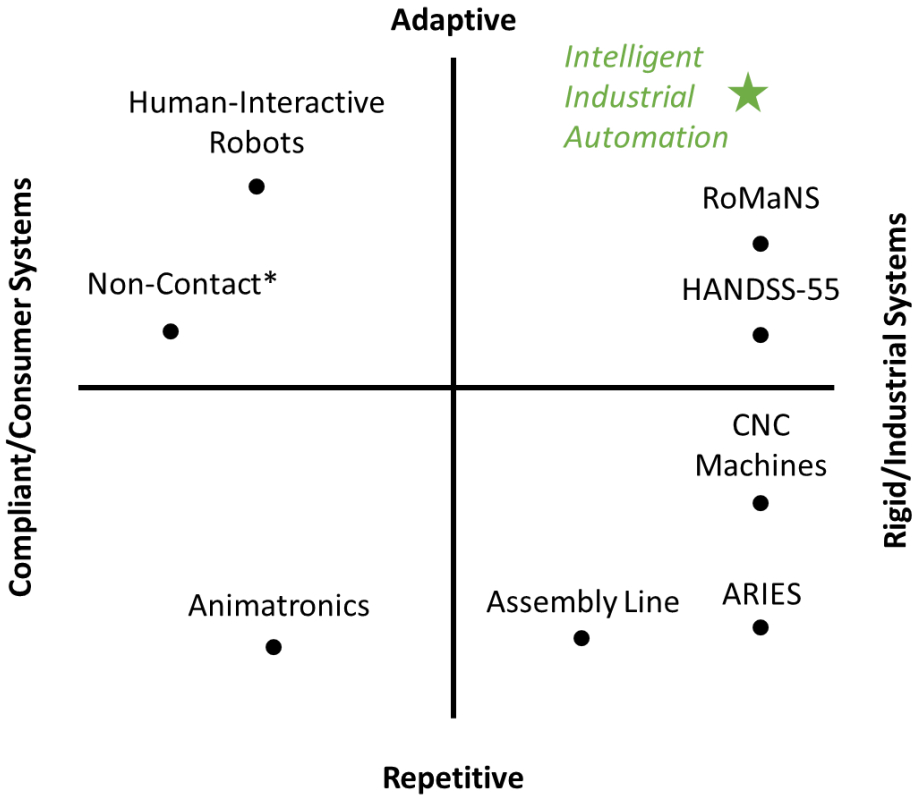

Figure 1-2: A comparison of hardware and software adaptability across different automation products. This chapter discusses a select subset of the systems shown here.

1.2 Technical Challenges in Intelligent Industrial Automation

IIA promises broad applicability and great gains in worker productivity and safety. So why is IIA rare in the DOE complex? DOE labs such as LANL easily innovate in fields such as theoretical science, but adopt new technology from other fields relatively slowly. A number of factors prohibit technology adoption in the DOE complex. Problems range from technical, to administrative, to social. These factors include:

- **Task Difficulty**: While robotics capabilities have grown greatly in the last five years, many tasks, especially open-ended ones such as waste management and cleanup, still remain challenging to automate. In addition, many procedures and

processes used in the nuclear industry were developed in the 1960s and earlier, when the field of robotics was just developing. As a result, even simple processes, such as machining, have not been designed with automation in mind.

- **Task Specificity**: Some of the tasks required in nuclear manufacturing are not encountered anywhere else in the world. LANL cannot devote the resources to develop specific automation solutions for each task in their pipeline, and because of the specificity of the tasks, commercial robotics companies have even fewer incentives.
- **Safety/Regulatory Requirements**: The nuclear industry highly safeguards both materials and processes to prevent disasters and nuclear weapons proliferation. Meeting restrictive regulatory and safety requirements hampers adoption of technologies (such as intelligent robotics) that have not yet been used in the DOE complex.
- **High Institutional Knowledge**: The domain and task specificity in the nuclear industry results in a workforce of experts in niche fields, but discourages a broader engineering or research focus. LANL's workforce is not well-versed in automation, and therefore does not have the infrastructure in place to easily incorporate robotics into its procedures.
- **Trust**: Many people, including those in the engineering and science disciplines, do not trust robots to perform high-stakes tasks such as weapons manufacturing, regardless of whatever safeguards are in place.

While the nuclear industry must overcome all of these challenges to implement IIA, the efforts in this thesis specifically target Task Specificity and Task Difficulty. Five recent robotics innovations, all of which are actively being researched at NRG, can help overcome these two challenges:

- **Machine Learning**: By learning from data and demonstrations, robots can adapt prior knowledge to new tasks.
- **High-Level Goal Planning**: Work in environment modeling and task planning has resulted in robots capable of performing more complex tasks and recovering from unforeseen errors.
- **Intelligent Control**: Low-level planning such as software-based compliant control allows systems to complete nuanced tasks, such as responding to contact forces or gently inserting a peg in a hole.
- **Human-Machine Interfaces**: Novel interfaces such as speech control, virtual/augmented reality, and natural teleoperation provide faster, more intuitive, or more ergonomic ways to interact with and control automated systems.
- **Computer vision**: 3D cameras, coupled with intelligent reconstruction algorithms, have opened entire new fields for robots by allowing them to more accurately perceive and characterize their environments.

This thesis focuses on the area of computer vision, which is arguably the newest of these five developments, having only begun to see significant, commercializable innovation since 2010. Although the DOE has occasionally tried to adopt more automation technology, it still lags behind most industries, especially in adoption of computer vision.

Previous efforts by the DOE attempted to move beyond ARIES into automating other critical tasks such as radioactive waste sorting. In 2002, the DOE began development of the Handling and Segregating System for 55-gallon drums of mixed waste, or HANDSS-55 [10]. The DOE Robotics Crosscutting Program designed HANDSS-55 to unpack, characterize, and sort the waste in 55-gallon drums, repackaging the appropriate waste for storage at a long-term storage site such as New Mexico's Waste Isolation Pilot Plant (WIPP). HANDSS-55's design even included basic vision systems, allowing operators to

direct the system to perform basic pick and place operations on objects on a conveyer belt [11]. Originally targeted for a 2005 deployment at Savannah River Site (SRS), the discontinuation of the DOE Robotics Crosscutting Program in 2002 halted the project, showing that technology adoption and trust were not yet strong enough to overcome administrative direction.

HANDSS-55 was a project ahead of its time. 15 years later, DOE still has not implemented IIA for waste sorting at SRS or elsewhere, and still has not incorporated vision-based manipulation into its waste management pipelines. Seemingly in direct reference to this problem, the ARIES report referenced above points out that next-generation robotics at LANL will need to implement "on-demand system models" and real-time recognition of the environment state [3]. DOE must bridge many gaps to turn systems like ARIES into IIA systems, and one of the largest gaps is that of advanced sensing. With years more research and development in perception than when HANDSS-55 was designed, DOE is in a better position than ever before to benefit from computer vision-enabled automation systems.

1.3 SIGNIFICANCE

The primary benefit of a flexible computer vision system is greater task flexibility for manipulators, ultimately leading to easier deployment in a variety of fixture-free tasks. By using computer vision and other environmental sensors, systems can reduce their reliance on rigid, task-specific fixtures and tooling, such as the custom end-effectors required for current tasks at LANL.

From a human factors perspective, increasing automation flexibility improves worker safety. The occupational radiation exposure is the most important safety concern in nuclear processes performed by humans [3], and by deploying automated manipulators

wherever possible, we can reduce this exposure as much as possible. This is in line with DOE's policy of keeping radiation levels As Low As Reasonably Achievable, known as the ALARA principle [12]. There is no better way to reduce human radiation exposure than to remove humans from the vicinity of radioactive material entirely. To this end, we must have robots that can perform a wide array of tasks.

In addition, reducing the numbers of required fixtures can assist other robotics research efforts by decreasing experiment development cycle time, allowing robots to easily work in unconstrained environments.

As a final benefit, robots with vision can perform an entirely different class of tasks, ones where fixtures are impossible to implement and traditional robots such as ARIES would be useless. For example, only vision-enabled systems would be able to dump out a jar of rags contaminated with radiation, pick up each rag individually, and sort them according to radioactivity type; this is exactly what HANDSS-55 was designed to accomplish. It is hard to imagine a fixture that would allow a conventional "blind" robot to complete such a task effectively and safely. This fact has not gone unnoticed by researchers in the European Union (EU), who face the problem of decommissioning parts of the United Kingdom's Sellafield nuclear site. To this end, a conglomeration of five EU research organizations is building the Robotic Manipulation for Nuclear Sort and Segregation, or RoMaNS [13]. RoMaNS, a mockup of which is shown in Figure 1-3, is designed to be a "beyond state-of-the-art" system (see Figure 1-2) to cut open, investigate, and sort legacy waste containers. By funding RoMaNS through the European Commission's Horizon 2020 innovation program, the EU has shown that it recognizes the need for IIA. Now is the time for DOE to do the same.

Figure 1-3: The RoMaNS segregation and sorting system testbed [14].

1.4 OBJECTIVES

The research in this thesis aims to develop a system able to overcome the strict positioning requirements of existing automation systems, with the end result of reducing the amount of tooling and fixtures required to complete tasks common in a nuclear manufacturing environment. A system capable of eliminating the need for fixtures must be able to accurately determine the identity, position, and orientation of objects in the workspace and determine the proper way to manipulate them given their current pose. This overall criterion includes minimum levels of accuracy, precision, and robustness to different environments and tasks, as well as the requirement that the automation system is efficient enough to complete tasks quickly enough to be useful in at DOE complex sites such as LANL.

To meet the criteria, a software library for object recognition and pose estimation (dubbed "ORP") has been developed and incorporated into multiple hardware demonstrations. By maintaining several different vision algorithms in parallel and using each technique where appropriate, the library can correctly identify and locate a wide variety of objects, regardless of their position or orientation. As a result, even complex tasks do not require prohibitive workspace restrictions or fixtures with tight tolerances. To show technology readiness, this thesis includes three demonstrations directly based on DOE needs in the topic areas of inspection, manipulation, and waste handling. The thesis evaluates ORP's success in these demonstrations as a way to analyze the effectiveness of computer vision in fixture-free manipulation.

As a corollary, this thesis also seeks to show that computer vision is ready to be implemented in DOE applications. This conclusion is based on the large number of successes that computer vision has seen recently due to advanced research, novel industrial automation tasks that would be impossible without reliable vision, and ORP's success in DOE-related tasks as shown in this thesis.

1.5 Organization of the Thesis

Chapter 2 contains a critical review of past and current solutions for accounting for unconstrained environments, many of which involve computer vision. The strengths and weaknesses of different approaches, combined with the specific tasks we seek to automate, help define the design of the ORP computer vision system, which is presented in Chapter 3. Chapters 4 through 6 describes how ORP assists in three different automation tasks at UT Austin's Nuclear and Applied Robotics Group (NRG), and contain quantitative results showing task performance when the vision system is utilized. Finally, Chapter 7 draws conclusions from both quantitative and qualitative data, evaluating the new vision system

based on the aforementioned criteria and determining the remaining advances necessary to meet the high-level objective of completing fixture free manufacturing processes in a DOE environment.

Chapter 2: Literature Review

This chapter begins with a brief history of solutions for object recognition and pose estimation, which have both been active research areas since the birth of computer vision. After reviewing several older methods, some of which are still in use today, current state-of-the-art machine learning-based techniques will be presented and discussed. Finally, the chapter will close with an overview of how computer vision has already been incorporated into some unconstrained industrial tasks.

Note: computer vision is historically a field rife with complicated algorithms. Readers are encouraged to use the glossary at the end of the thesis to help understand the many acronyms.

2.1 Historic Techniques for Object Recognition

Object recognition, or classification, is arguably the oldest topic in the discipline of computer vision. In 1966, Marvin Minsky assigned an undergraduate student to the computer vision problem as a summer project, tasking the student to "connect a camera to a computer and get the machine to describe what it sees" [15]. Decades later, researchers are still working on object recognition.

The world "recognition" comes from the Latin roots *re*, meaning "again", and *nosco*, meaning "to know." By its very definition, recognition implies having prior knowledge[1]. As a result, the vast majority of computer vision techniques implement recognition using a form of machine learning, which always consists of a *training* step, where the system learns a response or mapping function from collected data, and a *testing*

[1] One should note that some research focuses on so-called *zero-shot learning*, where a system recognizes objects it has never seen before based on previous descriptions. Humans are able to perform recognition this way, with the visual training step entirely omitted. This approach is well-suited to detecting novel objects in the unconstrained real world, but in a well-defined task such as glovebox automation, all the objects to be observed are known and can be well-defined. Therefore, in our use cases, zero-shot learning provides little benefit and does not merit further consideration.

step, where the system uses its mapping function to produce predictions. This approach describes all but the most basic forms of recognition used in computer vision literature.

The data used for object recognition is usually a three-channel, 8-bit RGB image or a four-channel, 8-bit RGBD image, where the fourth channel represents the 3D distance of a pixel from the image sensor in the real world. From this input, a system usually seeks to produce at least one object type (or class) that appears in the image—the recognition result. Object recognition can either be performed in real-time, analyzing images acquired from a camera as they are streamed from the sensor, or it can be performed offline, in which analysis runs on images saved in files or databases. In robotics, the training step usually runs offline on pre-collected data, and the testing step occurs online during the manipulation task.

Whether online or offline, an object recognition system often has to analyze hundreds, thousands, or even millions of images. An object recognition function must therefore map a potentially huge input space[2] into a small output space: the short list of different objects that we wish to detect. For manufacturing applications, 10 distinct object classes are usually sufficient for a given task, making the mapping problem extremely lopsided. And yet, because of viewpoint changes, lighting variations, and sensor noise, discovering the correct way to map a large number of input pixels to a small output space of object classes becomes a complex and nuanced task. To help simplify this problem, computer vision employs a variety of algorithms to detect important features in images, simplifying their representation to something that can be more easily analyzed. These simplified representations are called *descriptors* or *feature vectors*.

[2] Each color channel being represented by an 8-bit number requires 24 bits/pixel and allows $256^3 = 16777216$ different colors. A single 640x480 pixel image has $640*480*16777216 \approx 5.15 \times 10^{12}$ possible configurations.

Feature vectors can be categorized into *local* and *global* descriptors. A local descriptor is a simplified mathematical representation of a single point in an image. A global descriptor attempts to encode an entire image or 3D point cloud with a single set of values. Both types of descriptors have seen success in different research areas, as described in this chapter.

Beginning with low-level features such as edge detection and gradient detection, researchers soon developed algorithms capable of discerning mid-level features such as corners, objects, or even handwritten text. In 2004, Lowe developed the Scale-Invariant Feature Transform (SIFT), which generates local descriptors. SIFT begins by applying a Difference of Gaussian (DOG) filter, which detects sharp changes in pixel intensity (intensity gradients). Next, it finds "interesting" points in the image by finding local maxima and minima in the DOG filter response. At each of these *interest points*, SIFT calculates the gradient in multiple directions and over multiple distances, binning the gradients into 8 angular bins to generate a histogram representing the interest point [16]. This histogram is the descriptor, and it is designed to hold as much useful information about the interest point in question while being as simple as possible.

Once the algorithm generates descriptors for interest points in an image, they can be stored: this completes the training step. In the testing step, the descriptors from new images are compared to those from the training step to find a good match. This overall procedure is the same for nearly any feature-based object detection approach.

Because it analyzes gradients in multiple scales and directions, SIFT is well-suited for finding the same interest points in different images, even when objects are viewed from different angles [16]. A similar local descriptor, Histogram of Gradients (HOG), was

developed in 2005 to detect pedestrians[3] in still images [17]. When extended and coupled with machine learning techniques, HOG still sees use today for tasks such as optical character recognition [18], but it generally lacks the discriminative power required to recognize complex 3D forms.

A few years after the development of SIFT and HOG, depth-based recognition techniques began to rise in popularity. In 2009, the Kinect 3D sensor entered the market, making RGBD data easily available to researchers for under $200. Using the depth component of the data to make up for ambiguous 2D data, 3D classification algorithms can detect objects with less defining color and texture features. This is great news for fields such as automated manufacturing, where many of the objects to manipulate (such as machine parts) are made of textureless metal.

When detecting textureless objects, global descriptors are generally more appropriate because of their ability to characterize a 3D object's entire form. Rusu developed the Global Fast Point Feature Histogram (GFPFH) [19] and the Viewpoint Feature Histogram (VFH) [20] [21] during the development of the open-source Point Cloud Library (PCL) [22], released in 2011. In contrast to SIFT and HOG, which are local descriptors, GFPFH and VFH are global descriptors. In the case of GFPFH, the descriptor encodes information about the estimated surface normals (the orientation and shape of the surface in 3D space) of the objects in the point cloud. VFH is an extension to GFPFH, where feature vectors include an additional "viewpoint component," which is calculated as the angle between the surface normal at each point and a vector starting at the viewpoint and passing through the point cloud's center.

[3] Pedestrian detection is a *binary classification task*: every window in an image can be classified as either "pedestrian" or "non-pedestrian."

The VFH feature vector is a 308-bin histogram, and at test time it is treated as a 308-dimensional vector and compared to previously learned feature vectors, finding a close match to perform object recognition. Because of the difficulty of comparing many 308-dimensional vectors, the search technique usually uses a library for *approximate nearest neighbors*, such as the Fast Library for Approximate Nearest Neighbors (FLANN) [23]. Approximate nearest neighbor libraries allow feature descriptor matching to proceed quickly, and as a result this type of recognition can proceed in near real-time.

