

Copyright

by

Andrew Shropshire Boddiford

2013

**The Thesis Committee for Andrew Shropshire Boddiford
Certifies that this is the approved version of the following thesis:**

**IMPROVING THE SAFETY AND EFFICIENCY OF RAIL YARD
OPERATIONS USING ROBOTICS**

**APPROVED BY
SUPERVISING COMMITTEE:**

Supervisor:

Delbert Tesar

Pradeepkumar Ashok

**IMPROVING THE SAFETY AND EFFICIENCY OF RAIL YARD
OPERATIONS USING ROBOTICS**

by

Andrew Shropshire Boddiford, B.S.M.E

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

December 2013

Acknowledgements

I would like to acknowledge my advisor Dr. Delbert Tesar for his guidance and patience throughout my research, experimentation, and writing process. I also wish to thank Dr. Pradeepkumar Ashok for his constant guidance and support throughout the duration of my graduate studies. Thank you to my research sponsors for providing me with the financial support to complete my degree, creative freedom to maximize my learning potential, and input at each stage of this process. Many thanks also to the staff in the Center for Electromechanics and Microelectronics Research Center for their prompt and devoted assistance. A special thanks to Dan Rubin, Evan Freilich, Royce Connerley, David Braley, James Hitzfelder, and John Nodler for their invaluable guidance and support.

I would also like to thank all of my colleagues in the Robotics Research Group past and present. Without your advice, assistance, and friendship, this work would not have been possible. Finally, to my family and friends: thank you for allowing me to disappear for weeks at a time; your loving support and encouragement is greatly appreciated.

Abstract

IMPROVING THE SAFETY AND EFFICIENCY OF RAIL YARD OPERATIONS USING ROBOTICS

Andrew Shropshire Boddiford, M.S.E

The University of Texas at Austin, 2013

Supervisor: Delbert Tesar

Significant efforts have been expended by the railroad industry to make operations safer and more efficient through the intelligent use of sensor data. This work proposes to take the technology one step further to use this data for the control of physical systems designed to automate hazardous railroad operations, particularly those that require humans to interact with moving trains. To accomplish this, application specific requirements must be established to design self-contained machine vision and robotic solutions to eliminate the risks associated with existing manual operations.

Present-day rail yard operations have been identified as good candidates to begin development. Manual uncoupling, in particular, of rolling stock in classification yards has been investigated. To automate this process, an intelligent robotic system must be able to detect, track, approach, contact, and manipulate constrained objects on equipment in motion. This work presents multiple prototypes capable of autonomously uncoupling full-scale freight cars using feedback from its surrounding environment.

Geometric image processing algorithms and machine learning techniques were implemented to accurately identify cylindrical objects in point clouds generated in real-

time. Unique methods fusing velocity and vision data were developed to synchronize a pair of moving rigid bodies in real-time. Multiple custom end-effectors with in-built compliance and fault tolerance were designed, fabricated, and tested for grasping and manipulating cylindrical objects. Finally, an event-driven robotic control application was developed to safely and reliably uncouple freight cars using data from 3D cameras, velocity sensors, force/torque transducers, and intelligent end-effector tooling.

Experimental results in a lab setting confirm that modern robotic and sensing hardware can be used to reliably separate pairs of rolling stock up to two miles per hour. Additionally, subcomponents of the autonomous pin-pulling system (APPS) were designed to be modular to the point where they could be used to automate other hazardous, labor-intensive tasks found in U.S. classification yards. Overall, this work supports the deployment of autonomous robotic systems in semi-unstructured yard environments to increase the safety and efficiency of rail operations.

Table of Contents

Table of Contents	vii
List of Tables	x
List of Figures	xii
CHAPTER 1: INTRODUCTION	1
1.1 Background on Rail Operations.....	3
1.2 Rail Yard Design and Operation.....	7
1.2.1 Flat Yards.....	8
1.2.2 Hump Yard Operation.....	11
1.2.3 Yard Hazards	13
1.3 Motivation.....	14
1.4 Manual Rail Yard Tasks	18
1.4.1 Uncoupling Freight Cars.....	18
1.4.2 Air Brake Hose Operation	24
1.4.3 Handbrake Operation.....	28
1.4.4 Summary of Manual Tasks	32
1.5 Organization of the Report.....	33
CHAPTER 2: RELATED WORK.....	34
2.1 Existing Machine Vision Systems for Rail Applications	34
2.2 Modern Robotic Systems for Rail Applications	39
2.3 Existing Patents for Automating Yard Processes	46
CHAPTER 3: LITERATURE REVIEW.....	53
3.1 Object Detection & Recognition.....	53
3.1.1 Digital Imaging and Preprocessing.....	54
3.1.2 Edge Detection.....	57
3.1.3 Hough Transform.....	63
3.1.4 Pattern Recognition & Machine Learning.....	65
3.2 Robotic Path Planning.....	69

3.2.1 Traditional Path Planning Techniques	69
3.2.2 RRG Geometric-based Spatial Path Planning.....	70
CHAPTER 4: REVIEW OF ROBOTIC & SENSOR HARDWARE.....	87
4.1 Vision Systems.....	87
4.1.1 Two-Dimensional Vision Systems	89
4.1.2 Three-Dimensional Vision Systems	92
4.2 Robotic Manipulators & Mobile Platforms	105
4.2.1 Robotic Manipulators.....	105
4.2.2 Mobile Robotic Platforms.....	107
4.3 Robotic End-Effector Sensing & Tooling.....	110
4.3.1 Robotic Collision Sensor	110
4.3.2 Force/Torque Sensors	114
CHAPTER 5: APPLIED OBJECT DETECTION & RECOGNITION.....	119
5.1 Camera Perspective & Digital Image Acquisition.....	121
5.2 Object Detection Using Digital Images	123
5.3 Object Detection Using 3D Data	134
5.3.1 Lever Detection using Image Processing.....	136
5.3.2 Lever Detection using Machine Learning.....	139
CHAPTER 6: ROBOTIC SYSTEM DESIGN & IMPLEMENTATION.....	149
6.1 University of Texas Rail Test Bed.....	149
6.2 Uncoupling Demonstration	154
6.3 Robotic Test Bed.....	155
6.3.1 Robotic Arm.....	156
6.3.2 Motorized Linear Axis.....	157
6.3.3 RRG Modifications & Additions	158
6.3.4 Robot Controllers.....	159
6.3.5 Software	160
6.3.6 End-Effector Development	161
6.4 APPS Operation.....	167

6.4.1 Robot/Train Synchronization with Vision	168
6.4.2 Sequence of Events for APPS Operation.....	169
6.4.3 Align & Pull Sequence: J-Hook Design	171
6.4.4 Align & Pull Sequence: Rotary Design	175
6.4.5 Summary of APPS End-Effector Designs	177
6.5 Conclusion	179
CHAPTER 7: CONCLUSIONS AND FUTURE WORK.....	181
7.1 Summary	182
7.2 Recommendations for Future Work.....	183
7.3 Concluding Remarks.....	185
References.....	187
Vita	195

List of Tables

Table 1-1: LWD yard injuries by higher-level triggering event, 1997-98 [FRA, 2001]	15
Table 1-2: LWD yard injuries by higher-level physical act, 1997-98 [FRA, 2001]	16
Table 1-3: Human factor-attributed train accidents in railroad yards [FRA, 2001]	17
Table 1-4: LWD yard injuries specific to uncoupling cars in flat yards, '97-98	22
Table 1-5: LWD yard injuries specific to uncoupling cars in hump yards, '97-98	23
Table 1-6: LWD yard injuries specific to air brake hose & valve operation, '97-98	27
Table 1-7: LWD yard injuries specific to handbrake operation, '97-98	31
Table 3-1: Comparison of common edge detection methods (adapted from [Maini, 2009])	62
Table 3-2: Position and orientation at each point of defined trajectory [Tesar et al., 2013]	74
Table 3-3: Parametric constraints using varying tangent scales [Tesar et al., 2013]	75
Table 3-4: Parametric constraints using varying curvature scales [Tesar et al., 2013]	77
Table 3-5: Parametric constraints using varying torsion scales [Tesar et al., 2013]	78

Table 3-6: Parametric constraints using varying derivatives of curvature [Tesar et al., 2013]	79
Table 3-7: Parametric constraints using varying rates of torsion [Tesar et al., 2013]	81
Table 4-1: Summary of available 3D sensing methods [Branding, 2013]	104
Table 4-2: Geometric configurations for robotic arms (adapted from [Petriu, 2012])	106
Table 4-3: Qualitative comparison of the four categories of planar mobile platforms [Kulkarni, 2009]	107
Table 4-4: Commercially available mobile robotic platforms	108
Table 4-5: Overview of specifications for commercially available collision sensors	114
Table 5-1: Tabulation of prediction performance using different network configurations	146

List of Figures

Figure 1-1: Impact of spending on accident & injury rates [AAR-2, 2013].....	5
Figure 1-2: Schematic representation of a flat yard [Wong et. al., 1981].....	8
Figure 1-3: Switching cars in a flat yard [Dirnberger, 2006]	9
Figure 1-4: Kicking cars in a flat yard [Dirnberger, 2006].....	10
Figure 1-5: Hump yard schematic [Dirnberger, 2006, Wong et. al., 1981].....	12
Figure 1-6: Schematic representation of hump crest [Wong et. al., 1981]	13
Figure 1-7: Closed/open positions of Type E coupler (left) & internal parts (right) [McConway, 2013, REB, 2004]	19
Figure 1-8: Three internal states of a Type E coupler [AAR, 1965]	20
Figure 1-9: Standard configuration for an uncoupling lever assembly [REB, 2000]	21
Figure 1-10: Location of uncoupling lever on freight car [Edwards, 2006].....	21
Figure 1-11: Air brake hose assemblies connected between freight cars	24
Figure 1-12: Air brake hose assembly and coupling methods [REB, 2004]	25
Figure 1-13: Air brake system with handbrake connection [Wabtec, 2004]	28
Figure 1-14: Handbrake and brake step for employee use [Edwards, 2006, REB, 2004]	29
Figure 1-15: Handbrake operation on individual freight cars [Wabtec, 2004].....	30
Figure 2-1: Extreme deformation to sill steps on rolling stock [Edwards, 2006].....	35
Figure 2-2: Beena Vision wayside inspection system [Nayebi et al., 2012]	37
Figure 2-3: “TrackView-Profile” vehicular track inspection system [BeenaVision, 2013]	37

Figure 2-4: “LazerView” non-contact handheld laser wheel profile [BeenaVision, 2013]	38
Figure 2-5: Motoman robotic gantry system for welding coal cars [Motoman, 2007]	39
Figure 2-6: igm gantry robot welding a bogie cross frame [igm, 2013]	40
Figure 2-7: Plasser & Theurer APT 1500 R rail welding robot [P&T, 2013]	41
Figure 2-8: Integrated robot/lorry design (top) and resulting welds (bottom) [P&T, 2013]	42
Figure 2-9: Schematic of ultrasonic wheel inspection stations [Pagano, 2009]	43
Figure 2-10: Schematic of ultrasonic wheel inspection stations [Nordco, 2013]	44
Figure 2-11: An autonomous mobile robot for remote switch inspection [Loccioni, 2013]	45
Figure 2-12: RailHawk, autonomous robot unloading rail cars [Concept, 2013]	45
Figure 2-13: Patented concepts for local automated uncoupling systems (top-left [Heimaster, 1956] bottom-left [Sanyas, 1965] right [Kandoth, 2009])	47
Figure 2-14: Patented concepts for wayside mechanical uncoupling systems (top-left [Peterson, 1972] bottom-left [Nell, 1977] right [Lindow, 1974])	49
Figure 2-15: Patented concepts for wayside robotic uncoupling systems (left [Bruns, 1990] right [Cappelletti, 1996])	51
Figure 3-1: Mathematical representation of digital images [Grauman, 2011]	54
Figure 3-2: Histogram equalization [Ahn, 2006]	55