The angle-binned histograms used in GFPFH and VFH have similarities to SIFT's gradient histograms, showing that successful depth-based recognition approaches follow steps similar to those for robust 2D descriptors. More recent iterations on 3D depth feature vectors include Oriented, Unique, Repeatable Clustered VFH (OUR-CVFH) [24] and the Circular Projection Histogram (CPH) [25], which was developed by a former NRG student in 2013. CPH and CVFH are both used as 3D descriptors in this research; for more implementation details, see Section 3.3.3.

While researchers iterated on the idea of matching feature vectors, parallel efforts focused on *template matching* for object recognition. Template matching looks for small parts of an image which match a pre-trained object (the "template"). This procedure often involves an algorithm such as *sliding window search*, where a template is checked against a rectangular image patch which is sampled at different locations across the original image. Image patches that match the template well are recognized as instances of the template object, and if pose data is saved during training, this approach can also provide a pose estimate for the detected object.

In 1998, long before the current research surge in autonomous road vehicles, Gavrila and Philomin used template matching to detect road signs in real-world driving data. Their approach used sliding window search to minimize the chamfer distance [26] on

edge-detected images of highway scenes. This approach effectively found image patches that matched the template's appearance, detecting 3 different types of traffic signs at 8-10Hz with 90% accuracy [26].

While template matching works well for frontal images of road signs, it does not perform well for viewpoints even slightly different from those originally used for training. As such, it seems ill-suited for detecting objects from variable viewpoints because of the prohibitive amount of training data required to represent all possible viewpoints. But the simplicity and success of the template matching technique has continued to inspire new research, and its performance has improved significantly in the last 10 years, especially when detecting objects that lacked surface texture. Hinterstoisser *et al*. [27] developed a multi-modal template matching algorithm, LINEMOD, which uses a more robust and efficient template similarity measure than chamfer distance. Using a combination of SIFT-like gradient features and VFH-like features calculated from 3D surface normal and camera viewpoints, LINEMOD can perform an extremely discriminative sliding-window search, giving results that far outperform HOG.

2.2 Historic Approaches to Pose Estimation

Once recognition techniques have identified an object, the vision system must then determine its position and orientation. Pose estimation is essential for navigation, measurement, or manipulation, but historically has received less attention than object recognition in research literature, presumably because it depends on successful recognition.

Pose estimation falls into three general approaches. In one approach, object pose is stored alongside feature vectors, tightly linking pose to object classification. When using this approach, each different observed orientation of an object acts as a separate detection, and so when classification matches an observed object to one that has been previously

trained, it also automatically matches the object to a specific orientation. All of the PCL global descriptors fall into this first category of pose estimation. When collecting data for evaluation of GFPFH, VFH, CVFH, and CPH, the algorithms require the use of a pan-tilt table, which can rotate a training object to a desired orientation. The software then generates a classification/orientation pair, which is saved and used later for joint recognition and pose estimation as described above.

The second approach for pose estimation uses statistical techniques to align (or *register*) observed point clouds with ones that have been previously observed. Iterative Closest Point, or ICP, is the most commonly employed algorithm for this purpose, and many variants of ICP exist for different applications [28]. Some variations of ICP are included in PCL, making it a popular choice for systems looking for easy improvements in pose estimation. ICP is often used as a final filtering step to improve pose accuracy, as in [29] and [30]. Unfortunately, ICP can easily become "stuck" in local optimization minima, which results in an incorrect pose estimation. This behavior most commonly occurs when the algorithm begins with a poor initial guess. Many approaches use pre-matched pose estimations to provide a good starting point for ICP, only using ICP for fine-tuning the system's guess. After fine-tuning, the result is usually accurate enough for well-constrained robotic manipulation. More details on manipulation research are found later in the chapter.

The third approach treats pose estimation as an important step in the recognition process itself. Intuitively, it makes sense for pose estimation and object recognition to be performed together, since an object can look drastically different when viewed in different orientations. Only recent research has been able to merge these two steps, catalyzed by the emerging field of machine learning.

2.3 Current Techniques for Object Recognition and Pose Estimation

Through the early 2010s, (all of the research efforts surveyed to this point), object recognition mostly relied on "handcrafted" features and feature vectors, such as SIFT or CVFH. These feature representations were developed using human ingenuity and trial-and-error, and while computer vision could learn and compare these features, humans still defined the underlying representation. Meanwhile, machine learning, which had been developed in the 1970s, was being adopted for an ever-wider range of problems with less and less human input. Since 2012, machine learning techniques have routinely outperformed handcrafted features, quickly emerging as the premier set of tools for computer vision, and they promise new capabilities for the entire field, including 3D object detection and pose estimation.

In 2010, no machine learning approach had been powerful enough to complete a task as nuanced and difficult as general object recognition, but new research in neural networks continued to give rise to entire new types of network. Convolutional neural networks, or CNNs, are a specialized type of multi-layer neural network specifically designed for working with data that has spatial or temporal correlation, such as speech recognition and visual object recognition [31]. The nodes in the first layers of a CNN, the convolution layers, are only sparsely connected to the network inputs, with each node's connections overlapping with those next to it. On these first layers, all of the nodes' weights are shared, with the end result that the system can easily learn filter responses [32]. Higher layers may sometimes be *max pooling* layers, which output the maximum of a small set of input nodes, and the highest layers in the network are usually fully connected like a traditional neural network.

A CNN can be customized by choosing the connectedness of the convolution layers, as well as varying the number and positions of the convolution and max pooling

layers. Figure 2-1 shows an example architecture. Modern CNNs, such as the 2015 ImageNet winner, can have up to 150 layers [33]. As with other neural networks, a CNN is trained using backpropagation and large amounts of training data. The amount of data required to successfully train a CNN is proportional to the network size.

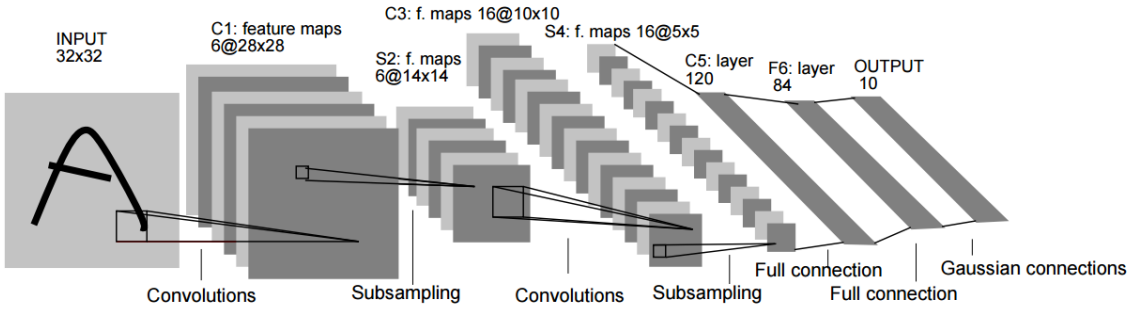

Figure 2-1: Architecture of the LeNet-5 CNN, which was developed in 1998 for robust optical character recognition [34].

The ImageNet challenge dataset is one of the largest sets of publicly available training data. The annual challenge, which began in 2010, is "the standard benchmark for large-scale object recognition" [35]. Consisting of approximately 1.4 million images and over 1000 classes, ImageNet competitors have pushed the limits of 2D object recognition. In 2012, a team from the University of Toronto used a deep CNN to achieve first place in the ImageNet image classification contest, beating the runner-up by over 10% recognition accuracy [36]. Since then, researchers have improved CNNs to achieve even greater performance gains over handcrafted approaches, with current ImageNet entries approaching 95% accuracy, about the same as human performance on the same task [35]. ImageNet has motivated hundreds of computer vision research papers, and it is no coincidence that most of the recent growth in computer vision can be directly attributed to CNNs.

Unfortunately, the results of the ImageNet challenge and the networks it has produced are not directly applicable to the problem of sensing for industrial robotic manipulation. ImageNet competitors attempt to make a classifier capable of sorting highly unconstrained images of brand-new objects into a huge set of potential object classes. For example, a classifier may be expected to detect a salmon swimming in a stream, or a salmon sitting in a marketplace in the background of a selfie.

For robotic manipulation in nuclear environments, we wish to re-identify the same objects over and over again in constrained environments and with extreme accuracy, and are usually not concerned with obscure cases. Furthermore, environment knowledge can be exploited to dramatically reduce the classification set from millions of objects to significantly less than a hundred. In addition, ImageNet only requires competitors to locate objects in a 2D image (drawing a bounding box around the object), and does not involve any orientation estimation. 6DOF pose estimation is an essential step of vision for manipulation. Instead of using competition-winning networks from other domains, vision for manipulation requires networks specifically designed for the task, and while researchers are currently investigating neural networks for 3D detection, they have produced fewer results than their counterparts working in 2D vision.

Wohlhart and Lepetit [37] used a CNN to learn feature descriptors from RGB and RGBD data, and showed that the learned descriptors are more discriminative than HOG or LINEMOD. Their approach achieved 99.0% classification accuracy within 5 degrees of correct pose estimation, and 99.9% classification accuracy when orientation information was not required. This technique has the additional benefit of performing well on images lacking depth information, although there is a small decrease in performance compared to the RGBD case. Schwarz *et al.* [38] showed that adding depth information to an image

could significantly increase the performance of a CNN for classification, even when the CNN was not originally trained using depth data.

Another way to generate pose estimates from RGB-only data is to detect and track specific parts of objects, then use these parts' positions and orientations to infer the overall object's pose. Crivellaro *et al.* [39] trained a CNN to predict the 2D projections of the interest points' 3D poses, which can then be combined to retrieve an accurate pose estimate for the entire object. This system works well for occluded objects, because as long as at least one interest point is visible, the algorithm can still detect the object.

As machine learning research has exploded, so has machine learning-based pose estimation techniques. *Random forests* [40] were developed in 2001 to improve classification and regression. Random forests consist of many *decision trees* (hence the *forest*), each of which is trained on a randomly-selected set of features from the training data. For example, for an RGBD image, one tree in the random forest might only be trained on the red and green channels, while another tree might use the green and depth channels. The different trees in the forest can then cast "votes" for different classification results.

Working with random forests, Tejani *et al.* [41] adapted LINEMOD into a scale-invariant descriptor for image patches instead of entire objects. Their random forest is trained on images that are subparts of objects, so when the system observes an object during testing, it can infer the object's presence by observing some of the parts, instead of the entire object. The approach also allows more accurate pose estimation in the presence of clutter and partial occlusion by fitting a whole-object transformation to the observed object parts. In addition, the scale-invariant descriptor allows objects to be detected at more extreme scale differences from the training data. Other random forest-based techniques include Brachmann *et al.* [42], Bonde *et al.* [43], and Krull *et al.* [44], all of which perform as well as LINEMOD, or better.

Recent work at UC Berkeley suggests that with enough neural network depth and training data, neural networks can eliminate the need for descriptor matching entirely. The Berkeley Robot for the Elimination of Tedious Tasks (BRETT) uses a CNN vision architecture that consists of "visual autoencoders" which detects the configuration of objects in the field of view, such as robot joint positions or the position of a loop in a piece of rope [45]. BRETT's architecture can then learn a motor control policy based on feedback output from the autoencoders to complete complex tasks reliably in the presence of large initial uncertainty.

These neural network-based approaches to computer vision may become a new standard in sensing for manipulation, but for now, the results do not improve enough on previous handcrafted approaches to warrant their use in an industrial or nuclear setting. The amount of data required to accurately train a neural network-based system is prohibitive, often numbering in the thousands or even millions of data points, and even the descriptor-free approach taken by UC Berkeley takes a long time to learn an effective control policy. This drawback is further discussed in Section 7.2.2.

Because reducing training, deployment, and cycle time are some of the primary goals for using industrial manipulators in fixture-free tasks, the current focus should be on bringing fast, proven methods to the industrial-nuclear complex. Once the merits of computer vision have been realized, work can proceed to integrate more cutting-edge detection algorithms and incrementally improve performance. Until then, DOE can achieve large gains in worker productivity by using vision to reducing process tooling and fixtures.

2.4 Fixture-Free Manipulation Using Computer Vision

Using vision to eliminate rigid fixtures in manufacturing is not a new idea, but early efforts had to rely on crude computer vision techniques. By using binary (black/white)

images, principal component analysis, and basic template matching, Yoon *et al.* [46] worked with Ford Motor Company in 2003 to determine the 6D pose of an engine cover using a stereo camera. Their results were impressive, achieving position accuracy within 2.4 mm and rotation accuracy of 1.5 degrees, but relied on many simplifying assumptions. Their system assumed that the entire engine cover was visible, and also that the pose was not so extreme that major features on the object could not be detected. In addition, the lighting conditions and background were well-controlled. The research is an admirable early example of how much can be achieved using computer vision, but we must progress far beyond these results to achieve flexible IIA.

Using a brute-force sliding window search on depth data collected with a laser scanner, Quigley *et al.* [47] were able to accurately find the location of objects such as coffee mugs and door handles using a mobile robot. Their mobile manipulator was able to find and grasp a door handle using this method, which was published in 2009. Unfortunately, the brute-force approach used in the paper required exhaustive analysis at multiple image scales, and as a result it took over 6 seconds to analyze a single frame. This processing rate would prove to be much too slow for all but the most basic tasks.

Vision techniques and algorithms continued to improve, and along with them, compute power increased, allowing systems to complete far more impressive (and useful) tasks. Cowley *et al.* [48] sought to use a PR2 robot to manipulate various household objects randomly oriented on a conveyer belt moving at 33 cm/s, using RGBD data, handcrafted filtering, and a series of increasingly fine-grained recognition checks. By using Kalman filtering and RANSAC estimation, they were also able to predict objects' motion paths as they moved along the belt. Unfortunately, the authors do not quantify the accuracy of their pose estimation, instead only reporting the pick and place success rate. Using this approach,

their robot was able to successfully pick and place objects from the conveyer with a 91% success rate and an average time per object of 6.7 s [48].

Previous work at NRG resulted in successful pick-and-place tasks in gloveboxes [49]. However, this work is restricted to a single object type (a metal canister) and does not take into account object orientation. The same work showed the ability to visually locate the handle on a cabinet door and open it using an adaptive gripper and arm, but this work required operator input for an initial guess at the handle's position, and also assumed that the visual sensor was oriented perpendicular to the cabinet door. While this prior work is encouraging, the methods used in the paper are too restricted by their environment assumptions for autonomously completing useful tasks.

Bin picking is another unconstrained task that represents unique challenges, such as heavy occlusions for some objects. Wu *et al.* [50] proposed a voting scheme to produce 6D pose estimations and object classifications, followed by ICP pose refinement and filtering to disregard occluded objects until the objects above them had been picked first. This approach discerned between three types of objects with 93.9% accuracy and achieved a pickup success rate of 89.7%.

In addition to using traditional handcrafted approaches, some laboratories and companies have begun to implement machine learning methods to perform automated tasks in unconstrained environments. Kujala *et al.* [51], working for ZenRobotics, a robotics recycling company, have autonomously sorted industrial and demolition waste (glass fragments, chunks of stone, etc.) using a 3D vision system, machine learning techniques, and autonomous gantry robot. Their approach naively calculates possible grasp configurations for objects using an overhead depth camera and calculating gripper positions that will place an object in between the gripper's fingers. Their success is augmented by using gripper position feedback and a machine learning algorithm to better

determine which grasps are likely to succeed. While this approach does not strictly use machine learning in the vision process, the learned grasp evaluator greatly improves the system accuracy. The gantry robot was able to pick all objects off of a conveyer except for those pushed outside of its work area. Although simple, this approach is effective for objects that are not fragile and for which the specific grasp configuration is unimportant, making it reasonable for a task such as waste sorting.

2.5 SUMMARY

Beginning with handcrafted feature descriptors such as SIFT, computer vision approaches have expanded to include a plethora of possible approaches for classification and pose estimation. The recent rise of machine learning and the unmatched results possible by using CNNs hold great promise for the future of computer vision for manipulation, and indeed many robots are already incorporating the technology. If we wish to completely eliminate the need for prohibitive tooling and fixtures in industrial manipulation tasks, machine learning-based techniques may be the only way to achieve the required accuracy. But even before machine learning-based vision is considered, a gap remains between proven academic computer vision research and the sensing used in robust industrial manipulation. Without waiting to validate brand new computer vision methods, we can still see great gains in industrial manipulation from implementing well-understood computer vision in the context of a robust reliable architecture.

Chapter 3: Implementation Details

This chapter will present in detail the implementation of the Object Recognition and Perception (ORP) library, which is a new object-oriented library for robust recognition and pose estimation that can use a variety of computer vision algorithms interchangeably. The library consists of a core set of recognition tools, as well as an extensible system that can be adapted to implement any number of current or future object detection methods.

3.1 Supporting Software

ORP is written in C++, and is heavily integrated with Robot Operating System (ROS) [52] to allow for easy interoperability with existing systems. ROS is "a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms" [53]. ROS implements a modular architecture based on the concept of *nodes*, which are independently-running programs which may be written in any language. A node may implement an interface to a hardware driver, such as a camera or a robot arm, but it can also implement software-only libraries or any other self-contained code. By performing all cross-library communication using nodes, ROS applications offer an unprecedented level of modularity.

ROS nodes can communicate with each other in two ways: *topics* and *services*. An example of these communication channels is shown in Figure 3-1. Topics are uniquely-named communication channels that any node can publish data to, and any number of nodes may subscribe to a topic. Some examples of data published on topics are camera images and robot joint states. Services are a more strictly-defined communication channel published by a single node, and implement both a request and a response. When a node calls a service, it results in code execution on the server node, which then returns the

response to the caller. A node may implement any number of service clients, service servers, topic publishers, and topic subscribers.