Figure 3-3: Typical intensity changes in rail yards (physical, material, & lighting).....	58
Figure 3-4: Edges types (a) step, (b) ramp, (c) roof, and (d) smooth [Cattin, 2012].....	59
Figure 3-5: Edges detected using Canny (physical, material, or lighting).....	61
Figure 3-6: Representation of a line in Cartesian (left) & Hough (right) coordinates [Hough, 2013].....	63
Figure 3-7: Straight lines detected by Hough transform algorithm [OpenCV-2, 2013].....	65
Figure 3-8: Three-layer perceptron (left) & input function to an individual neuron (right) [OpenCV, 2013].....	67
Figure 3-9: Frenet Frame on a spatial curve [Tesar et al., 2013].....	72
Figure 3-10: Effect of varying tangent scale on a planar curve [March, 2008].....	74
Figure 3-11: Spatial example of varying tangent scale [Tesar et al., 2013].....	75
Figure 3-12: Planar example of varying curvature [March, 2008].....	76
Figure 3-13: Example spatial trajectories with varying curvature [Tesar et al., 2013].....	77
Figure 3-14: Example spatial trajectories with varying torsion [Tesar et al., 2013].....	78
Figure 3-15: Example trajectories with varying rates of curvature [Tesar et al., 2013].....	80
Figure 3-16: Example trajectories with varying rates of torsion [Tesar et al., 2013].....	81
Figure 3-17: Typical workspace for an industrial manipulator [MH80, 2013].....	83
Figure 3-18: Path trajectories constrained by $\tau = 10$	84

Figure 3-19: Arc lengths of paths defined by $\tau = 10$ over robot's workspace	85
Figure 3-20: Percent difference b/t arc length of straight paths & paths defined by $\tau = 10$	86
Figure 4-1: Operation flow and cycle times for machine vision systems [SICK, 2006].....	88
Figure 4-2: a) Illumination, b) reflection, c) resolution, & d) field of view [SICK, 2006].....	90
Figure 4-3: Digital 2D cameras a) SICK smart camera [SICK, 2006], b) Logitech Pro 9000 webcam [Logitech, 2013], & c) PlayStation Eye [Sony, 2013].....	92
Figure 4-4: 3D SICK VP line scan camera [SICK, 2006]	93
Figure 4-5: Stereo vision principle of operation (left) [PMD, 2008] & Point Grey Bumblebee stereo camera (right) [Point Grey, 2013].....	94
Figure 4-6: Stereo vision system operation	95
Figure 4-7: Time-of-flight principle of operation [Metrilus, 2013].....	96
Figure 4-8: Time-of-Flight cameras a) Mesa Imaging SwissRanger 4000 [Mesa, 2013] b) PMD CamBoard Nano [PMD, 2013] & c) Creative Senz3D [Creative, 2013]	98
Figure 4-9: SICK LMS 500 Pro laser rangefinder [SICK, 2012].....	99
Figure 4-10: Laser rangefinder principle of operation [SICK, 2012].....	100
Figure 4-11: Structured light camera principle of operation [Hecht, 2011]	101
Figure 4-12: (Left to Right) a) IR image of scene, b) Grayscale distance image, & c) Color coded point cloud [Metrilus, 2013]	102
Figure 4-13: Microsoft Kinect schematic [Beyer, 2013] (left) & ASUS Xtion Pro Live (right) structured light sensors	103
Figure 4-14: Robotic collision sensor diagram [ATI-1, 2013]	110

Figure 4-15: Robotic collision sensor diagram [ATI-1, 2013]	111
Figure 4-16: Internal components and circuitry for collision sensor [ATI-1, 2013].....	112
Figure 4-17: Collision sensor response modes [ATI-1, 2013].....	112
Figure 4-18: Commercially available collision sensors (AGI, RAD, ATI, IPR).	113
Figure 4-19: Force and torque vectors of Omega160 transducer [ATI-2, 2013].	115
Figure 4-20: Electronics and software used to process F/T data [ATI-2, 2013].	116
Figure 4-21: LabVIEW interface to view and log F/T data	117
Figure 5-1: Freight car components & safety appliances	119
Figure 5-2: Uncoupling lever handle types.....	120
Figure 5-3: Camera perspectives, perpendicular (left) & angled (right)	122
Figure 5-4: Original image (left) & detected edges using Canny (right).....	124
Figure 5-5: Straight lines detected by Hough transform.....	125
Figure 5-6: Pairs of parallel lines detected by Hough transform	126
Figure 5-7: Filter line pairs for vertical orientation, enclose boundaries, and fill	127
Figure 5-8: Uncoupling lever handle detection – blob creation and tracking.....	128
Figure 5-9: Uncoupling lever assembly detection – geometric fitting	130
Figure 5-10: Contour extraction from digital images	131
Figure 5-11: Lever detection using binary masks & contour extraction algorithms	132
Figure 5-12: Inconsistent edge detection results using digital images	133
Figure 5-13: Three-dimensional point cloud acquired using SwissRanger ToF camera	134
Figure 5-14: Digital image (left) and depth map (right) of a passing train.....	135

Figure 5-15: Full depth map (left) and filtered depth map (right) of a passing train	136
Figure 5-16: Lever detection process using depth maps a) original map, b) fill enclosed areas, c) polygon approximation, and d) find parallel lines & centroid	137
Figure 5-17: Handle detections over 20 video frames (true detections circled in blue).....	138
Figure 5-18: Filtered depth map (pixels shown are 0 to 1.5 m from camera)	139
Figure 5-19. Three true images (left) & three false images (right) used for ANN training.....	140
Figure 5-20: “Training” image, 300 true, 400 false images (row vectors = 600 pixels)	141
Figure 5-21: Visual representation of the sliding window algorithm	143
Figure 5-22: Predicted vs. actual outputs for true (left) and false (right) test images	144
Figure 5-23: Bounding box over prediction: good (left) and false positives (right).....	145
Figure 5-24: Uncoupling lever detected on UT Austin test bed during successful uncoupling procedure.....	148
Figure 6-1: Rolling stock test bed for railroad robotics research.....	150
Figure 6-2: Comparison of 3D vision data from rail yard (left) and lab (right) ..	151
Figure 6-3: Electric winch (left) and wheel stops (right).....	151
Figure 6-4: Custom push-pull configuration for winch to actuate rail test bed ...	152
Figure 6-5: Custom brake on non-actuated cart.....	153
Figure 6-6: Rail test bed – uncoupling lever handle vertical - coupled	154

Figure 6-7: Rail test bed – uncoupling lever handle horizontal - uncoupled.....	154
Figure 6-8: Initial platform Motoman MH80 on a Güdel floor mounted linear axis	155
Figure 6-9: Working volume for MH80 manipulator and sample lever pull forces.....	156
Figure 6-10: Güdel floor mounted linear axis.....	157
Figure 6-11: Carriage plate extension and sensor mounting	158
Figure 6-12: Motoman DX100 with teach pendant (left), internal components of DX100 (middle), & internal components of Agile Planet AX-I3 controller (right).....	160
Figure 6-13: Custom J-Hook gripper and end-effector designed to pull lever	162
Figure 6-14: Robotic tool changer schematic	163
Figure 6-15: Operation and passive actuation of J-Hook gripper as it contacts handle.....	164
Figure 6-16: Custom rotary end-effector designed to pull lever.....	165
Figure 6-17: Start and end positions of the custom rotary end-effector	166
Figure 6-18: Position tracking using vision: offset error (left) and position tracking (right)	168
Figure 6-19: Align sequence with J-Hook gripper	171
Figure 6-20: Gripper align with active compliance enabled.....	173
Figure 6-21: Circular lever pull with J-Hook gripper.....	174
Figure 6-22: Alignment and extension of pneumatic cylinder for handle pull....	175
Figure 6-23: Lever pull with rotary end-effector design, simulated (top), lab test (bottom).....	177
Figure 6-24: Robot in “red-zone” between rolling stock.....	178

CHAPTER 1: INTRODUCTION

Since the invention of the first commercial locomotive by Englishman George Stephenson at the beginning of the 19th century, railroads have been vital to the economic development of countries around the world. The railroad had a particularly large impact on the growth of the United States due to the physical size of the country and its role in the nation's Industrial Revolution. From 1820 to 1860, thousands of miles of track were laid to connect nearly all major cities in the North and Midwest. Shortly after the end of the American Civil War, Central and Union Pacific railroads worked tirelessly to connect the rail networks in the Eastern U.S. to the West Coast by completing the First Transcontinental Railroad between Omaha, Nebraska and Sacramento, California in 1869. This incredible achievement unified a divided nation and led to massive expansion in the late 19th and early 20th centuries.

The development of rail-related technology from the 1820s onward was rapid. Many of the world's best engineers were constantly working to invent or improve the next iteration of rolling stock during this time period. At first, new ideas disseminated quickly to competitors and large railroad firms agreed to share innovation and move the industry forward to everyone's benefit. This culture began to change in the late 19th century, as outside inventors began to file patents for technology that promised to improve railroad equipment and operations. The patent industry really took off when the courts began awarding huge settlements to patent owners that had sued railroads for taking their ideas; a trend very similar to one seen in the modern software market [Merges, 2003]. Contrary to popular opinion, the widespread introduction of patents did not have a negative effect on the rail industry due to the severe need for increased safety and efficiency [Merges, 2003].

The surge in rail traffic due to the rapid growth of rail networks in the late 19th century led to an increase in railroad related fatalities and injuries, particularly among brakemen. A brakeman's responsibilities included applying brakes on individual cars, properly setting couplings between cars, and throwing switches to properly route trains to their destinations. Many accidents were directly connected to the coupling and uncoupling of railroad stock due to the deficiencies of the widely used link-and-pin couplers. These devices required brakeman to enter the area between moving freight cars to insert a pin into an aligned pair of couplers to successfully join rolling stock. Between 1883 and 1892, there were 5,623 rail-related fatalities and 20,445 injuries, making the railroad the second most dangerous industry in the United States, second only to coal mining [McDonald, 1993]. Due to public pressure, the U.S. Congress passed the Railroad Safety Appliance Act (SAA) in 1893 that enforced the adoption of safety appliances, automatic couplers, and handbrakes. This act was the first of many measures to make the brakeman's job safer and easier. In 1966, the Federal Railroad Administration (FRA) was created in large part to define, promote, and regulate rail safety. The FRA has since passed numerous acts, endorsed certification programs, and sponsored research initiatives that have all made the rail industry safer than ever before.

Many of the improvements responsible for increasing safety in the rail sector have involved thorough studies of existing procedures to redefine job responsibilities to reduce work related injuries or design and implement new technology to fully remove rail personnel from hazardous environments. Although large strides have been made over the past 150 years, certain aspects of today's rail yard operations are still labor-intensive and potentially hazardous, particularly those that require human interaction with moving trains. With modern advancements in robotics and sensing, the opportunity exists to further automate rail yard operations to make it a safer and more efficient environment.

1.1 BACKGROUND ON RAIL OPERATIONS

The U.S. freight rail network is widely considered one of the most dynamic freight systems in the world and represents a \$60 billion industry [FRA, 2012]. In 2012, U.S. Class I railroads totaled 1.71 trillion ton-miles of domestic intercity freight, accounting for nearly 40% of the market share between all modes of transportation (rail, water, pipeline, truck, and air) [AAR-1, 2013]. Over 24,000 locomotives and 1.1 million active freight cars were used to transport this freight across 140,000 miles of rail [AAR-1, 2013]. Nearly any type of product can be transported by rail. The FRA estimates that each U.S. citizen requires the movement of nearly 40 tons of freight every year. Approximately 91% of this freight is bulk commodities that people use on a daily basis, such as agricultural products, automobiles, energy, food, and paper while the remaining 9% is intermodal traffic that typically consists of smaller consumer goods [FRA, 2012]. Railroads are particularly effective at moving large, heavy freight over long distances whereas trucks excel in providing time-sensitive delivery services for high value goods hauled over smaller distances [FRA, 2012]. The rail industry is constantly looking to increase capacity and improve upon existing performance metrics, such as train velocity and terminal dwell time, to maintain their presence in bulk freight while simultaneously earning more business from the trucking market.

Progression in the rail industry comes at a cost. U.S. railroads in the freight sector invest more of their capital into maintenance and improvement projects than a majority of their counterparts in other industries. Approximately 80% of this investment is spent on upkeep to preserve existing equipment while the remaining funds are typically used for expansion, enhancement, or research initiatives [FRA, 2012]. Safety and efficiency are often key components of both maintenance and improvement projects. An overview and

evaluation of current initiatives with regards to rail safety and efficiency as they pertain to the freight sector will be provided in the following sections.

Status of Freight Rail Safety

Whether it's navigating a city via a metro system or waiting at a train crossing on one's daily commute to work, railroad operations will always be a part of everyday life. With nearly 220,000 private and public grade crossings in use over 140,000 miles of track today in the United States, railroads work constantly with their employees, government officials, and the general public to rigorously promote safety [Operation, 2013]. Over the last two decades, this dedication to safety has reduced the train accident rate by 80%, grade crossing collision rate by 82%, and the rail employee injury rate by 85%, making 2012 the safest year on record for freight railroads [AAR-2, 2013]. This was accomplished through increased investment in rail infrastructure and innovation, employee safety training, and public safety education.

Since 1980, privately owned railroads have invested \$525 billion into the U.S. rail network, and these same companies plan to invest \$25.5 billion in 2013 [AAR-2, 2013]. A significant portion of the spending was used to replace aging track and rail equipment. However, billions of dollars were also invested in new technologies, including ground-penetrating radar capable of detecting problems beneath the Earth's surface, specialized rail cars that use sophisticated instruments to identify defects in tracks, and wayside sensor networks to monitor the health of rolling stock [AAR-2, 2013]. Remote Control Locomotives (RCL) have also been implemented in many rail yards to make operations safer. The perceived safety benefit of RCL technology is that a switchman on the ground can directly control the locomotive without having to pass signals to an engineer in the locomotive, thus eliminating the chance of miscommunication between the switchman

and engineer [FRA, 2006]. Although the use of RCLs has historically been somewhat controversial, the FRA reported that employee injury rates were 20% lower for RCL operations than for conventional switching operations in March 2006 [FRA, 2006]. Lastly, railroads have also spent \$2.7 billion to date for the implementation of positive train control (PTC). PTC is a broad set of technologies, one of which is designed to automatically decelerate a train to a stop if the supervisory control system believes an accident will occur. PTC is primarily designed to prevent train-to-train collision, derailments caused by excessive speeds, and unauthorized routings of trains where incorrect switch positions or repairs work could lead to a disaster [AAR-3, 2013]. Further information on these technologies can be found in Chapter 2. Railroads continue to invest record amounts into infrastructure upgrades due to the impact it has had on safety based on data collected by the AAR and FRA in Figure 1-1.

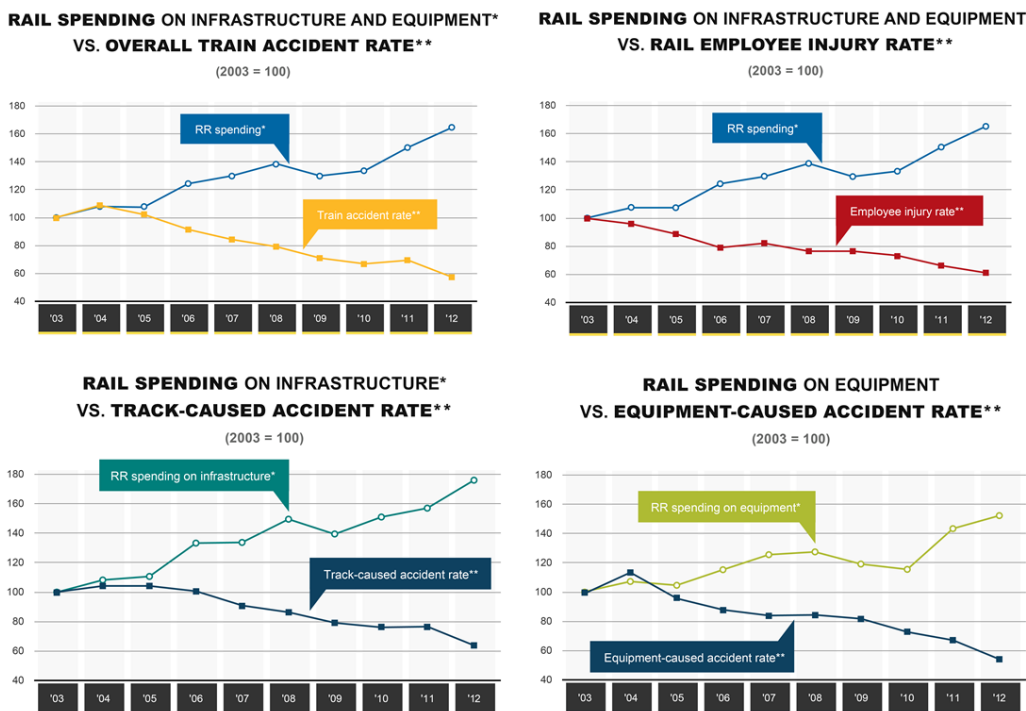


Figure 1-1: Impact of spending on accident & injury rates [AAR-2, 2013]

The data in Figure 1-1 clearly demonstrates that increased spending on rail infrastructure and equipment will increase safety, thereby reducing employee injury, train accident, track-caused accident, and equipment accident rates. The more revenue that railroads are able to re-invest in infrastructure and innovation, the safer railroad operations will become.

Railroads have increasingly dedicated time and resources into training programs that provide the essential skills and knowledge to 180,000 employees who are responsible for keeping the nation's freight moving safely [AAR-2, 2013]. These programs also help instill a workforce culture where safety is always top priority. Some examples of employee training programs include: railroad safety, locomotive operations, inspection and maintenance, track safety, hazardous materials handling, and first responder training. Railroads also strive to educate the general public to improve awareness about safe behavior around grade crossing and railroad property [AAR-2, 2013]. Non-profit organizations, such as "Operation Lifesaver," and state/local law enforcement agencies often partner with railroads to collectively reduce the frequency of rail-related accidents involving the general public.

Status of Freight Rail Efficiency

Many of the aforementioned technologies that were designed and implemented to increase safety also improve rail efficiency. Wayside sensor networks that actively monitor passing trains provide substantial amounts of useful data. This data can be processed quickly to predict when components will fail weeks in advance to prevent derailments that risk lives and cause massive delays. Additionally, this knowledge can be used to optimize scheduling for preventative and condition-based maintenance depending on a piece of equipment's proximity to a yard or repair facility. Parts can then be ordered

or prepared in advance in minimize dwell time when the car reaches its destination. Automated systems, such as RCLs and sophisticated yard control systems have had a positive impact on the efficiency of rail yard operations. According to a Congressional report in 2012, some implementations of PTC have been projected to boost the efficiency of rail operations through an increase in capacity and a reduction in fuel consumption [Peters, 2012]. Although this outlook is positive, it will likely be another decade before these claims can be validated once PTC is in place.

Background Summary

Overall, it is abundantly clear that U.S. freight railroads are taking the necessary steps to make operations safer and more efficient across the board. The amount of capital invested in increased infrastructure, training, and education demonstrates the freight industry's level of commitment to maximizing safety and performance. However, there is room to improve, and the freight rail industry will continue targeting areas where emerging technologies can be used to enhance safety and efficiency. With this in mind, the focus of this work is to identify areas and applications within the freight rail industry where modern advancements in robotics and automation can be leveraged. After a careful review of present operations in the freight rail industry, it was determined that rail yard operations, particularly those in classification yards, can benefit the most from robotic technology. The following sections will present the findings that led to this conclusion.

1.2 RAIL YARD DESIGN AND OPERATION

The purpose of a rail yard is to sort, store, or load/unload freight cars. Most yards dedicated to loading/unloading rolling stock are located at or near industrial facilities that require periodic delivery of bulk commodities. The configuration of these yards is often unique to the business and operations are generally not time sensitive. Once the

loading/unloading operation is complete, a set of cars may be stored temporarily until their departure is scheduled. Rail yards used to divide and sort incoming trains by their future destinations are called “classification” yards. Classification yards can be separated into two basic types: flat yards and hump yards. Flat yards have little to no elevation change, whereas a hump yard has an artificial “hill” with carefully designed grades to facilitate the uncoupling and sorting of freight cars [Wong et. al., 1981]. Flat yards are typically used for small to medium volume operations whereas hump yards are designed to handle high volumes of up to 3,000 freight cars daily. Similarly, flat yards are often very labor intensive while modern hump yards are highly automated.

1.2.1 Flat Yards

A flat yard typically consists of a series of parallel tracks connected by a ladder track and a switching lead. This configuration is useful for storing and sorting rolling stock off the mainline, so that they do not obstruct the flow of traffic. Most flat yards use the same tracks for receiving, classifying, and dispatching trains although some yards do have separate areas for dedicated procedures [Wong et. al., 1981]. Since the flat yard does not have any change in elevation, an active locomotive must be used to move all rolling stock. A standard layout for the design of a flat yard can be seen in Figure 1-2.

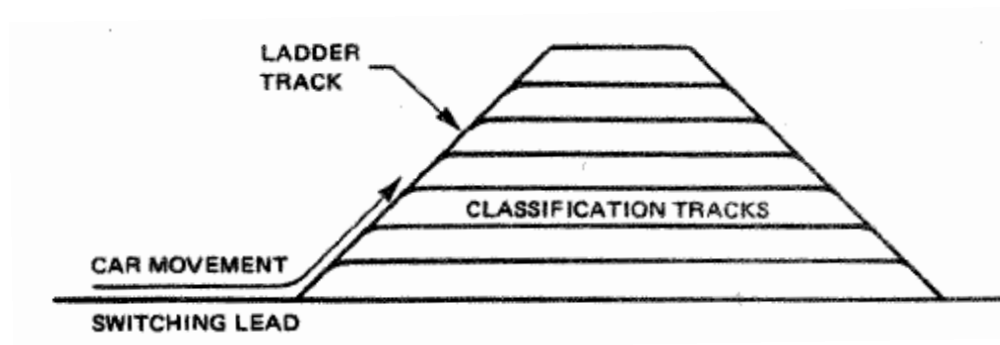
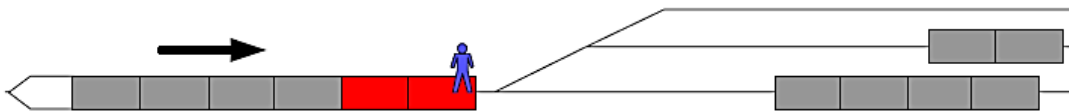


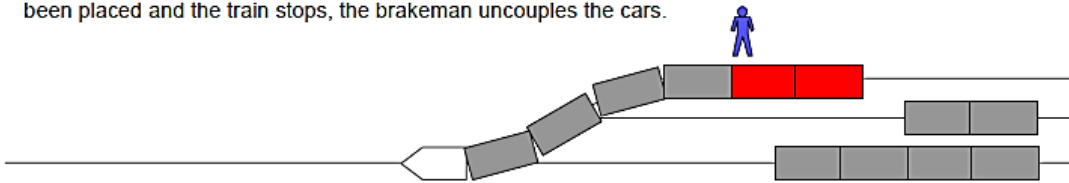
Figure 1-2: Schematic representation of a flat yard [Wong et. al., 1981]

To sort or “classify” rolling stock in a flat yard, a group of cars to be switched must be transported to the switching lead shown in Figure 1-2 using a locomotive, also known as a switch engine, positioned at the rear. Once the switch engine is in place, the classification of cars can be performed using one of two methods: “flat switching” or “kicking”. Figure 1-3 below depicts three stages of a flat switching process with accompanying descriptions for each.