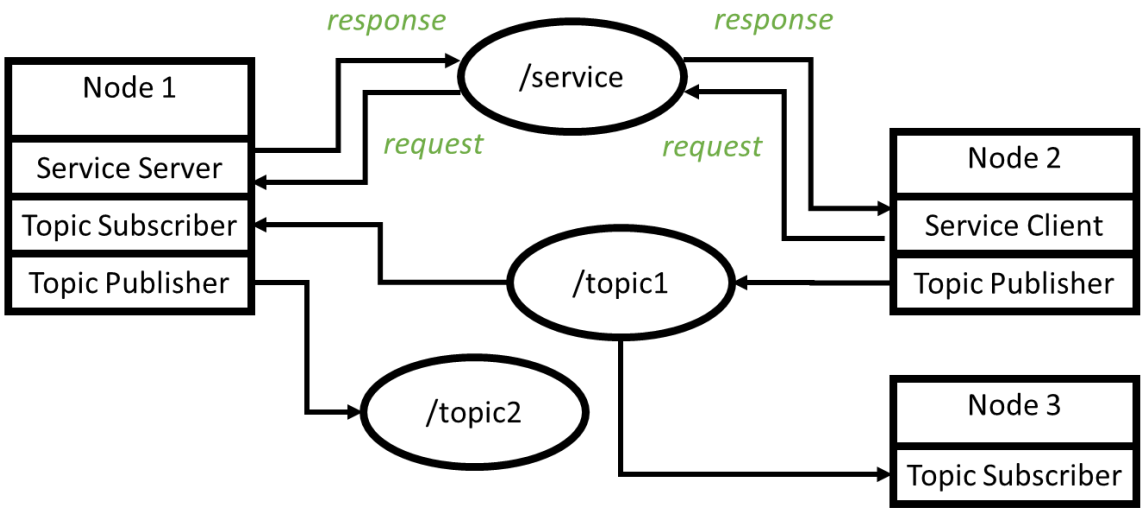

Figure 3-1: An example ROS architecture consisting of 3 nodes, 2 topics, and 1 service. The arrows show the data flow between the nodes. In this system, any node can be stopped, restarted, or replaced independently of the others.

ROS was chosen for three reasons:

1. **Flexibility**: ROS's ability to write nodes in different languages and styles is unparalleled, and nodes can easily be turned off, restarted, or switched out for replacements without a loss of overall system functionality. This also improves the overall system reliability.
2. **Interoperability**: thousands of other systems, from aerial drones to autonomous cars, already use ROS, and ORP could easily be integrated with any of them. Most importantly, the three hardware platforms at NRG already use ROS.
3. **Pre-existing support**: Publicly-available packages exist for reading data from RGB and RGBD cameras (e.g. [54]–[56]), and ROS also has C++ native bindings to the open-source Point Cloud Library (PCL) [57], discussed below. These

packages are a strong foundation of "boilerplate" code on which to build an object recognition library.

PCL is the second important supporting library for ORP. PCL is a C++ library for processing point clouds [22], and it contains many useful functions for signal conditioning, noise removal, and higher-level functions such as grouping points into clusters.

3.2 Recognition Library

ORP is a library designed to allow a robotic system to recognize and locate objects in the environment using any one of a number of object detection algorithms. ORP contains two main parts: object recognition code, and a reusable world object description. Figure 3-2 outlines the communication channels between ORP's various components.

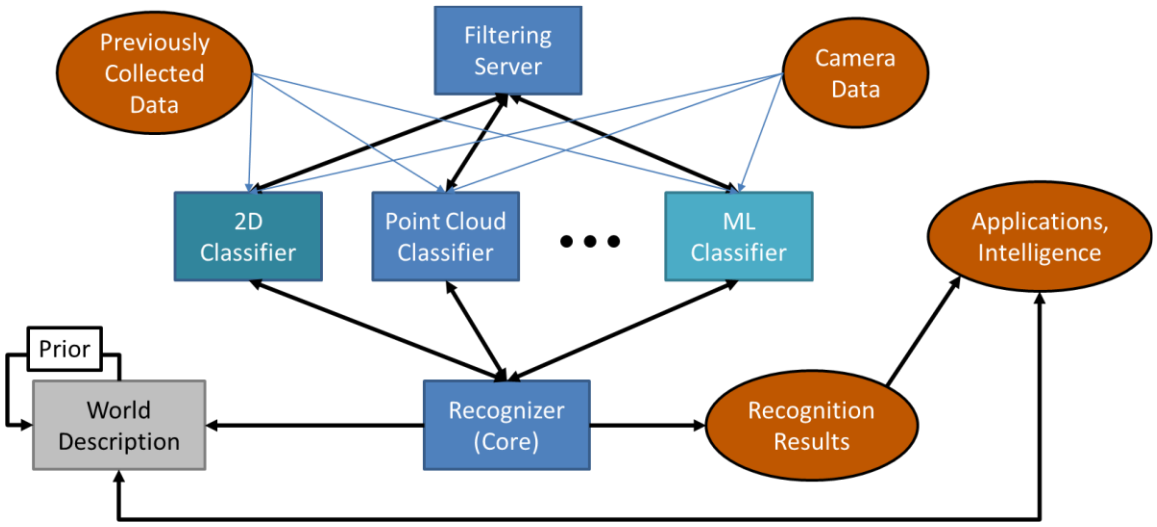

Figure 3-2: An overview of the high-level systems in ORP.

ORP's primary feature is its flexible point cloud-based classifier system that allows multimodal sensing for different objects in the world. A generic classifier is treated as a "black box" which accepts RGBD data and produces object classifications and pose

estimations from that data. This classifier may discard the 3D information from the point cloud and only use the RGB channels, it may only use the depth channel, or it may use some combination of the two. Classifiers can also use additional data from previous training steps, which is usually persistently stored to disk. Various classifiers can implement the internal recognition and estimation functionality however they choose, and a recognition system can choose to run any set of classifiers as dependent on application needs. Because each classifier is implemented as a ROS node, implementing new classifiers has little to no effect on the overall application framework for a given task.

The different classifiers are orchestrated by a core recognition node ("recognizer"), which is another ROS node that combines the recognition results from the classifiers to produce a standardized world representation. The recognizer is also responsible for determining what filtering (if any) occurs from one frame to the next, improving overall recognition and pose estimation results. The recognizer is also responsible for publishing the forward-facing ROS messages with the final recognition results. These results can then be observed by an application, also using ROS, to make intelligent manipulation decisions.

3.2.2 Classifiers

ORP currently consists of five "official" classifiers (Not counting tests, classifiers directly superseded by newer ones, and niche classifiers written by other NRG researchers), each of which is designed for specific tasks. Table 3-1 summarizes details on the various classifiers, including which NRG applications use them. The classifiers are sorted by the age of the algorithms they use, with the oldest approaches listed first. Also shown is whether or not each classifier requires pre-collected data to train the system—"Yes" indicates a machine-learning based classifier.

Table 3-1: Classifiers currently implemented in ORP.

Name	Category	Used in	Training?
RGB	Histogram Analysis	Glovebox Color Sorting	No
CPH	Feature Descriptor	Mobile Inspection	Yes
VFH	Feature Descriptor	Glovebox Object Detection	Yes
CVFH	Feature Descriptor	Glovebox Object Detection	Yes
Cup Detector	Hough Transform	Detonator Cup Detection	Yes

3.2.3 Supporting Components

To support the various classifiers, ORP includes the ability to filter point clouds via a variety of methods, most of which are internally supported by PCL. This is facilitated by yet another ROS node that advertises a service, making filtering standardized and available for use by any classifier. All point-cloud based recognition approaches utilize some form of pre-filtering, as it helps isolate the objects being detected and achieves better recognition results. The filtering process begins with the 3D points from the RGBD sensor, and ends with a set of well-behaved point clouds representing the different objects in the scene. The filtering algorithm defaults to the assumption that the objects in the scene are sitting on a planar support surface, and filters this surface out of the point cloud. However, this behavior can be disabled if desired.

Filtering the point cloud consists of five steps, which are shown in Figure 3-3. First, all points that lie outside the region of interest are removed. Second, the points are sampled at regular intervals, converting the point cloud into a rectangular grid of voxels that has a lower resolution than the original data. This step reduces the number of points in the cloud while still maintaining as much spatial information as possible. Any voxel containing at least one point in the point cloud will be assigned a point in the resulting filtered cloud. In the third step, large planar surfaces (which are assumed to be floors or walls unimportant to the classifier) are detected via the RANSAC algorithm [58] and removed. Next, the remaining points are grouped into clusters by a threshold on the distance between adjacent

points. Finally, any groups that are too large or too small are discarded.

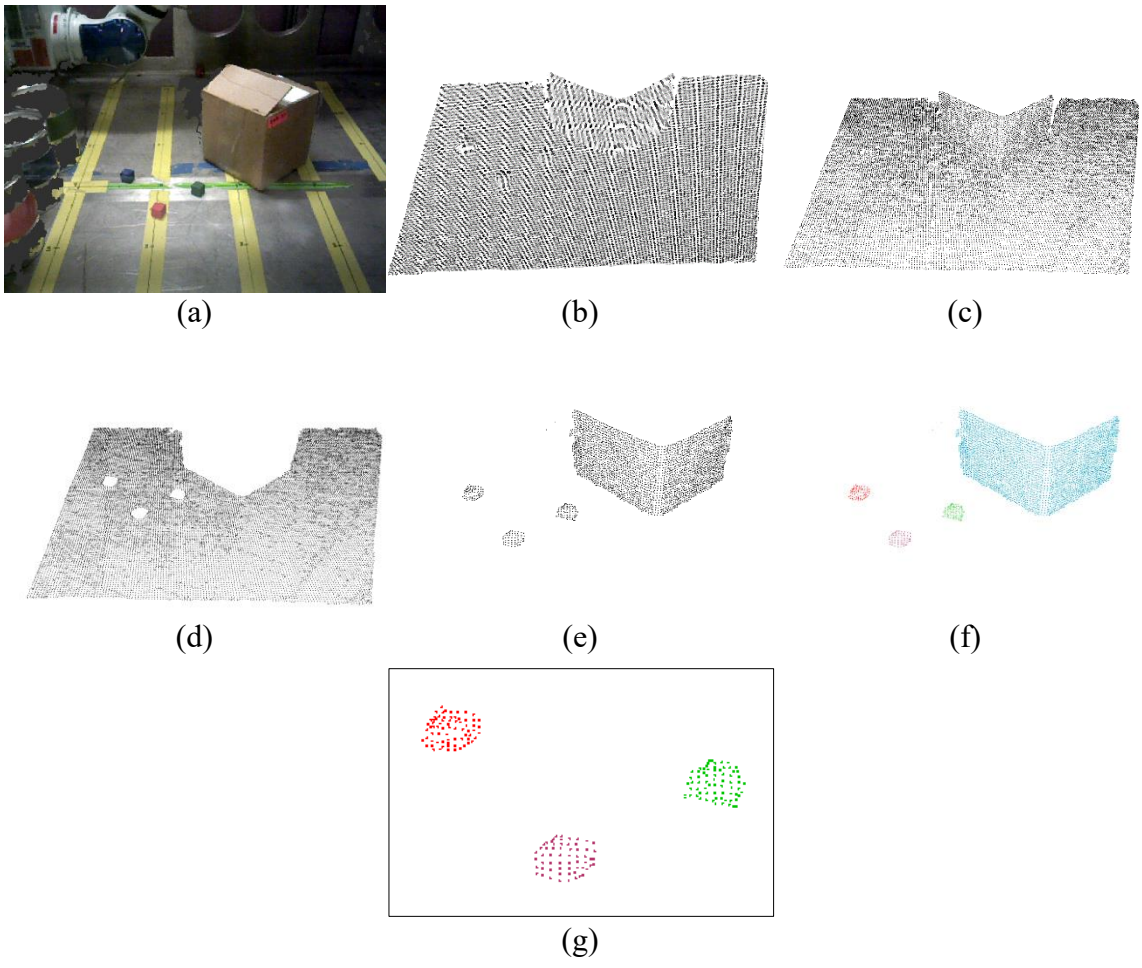

Figure 3-3: Steps in the point cloud filtering process. The goal is to detect the small cubes in the image in the upper left. (a) original point cloud, (b) after spatial clipping, (c) after applying voxel grid, (d) the primary plane detected and removed from analysis, (e) after removing the plane from the point cloud, (f) after Euclidean segmentation, (g) close-up after outlier removal, which removes the large point cloud from the cardboard box.

The CPH, CVFH, VFH, and RGB classifiers in ORP apply the five filtering steps, but each of them use different filter parameters. ORP's interface, which was developed in prior work [59], allows the operator to tweak the filtering parameters in real-time to account for sensor noise, changes in lighting, or other unpredictable factors. Once the user makes

any necessary adjustments to achieve good filtered sensor data, the system can proceed just as it would have before, with no code changes required.

3.3 Classifier Details

Each of the five classifiers implements different algorithms, including 2D, 3D, handcrafted, and machine learning-based analysis. As mentioned above, each classifier is designed with a specific use case in mind.

3.3.1 RGB Classifier

The RGB classifier uses OpenCV to analyze the histograms of the color channels of each object in a point cloud. It calculates three values by summing the red, green, and blue channels of all pixels in the cloud, then taking the maximum of these three values. The strongest channel is returned as the object's classification: "red," "green," or "blue." This method, while old and simplistic, is by no means useless. The classifier is extremely fast to calculate and, when coupled with ORP's filtering node, performs with near-perfect accuracy for objects in the NRG glovebox, as will be discussed in Chapter 6. In addition, this classifier is a good surrogate for a radiation-based classifier that detects objects based on their radiation type (alpha, beta, or gamma radiation).

3.3.2 Cup Detection Classifier

The cup detector classifier is a task-specific classifier designed to detect small (<1 cm) circular detonator components for accurate manipulation. In lieu of actual detonator components, the system was designed to detect stainless steel cups with a diameter of 6.4 mm (1/4 in.). The classifier consists of two parts: a 2D circle detector, and a 2D-to-3D neural network.

The *Circle Detector* uses the Hough transform [60] to detect circular patterns in images. The detector uses the open-source OpenCV implementation of a Hough circle

detector [61]. The camera is placed overhead at a vertical distance of approximately 0.3 m from the cup's position to collect images for the detector. In the case of a cup pick-and-place task, the initial position of the cup is known to within 40 mm, so the image is cropped to the central 200 pixels to make calculating the Hough transform faster.

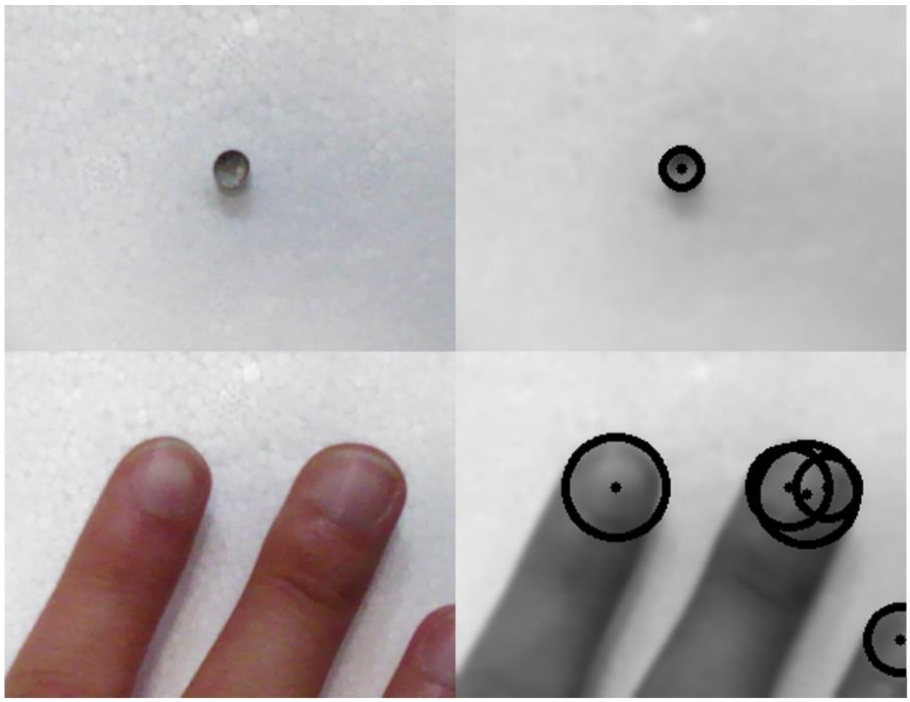

Figure 3-4: Examples from the cup classifier's circle detector. The left column shows input images. The right column shows the results after the detector has been applied, with circle boundaries and centers denoted in bold black. Note that although more complex objects such as fingers and fingernails result in many overlapping and spurious detections, a single cup in the field of view creates a single correct detection. The background is white polystyrene.

The detector changes the cropped images to grayscale and applies a 2 px Gaussian blur to reduce noise that could cause spurious detections. The Hough circle transform is then called, using a resolution of 1, a minimum distance between detected centers of 10 px, internal Canny edge detector parameters of 100 and 10, and restricting the detected circles

to have a radius between 10 and 30 pixels. The end result is a set of (x, y) pixel coordinates for each circle detected in the image. Sample collection images and detected circles are shown in Figure 3-4.

Once the circle detector has extracted the circle center, the 2D coordinate in pixels, $(x, y)_{camera} \in \mathbb{R}^2$ must then be converted into a 3D pose in meters, $(x, y, z)_{world} \in \mathbb{R}^3$. To perform this conversion, the 2D coordinates are passed through a *2D-3D neural network* that has been pre-trained using data from a previous collection step, where the transformation between the cup and camera was known. The network has three layers and 10 hidden nodes, trained in MATLAB using the default training method[4]. The neural network learns the mapping function $F: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ from the labeled training data. This mapping can be easily achieved because of the structure inherent in the data due from the cup always sitting on a flat surface. Finally, to improve the pose detection accuracy, the system takes advantage of the fact that the cup always is at the same z-coordinate, since it is on a flat surface, and set $z_{world} = Z + h/2$, where Z is the z-coordinate of the support surface and h is the height of the cup.

3.3.3 VFH, CPH, and CVFH Classifiers

The VFH, CPH, and CVFH classifiers all implement classifiers that have been previously developed by other researchers, and the ORP versions seek to leverage the classifiers' success in specific domains.

The VFH classifier users the global VFH descriptor included in PCL. VFH is based on GFPFH, a simpler global descriptor. In GFPFH, the surface normal for each point is compared with those of nearby neighboring points to calculate 3 metrics (for details, see [64]), which are binned and combined for all points in the cloud to create a feature vector.

[4] MATLAB's default neural network optimization uses the Levenberg-Marquardt algorithm [62] [63].

VFH is an extension to GFPFH, and its feature vectors are identical to GFPFH's except that they append an additional *viewpoint component*, which is calculated as the angle between the surface normal at each point and a vector starting at the viewpoint and passing through the point cloud's center. The angle for each point is added to an additional binned histogram to create the VFH feature vector (see Figure 3-5).

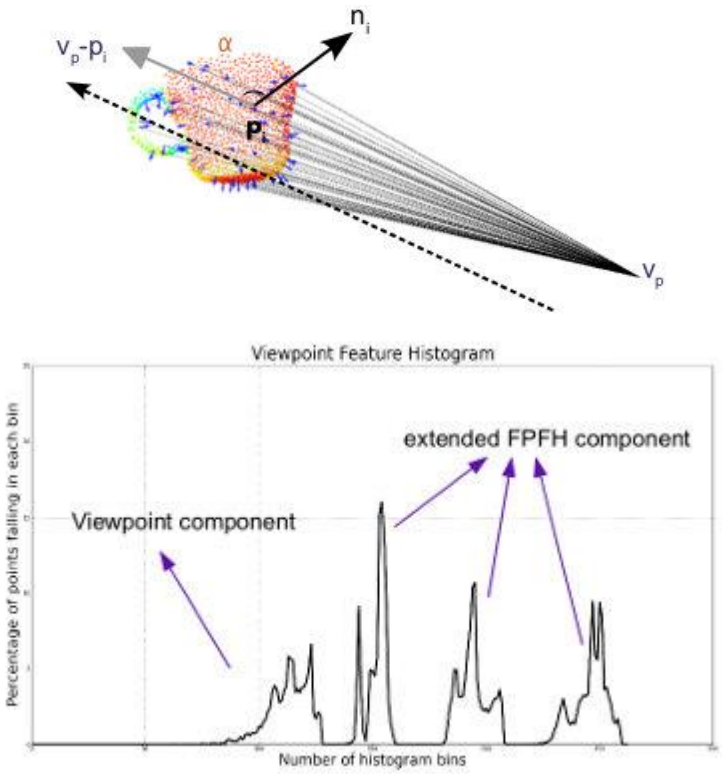

Figure 3-5: The Viewpoint Feature Histogram. Above: The "viewpoint component" of the VFH descriptor is calculated using the angle α between the surface normal n_i and the vector from the viewpoint to the point cloud's center, v_p-p_i [21].

The CVFH classifier uses the global CVFH descriptor included in PCL, which was developed as an extension to VFH to work better with incomplete point clouds and occluded objects. The CVFH classifier works in a similar fashion to the VFH and CPH

descriptors, but includes additional steps to cluster points with similar curvature characteristics. This classifier also includes a calculation of the camera roll histogram, which improves the 6D pose estimation [65]. Because CVFH outperforms VFH in accuracy, this thesis does not consider the VFH classifier further, except to point out CVFH's improvements.

The CPH classifier uses the global CPH descriptor developed by O'Neil [25]. This classifier runs faster than VFH, and seeks to leverage CPH's ability to outperform VFH for point clouds in noisy environments. Other than the different descriptor, the classifier's behavior is identical to that of the VFH classifier.

During the training stage, point cloud data is gathered using one or more cameras positioned around a pan table, which slowly spins through a 360-degree rotation to afford views of the object from any number of angles. Synthetic training data can also be gathered in software by rendering artificial RGBD views of a 3D model or point cloud from any desired angle. After the images are collected (by either method), the feature vectors generated by the classifier are saved to a file, along with information about the object's pose when the data was collected., along with the known transformation between the center of the object and the center of the object's point cloud.

When the classifier is used for training, it begins by loading all previously saved feature vectors for fast lookup. When point clouds are passed to the system (usually by a camera), they are first filtered by the segmentation procedure described above. The classifier then processes each object point cloud separately. It generates a feature vector from the point cloud and finds the approximate nearest neighbor from the loaded training data using FLANN [23]. This nearest neighbor feature vector represents a specific class of object viewed from a specific viewpoint, providing both a classification result and orientation information. In the CPH and VFH classifiers, the coordinates of the centroid of

all the points in the point cloud is taken as the object's position in camera space, which often does not coincide with the actual center of the object (see Figure 3-6). The transformation we wish to determine is $T_{obj/cam}$, but we must make the approximation $T_{obj/cam} \approx T_{cloud/cam}$, as shown.

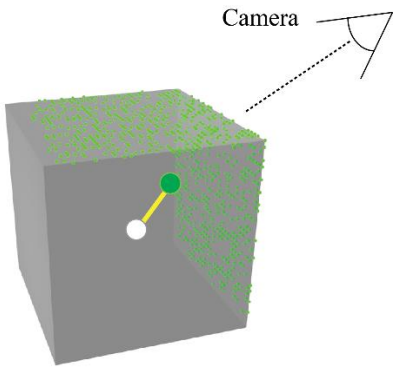

Figure 3-6: Illustration showing incorrect calculation of object centroid. The cube shown is observed from one side, producing an incomplete point cloud (small green dots). The centroid of this point cloud (green circle) is different from the actual centroid of the cube (white circle).

This incorrect center position makes these classifiers unsuited for all but the most basic of manipulation tasks. However, if a user is interested in how an object's position changes over time, rather than the absolute position, this point cloud centroid method can still provide good results. In some cases, NRG's parallel efforts in force/contact control could provide the additional information needed to improve these pose estimates by "feeling out" the exact object position, but this approach is beyond the scope of this effort.

The CVFH classifier requires more work during testing, but is able to overcome the object center issue discussed above. The CVFH pan table is supplemented with three augmented reality (AR) tags offset from the table's center of rotation by a known distance. The object to be trained is centered on the table's axis of rotation, so by determining the

orientation of the AR tags, the system can infer the absolute position of the object in the camera's frame of reference, T_{cam}^{obj}. Each time the classifier processes a point, it calculates the transformation from the center of the observed point cloud, T_{cam}^{obj}, to the table's center of rotation. Since T_{cam}^{table} is known, the system can then calculate the object's position, as shown in Figure 3-7. This difference, T_{cloud}^{obj}, is then saved alongside the pose information in plaintext format.

When an object is observed during testing, the previously stored transformation for the closest pose is applied to find the center of the object being observed, using the equation $T_{cam}^{obj} = T_{cam}^{cloud} * T_{cloud}^{obj}$. In this way, the classifier can accurately return the center position and 6DOF pose of observed objects, as well as an object classification based on point cloud shape. In practice, CVFH also performs better than CPH for noisy and incomplete point clouds, making it an ideal classifier for use in glovebox sensing applications.

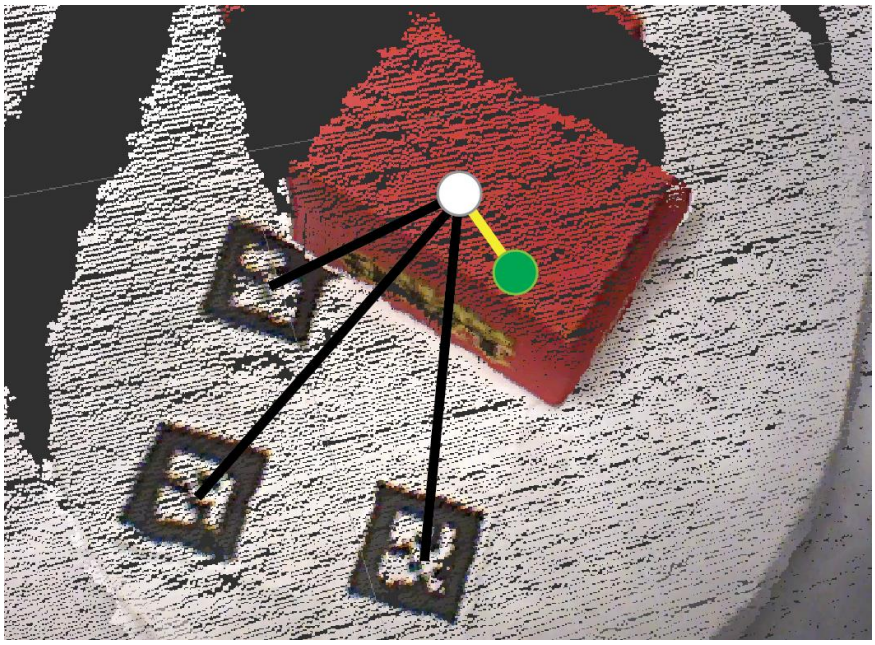

Figure 3-7: Calculation of the true object center using AR tags. The observed point cloud for the red box has a centroid (green circle) that is different from the box's actual center (white circle). The object center is inferred from the position of the three AR tags (black lines). The transformation from the point cloud centroid to the object centroid (yellow line) can be calculated and saved, allowing the true centroid to be inferred from the point cloud during later observation.

3.4 World Object Description

In addition to the classifier system, ORP contains new code for a universal *world object description*. ROS contains many libraries for interfacing with hardware components, but does not provide a canonical way to represent a scene in the world – the implementation is left to the user. ORP augments ROS by adding the notion of a world object type. Each type of world object has the following attributes:

- Name, which is used to unique identify the world object type.
- Shape, which determines the set of valid grasp configurations for the object.
- Size in X, Y, and Z dimensions.
- The object's color in the simulation environment for user visualization purposes.

As objects are recognized by the object recognition pipeline, they are associated with a world object type from a canonical list. By using a single list of objects, and rejecting objects of which it has no prior knowledge, ORP can avoid "guessing" information about objects it cannot recognize.

World objects keep track of their own position from frame to frame, automatically updating as new measurements arrive. A Kalman filter is used to update the 3D object position vector μ_t based on new measurements by making simplifying assumptions on the system model (a system similar to the one implemented in [25]). Beginning with the standard set of Kalman filter equations [66], the following assumptions are made:

- No control input: $u_t = B_t = R_t = \vec{0}, A_t = \vec{I}$
- Quasi-static system: $\bar{\mu_t} = \mu_{t-1}, \bar{\Sigma_t} = \Sigma_{t-1}$
- Well-behaved covariance: $C_t = \vec{I}$

The covariance matrix Σ_t is initialized by default as $\Sigma_0 = 1\ cm * \vec{I}$, corresponding to a Gaussian noise model with variance of 1 cm. If desired, individual classifiers can provide their own covariance value. Making the above substitutions gives the following set of equations for the Kalman gain K_t, the system state (object position) μ_t, and the updated covariance Σ_t:

$$K_t = \Sigma_{t-1}(\Sigma_{t-1} + Q_t)^{-1}$$

$$\mu_t = \mu_{t-1} + K_t(Z_t - \mu_{t-1})$$

$$\Sigma_t = (I - K_t)\Sigma_{t-1}$$

Using a Kalman filter ensures that sensor noise will not unduly affect the quality of the pose estimation, while still allowing the system to take advantage of multiple measurements and respond to object motion.

Finally, each object in the world generates its own grasp configurations based on its position, orientation, and shape. This feature allows a manipulation system to grasp any

type of object without requiring specific knowledge about the object, and the same object can be grasped using the same configuration on any manipulation platform automatically.

The world object type definitions are stored persistently in a SQLite database, which can be easily edited to modify or create world objects. The database is loaded via a special ROS node (written in Python) when ORP is launched. ORP also has the ability to handle multiple world object databases, and the user can choose to only load databases that are relevant to the current task. This allows the system to leverage domain-specific knowledge to reduce the complexity of the object recognition mapping function, thereby improving the recognition accuracy (see Figure 3-8).

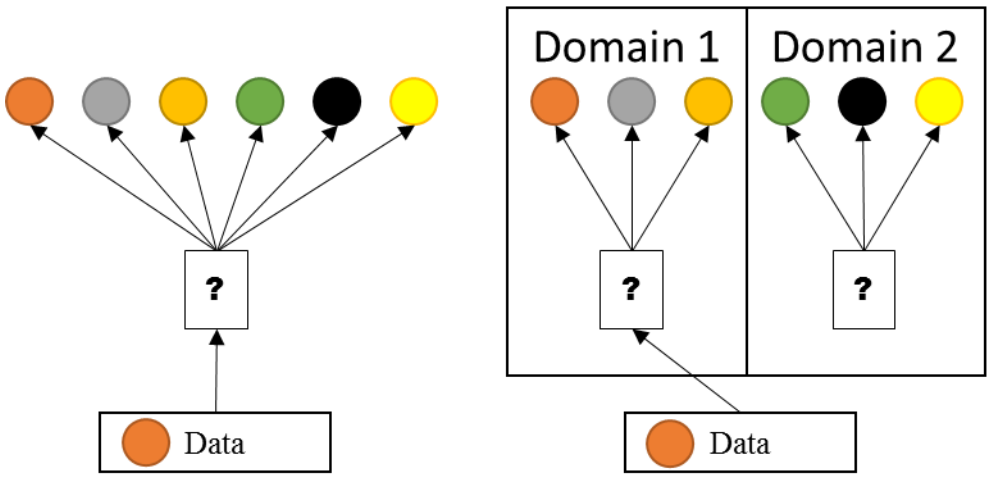

Figure 3-8: On the left, trying to classify one object given a large set of possibilities. On the right, using separate object databases built from domain knowledge simplifies the recognition problem.

3.5 Platform Integration

ORP's modular classifier system shows its strength in the diverse range of platforms that incorporate it. These platforms include two different mobile robots and one stationary robot, and have been specifically designed by our lab to carry out specific

automated tasks for the nuclear industry. The hardware platforms are discussed further in each demonstration chapter.

3.5.1 Software Requirements

ORP is a C++ library and exposes a forward-facing API using ROS nodes and services, and as such has software platform requirements linked to those of ROS. ORP is compatible with two versions of ROS, Hydro and Indigo, and has been tested on two versions of Ubuntu Linux, 12.04 and 14.04.

Because of the high throughput rate of most depth cameras and the complex processing required for full point cloud recognition, ORP is designed to run on capable hardware. A computer with a 4th-generation Intel i3 processor, 8GB of RAM, and USB3.0 capability is required to be able to run full recognition in real-time or near real-time. ORP can be run on slower machines, but with a reduced frame processing rate.

3.6 SUMMARY

This chapter described the ORP flexible classifier system and how it is used to accomplish a variety of sensing tasks on different hardware platforms. ORP is a modular system built in the framework of ROS (as shown in Figure 3-2), which allows it to easily adapt to different sensing situations.

ORP's array of classifiers provides the necessary detection and pose estimation capabilities required by NRG's use cases, as shown in Table 3-2. It should be noted that other researchers have built ORP classifiers to take advantage of the framework for niche cases, but these classifiers apply to different demonstrations, and are therefore outside the scope of this study. One such example is a "cylinder classifier," which uses RANSAC to fit a cylindrical model to a point cloud, and provides extreme robustness in the presence of sensor noise.

Table 3-2: Overview of ORP classifiers' applicability.

Classifier	Designed for	Works well on	Shortcomings
RGB Classifier	Waste sorting	Small objects	Does not classify shape
CPH Classifier	Inspection	Larger objects	Does not estimate object centers
CVFH Classifier	Manipulation	Larger objects	Requires extensive training data
Cup Detector	Detonator cups	Circular parts	Assumes objects lie on flat plane

Each classifier included in ORP has its own strengths and weaknesses. By using each component as a ROS node, systems gain the ability to mix and match components to meet specific task needs, as shown in the following three chapters.

Chapter 4: Use Case: Detonator Cup Picking

ORP's key strength is its flexibility. By enabling and disabling different combinations of the features described in Chapter 3, the library can supply the specific sensing needs for each platform's target tasks. To illustrate this flexibility, the next three chapters explore how ORP can be used in three diverse DOE-related use cases.

The first task is picking and placing detonator cups, which are small metal parts processed at LANL that require extensive manufacturing attention, but also delicate handling. The majority of this chapter focuses on the detonator picking task. The second task is remote inspection in a nuclear materials storage vault, which is discussed in Chapter 5. Although this task does not involve manipulation, ORP still proves useful in getting a position estimate for the various objects in the storage vault. The final task, as analyzed in Chapter 6, is a mixed-waste sorting task meant to mimic RoMaNS or HANDSS-55. Table 4-1 shows which features of ORP are used for each of the demonstrations.

Table 4-1: Usage matrix for DOE demonstrations.