A. A brakeman typically rides the leading car to ensure there are no obstacles and to line switches as the movement proceeds.



B. The brakeman steps down near the point where the cars are to be uncoupled. When the cars have been placed and the train stops, the brakeman uncouples the cars.



C. The switch engine reverses, leaving the cut of cars on the desired track.

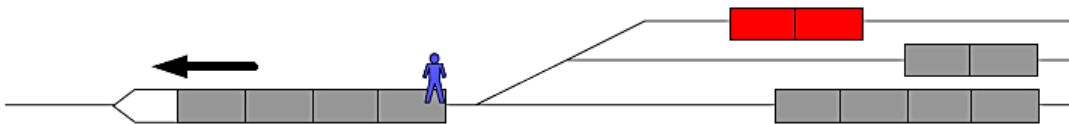


Figure 1-3: Switching cars in a flat yard [Dirnberger, 2006]

In Figure 1-3 above, the brakeman first climbs onto the lead car of a train to ride until the designated classification track for a cut of cars is reached (step A, shown in red). Once the brakeman is safely aboard, the switch engine begins to push the rolling stock toward the classification tracks until it reaches the designated area where the cut of cars needs to be uncoupled [Dirnberger, 2006]. The brakeman steps off the train once it is completely at rest and helps the engineer adjust the position of the cars (see Figure 1-3,

step B) [Dirnberger, 2006]. Upon successful placement, the brakeman manually uncouples the desired cut of cars from the remaining length of the train. The brakeman then re-boards the new lead car of the train and the switch engine reverses to leave the cut of cars (see Figure 1-3, step C) [Dirnberger, 2006]. Finally, as the train pulls away, the brakeman is responsible for confirming coupler separation. Flat switching is an inefficient, time consuming, and unsafe process due to the repeated back-and-forth trips to switch cuts and the extra handling of cars near the engine [Dirnberger, 2006].

“Kicking” attempts to reduce the number of back-and-forth trips required for classification by uncoupling cuts of cars while the train is moving, thus utilizing the train’s forward momentum to propel cars to their destinations [Dirnberger, 2006]. Figure 1-4 depicts three stages of a “kicking” process with accompanying descriptions for each.

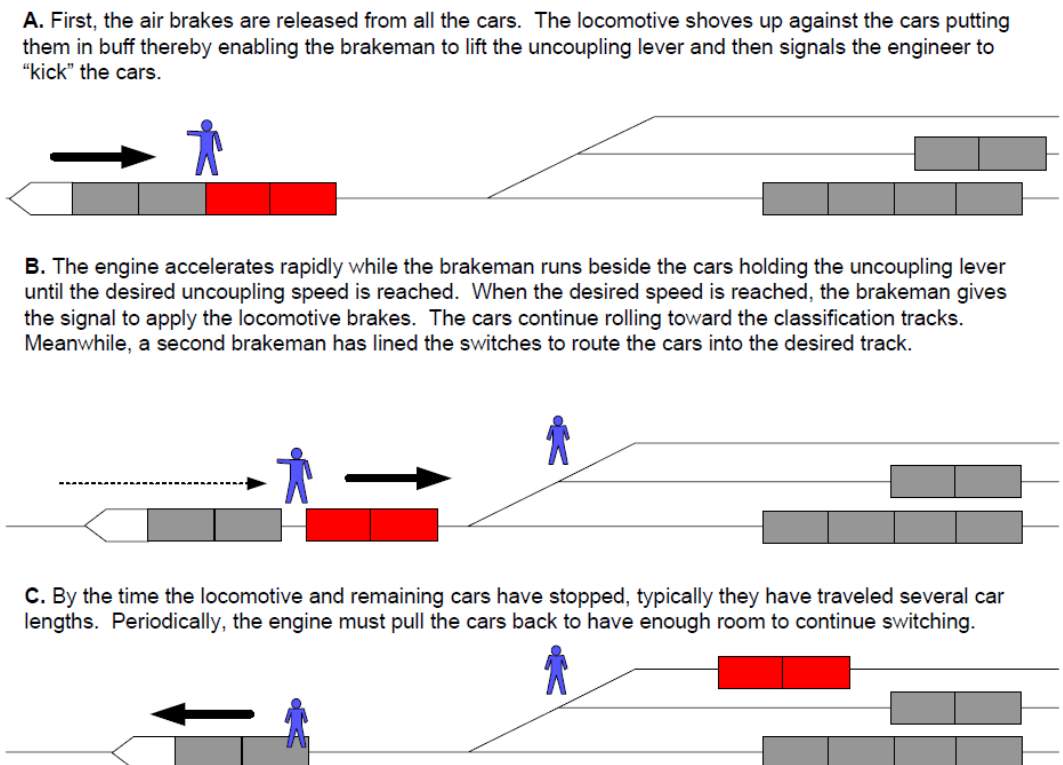


Figure 1-4: Kicking cars in a flat yard [Dirnberger, 2006]

Before the kicking process is started, the brakeman releases the air brakes from all the cars in the train. Next, the switch engine slowly pushes the cars to compress all couplers to allow the brakeman to lift the lever required to uncouple a cut of cars (see Figure 1-4, step A) [Dirnberger, 2006]. Once the brakemen signals that he is ready, the switch engine is commanded to accelerate until it reaches uncoupling speed. All the while, the brakeman is running next to the train, lifting the uncoupling lever until he gives another signal to quickly apply the engine's brakes [Dirnberger, 2006]. This sudden deceleration releases the cut of cars from the train to roll to a destination determined by a second brakeman that is operating the switches (see Figure 1-4, step B). If required, the switch engine will pull the remaining length of cars back to ensure that it has enough space to safely and effectively repeat the "kicking" process for the next cut (see Figure 1-4, step C) [Dirnberger, 2006].

This form of classification in a flat yard is more efficient because it reduces the required length of travel by the switch engine for each cut of cars. The larger the cut, the more efficient the process becomes. Kicking speeds ranging from 2 to 8 mph are required to achieve desired rates of switching measured in cars/minute [Wong et. al., 1981]. The greater the speed, the more dangerous kicking becomes for the brakeman responsible for walking or running next to a moving train to successfully uncouple cuts of cars.

1.2.2 Hump Yard Operation

Hump yards are commonly used for high volume operations due to the minimization of locomotive travel for individual cars and the constant, predictable nature of uncoupling. Hump yards often have separate receiving, classification, and departure yards, as shown in Figure 1-5. Inbound trains are stored in the receiving yard until a hump plan is scheduled. Rolling stock is then classified according to the hump plan using

a switch engine that takes a group of cars from the receiving yard and pushes them over the artificially raised portion of track called the “hump” [Wong et. al., 1981]. Figure 1-5 below illustrates a standard layout of a hump yard, including the locations of the different yards, hump crest, and retarders.

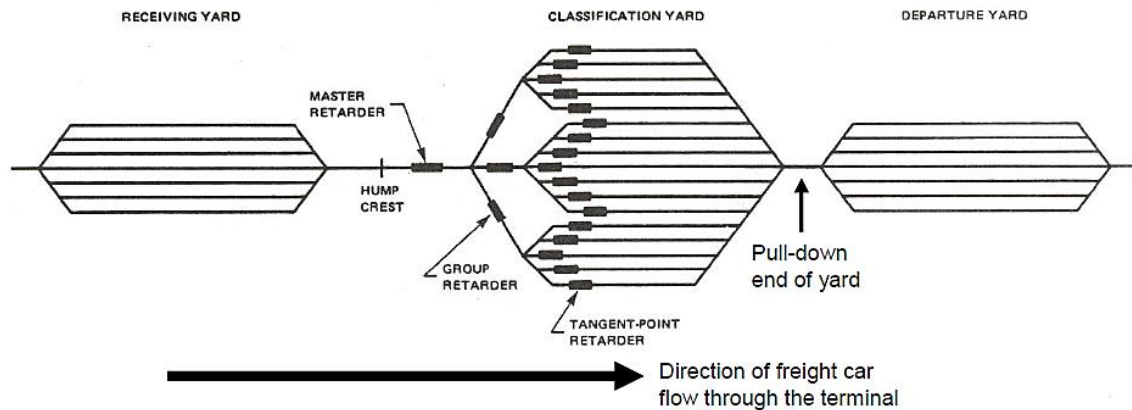


Figure 1-5: Hump yard schematic [Dirnberger, 2006, Wong et. al., 1981]

Cars are uncoupled at the hump crest by a brakeman that pulls an uncoupling lever to release the lead car. As the center of gravity of the lead car passes over the crest of the hump (see Figure 1-6), the car begins to accelerate down the hump grade, thereby separating it from the hump engine and attached cars. Wayside braking devices, known as retarders, are used to control the speed of the cars as they accelerate down the hump and the appropriate switches are thrown automatically to direct rolling stock to the correct classification to build trains according to the hump plan. The speed of the cars accelerating down the hump is carefully regulated by the retarders according to required travel distance, position in the train being built, and wind speed so that cars will successfully couple to other cars without damage to any of the coupler’s components [Wong et. al., 1981]. Once the humping process is completed, assembled trains are moved from the classification yard to the departure yard using a makeup engine.