	Demonstration		
Feature	**Detonator Picking**	**Remote Inspection**	**Mixed-Waste Sorting**
World Description	✓	✓	✓
Recognition Core		✓	✓
RGB Classifier			✓
CPH Classifier		✓	
CVFH Classifier			
Cup Detector	✓		

Each task is designed to target a different need prevalent at DOE laboratories (such as LANL), and all of them involve IIA. Two of the tasks involve manipulation, but do not use any custom fixtures; as will be seen in this chapter, simple custom gripper fingertips account for the majority of the design work in any of the demonstrations.

4.1 Performance Metrics

In each demonstration, the goal is to achieve the highest success rate possible on the overall task, as calculated by the following simple task completion reliability equation.

$$R_{task} = R_{vision} * R_{manipulation}$$

The reliability scores lie in the range [0,1]. As evidenced by the equation, even a perfect manipulation system cannot make up for a poor visual result, and good vision results can be marred by poor manipulation performance. For non-contact tasks, $R_{manipulation}$ is not applicable and can be thought of as equal to one.

In pursuit of quantifying R_{task} and providing context for the scores, this chapter focuses on measuring three metrics. The first metric is vision accuracy, R_{vision}: how effective are ORP's classifiers in sensing the environment? The second metric, which applies only to detonator cup picking and mixed-waste sorting, is manipulation accuracy $R_{manipulation}$: given the observed state of the environment (whether from ORP or any other system), how well can the robotic system complete the task required of it?

The final metric is speed: how quickly can a vision-enabled system complete a task? Task success is paramount, but DOE-critical applications are manufacturing processes, not single experiments. While some applications, such as plutonium pit production, consist of low-volume processes, the three tasks surveyed in this chapter have the potential to be performed hundreds or thousands of times, making speed a worthwhile metric to measure and optimize.

4.2 Detonator Cup Picking: Background

LANL subjects weapon detonation systems to the same high standards as the rest of the nuclear-industrial complex, with each part requiring multiple types of verification and validation to pass Lot Acceptance Testing (LAT). Detonator cups, which are small (<20 mm) metal parts of varying sizes, require LAT techniques including repeated visual

inspection for surface damage or blemishes. “Cross-sectioning” is another time-consuming LAT process wherein multiple lot-representative parts are mounted in resin, cut in half, and polished so that critical dimensions and tolerances can be checked.

The current method of detonator LAT has a number of shortcomings. First, LAT must assume that the tested parts make up a representative sample of the lot population, which is not always the case. In addition, methods such as cross-sectioning produce process waste, and destroy parts instead of non-destructively scanning using a method such as computed tomography (CT). Also, LAT is one of the most time-consuming parts of the manufacturing process—just the process of mounting parts for cross-sectioning can take hours of worker time and additional overnight periods for resin to harden; cross-sectioning just five cups can take over an hour per cup of worker time [67]. Any mundane task that takes this many man-hours is an ideal candidate for automation. Finally, labs such as LANL are focused on reducing worker injuries of all types, including Repetitive Stress Injuries (RSI) due to ergonomic issues. Tasks such as cup picking are a prime target for reducing ergonomic safety risks.

The goal of this automation task is to pick detonator cups out of a polystyrene part tray and place them on a metal peg nearby, which simulates the loading point on a part inspection machine. The details of this goal are driven by the fact that LANL has designed detonator LAT processes for humans, and not for automation. Cups arrive at LANL from a subcontractor in polystyrene trays that have shallow depressions. The cups may be lying in the depressions at any position or orientation, and workers must carefully pick the cups from the tray to load them into metrology machines, cross-sectioning mounts, or any other process fixtures.

The freedom of the cups to be in any position would pose a challenge to a traditional assembly line-style robot, and so the most straightforward way to automate the cup loading

process would be to redesign the tray in which the parts arrive. However, this may not be the most cost-effective solution, and it might require significant procedure changes for LANL or the subcontractor. In addition, the redesigned tray would have to be customized to each type of cup. This is exactly the type of process-specific fixture that ORP and IIA can eliminate.

4.3 Experimental Setup

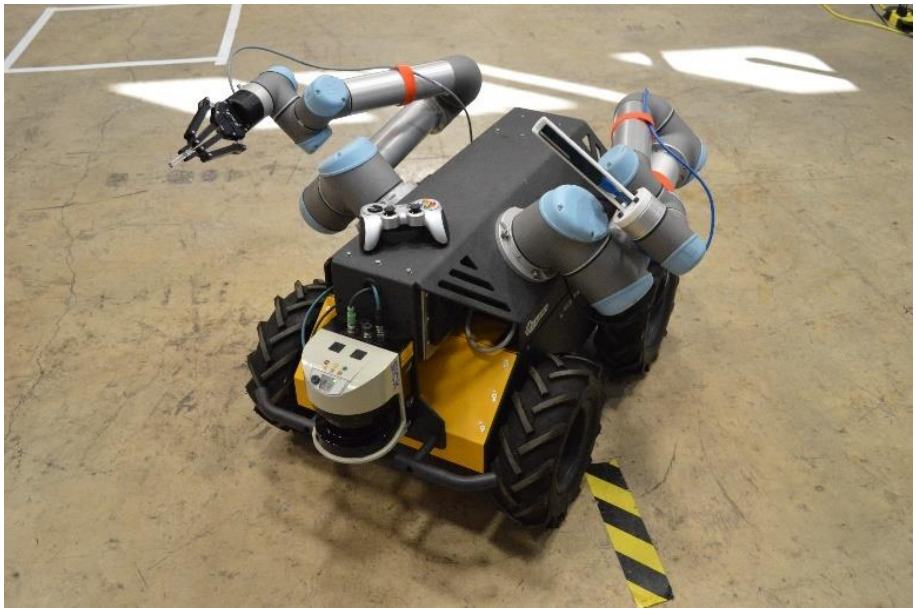

Figure 4-1: The Vaultbot mobile manipulator. Image courtesy of Andrew Sharp.

The apparatus for this experiment is based on an existing manipulator at NRG. The Vaultbot (Figure 4-1) mobile robot consists of two Universal Robotics UR5 6DOF serial manipulators mounted to a Clearpath Husky base. The left manipulator holds an Intel RealSense R200 camera, and the right arm can be fitted with a RealSense F200 camera and Robotiq 2-finger tripper, or a Robotiq adaptive 3-finger gripper. The Vaultbot also carries a SICK LIDAR sensor, laptop computer, and an internal router to connect the various

components. NRG uses this robot for mobile manipulation research, but it can also be used as a stationary platform, as is the case with this demonstration.

For this task, we use a single UR5 arm on the Vaultbot, equipped with an Intel F200 RealSense depth camera and a Robotiq 140 2-finger gripper. The gripper is capable of "adaptive" grasping, where the fingers' rotation is spring-loaded to allow grasping various object shapes, but the adaptive grasping was disabled for this task to achieve more precise manipulation. The mock polystyrene tray and cup surrogates sit on the bulkhead of the Vaultbot itself, removing the need to deal with robot navigation and placement. The task setup is shown in Figure 4-2.

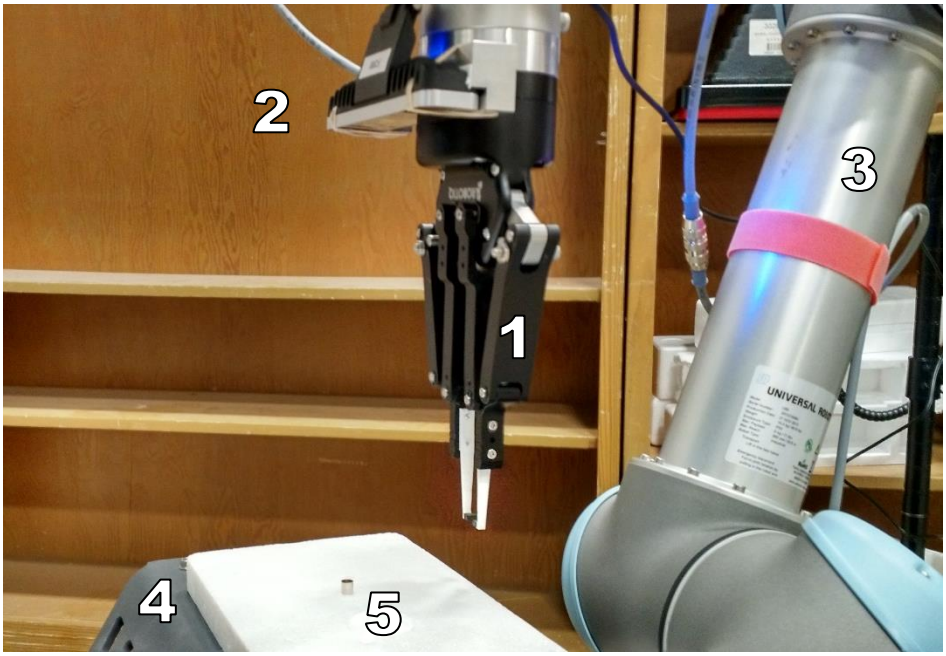

Figure 4-2: The hardware setup for the detonator cup manipulation task. 1: Robotiq 2-finger gripper. 2: Downward-pointing Intel RealSense F200 camera. 3: UR5 manipulator. 4: VaultBot bulkhead and polystyrene tray. 5: detonator cup.

4.4 PROCEDURE

The robot detects and moves cups from two positions on the tray onto two pegs mounted nearby on the Vaultbot's bulkhead. This demonstration runs continuously, and will continue to run without human input as long as cups exist in the workspace. The pick and place procedure proceeds as described by Figure 4-3.

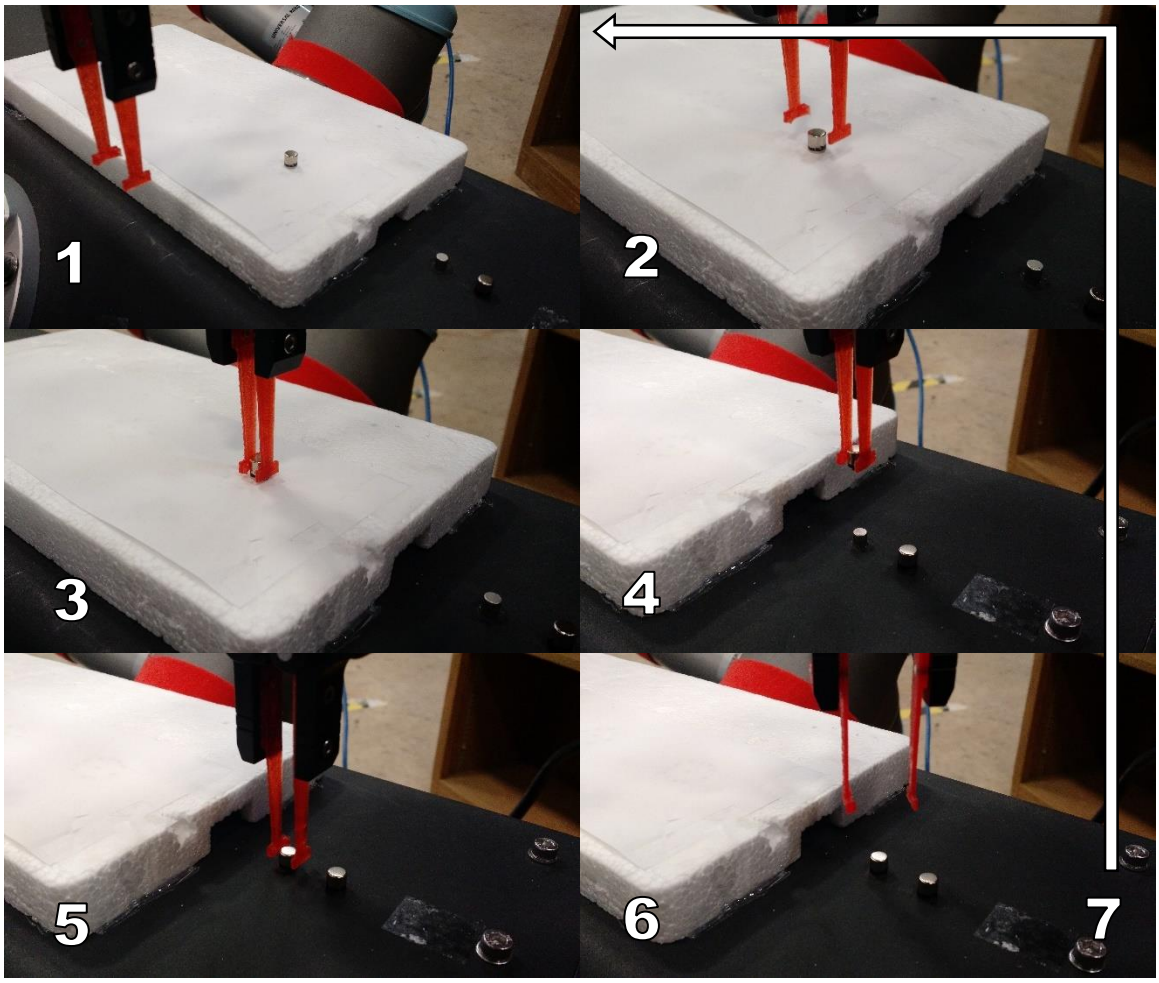

Figure 4-3: The 7 steps of the cup picking task. 1: Visual inspection pose. 2: Cup approach. 3: Grasp position. 4: Peg approach. 5: Peg placement. 6: Release and lift. 7: Repeat for next cup in tray.

In lieu of actual detonator parts, the system detects and manipulates 6.4 mm (1/4 in.) thin-walled stainless steel cups. The cups sit on a polystyrene platform, similar to the ones used at LANL, but without depressions. The cups are located randomly inside a 25 mm square area, which is the approximate size of the depressions in the trays used at LANL. Finally, we assume that the cup is oriented either with its opening pointing straight down or straight up to prevent the cup from rolling as it is manipulated.

To detect the cups, ORP uses the Cup Detector Classifier described in Chapter 3, which was designed specifically for this task. Overhead images of the cups were assumed to resolve the circular parts to a size between 30 and 60 pixels in diameter. These bounds dictated the tolerances of the Hough transform circle detector used in the classifier.

Under normal circumstances, the gray stainless steel stands out as darker than the light background, but under direct light, specular reflection on the polished cup surface causes the cup to appear brighter than the white tray. Therefore, both light circle and dark circle detections were used so cups could be located, even in diverse lighting conditions.

4.5 RESULTS

The first run of the detonator picking program resulted in a R_{task} of 74% and an average time of 26.13 s per cup, as seen in Table 4-2. The time measurement includes the time to perform a post-placement "alignment pinch" to improve the placement rate in the case of a slight cup misalignment on the target pin.

Table 4-2: Detonator pick and place statistics.

Trials	50
Total Time (s)	1307 s
Avg. Time per cup	**26.13 s**
Correct Pick/Place	37
Pick Errors	2
Place Errors	11
R_{vision} (from pick error count)	96.0%
$R_{manipulation}$	**77.1%**
$\boldsymbol{R_{task}}$	**74.0%**

The accuracy is grossly inadequate for LANL's mass-production of detonators. However, observing the reliability of the individual components, R_{vision} and $R_{manipulation}$, it was clear that most inaccuracies came from manipulation, not the vision system. The gripper needed to be redesigned to improve manipulation accuracy.

The gripper fingertips had been designed to pick small objects, but the contact surfaces were convex, meaning that even small errors in vision became exaggerated when attempting to pick up the round detonator cups using the original fingertip design. This is illustrated more clearly in Figure 4-4.

The finger redesign, performed in conjunction with another UT researcher, resulted in two different fingertips, one concave, and one flat (see Figure 4-5). The redesign was completed in less than 24 hours thanks to rapid prototyping, and the new fingertips allowed the gripper to self-center the object in its grasp, providing a stable, repeatable in-hand position. The redesign had the side effect of completely eliminating pick errors, because the new fingertips could even compensate for a slight inaccuracy in position estimation from the vision system. As a final benefit, after this change, the system could run without the need for an "alignment pinch" because of the increased precision, allowing faster times.

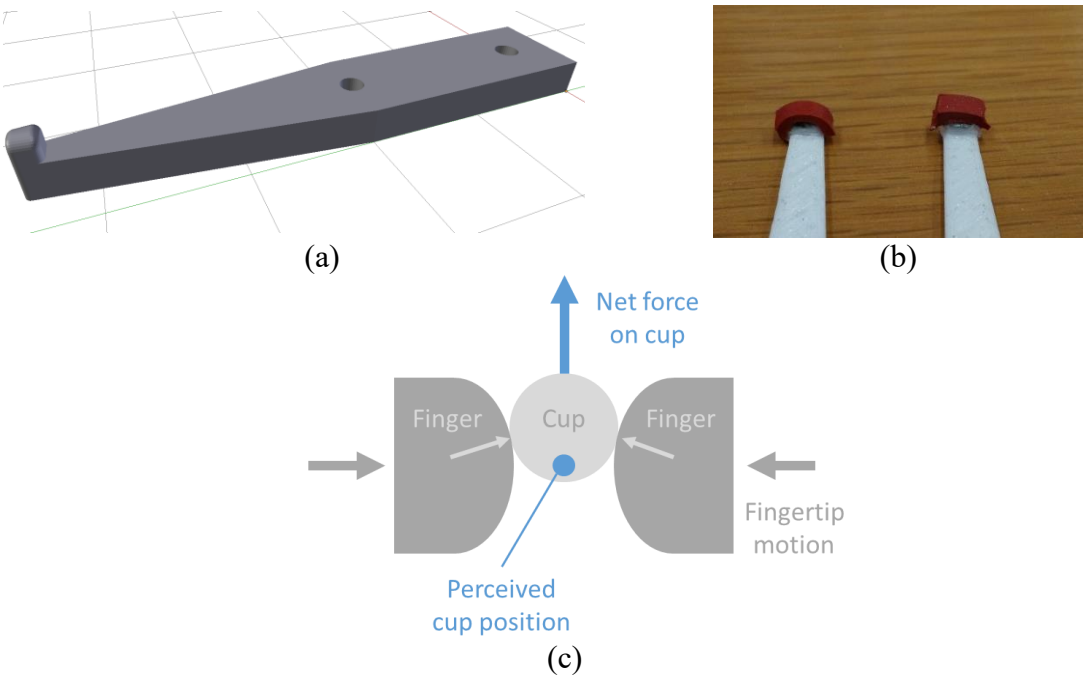

Figure 4-4: Diagrams showing fingertip design 1. (a) CAD model of fingertip. (b) Fingertip after adding tacky rubber pads for grip. Note the convex surfaces. (c) Illustration showing how convex gripping surfaces exaggerate position inaccuracy.

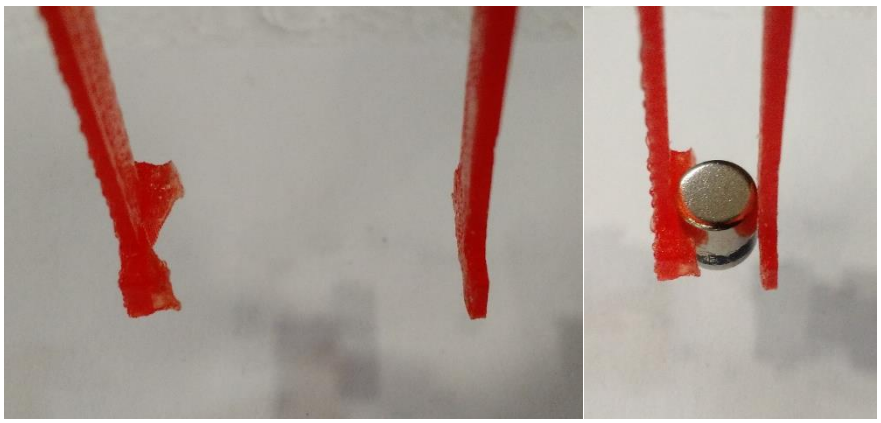

Figure 4-5: Fingertip design 2. The shallow "V" shape on the left fingertip causes the gripper to self-center cups as it grasps them. Left: gripper open. Right: grasping a cup.

The results of the finger redesign are apparent in Table 4-3. The pick accuracy improved to 100%, and the placement accuracy increased to 98%. Without the need for an alignment move, the task speed also increased by 48%, for a rate of 13.54 seconds per cup.

Table 4-3: Detonator pick and place statistics for two different finger designs.

Fingertip Design	**#1**	**#2**
Trials	50	100
Total Time (s)	1307 s	1354 s
Avg. Time per cup	**26.13 s**	**13.54 s**
Correct Pick/Place	37	98
Pick Errors	2	0
Place Errors	11	2
R_{vision} (from pick error count)	96.0%	100.0%
$R_{manipulation}$	77.1%	98.0%
$\boldsymbol{R_{task}}$	**74.0%**	**98.0%**

The final results are encouraging, because the success rate and speed are such that the system would make a suitable replacement for a human. A human operator can tend the machine and only step in when the robot makes a mistake, drastically reducing the ergonomic strain on the worker. This near-perfect success rate also opens the possibility of fully autonomous detonator cup manipulation and verification.

4.6 SUMMARY

There are two key takeaways from this use case. The first is that ORP can be successfully applied in the domain of small-part manipulation, making it suitable for machine loading and unloading without the need for expensive fixtures for each type of part. The second takeaway is that, while this thesis focuses on the vision systems in fixture-free manipulation, the gripper design can be as important as the vision algorithms in determining overall task performance. Or, put more generally, any automated system must ultimately achieve reliability in all subsystems to ensure task success. In some fixture-free tasks (such as this one), a good gripper can reduce the vision system's accuracy requirements and expand the overall system's applicability.

Chapter 5: Use Case: Remote Inspection

LANL maintains a storage vault for special nuclear materials, including radioactive samples and materials that date back to the Cold War era. This storage vault, although vital for national security, is dangerous for workers because of its high levels of radiation. Despite worker risk, the sensitive nature of the materials means that the vault must be inspected often to ensure that it has not been compromised by environmental or human actions. To achieve ALARA dose rate for workers, LANL and NRG are working to deploy a mobile inspection robot into the vault which can autonomously perform radiation scans and inspect the shelves of the vault, verifying the locations of the samples stored there [68].

To support the goal of remote inspection, this task seeks to perform a comprehensive scan of the objects in the vault, identifying and locating the storage containers on the various shelves of the vault. In this demonstration, ORP is applied to a less constrained position detection task, but still performs reliably, providing the ability to repeatably verify sample positions in the vault.

5.1 Experimental Setup

The newest mobile robot at NRG is a heavily-augmented Adept Pioneer LX (shown in Figure 5-1). The system includes the mobile base and a wide variety of peripherals, including Ludlum alpha radiation detectors, an Intel Next Unit of Computing onboard vision processing computer, a slower navigation processing computer, and an Arduino-monitored bump sensor for collision detection. NRG uses the Pioneer LX as a mobile platform for research autonomous behaviors that do not involve manipulation, such as remote inspection and surveying. Despite this, the Pioneer still needs an object detection system such as ORP. Although it does not require manipulation capabilities, during an autonomous inventory task, the robot must be able to accurately classify objects and

determine their position.

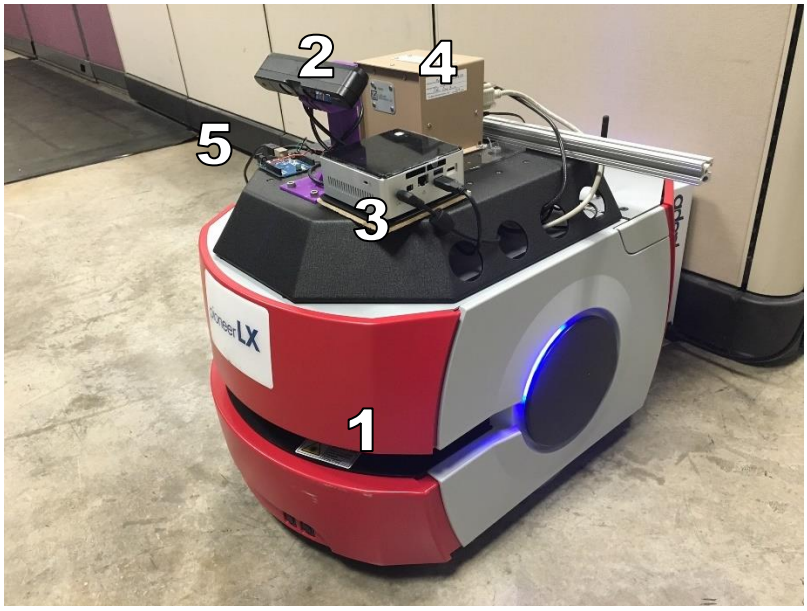

Figure 5-1: The Pioneer LX mobile robot. 1: LIDAR sensor aperture. 2: Asus Xtion Pro RGBD camera. 3: Vision processing machine (Intel NUC). 4: Ludlum 3-channel radiation counter box. 5: Arduino Uno bump sensor controller.

5.2 PROCEDURE

The robot undocks itself and autonomously navigates to one shelf at a time, scanning the objects on each shelf and comparing them with an internal database of known containers. By monitoring position in addition to serial numbers, the system can tell if an object on a shelf has been moved, tampered with, or replaced with a different object. This is a critical capability to ensure that nonproliferation requirements are met at LANL. As the scan proceeds, the robot saves images of each shelf, and objects are flagged as anomalies if the data does not match that collected during the last inspection[5].

[5] During this time, the robot passively scans the floor using its alpha radiation detector. If the detector exceeds a radiation threshold, the robot will immediately throw an alarm. While this behavior is not important to the analysis presented in this thesis, interested readers can learn more about the capabilities of the NRG Pioneer system in [68]

For this inspection task, the system enables the CPH classifier, which has been pre-trained on cylindrical metal canisters. CPH is the ideal classifier for this task because it has been shown to work well on point clouds of cylindrical objects, even in the presence of noise and incomplete data, and is extremely fast to calculate. After filtering removes the bottom surface of the shelf from the data, the cans resolve into clean detections that can be used to infer can position. By matching the position with the barcodes attached to the can, the higher-level task architecture can keep track of the position of different sample canisters between inspections.

The careful reader may point out that, as described in Chapter 3, the CPH classifier does not give an accurate position estimate of an object, instead only calculating the center of the observed point cloud. While this is true, the inspection is only concerned with the relative motion of an object between scans (to check if the object has been tampered with or removed). The robot observes the can from nearly the same vantage point every time, making the point cloud center a stable and acceptable position estimate for the can.

As in the previous use case, we do not use the actual specimens encountered at LANL—radioactive sample canisters are replaced with surrogates. The surrogates are stainless steel 137 mm (5.4 in) paint cans. The cans are labeled with a 2D black and white augmented reality barcode to allow the system to differentiate between the unique cans. ORP's CPH classifier assumes that the cans are all of the same size. LANL has recently standardized the containers used for storage to the SAVY-4000 nuclear material storage container [69], so this is a reasonable assumption. The SAVY-4000 cans are not as polished as the surrogate paint cans, which actually makes the task more difficult when using surrogates. The mirror-finish paint cans result in poor-quality point clouds compared to those obtained for a SAVY-4000; however, the CPH classifier is able to compensate for

this issue, as will be seen in the next chapter. As a final simplification, the system assumes that all the cans stand upright on their shelves.

The current hardware design of the robotic system does not allow inspecting any shelves besides the bottom one. However, the inspection procedure would be identical for the higher shelves. Readers are referred to the upcoming paper [68] for more information on the overall inspection process. For the purposes of this thesis, ORP is integrated only with the bottom-shelf inspection system.

5.3 RESULTS

The same task metrics (speed and accuracy) as in Chapter 4 are relevant to this task, with two minor adjustments. In contrast to detonator cup picking, remote inspection does not involve any manipulation. Rather, the system needs to verify the location of objects in comparison to the last inspection. Therefore, repeatability, not accuracy, is the goal of the computer vision system in this context. Also, the lack of manipulation means that the manipulation success rate is not applicable for this task.

During the task, the Pioneer mobile robot was tasked to repeatedly drive between two shelves. At each shelf, the system collected 10 seconds of vision data and located the storage canister in the field of view. Because of sensor noise, the position estimations varied from frame to frame, and we seek to minimize this noise-induced error.

The CPH classifier used in this task calculated point cloud centers to return the object locations in 3D space. Figure 5-2 shows a graph of a typical set of pose estimations. For each point, the Euclidean distance from the mean was calculated, and this was taken as the estimation error. The error across the different runs is tabulated in Table 5-1.

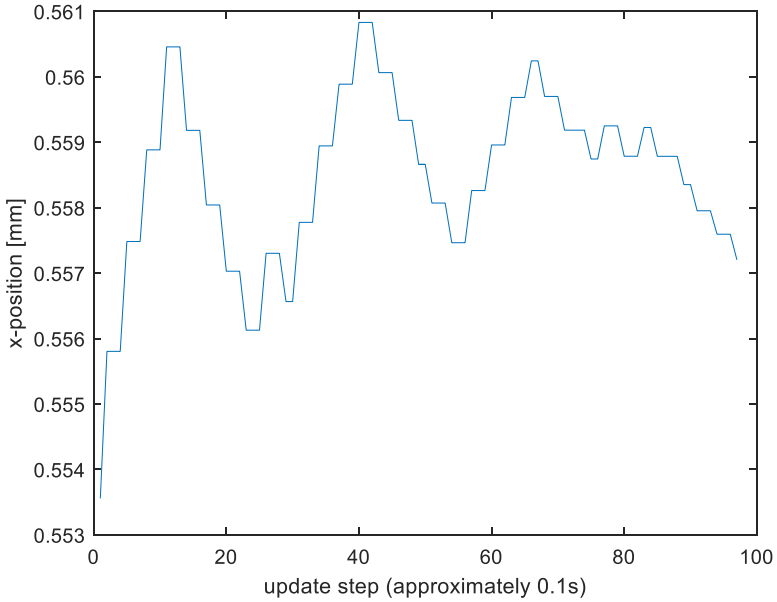

Figure 5-2: A typical graph of canister pose estimation. Note the small scale of the y-axis. The data shown is from shelf 4, trial 4.

Table 5-1: Precision values for the remote inspection task.

		Shelf 2				**Shelf 4**	
Trial	n	Std Dev of Error [mm]	Max Error [mm]	Trial	n	Std Dev of Error [mm]	Max Error [mm]
1	96	2.39	14.09	1	96	0.21	1.01
2	96	0.11	0.53	2	97	0.66	3.06
3	95	0.13	0.71	3	97	0.88	5.77
4	95	0.17	0.80	4	96	2.04	11.34
5	95	0.24	1.19	5	95	1.10	5.38
6	50	1.65	4.48	6	97	0.72	2.71
7	95	0.36	1.90	7	96	1.80	8.06
8	96	0.33	1.19	8	96	0.66	3.76
9	95	0.17	0.84	9	97	0.82	3.97
10	94	0.26	1.17	10	97	0.73	3.73
Max (worst-case)		2.39 mm	14.09 mm			2.04 mm	11.34 mm
Average		**0.58 mm**	**2.69 mm**			**0.96 mm**	**4.88 mm**

ORP's recognition core runs at a rate of approximately 10 Hz, so each run resulted in a maximum of 100 data points. At some update steps, less data was observed because

the RGBD sensor did not have data available for processing. In this case, ORP was still able to maintain the object position using the prior knowledge.

In both shelf inspections, the pose estimations were extremely stable, with an overall maximum error of 14.09 mm across all measurements. In general, operators could expect this system to be repeatable within +/- 3 mm at a confidence level over 99% (this is the approximate 3σ threshold as determined from Table 5-1).

5.4 SUMMARY

The results of this demonstration are encouraging for detecting canisters that have been moved or tampered with. Being able to detect any motion greater than 3 mm is most likely better than human performance, and this accuracy comes in the presence of a major source of experimental error. As mentioned earlier, the surrogate canisters used in this task are polished to a near-mirror finish. This is in contrast to the actual SAVY cans used at LANL. The polished surface of the surrogates results in a poor-quality point cloud, shown in Figure 5-3. The point clouds for SAVY cans, while still imperfect, give significantly more depth data. Despite this noisy and incomplete depth reading, the CPH classifier performs extremely well, and so operators could expect even better performance on a SAVY-4000.

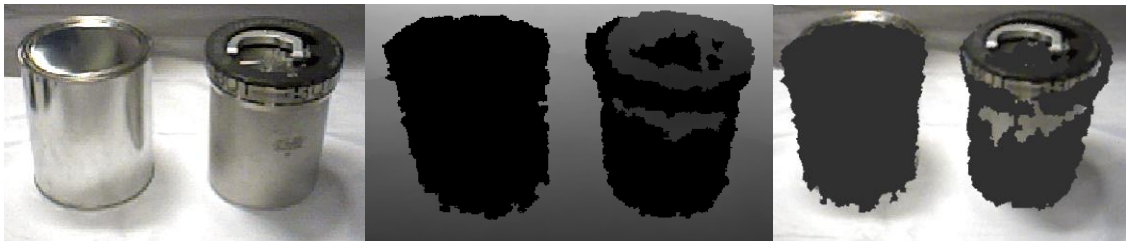

Figure 5-3: Differences in point cloud data between the SAVY-4000 storage canister and the paint can surrogate. In each image, the surrogate is on the left and the SAVY-4000 on the right. From left to right: color image, raw infrared (IR) image (black indicates no data), and combined color/IR image.

This chapter has shown a feature of ORP that extends beyond the 2D-based vision used in the detonator picking task: the ability to repeatedly and reliably detect and locate objects, even from diverse viewpoints in 3D space. The next chapter will expand this functionality further, performing object classification as well as pose estimation.

Chapter 6: Use Case: Glovebox Mixed Waste Sorting

The uncompleted HANDSS-55 waste repackaging system was designed for 4 primary functions in the repackaging of 55-gallon drums of mixed waste [70]:

1. An automatic drum and liner opener (AD&LO)
2. A visual inspection and sorting table
3. A transuranic-waste repackaging module (TWRM)
4. A process waste reduction module (PWR)

Step 2 in this procedure, the inspection and sorting table, is the task that best displays the benefits of using ORP in the automation pipeline. In this step, the system must sort pieces of mixed waste based on the level and type of radiation they emit, and pack these items into containers. HANDSS-55 was designed in 2002 to use human supervision of the vision pipeline, but now vision systems are robust enough to allow for fully autonomous sorting.

The goal of this demonstration is to sort objects, which lie on the floor of a glovebox in a random configuration, into different containers based on their radiation type. This goal seeks to emulate one of the processes targeted for automation by HANDSS-55 and the in-development RoMaNS sorting and segregation system.

6.1 Experimental Setup

The glovebox-deployed manipulator at NRG can represent automated glovebox tasks of all types, from nondestructive assay to pit manufacturing/processing to waste sorting/handling. As shown in Figure 6-1 and Figure 6-2, the system consists of a Yaskawa Motoman SIA5D 7-degree of freedom manipulator semi-permanently mounted in the transfer port of a one-ton lead-lined glovebox. The manipulator can use a number of end effectors, including a wrist-mounted RGBD camera and a 3-finger Robotiq gripper. A

variety of different cameras, such as the Asus Xtion Pro RGBD sensor, can also be mounted outside the lead-lined windows of the glovebox, allowing for reliable data for ORP while still maintaining environment realism and applicability as a surrogate research testbed.