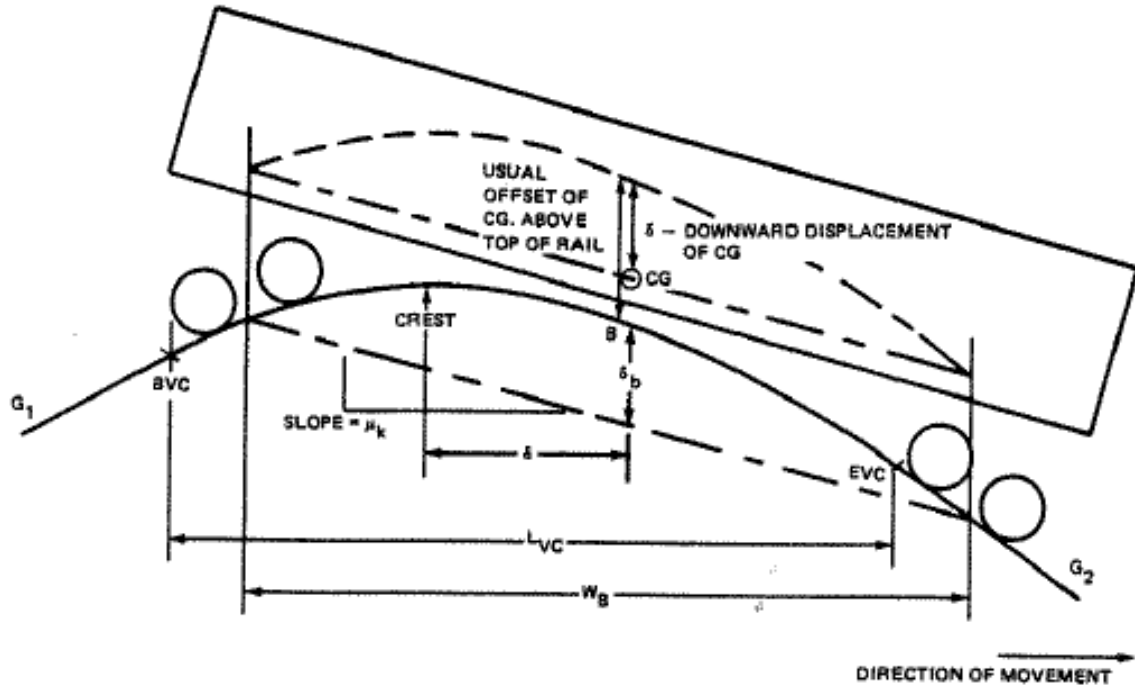


Figure 1-6: Schematic representation of hump crest [Wong et. al., 1981]

1.2.3 Yard Hazards

Independent of yard type, there are numerous hazards with railroad yard operations due to the “high traffic volume, the physical layout of the yard, and the labor-intensive nature of switching cars. These job hazards include a high volume of large, heavy, moving equipment; uneven surfaces; manual switches; equipment and debris on the ground; noise and visual distractions; inclement weather; and greasy and slippery work areas and surfaces (e.g., a switching step or vertical handhold of a locomotive)” [FRA, 2001]. These and many other hazards associated with specific yard tasks will be discussed later in the following sections of this work.

1.3 MOTIVATION

Although safety in the railroad industry has improved significantly over the last three decades, two separate studies have shown that several job categories within the rail sector experience injury rates and severities “far in excess” of the industry average [Drudi, 2007, FRA, 2001]. A majority of these jobs were found to be associated with rail yard operations; 48% all train accidents and 31% of all injuries to rail employee occurred in rail yards in 1998 [FRA, 2001]. Section 1.2 provided an overview of rail yard types and operations. The following sections will delve into the safety of specific rail yard tasks to make a case for automating them using state-of-the-art sensing and robotics technology.

The first study, performed by the FRA in 2001, analyzes employee injury data from 1997 to 1998 and train accident data from 1994 to 1998. The objectives [FRA, 2001] of this study were to:

1. Identify injury and accident trends using the FRA accident/injury database
2. Examine the effect of work schedule on yard injuries
3. Solicit and evaluate railroad labor and management opinions and experiences regarding safety strategies
4. Explore methods for collecting additional injury causal factor data

The scope of this study was limited to accidents and injuries occurring in U.S rail yards. Included analyses of personal injuries focused on those injuries that were sustained by on-duty employees and that resulted in one or more lost workdays (LWD) [FRA, 2001]. A LWD injury is one that results in at least one day away from work and/or one day of restricted duty [FRA, 2001]. This metric represents the most serious of injuries sustained by rail employees with the largest economic impact to both the employer and employee due to the costs associated with the required litigation [FRA, 2001]. A 1994

report from the Transportation Research Board estimated that job related injuries cost American railroads over \$1 billion annually [TRB, 1994]. Hence, the increased implementation of rail safety measures will not only reduce the frequency and severity of worker injuries, but will also result in significant savings for American railroads [FRA, 2001].

The following two charts display archived LWD rail yard injuries sustained between 1997 and 1998; categorized respectively by the event that triggered the injury (Table 1-1) and the physical act being performed when the injury occurred (Table 1-2). The FRA provides railroads with over 65 codes from which to choose and categorize the triggering event in an accident/incident report form [FRA, 2001]. Table 1-1 presents a condensed set of higher-level triggering events (each listing includes multiple event codes) that are especially relevant to everyday yard tasks.

Table 1-1: LWD yard injuries by higher-level triggering event, 1997-98 [FRA, 2001]

Triggering Event	Injuries	% of Total Injuries	Median LWDs	Total LWDs	% of Total LWDs
Slips, Trips, Falls and Lost Balance	1,516	41.6	30.0	98,743	44.1
Overexertion	726	19.9	30.0	49,089	21.9
Struck by or Against Object	562	15.4	18.0	27,382	12.2
Sudden/Unexpected Movement of Equipment	217	6.0	40.0	15,972	7.1
Collision/Impact (Includes Derailment)	186	5.1	45.0	16,778	7.5
Caught in/Compressed/Crushed	172	4.7	19.5	6,433	2.9
RSI	125	3.4	22.0	6,608	3.0
Exp. to Harmful Substances (Fumes/Chemical)	69	1.9	7.0	1,320	0.6
Climate/Natural Environment	48	1.3	4.0	982	0.4
Security Issues	20	0.5	9.5	630	0.3
Horseplay/Thrillseeking/Sustained Viewing	3	0.1	4.0	11	0.0
Total	3,644	100	25.0	223,948	100

As shown in Table 1-1, “slips, trips, falls, and lost balance” account for the largest percentage of *injuries* to rail employees (42 percent) in addition to the largest proportion of LWDs (44 percent). Another important metric, median number of LWDs, was calculated to determine the severity of each triggering event listed above. The events that caused employees to miss the most days of work were “collisions and impacts with equipment” and “sudden or unexpected movement of equipment” (medians of 45 and 40 LWDs, respectively) [FRA, 2001].

Similar to the triggering events above, the FRA has defined more than 70 codes for physical acts that employees were engaged in at the time of each accident/incident. Table 1-2 contains a condensed set of higher-level physical acts (each listing includes multiple physical act codes) that are commonly required to perform yard tasks.

Table 1-2: LWD yard injuries by higher-level physical act, 1997-98 [FRA, 2001]

Physical Act	Injuries	% of Total Injuries	Median LWDs	Total LWDs	% of Total LWDs
Walking/Running/Stepping Over	816	25.0	28.0	52,040	25.3
On/Off/Up/Down	731	22.4	26.0	43,841	21.3
Lining Switches/Other	357	10.9	29.0	25,234	12.3
Sitting/Riding	342	10.5	36.0	24,204	11.8
Pull/Push	308	9.4	29.5	18,630	9.1
Lift/Load	219	6.7	18.0	10,552	5.1
Handling (Materials)	198	6.1	19.0	9,431	4.6
Coupling/Uncoupling	190	5.8	36.0	13,608	6.6
Standing	109	3.3	34.0	7,946	3.9
Total	3,270	100	27.0	205,486	100

As seen in Table 1-2, “walking/running/stepping over” accounts for the largest percentage of *injuries* to rail employees (25 percent) as well as the largest proportion of LWDs (25 percent), with “climbing on/off/up/down” coming in a close second [FRA, 2001]. The largest median number of LWDs was shared by “sitting/riding” and

“coupling/uncoupling,” indicating that the severity of injuries associated with these acts are higher than those associated the largest percentage of LWDs [FRA, 2001].

Finally, Table 1-3 below shows the number and percentage of yard accidents attributable to a list of categorized human factors between 1994 and 1998. Nearly 60% of these accidents have been associated with problems related to switching. These accidents include “cars shoved out and left out of the clear, cars left foul, failure to couple, improperly applied portable derail, improperly lined switch, and spring switch not cleared before reversing [FRA, 2001].” Brake-related problems were the second leading cause of human-factor attributed yard accident over the same time period.

Table 1-3: Human factor-attributed train accidents in railroad yards [FRA, 2001]

Human Factor-Attributed Cause	Human Factor-Attributed Train Accidents	% of Total Train Accidents
Switching	1,810	59.8
Use of Brakes	434	14.3
Train Handling/Make-Up	386	12.8
Speed	173	5.7
Flag, Fixed, Hand and Radio Signals	84	2.8
Other	82	2.7
Main Track Authority	53	1.8
Employee Physical Condition	4	0.1
Cab Signals	0	0.0

The detailed information in Tables 1-1 to 1-3 and many others in [FRA, 2001] illustrate the need to augment and/or automate several manual rail yard tasks if the required technology exists. To fully understand the causes for the accidents and injuries that occur in rail yards, more background on specific tasks must be provided. Several rail yard tasks will be analyzed in the following section. Additionally, the triggering events and physical acts provided in Tables 1-1 to 1-3 will be revisited in the following sections as they pertain to the following manual tasks.

1.4 MANUAL RAIL YARD TASKS

Rail yards are used to divide, sort, store, repair, or inspect incoming trains. To accomplish these tasks, brakemen must walk alongside, reach in-between, climb onto, and step off of moving freight cars. These responsibilities can be monotonous, laborious, and potentially hazardous for a brakeman. Additionally, yard operations are often performed in outdoor settings 24 hours a day, 365 days a year. Brakemen are often exposed to all the elements while on the job, including heavy snowfall, pouring rain, and extreme temperatures at both ends of the spectrum. Even under normal operating conditions, these tasks can be dangerous if brakemen are not well-rested and focused on the task at hand due to limited workspace, ambient noise, and proximity to large moving equipment. An overview of applicable manual tasks, including freight car uncoupling, pneumatic brake hose assembly, and handbrake operation will be provided in the following sections.

1.4.1 Uncoupling Freight Cars

Couplers are mechanisms designed to connect one car to another for the purpose of pulling and pushing trains. All modern coupler designs used on freight cars in North America are capable of coupling automatically without intervention from a brakeman. However, couplers must be released manually using a lever that extends from the coupler to a bracket located near the exterior of a car (see Figure 1-9). Lifting the lever disengages a lock pin inside the coupler, which allows the coupler's knuckle to release. Different coupler designs are manufactured for specific freight car types. Type E couplers are the most common and are regulated by the AAR to ensure that all have similar head designs and interchangeable internal components [REB, 2000]. Diagrams depicting the open and closed positions for a pair of Type E couplers and their internal components are shown in Figure 1-7 below.

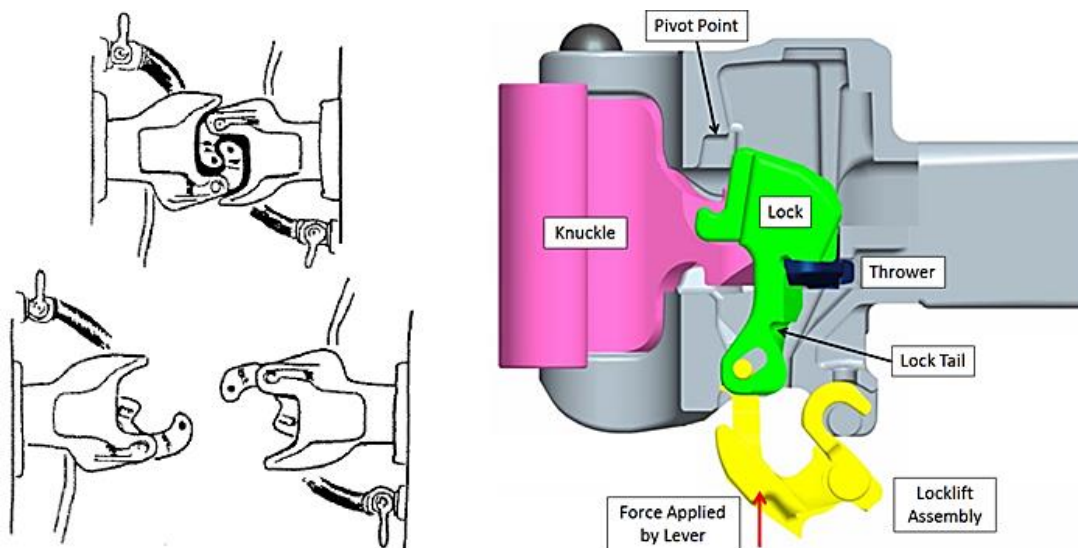


Figure 1-7: Closed/open positions of Type E coupler (left) & internal parts (right)
 [McConway, 2013, REB, 2004]

To successfully couple a pair of freight cars, the knuckle of at least one coupler must be rotated fully in the clockwise direction to its open position (see Figure 1-7, bottom-left). The two couplers should then be moved toward one another with enough momentum for their knuckles to rotate counter-clockwise about their respective pin joints to the closed position (see Figure 1-7, top-left).