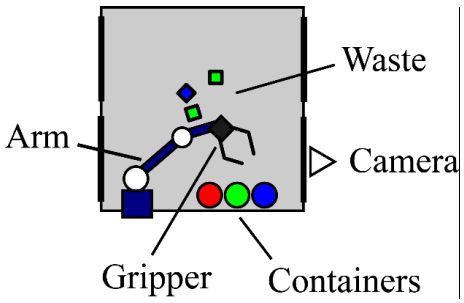

Figure 6-1: Diagram (top view) of the automated glovebox configuration.

Figure 6-2: The glovebox port-deployed SIA5 manipulator with Robotiq 3-finger gripper.

6.2 PROCEDURE

Three metal "nuclear material cans" (of the same type used in the remote inspection task) are placed at predetermined positions in the glovebox. Each can stores one color (radiation type) of object—red, green, or blue. The "waste" objects begin randomly arranged in an area that is both accessible by the robot and visible to the camera. This

region has an approximate area of 0.5 square meters. The system then sorts the waste according to the pseudocode listed in Figure 6-3 [71].

```
1  detect objects
2  classify objects
3  while(object list is not empty):
4    try
5      move over object
6      grasp object
7      verify grasp
8      lift object
9      move over appropriate receptacle
10     release object
11     remove object from list //success, next
12   catch
13     if(object grasped):
14       put object back
15       remove object from list //failure, skip
16   move to home
17   wait for user confirmation //optional
18 end
```

Figure 6-3: Mixed waste sorting algorithm.

The system begins by classifying objects in the scene. This is where ORP performs color classification and pose estimation of the waste in the glovebox. The loop that begins on line 3 sorts each object using robot motion controlled by ROS and MoveIt!, coupled with custom libraries written at NRG to control the Robotiq 2-finger gripper. In line 16 of the pseudocode, the robot moves out of the way after each item is sorted so that the operator can inspect the workspace before proceeding. Figure 6-4 and Figure 6-5 show the robot during the sorting process.

Figure 6-4: The robot shortly after sorting an object into a waste container (pseudocode line 10). This picture was taken from the approximate viewpoint of the RGBD camera mounted on the glovebox.

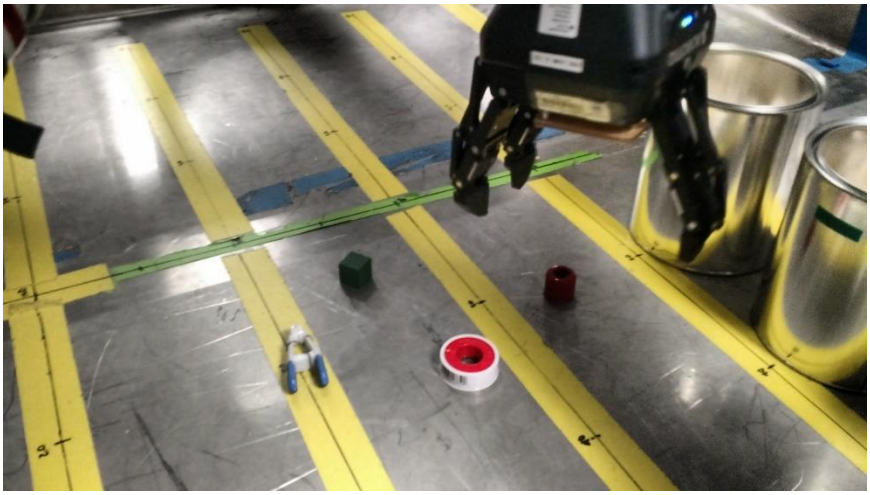

Figure 6-5: The robot moves over an object before sorting it (pseudocode line 5).

Line 17 of the pseudocode provides an optional operator confirmation before continuing each step of the sorting task. When used, the system's autonomy decreases, but allows a human operator to exercise judgement similar to the way that HANDSS-55 was designed. This option allows the system to perform extremely dangerous tasks that require

close supervision, or simply to afford greater trust in the program and machine, but removing this line allows the system to proceed fully autonomously.

In this task, ORP runs with the RGB classifier enabled. By using 3D vision, the brightly colored red, green, and blue objects can be segmented from the yellow and green tape on the glovebox floor before 2D color classification. This allows the task to run without being "tricked" by patterns or shapes on the glovebox floor. The only errors occur during manipulation, or because of incorrect color classification.

This demonstration depends on a number of assumptions due to the danger of the actual task being emulated. Waste sorting can be almost entirely unstructured, making the automation process challenging. The objects being sorted may have a wide variety of shapes, sizes, and weights. However, "the system must be capable of identifying an object well enough to quickly pick it up—even though there is no reference to compare it with" [11]. To help simplify the problem to be achievable using the relatively small working range of the SIA5D robot in the automated glovebox, all objects in the waste sorting process are assumed to be small (less than 50 mm at their largest extent), while still being large enough to be sensed by the Xtion RGBD camera (at least 10 mm). This still allows a diverse set of objects to be sorted, including drill bits, wooden blocks, pieces of tape, etc. while reducing the problem to manageable levels of size uncertainty.

To make the development of the automation system less hazardous, color stands in for radiation type. For example, red-colored objects can be treated as alpha particle-emitting waste, and green objects can be thought of as gamma ray-emitting waste (the exact mapping of color to radiation is irrelevant). Objects can have multiple colors, but the system will use the most dominant color when determining categorization. We assume that color is a suitable surrogate for radiation signature, which implies that radiation signatures are easily discerned from each other—not a large leap given the current state of radiation

measurement. However, other researchers at NRG are also researching radiation detection and surveying [72], which could eliminate the need for color as a surrogate in the future.

6.3 RESULTS

The remote inspection task in Chapter 5 shows the applicability of ORP to a different domain than Chapter 4's detonator picking demonstration: the steady, reliable detection of objects from an autonomous mobile system. But what about an environment where the objects' identities are unknown, and the system must not only locate them but classify them? The mixed-waste sorting task is such an environment, and ORP performs well here, too. Table 6-1 summarizes the results of running the mixed-waste sorting task 10 times, each time with 9 objects in the workspace. The 9 objects, three of each color, are shown in Figure 6-6. Overall, the system achieved a classification accuracy of 95.9% and a sorting success rate 94.6% of the objects correctly. In four cases, an object was sorted into the wrong container. Three of these incorrect sorts were the same object, labeled as 6 in Figure 6-6. This object contains the least color information, and so presents the largest challenge to the color classifier used in this task. The final incorrect sort was object 6, which means that 7 of the 9 objects had a 100% classification accuracy.

In one case, object 7 became lodged between the two fingers of the Robotiq gripper and failed to drop into the sorting container when the gripper was opened. This error (listed as "Other" in Table 6-1) reduces the overall task accuracy compared to the visual classification accuracy.

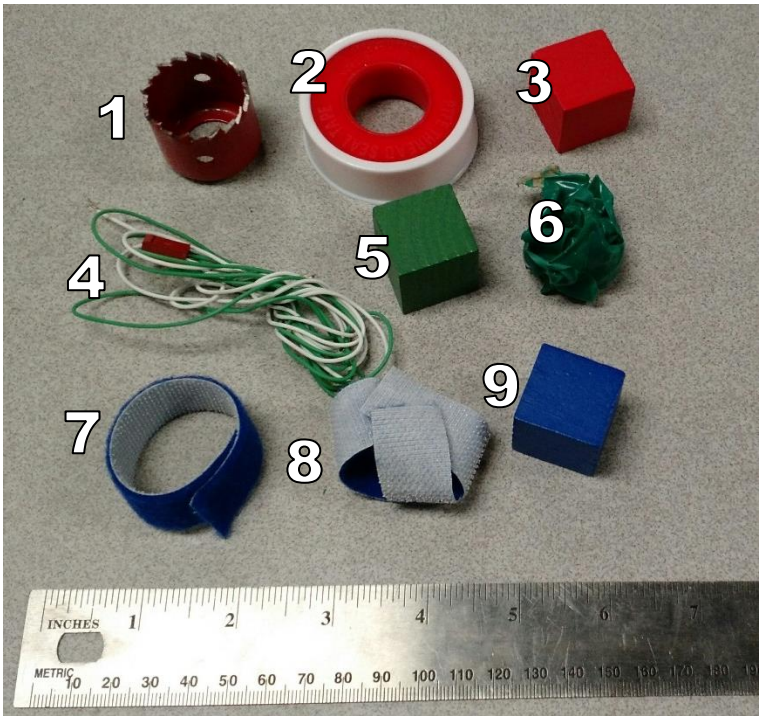

Figure 6-6: The 9 objects used in the mixed-waste sorting task. All objects had a 100% sorting success rate, with the exception of objects 6 and 8.

Table 6-1: Summary of results for the mixed waste sorting task over 10 runs.

Run #	Time (s)	Items	Correct	Incorrect	Other	Correct	Incorrect	Other	Time per Item (s)
1	247	6	6	0	0	100.0%	0.0%	0.0%	41.2
2	590	5	5	0	0	100.0%	0.0%	0.0%	118.0
3	432	9	9	0	0	100.0%	0.0%	0.0%	48.0
4	628	8	7	1	0	87.5%	12.5%	0.0%	78.5
5	223	9	8	1	0	88.9%	11.1%	0.0%	24.8
6	258	9	9	0	0	100.0%	0.0%	0.0%	28.7
7	283	6	6	0	0	100.0%	0.0%	0.0%	47.2
8	245	8	8	0	0	100.0%	0.0%	0.0%	30.6
9	229	8	6	1	1	75.0%	12.5%	12.5%	28.6
10	249	6	6	1	0	100.0%	16.7%	0.0%	41.5
Overall	338.4	74	70	4	1	**94.6%**	**5.4%**	**1.4%**	**48.7 s**

The system took 48.7 seconds on average to sort each item. This was calculated by dividing the total time for the sorting run by the number of objects sorted on that run. The total time for the sorting run is controlled by the sorting algorithm, which is reproduced in Figure 6-7 from earlier in the chapter for clarity. In some trials, the robot could not reach one or more objects because of kinematic constraints. In this case, the planning time to determine that the object was out of the workspace was still counted in the overall task time. Executing motion plans accounts for most of the task time (lines 5-9 in Figure 6-7), with perception and path planning completing quickly in comparison. The total time for planning all motion paths associated with a single object varied, but was always less than four seconds, or less than 10% of the average time per object.

```
1  detect objects
2  classify objects
3  while(object list is not empty):
4    try
5      move over object
6      grasp object
7      verify grasp
8      lift object
9      move over appropriate receptacle
10     release object
11     remove object from list //success, next
12   catch
13     if(object grasped):
14       put object back
15       remove object from list //failure, skip
16   move to home
17   wait for user confirmation //optional
18 end
```

Figure 6-7: Mixed waste sorting algorithm.

6.4 SUMMARY

Success rates for this task are nearly as good as those for the detonator picking task, where the objects identities were already known. The detonator cups were confined to a

very small area, but this task showed that ORP works for manipulation in larger domains as well. The automation system used in this demonstration (coupled with ORP software) could well represent a proof-of-concept for the HANDSS-55 sorting table, finally developing the capabilities meant for a system designed 15 years ago.

The last three chapters laid out three nuclear-industrial automation tasks motivated by the requirements of nuclear-industrial manufacturing, nuclear materials monitoring, and waste management. The results showed ORP's performance across different domains, including achieving a vision reliability above 95% in two different manipulation tasks. In all cases, ORP's flexible classifier system allows easy integration of perception technology without reinventing the underlying computer vision framework. None of the tasks in this chapter could be performed with a traditional assembly-line type industrial machine, but by incorporating ORP, the systems can "abstract away" perception to allow a system to intelligently perform specific tasks based on changes in its environment.

Perhaps more importantly than completing specific tasks, ORP's success helps show the strength of IIA. The reduction of repetitive motions (as seen in the detonator cup picking task) will reduce RSI injuries, making labs safer work environments. Using IIA to enable robots to autonomously operate in radioactive environments (as seen in the remote inspection and mixed-waste sorting tasks) is in line with ALARA principles. Each application of IIA offers the opportunity improve safety and productivity in different regions of the nuclear-industrial complex. Once we have proven their robustness, IIA systems such as the Vaultbot, Pioneer LX, and automated glovebox can make up the first wave of IIA deployed at DOE sites, such as LANL and SRS.

Chapter 7: Conclusions and Future Work

The objective of this thesis, as defined in the introduction, was to develop a system able to overcome the strict positioning requirements of existing automation systems, with the end result of reducing the amount of tooling and fixtures required to complete tasks common in a nuclear manufacturing environment. The thesis introduced the Object Recognition and Pose Perception library (ORP), which is designed to meet these goals.

7.1 Research Summary

ORP is a flexible ROS-integrated software package that allows any number of sensor-based classifiers to work together to provide a representation of an automation system's environment. By locating various objects in the environment visually, the package allows the automation system to reduce its dependence on a strictly controlled environment. In the future, new classifiers, including those that use other sensing modalities (such as radiation, barcode recognition and text recognition) could be integrated into ORP to help provide further details about the environment state.

ORP automatically maintains a list of objects in the world and does much of the "legwork" required for a robust vision system, such as point cloud filtering. This means that individual classifiers can quickly be designed, implemented, and interchanged to provide the best vision results. ORP is currently includes a recognition core, 3D point cloud utility library, world description and 4 classifiers: RGB, CPH, CVFH, and cup detection. The diverse set of tools contained in the ORP toolbox allows it to apply to DOE use cases such as part manipulation and machine tending, remote inspection, and waste sorting.

None of the ORP demonstrations required special tooling or fixtures. The custom fingertips required to improve the detonator picking task to acceptable levels is the only exception, and even that task was not design-intensive. Therefore, we can conclude that

computer vision (as provided by ORP) can allow robots to work in much more unconstrained environments than before, by enabling systems to meet high standards of accuracy, precision, and robustness.

The ORP-enabled systems achieved high levels of reliability and repeatability, as shown in Table 7-1, which summarizes the results gathered in Chapters 4-6. The reliability values for the demonstrations present a strong case for technology adoption. The systems also operate quickly enough to be suitable for deployment in a nuclear-industrial environment.

Table 7-1: Summary of reliability and speed for the three ORP-enabled demonstrations.

Task	System	Reliability	Time per Task
Detonator Pick/Place	Vaultbot & UR5	98.0%	13.54 s
Remote Inspection	Pioneer LX	±2.69 mm	10 s
Mixed-Waste Sorting	Glovebox & SIA5	94.6%	48.7 s

As a corollary, the thesis also sought to show that computer vision is ready to be implemented in DOE applications. Indeed, at least two of the three demonstrations in this thesis would have been impossible for a machine without computer vision. DOE will need to implement reliable computer vision to realize a vision of IIA, and ORP has shown that this goal can be achieved using current technology.

7.2 Future Work

ORP explores capabilities previously unused in the nuclear research sector, and the possibilities for improvements and extensions span all of computer vision and robotics. For example, while the software package shows great promise for integration into several tasks relevant to DOE, it still has not been applied to many other domains. Co-robotics, where the robot works in tandem with a human, is one such domain. Another is complex procedures, in which a robot needs to use multiple tools or end effectors to complete a task.

A third domain would be “hands-off” tasks such as the DARPA Robotics Challenge [73], where teleoperation is not possible (whether because it is forbidden by rules or limited by other extenuating factors), or would be too slow to complete critical missions at satisfactory speeds. ORP could grow to include classifiers and supporting packages for these use cases as well. However, computer vision and sensor-based environment modeling comes with their own unique challenges.

7.2.1 Pose Estimation Improvement for Complex Objects

As seen in Chapter 4, ORP’s use cases are restricted to environments where the objects being detected are fairly uniform, such as identical detonator cups or mixed waste of a specific size. The somewhat simplistic nature of the classifiers used dictates this limitation. CVFH, the most state-of-the-art classifier currently implemented in ORP, was developed in 2011, and since then the field of 3D pose estimation has improved considerably. Many of the techniques discussed in Chapter 2 offer far better pose estimation results and classification discernment than those used in ORP’s relatively simply classifiers. However, many of these techniques are as-yet unproven in the commercial sector, and would need to be verified before being used in a high-consequence nuclear environment.

7.2.2 Reducing the Data Burden

Another challenge that newer machine learning-based vision systems face is the “data burden.” This thesis has shown that simpler computer vision techniques are often adequate for basic tasks, but for more complex procedures, even more intelligent automation may be required. The newest of computer vision techniques, such as the random forest-based detectors mentioned in Chapter 2, far outperform older methods such as CPH in terms of raw results. However, the accuracy of computer vision results in state-

of-the-art research is strongly correlated with the amount of data used to train the machine learning-based portion of the pipeline. This is not a problem in many domains, such as 2D image classification, where the Internet provides millions of diverse images for free. However, in the 3D domain, more data is required to train a high-quality neural-network based vision system. This requirement is related to the "curse of dimensionality" [74], where training a high-dimensional inference system, such as an advanced neural network, requires a dataset that grows exponentially with the size of the representation. While using data-driven techniques, RGBD sensors, and classifiers such as CVFH may theoretically give the best results, this may be an unrealistic goal given the amount of data required.

While some prior work has looked into generating "synthetic" data sets by rendering CAD models from various viewpoints ([20], [75]), these approaches cannot accurately generate the RGBD data encountered in the real world. For example, mirrored surfaces (or highly reflective ones such as polished metal) often result in poor depth readings from an RGBD camera, but a simulated rendering is unable to accurately model this phenomenon. In addition, objects found in the real world are often partially occluded, and synthetic training approaches do not take occlusion into account.

Active recognition, where a robot is able to manipulate its environment to gather training data it needs, also holds promise for reducing the data collection burden. For example, researchers have developed systems that can use vision, motion data, and even sound collected while an object is being picked up, poked, or dropped to identify the object [76]. Other techniques exist for determining the "next best view" of an object, turning the object to get a more accurate classification result [77], [78]. If DOE wishes to deploy a vision system that can quickly be adapted to new task domains, more work is needed in the areas of low-data learning, active recognition, or generating simulated/synthetic data.

7.3 Concluding Remarks

In this thesis, robots manipulated parts, sorted waste, and inspected high-security facilities. These capabilities currently exist only in the laboratory, but we now see they are robust enough for adoption, verification, and deployment. Computer vision holds great promise for the future of intelligent industrial automation. Once the other challenges to technology adoption have been overcome, we can use computer vision and IIA in dangerous or repetitive tasks across the DOE, reaching unprecedented levels of safety and productivity in the process.

Glossary

This glossary is divided into two parts. The first part lists general terms and acronyms relevant to robotics and this thesis. The second part contains computer vision algorithms and techniques, including citations to source material where appropriate.

General Terms

ALARA	As Low As Reasonably Achievable [12], the desired worker radiation dose, as defined by the US Code of Federal Regulations
ARIES	Advanced Recovery and Integration Extraction System [3], an automated plutonium weapons dismantlement system at LANL
BRETT	Berkeley Robot for the Elimination of Tedious Tasks
DOE	Department of Energy
DOF	Degrees of Freedom
EU	European Union
HANDSS-55	Handling and Segregating System for 55-gallon drums of mixed waste [10], a canceled 2002 DOE project
IIA	Intelligent Industrial Automation
LANL	Los Alamos National Laboratory, a DOE lab in New Mexico
LAT	Lot Acceptance Testing
NRG	University of Texas Nuclear and Applied Robotics Group [79]
RoMaNS	Robotic Manipulation for Nuclear Sort and Segregation [13], an EU-funded mixed-waste processing project
ROS	Robot Operating System [80], a flexible framework for writing robot software
RSI	Repetitive Stress Injury

SRS	Savannah River Site, a DOE site in South Carolina. The site includes both Savannah River National Lab and older facilities that are in the process of being decommissioned
ORP	Object Recognition and Pose Perception, a C++ library for easy integration of multiple computer vision techniques

COMPUTER VISION CONCEPTS

CNN	Convolutional Neural Network [31] [32], a special type of neural network designed to work on spatially connected data, including images
CPH	Circular Projection Histogram [25], a global 3D descriptor developed at UT-NRG
CVFH	Clustered Viewpoint Feature Histogram [30], a global 3D descriptor that extends VFH
Descriptor	A mathematical representation 2D or 3D image data at a single point (local) or over an entire image (global), designed to be more compact and easier to work with than raw pixel or point cloud information
Chamfer Distance	[81] A metric used to determine how closely two parts of an image match. The chamfer distance is lowest when the two images are identical
DOG	Difference of Gaussian, an image filter that detects sharp changes in pixel intensity

FLANN	Fast Approximate Library for Nearest Neighbors [23], a multi-language library that allows high-dimensional data to be quickly searched with high accuracy
FPFH	Fast Point Feature Histogram [64], a local 3D descriptor
GFPFH	Global Fast Point Feature Histogram [19], a global 3D descriptor based on FPFH
HOG	Histogram of Gradients [17], a local 2D descriptor
ICP	Iterative Closest Point [28], an algorithm often used for aligning point clouds
LINEMOD	Multimodal Memory Linearization [27], a global 3D descriptor
OUR-CVFH	Oriented, Unique, and Repeatable Clustered Viewpoint Feature Histogram [24], a global 3D descriptor that extends CVFH
RANSAC	Random Sample Consensus, an iterative method for fitting a model to data points, usually used to find planar surfaces in a point cloud
RGB	Red, green, and blue, the three color channels in a digital image
RGBD	Red, green, blue, and depth, the type of data produced by cameras such as the Microsoft Kinect
PCL	Point Cloud Library [22], a software library for processing RGBD data. PCL includes implementations of FPFH, GFPFH, VFH, and CVFH
Point Cloud	A set of 3D points, usually obtained from an RGBD sensor such as the Intel RealSense
SIFT	Scale-Invariant Feature Transform [16], a local 2D descriptor
VFH	Viewpoint Feature Histogram [20], a global descriptor that extends GFPFH

Bibliography

[1] U.S. Government Printing Office, *Senate Hearing 108-97*. Washington, DC: U.S. Government Printing Office, 2003.

[2] Los Alamos National Laboratory (LANL), "Radiological Worker II Training Manual." LANL, 2008.

[3] C. J. Turner, T. A. Harden, and J. A. Lloyd, "Robotics in Nuclear Materials Processing at LANL: Capabilities and Needs," pp. 701–710, Jan. 2009.

[4] Yaskawa Motoman Robotics, "SIA5D." Jul-2014.

[5] J. L. Thompson, *Redesigning the human-robot interface: intuitive teleoperation of anthropomorphic robots*. Austin, Tex.: University of Texas, 2014.

[6] "UR Robots Benefits," *Universal Robots*, 2015. [Online]. Available: http://www.universal-robots.com/products/ur-robot-benefits/. [Accessed: 02-Apr-2016].

[7] "Baxter Research & Manufacturing Robotics," *Rethink Robotics*. [Online]. Available: http://www.rethinkrobotics.com/products-software/. [Accessed: 14-Mar-2016].

[8] "UR5 robot," *Universal Robots*, 2015. [Online]. Available: http://www.universal-robots.com/products/ur5-robot/. [Accessed: 02-Apr-2016].

[9] T. E. Samspon, "Integrated nondestructive assay solutions for plutonium measurement problems of the 21st century," Los Alamos National Laboratory (LANL), Los Alamos, NM, 1997.

[10] C. M. Frazee and M. E. Brennan, "Handss-55: A Tru Waste Repackaging System for the Savannah River Site," presented at the Waste Management 2001 Symposium, Tuscon, AZ, 2001.

[11] R. Smith, "Helping HANDSS for Sorting Waste," *Radwaste Solutions*, vol. 9, no. 6, pp. 47–52, Nov-2002.

[12] "US Code of Federal Regulations 10CFR20.1003." 1991.

[13] European Commission, "European Commission: CORDIS: Projects & Results Service: Robotic Manipulation for Nuclear Sort and Segregation (RoMaNS)." European Union, 15-Feb-2016.

[14] University of Birmingham, "RoMaNS: Robotic Manipulation for Nuclear Sort and Segregation," *RoMaNS*, 2015. [Online]. Available: http://www.h2020romans.eu/. [Accessed: 09-Apr-2016].

[15] D. Crevier, *AI: The Tumultuous History of the Search for Artificial Intelligence*. New York: HarperCollins, 1993.

[16] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, 2005, vol. 1, pp. 886–893 vol. 1.

[18] B. Su, S. Lu, S. Tian, J. H. Lim, and C. L. Tan, “Character Recognition in Natural Scenes Using Convolutional Co-occurrence HOG,” in *2014 22nd International Conference on Pattern Recognition (ICPR)*, 2014, pp. 2926–2931.
[19] R. B. Rusu, A. Holzbach, G. Bradski, and M. Beetz, “Detecting and Segmenting Objects for Mobile Manipulation,” in *Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE International Conference on Computer Vision (ICCV)*, 2009.
[20] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3D recognition and pose using the Viewpoint Feature Histogram,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2155–2162.
[21] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3d recognition and pose using the viewpoint feature histogram,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 2155–2162.
[22] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1–4.
[23] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *VISAPP International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.
[24] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze, “OUR-CVFH – Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation,” in *Pattern Recognition*, A. Pinz, T. Pock, H. Bischof, and F. Leberl, Eds. Springer Berlin Heidelberg, 2012, pp. 113–122.
[25] B. E. O’Neil, “Object recognition and pose estimation for nuclear manipulation in nuclear materials handling applications,” Mar. 2013.
[26] D. M. Gavrila and V. Philomin, “Real-time object detection using distance transforms,” in *Proc. Intelligent Vehicles Conf*, 1998, p. 998.
[27] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, “Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes,” in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 858–865.
[28] S. Rusinkiewicz and M. Levoy, “Efficient Variants of the ICP Algorithm,” in *3-D Digital Imaging and Modeling*, 2001, pp. 145–152.
[29] Y. D. Shin, J. H. Park, and M. H. Baeg, “6DOF pose estimation using 2D-3D sensor fusion,” in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, 2012, pp. 714–717.
[30] A. Aldoma, F. Tombari, J. Prankl, A. Richtsfeld, L. D. Stefano, and M. Vincze, “Multimodal cue integration through Hypotheses Verification for RGB-D object recognition and 6DOF pose estimation,” in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2104–2111.
[31] L. E. Atlas, T. Homma, and R. J. Marks II, “An artificial neural network for spatio-temporal bipolar patterns: Application to phoneme classification,” in *Proc. Neural Information Processing Systems (NIPS)*, 1988, p. 31.

[32] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Advances in neural information processing systems*, 1990.
[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *ArXiv151203385 Cs*, Dec. 2015.
[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Apr. 2015.
[37] P. Wohlhart and V. Lepetit, “Learning Descriptors for Object Recognition and 3D Pose Estimation,” *ArXiv150205908 Cs*, Feb. 2015.
[38] M. Schwarz, H. Schulz, and S. Behnke, “RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1329–1335.
[39] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, “A Novel Representation of Parts for Accurate 3D Object Detection and Tracking in Monocular Images,” *Proc. Int. Conf. Comput. Vis. ICCV*, 2015.
[40] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[41] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim, “Latent-Class Hough Forests for 3D Object Detection and Pose Estimation,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 462–477.
[42] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, “Learning 6D Object Pose Estimation Using 3D Object Coordinates,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 536–551.
[43] U. Bonde, V. Badrinarayanan, and R. Cipolla, “Robust Instance Recognition in Presence of Occlusion and Clutter,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 520–535.
[44] A. Krull, E. Brachmann, F. Michel, M. Y. Yang, S. Gumhold, and C. Rother, “Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images,” *ArXiv150804546 Cs*, Aug. 2015.
[45] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, “Deep Spatial Autoencoders for Visuomotor Learning,” *ArXiv150906113 Cs*, Sep. 2015.
[46] Y. Yoon, G. N. Desouza, and A. C. Kak, “Real-time tracking and pose estimation for industrial objects using geometric features,” in *IEEE International Conference*

on Robotics and Automation, 2003. Proceedings. ICRA '03, 2003, vol. 3, pp. 3473–3478 vol.3.
[47] M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. V. Le, A. Wellman, A. Y. Ng, and others, "High-accuracy 3D sensing for mobile manipulation: Improving object detection and door opening.," in *ICRA*, 2009, pp. 2816–2822.
[48] A. Cowley, B. Cohen, W. Marshall, C. J. Taylor, and M. Likhachev, "Perception and motion planning for pick-and-place of dynamic objects," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 816–823.
[49] C. Brabec, K. Schroeder, J. Williams, B. O'Neil, J. Hashem, and M. W. Pryor, "Reducing the operator's burden during teleoperation involving contact tasks," presented at the 3rd Int. Joint Topical Meeting on Emergency Preparedness and Response and Robotics and Remote Systems 2011, EPRRSD, 13th Robotics and Remote Systems for Hazardous Environments and 11th Emergency Preparedness and Response, 2011.
[50] C.-H. Wu, S.-Y. Jiang, and K.-T. Song, "CAD-based pose estimation for random bin-picking of multiple objects using a RGB-D camera," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, 2015, pp. 1645–1649.
[51] J. V. Kujala, T. J. Lukka, and H. Holopainen, "Picking a Conveyor Clean by an Autonomously Learning Robot," *ArXiv151107608 Cs*, Nov. 2015.
[52] Open Source Robotics Foundation, "Robot Operating System," *Robot Operating System*. [Online]. Available: http://www.ros.org/. [Accessed: 08-Mar-2016].
[53] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS*. O'Reilly, 2015.
[54] Open Source Robotics Foundation, "ROS libuvc_camera Package Summary." [Online]. Available: http://wiki.ros.org/libuvc_camera. [Accessed: 08-Mar-2016].
[55] Open Source Robotics Foundation, "ROS RealSense Package Summary." [Online]. Available: http://wiki.ros.org/RealSense. [Accessed: 08-Mar-2016].
[56] Open Source Robotics Foundation, "ROS openni2_camera Package Summary." [Online]. Available: http://wiki.ros.org/openni2_camera. [Accessed: 08-Mar-2016].
[57] Open Source Robotics Foundation, "ROS pcl Package Summary." [Online]. Available: http://wiki.ros.org/pcl. [Accessed: 08-Mar-2016].
[58] K. G. Derpanis, "Overview of the RANSAC Algorithm," *Image Rochester NY*, vol. 4, pp. 2–3, 2010.
[59] A. Allevato, T. Lu, and M. Pryor, "Using a Depth Camera for Object Classification in Nuclear Gloveboxes," in *American Nuclear Society Student Conference*, College Station, TX, 2015.
[60] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Commun ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.
[61] OpenCV Dev Team, "Hough Circle Transform." [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html. [Accessed: 08-Mar-2016].

[62] M. T. Hagan and M. B. Menhaj, “Training feedforward networks with the Marquardt algorithm,” *Neural Netw. IEEE Trans. On*, vol. 5, no. 6, pp. 989–993, 1994.
[63] “Levenberg-Marquardt backpropagation - MATLAB trainlm.” [Online]. Available: http://www.mathworks.com/help/nnet/ref/trainlm.html?refresh=true. [Accessed: 03-Apr-2016].
[64] R. B. Rusu, N. Blodow, and M. Beetz, “Fast Point Feature Histograms (FPFH) for 3D Registration,” in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2009.
[65] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, “CAD-model recognition and 6DOF pose estimation using 3D cues,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 585–592.
[66] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Mass: MIT Press, 2005.
[67] A. Allevato and N. Hashem, “Automation Design for Detonator Component Metrology,” presented at the Los Alamos, NM, Los Alamos National Laboratory Student Symposium, 04-Aug-2015.
[68] B. Anderson, M. Pitsch, S. Wanna, D. Park, S. Landsberger, and M. Pryor, “Autonomous Inventory in Nuclear Environment Using a Remote Platform,” presented at the American Nuclear Society Decommissioning and Remote Systems Joint Topical Meeting, Pittsburgh, PA, 2016.
[69] J. C. Winter, P. H. Smith, and T. A. Stone, “Analysis of the SAVY-4000 nuclear material storage container for transportation,” 2011.
[70] International Union of Operating Engineers National Hazmat Program, “HANDSS-55 Transuranic Waste Repackaging Module Human Factors Assessment Report.” Aug-2001.
[71] A. Allevato, M. W. Horn, and M. Pryor, “Demonstrating Autonomous and Robust Sorting in a Glovebox Environment,” presented at the American Nuclear Society Decommissioning and Remote Systems Joint Topical Meeting, Pittsburgh, PA, 2016.
[72] B. Anderson, A. Sharp, and M. Pryor, “Low-cost mobile platform and sensor suite for remote radiation surveying,” in *Embedded Topical Meeting on Decommissioning and Remote Systems 2014, Held at the American Nuclear Society 2014 Annual Meeting*, 2014, pp. 39–42.
[73] DARPA, “Darpa Robotics Challenge 2015,” 2015. [Online]. Available: http://www.theroboticschallenge.org/. [Accessed: 17-Apr-2016].
[74] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton, NJ: Princeton University Press, 1961.
[75] B. Sun, X. Peng, and K. Saenko, “Generating Large Scale Image Datasets from 3D CAD Models,” in *CVPR Workshop*, 2015.

[76] J. Sinapov, C. Schenck, and A. Stoytchev, “Learning relational object categories using behavioral exploration and multimodal perception,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 5691–5698.
[77] M. Malmir, K. Sikka, D. Forster, J. Movellan, and G. W. Cottrell, “Deep Q-learning for Active Recognition of GERMS: Baseline performance on a standardized dataset for active learning,” in *Proceedings of the British Machine Vision Conference (BMVC), pages*, pp. 161–1.
[78] N. Govender, J. Claassens, F. Nicolls, and J. Warrell, “Active object recognition using vocabulary trees,” in *2013 IEEE Workshop on Robot Vision (WORV)*, 2013, pp. 20–26.
[79] “Nuclear Robotics.” [Online]. Available: http://robotics.me.utexas.edu/. [Accessed: 03-Apr-2016].
[80] “ROS.org | Powering the world’s robots.” .
[81] G. Borgefors, “Hierarchical chamfer matching: a parametric edge matching algorithm,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 849–865, Nov. 1988.

Vita

Adam David Allevato grew up in Southern California and Colorado, where he was homeschooled throughout grade school. After developing a passion for math and programming in high school, Adam attended Colorado State University, graduating summa cum laude with a Bachelor of Science in mechanical engineering. He then began a graduate education at UT Austin's Nuclear and Applied Robotics Group, mainly as an excuse to play with robots and live near world-class barbecue restaurants. Adam can be reached by visiting his personal website at www.allevato.me.

Permanent address (or email): adam.d.allevato@gmail.com

This thesis was typed by the author.