A cross section of a Type E coupler and its internal components is shown in Figure 1-7 (right). The internal operation of Type E couplers during uncoupling can be described in three states: (1) locked, (2) lockset, and (3) thrown. The lock pin begins in the “locked” position (see Figure 1-8, left), where its weight rests on the knuckle of the coupler to prevent it from rotating clockwise to the open position. As the uncoupling lever is lifted, the lock lift assembly forces the lock pin upward until it makes contact with the pivot point (see Figure 1-7, right) inside the coupler. This designates the “lockset” position (see Figure 1-8, middle). The continued rotation of the lever forces the lock pin to rotate about its fulcrum. The lock pin reaches the “thrown” (unlocked)

position once its leg clears the tail of the knuckle (see Figure 1-8, right). The complete motion of the lock pin inside the coupler as the uncoupling lever is pulled is shown in Figure 1-8. It should be noted that couplers will only disengage under compression. If a pair of couplers is in significant tension, the knuckle will prevent the lock pin from disengaging.

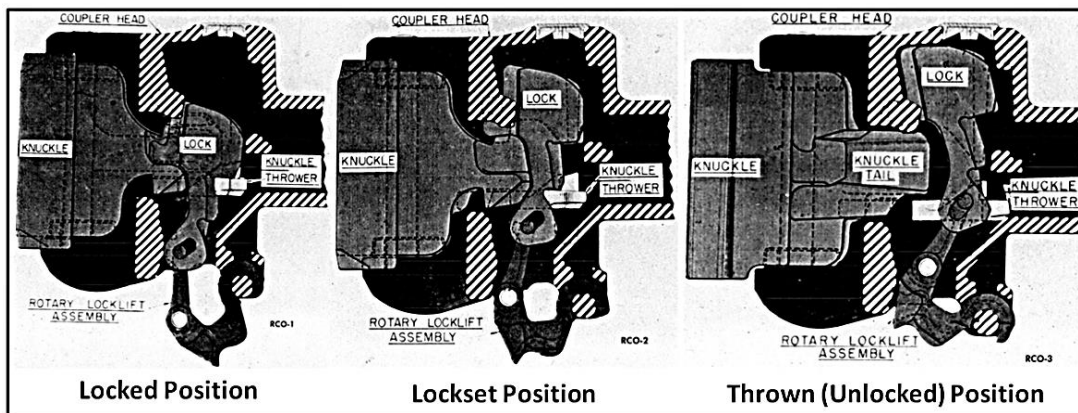


Figure 1-8: Three internal states of a Type E coupler [AAR, 1965]

As the lock pin transitions from the lockset to the thrown position on the right side of Figure 1-8, the lock tail (located on the rear of the lock pin) makes contact with the knuckle thrower (Figure 1-7, right). This forces the knuckle thrower to rotate counterclockwise until it makes contact with the knuckle, causing it to rotate slightly in the clockwise direction toward its open position, which allows the knuckle tail to swing out from below the lock pin. In this state, the coupler will remain open even if the lock pin falls back down unintentionally.

To initiate the uncoupling process detailed above, the handle of an uncoupling lever must be manually lifted to disengage the lock pin inside the coupler. The handle is attached to a rod with an eye at the end that connects the uncoupling lever assembly to a coupler. Many different types of uncoupling levers exist with varying handle and rod

geometries, which is dictated by type of freight car and coupler that it is attached to. A standard uncoupling lever assembly is shown in Figure 1-9 below.

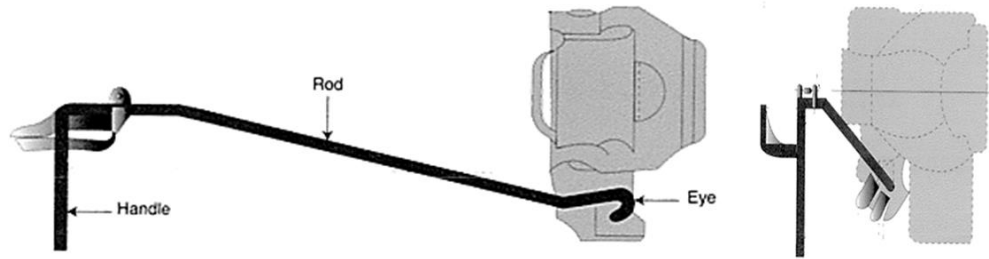


Figure 1-9: Standard configuration for an uncoupling lever assembly [REB, 2000]

Each freight car has two uncoupling levers located at opposite ends of the car. FRA provision 2004b requires that the handle of the uncoupling lever should be no more than 12", preferably 9", from the side of the car and should have a minimum of 2" clearance around the handle for grasping [Edwards, 2006]. Figure 1-10 illustrates the correct location for an uncoupling lever assembly as designated by the FRA.

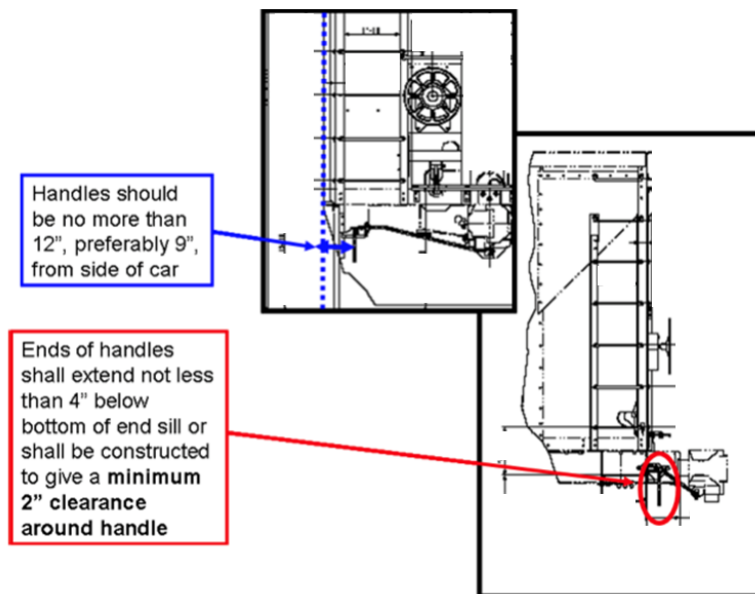


Figure 1-10: Location of uncoupling lever on freight car [Edwards, 2006]

Manual uncoupling of freight cars requires a brakeman to walk or run next to a moving train in a flat or hump yard and reach up to one foot between a pair of 100 ton freight cars to grasp, pull, and hold the uncoupling lever handle until car separation is achieved. Levers can require more than 75 pounds of force to successfully uncouple and often require multiple pulls to disengage the lock. This description alone is justification enough to automate this process to eliminate the risk of injury to a brakeman due to the physical requirements demanded of uncoupling. Nevertheless, the safety of manually uncoupling cars in flat and hump yards will now be evaluated using the injury data introduced in the previous section. Tables 1-4 and 1-5 below list the injury statistics categorized by physical acts related to manual uncoupling in 1997-1998 [FRA, 2001].

Table 1-4: LWD yard injuries specific to uncoupling cars in flat yards, '97-98

Uncoupling Freight Cars in Flat Yards					
Physical Act	Injuries	%Total of Injuries	Median LWDs	Total LWDs	% of Total LWDs
Walking	762	16.7	27.5	47,994	17.4
Lining Switches	334	7.3	29	23,801	8.6
Getting Off	283	6.2	28	17,413	6.3
Riding	239	5.2	33	16,699	6
Stepping Down	233	5.1	22	12,158	4.4
Pulling Pin	164	3.6	24	8,676	3.1
Operating	135	3.0	30	8,987	3.3
Getting On	111	2.4	28	6,887	2.5
Pulling Pin Lifter/Operating Uncoupling Lever	101	2.2	43	7,294	2.6
Climbing Over/On	99	2.2	21	5,061	1.8
Lifting Equipment (Tools, Parts, etc.)	78	1.7	19.5	4,663	1.7
Stepping Up	64	1.4	28	4,378	1.6
Adjusting Coupler	52	1.1	21	2,660	1.0
Reaching	46	1	17	1,830	0.7
Jumping From	35	0.8	53	2,899	1
Handling Car Parts	25	0.5	30	1,907	0.7
Handling Locomotive Parts	17	0.4	17	393	0.1
Totals (Direct Correlations)	930	20.3	156	56,419	20.4
Totals (Direct & Indirect Correlations)	2,778	60.8	471	173,700	62.8

The orange items highlighted in Tables 1-4 and 1-5 are physical acts defined by the FRA that are directly associated with uncoupling operations in the listed yard type. According to Table 1-4 above, approximately 20% of employee injuries sustained in 1998 were caused by physical acts that are nearly exclusive to manual uncoupling in flat yards (e.g. pulling pin lifter, reaching, lining switches, etc.). It should be noted that the remaining physical acts listed in Table 1-4 (e.g. walking, handling car parts, etc.) are also required by, but not exclusive to manual uncoupling. It is impossible to know whether any of the injuries listed in white occurred while performing manual uncoupling without the original incident reports, but one could assume the total percentage of injuries related to manual uncoupling was likely higher than 20% in 1998 [FRA, 2001].

Table 1-5: LWD yard injuries specific to uncoupling cars in hump yards, '97-98

Uncoupling Freight Cars in Hump Yards					
Physical Act	Injuries	%Total of Injuries	Median LWDs	Total LWDs	% of Total LWDs
Walking	762	16.7	27.5	47,994	17.4
Pulling Pin	164	3.6	24	8,676	3.1
Operating	135	3.0	30	8,987	3.3
Pulling Pin Lifter/Operating Uncoupling Lever	101	2.2	43	7,294	2.6
Lifting Equipment (Tools, Parts, etc.)	78	1.7	19.5	4,663	1.7
Adjusting Coupler	52	1.1	21	2,660	1.0
Reaching	46	1	17	1,830	0.7
Handling Car Parts	25	0.5	30	1,907	0.7
Handling Locomotive Parts	17	0.4	17	393	0.1
Totals (Direct Correlation)	363	7.9	105	20,460	7.4
Totals (Direct & Indirect Correlations)	1,380	30.2	229	84,404	30.6

Table 1-5 shares many of the same physical acts as they relate to manual uncoupling in hump yards. The percentage of injuries directly related to manual uncoupling in hump yards (approximately 7%) is lower than in flat yards. This decrease is likely correlated to the reduced uncoupling speeds that do not require brakeman to jump on, off, or run adjacent to moving trains. In summary, Tables 1-4 and 1-5 clearly

illustrate the hazards associated with physical acts required to manually uncouple freight cars. This analysis, in addition to the brakeman's constant proximity to large moving objects justify the need to automate uncoupling to substantially increase yard safety.

1.4.2 Air Brake Hose Operation

Locomotives and freight cars are equipped with air brakes that are capable of stopping immense moving loads. The main air supply for these brakes is located on the locomotive body. Therefore, a pneumatic hose must be connected between each freight car to supply air to the entire train with air for braking. Brakemen are required to couple pneumatic brake hoses on a frequent basis. To access the brake hoses, brakemen must enter the area between two freight cars, as shown below in Figure 1-11. *This is one of the most hazardous areas to be in a rail yard due to the potential of the car or train to move unexpectedly and potentially crush the brakeman.* As such, brakemen are trained to expect cars to move on any track at any time in either direction [REB, 2004]. Figure 1-11 shows a pair of connected air brake hoses between freight cars in a rail yard.

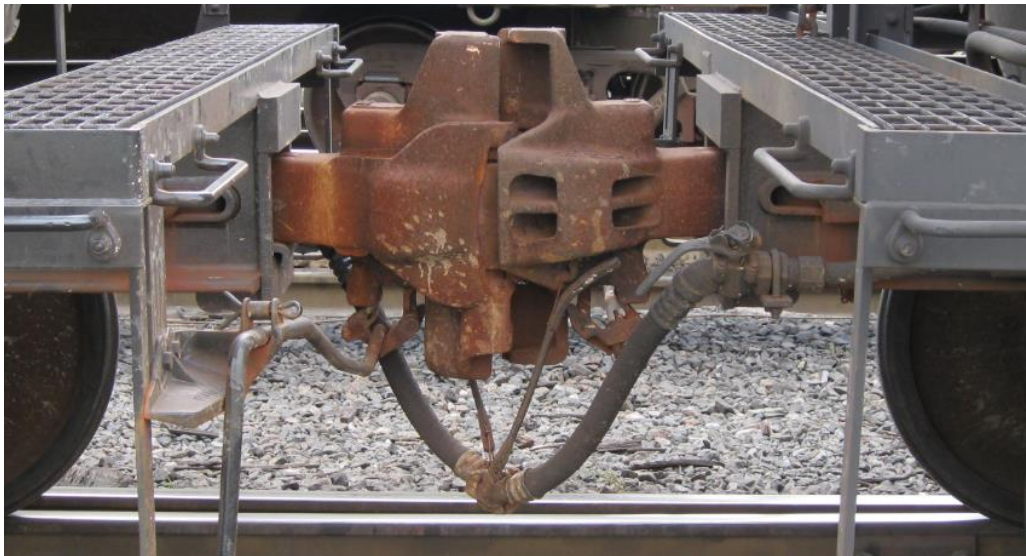


Figure 1-11: Air brake hose assemblies connected between freight cars

Figure 1-12 below depicts a manual angle cock valve (top, left), an air brake hose assembly (bottom left), and a sequence of instructional diagrams for coupling a pair of air brake hoses (right). To connect two air brake hoses, a brakeman first checks the state of the manual angle cock valve before stepping in between two static freight cars. If the angle cock valve is open, the brakeman will rotate the handle to the closed position, regardless if the hoses are coupled or uncoupled to prevent an unexpected flow of air from passing violently through the hose while the brakeman is in close proximity [REB, 2004].

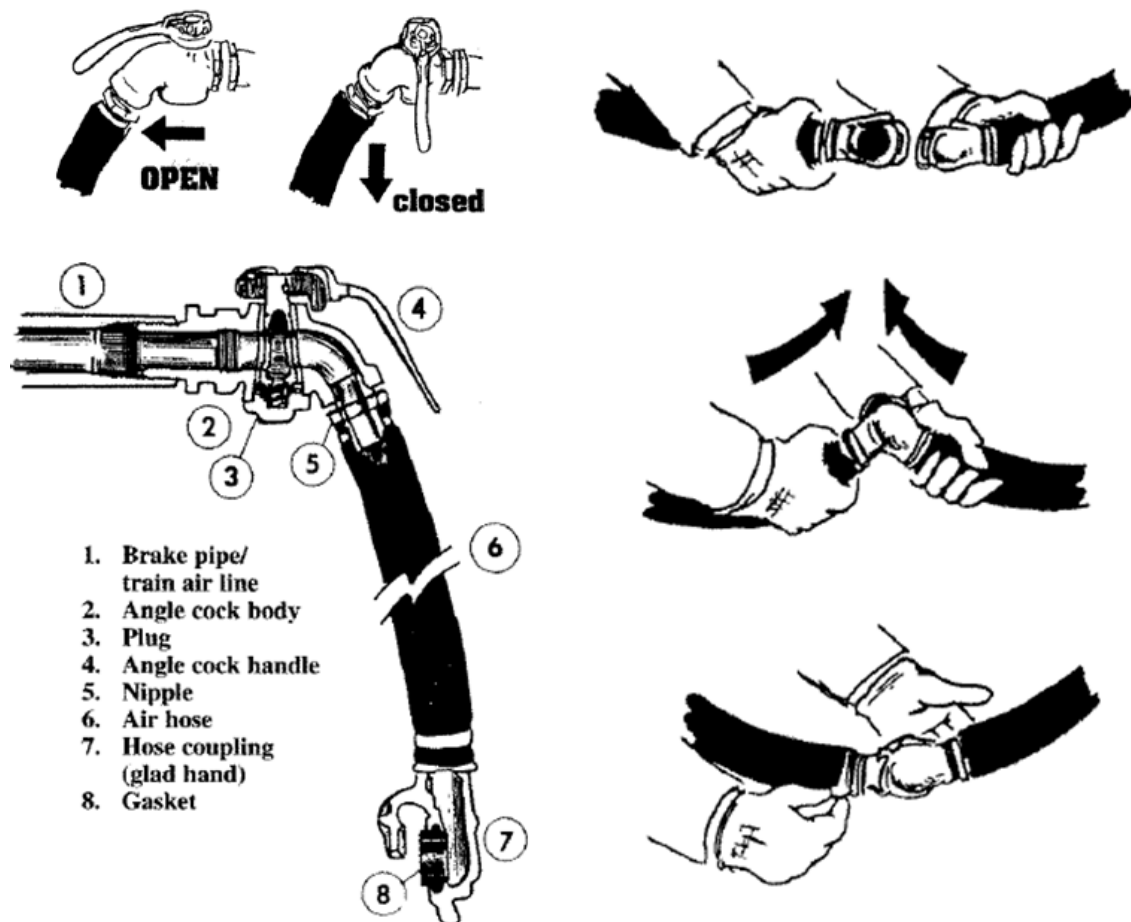


Figure 1-12: Air brake hose assembly and coupling methods [REB, 2004]

Next, the air hoses on each car must be coupled together as illustrated in Figure 1-12 (right) above using the following instructions from a training manual: “Take the hose on your right, bend it back toward you right shoulder, and hold it with your right hand near the metal hose coupling (also known as a “glad hand”). Then take the other air hose in your left hand and raise the hose from the other car so the two hose couplers come together. As you lower the touching hoses, the coupling will be made” [REB, 2004]. Once the connection between the two hoses is confirmed, the brakeman will slowly open the angle cock valves on both hose assemblies to allow air to pass through freely. For review, a step by step procedure [REB, 2004] for coupling air brake hoses is:

1. Be sure that all rolling stock is stationary.
2. Close the nearest angle cock.
3. Place left foot between rails.
4. Lift hose on right, bending back and holding same with right hand.
5. Reach under couplers for hose on other car, raising to meet other hose coupling.
6. Lower hoses so they join securely.
7. Reach across the top of the couplers (without placing your free hand on the knuckles) to open angle cock on the other car.
8. Open angle cock on the right while either holding hose or having leg against hose, in case hose coupling fails.
9. Decoupling brake hoses is typically not required. They are designed to separate automatically in tension (e.g. during uncoupling)

Injuries due to physical acts related to coupling air brake hoses and manipulating angle cock valves are listed in Table 1-6 below [FRA, 2001].

Table 1-6: LWD yard injuries specific to air brake hose & valve operation, '97-98

Coupling Air Brake Hoses & Turning Valves					
Physical Act	Injuries	%Total of Injuries	Median LWDs	Total LWDs	% of Total LWDs
Bending, Stooping	69	1.5	13	3,681	1.3
Coupling Air Hose	68	1.5	32.5	5,121	1.9
Inspecting	60	1.3	23	2,789	1
Repairing	56	1.2	20	2,973	1.1
Handling, Other	49	1.1	22	3,138	1.1
Reaching	46	1	17	1,830	0.7
Pushing	43	0.9	33	2,660	1
Stepping Over	40	0.9	24.5	2,517	0.9
Carrying	38	0.8	19.5	1,717	0.6
Crossing Over	25	0.5	9	1,781	0.6
Handling Car Parts	25	0.5	30	1,907	0.7
Opening/Closing Angle Cock	24	0.5	16	750	0.3
Uncoupling Air Hose	17	0.4	63	1,687	0.6
Crossing Between	6	0.1	41	709	0.3
Totals (Direct Correlations)	218	4.8	149	13,756	5.0
Totals (Direct & Indirect Correlations)	566	12.2	364	33,260	12.1

According to Table 1-6 above, approximately 5% of employee injuries sustained in 1998 were caused by physical acts that are nearly exclusive to the assembly of air brake hoses and the opening/closing of angle-cock valves (e.g. bending, uncoupling air hose, opening/closing angle cock, etc.). The remaining physical acts listed in Table 1-6 (e.g. repairing, handling, inspecting, etc.) are also required by, but not exclusive to air brake hose assembly and manipulation. It is impossible to know whether any of the injuries listed in white occurred while coupling air brake hoses or turning an angle cock valve without the original incident reports, but one could assume that total percentage of injuries related to these tasks was likely higher than 5% in 1998 [FRA, 2001].

1.4.3 Handbrake Operation

Each locomotive and freight car has its own brake assembly that can be operated automatically by air pressure or mechanically by hand [REB, 2004]. Many tasks in the rail yard require brakemen and switchmen to set or release handbrakes. Figure 1-13 depicts a schematic for a brake assembly found on a standard freight car. The brake cylinder in the diagram has a connection dedicated to a handbrake that operates the car brakes in the same manner as if they were engaged by air pressure [REB, 2004].

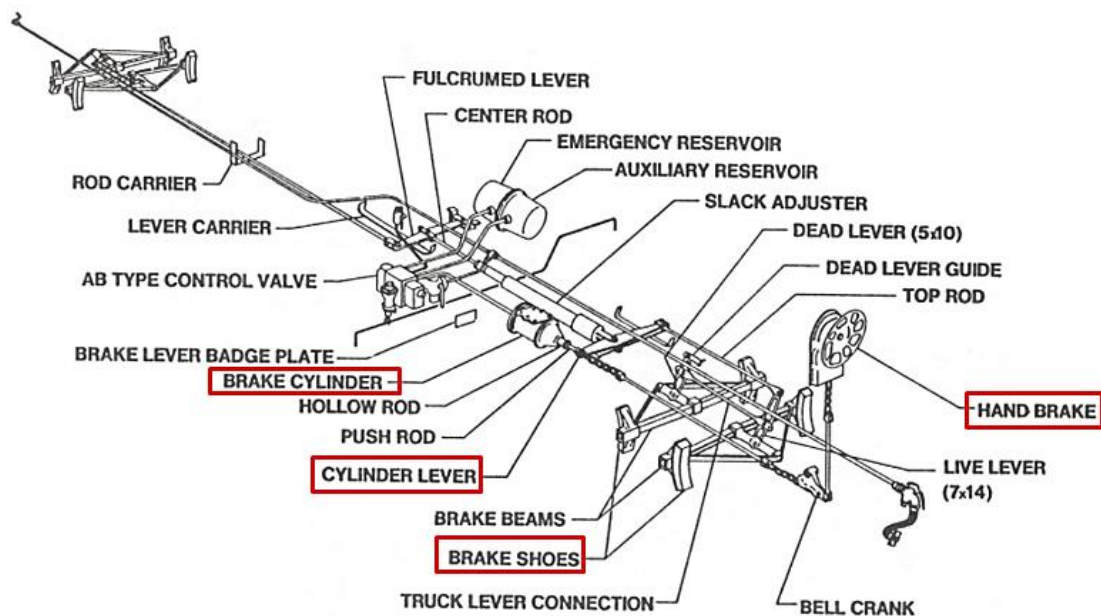


Figure 1-13: Air brake system with handbrake connection [Wabtec, 2004]

Handbrakes are located on only one end of the freight car and are typically mounted in a vertical orientation, as shown above in Figure 1-13. To access the handbrake, switchmen and brakemen must carefully climb aboard the car using available side ladders and sill steps to reach the brake platform (brake step). The FRA regulates that all freight cars should have brake steps located immediately below and extending beyond the handbrake on either side (Figure 1-14 right).

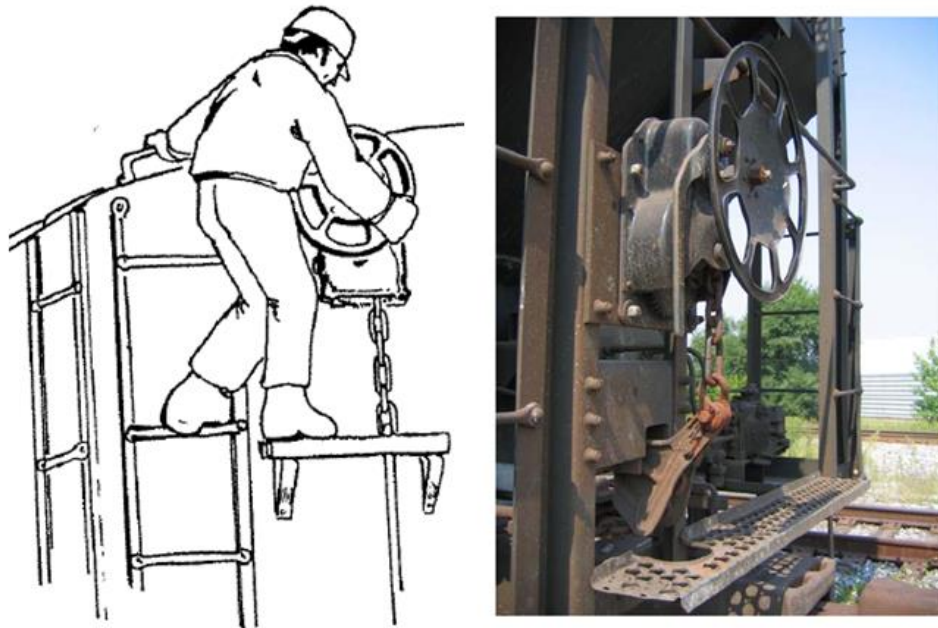


Figure 1-14: Handbrake and brake step for employee use [Edwards, 2006, REB, 2004]

Once the brakeman reaches the platform, he/she should place one hand on the brake wheel in preparation for engaging the brake. A firm grip on the ladder adjacent to the brake wheel should be maintained at all times (Figure 1-14, left) to prevent a fall during an unexpected jolt of the cars [REB, 2004]. The brakeman should then turn the wheel clockwise to apply the brake. The first few revolutions shouldn't provide much resistance due to slack in the chain [REB, 2004]. As the chain tightens, the brakeman should begin to pull the brake wheel from the bottom, using one's legs for leverage. Two or three hard pulls using this technique will adequately secure the brake. With few exceptions, one hand should provide sufficient force to engage the hand brake [REB, 2004].

To provide further insight into the braking process, a subsection of the brake assembly shown previously in Figure 1-13 is depicted in Figure 1-15. As the brake wheel is rotated, the chain directly beneath it tightens and sets a series of linkages into motion.

Four brake shoes are attached to a pair of beams (two shoes per beam) that are forced to move opposite one another by the moving linkage assembly [Wabtec, 2004]. The brake shoes make contact with a rail truck's wheelsets to apply the required braking force. The geometry and configuration of the linkage system is designed to amplify the moderate level of force provided by the brakeman to a level that can safely secure one or multiple stationary freight car(s) or decelerate cars that are moving to a stop. The red arrows in Figure 1-15 indicate the motion of each component in the assembly as the brakeman rotates the brake wheel in the clockwise direction.

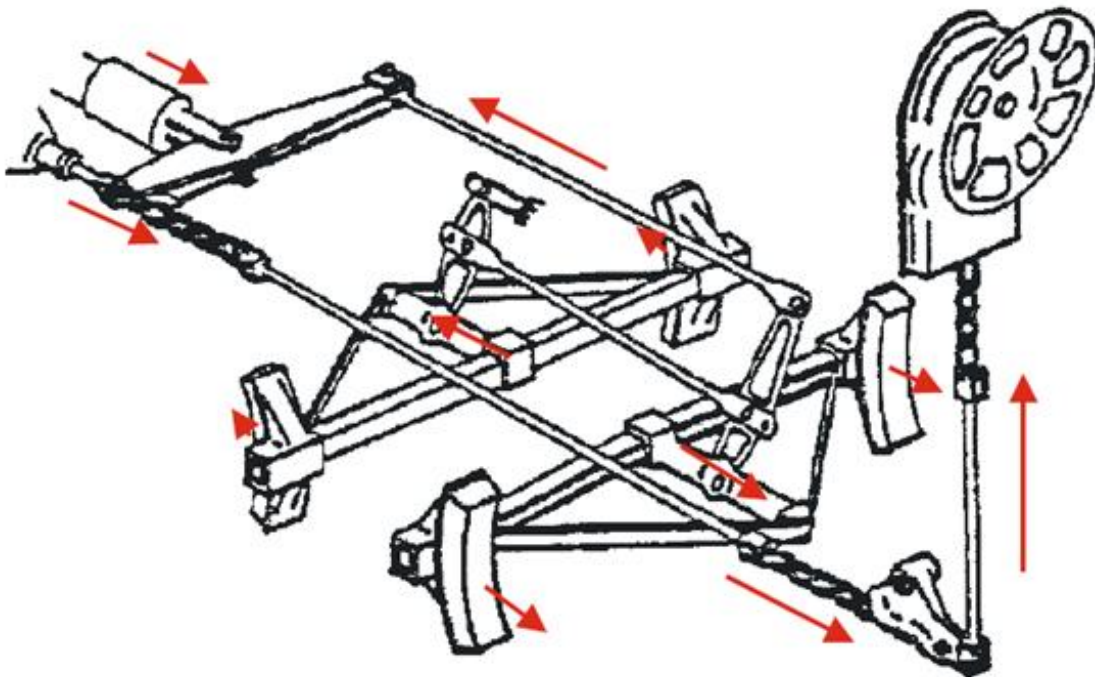


Figure 1-15: Handbrake operation on individual freight cars [Wabtec, 2004]

To release hand brakes, the vertical lever present on the right side of Figure 1-14 must be pushed to the right. If the wheel does not release, the brakeman should apply a slight downward force to free the handbrake [REB, 2004]. The wheel does have the

potential to unload abruptly, so the brakeman should never reach through the wheel to access or push the release lever to avoid serious injury [REB, 2004].

Table 1-7: LWD yard injuries specific to handbrake operation, '97-98

Handbrake Access and Operation					
Physical Act	Injuries	%Total of Injuries	Median LWDs	Total LWDs	% of Total LWDs
Getting Off	283	6.2	28	17,413	6.3
Riding	239	5.2	33	16,699	6
Stepping Down	233	5.1	22	12,158	4.4
Getting On	111	2.4	28	6,887	2.5
Standing	109	2.4	34	7,946	2.9
Climbing Over/On	99	2.2	21	5,061	1.8
Stepping Up	64	1.4	28	4,378	1.6
Reaching	46	1	17	1,830	0.7
Stepping Over	40	0.9	24.5	2,517	0.9
Handbrake, Applying	37	0.8	48	3,454	1.3
Jumping From	35	0.8	53	2,899	1
Crossing Over	25	0.5	9	1,781	0.6
Handling Car Parts	25	0.5	30	1,907	0.7
Handbrake, Releasing	25	0.5	86	3,230	1.2
Handbrake, Other	4	0.1	21	163	0.1
Totals (Direct Correlations)	896	19.6	307	55,261	20.1
Totals (Direct & Indirect Correlations)	1,375	30.0	483	88,323	32.0

According to Table 1-7 above [FRA, 2001], approximately 20% of employee injuries sustained in 1998 were the result of physical acts that are predominately required by setting and releasing the handbrake (e.g. climbing on/off cars, stepping over rails, etc.). The remaining physical acts listed in Table 1-7 (e.g. reaching, standing, crossing, etc.) are also required, but are not exclusive to accessing and applying the handbrake. It is impossible to know whether any of the injuries listed in white occurred while operating a hand brake without the incident reports, but one could assume that total percentage of injuries related to these tasks was likely higher than 20% in 1998 [FRA, 2001].

1.4.4 Summary of Manual Tasks

The RRG has identified three main tasks critical to rail yard operations that are also leading causes of injury to employees, which are: (1) manual uncoupling of rail cars, (2) air brake hose assembly and valve operation, and (3) handbrake application/release. All three of these tasks are good candidates for automation and would significantly reduce the risk of injury to rail yard employees. Chapter 2 will focus on existing robotic systems used for railroad applications and proposed methods to automate uncoupling. Many of these concepts involve making changes to existing freight car hardware to automate uncoupling, air brake hose assembly, or handbrake actuation. Although this is an effective solution, it requires an exorbitant amount of capital to retrofit 1.4 million active freight cars. Therefore, this work primarily focuses on the use of robotic hardware, 3D sensing technology, and custom end-effectors to make flexible mobile or wayside systems to automate these tasks.

To prove the feasibility of such a system, the RRG has designed, fabricated, and tested a prototype capable of autonomously uncoupling full scale rolling stock at speeds up to 2 mph. The system is capable of autonomously detecting and tracking a moving uncoupling lever to approach the lever, pull the handle, confirm successful uncoupling, and retract safely from the train in motion. Some aspects of this prototype are designed specifically for autonomous uncoupling while others are modular and can be used for numerous applications throughout rail yards, such as air brake hose assembly and/or hand brake operation. The following chapters will cover background information related to the development of the uncoupling system, subcomponents of the design, test results, and avenues for future work.

1.5 ORGANIZATION OF THE REPORT

The remainder of this report is organized as follows:

Chapter Two surveys previous work related to the augmentation or automation of manual rail yard processes. A review of available patents and existing commercialized systems will be provided to illustrate what has already been explored, and key items the Robotics Research Group at the University of Texas at Austin plans to achieve.

Chapter Three is a literature review of modern computer vision and robotic control techniques that can be utilized to design an autonomous system for rail yard applications. Object detection, object tracking, and robotic path planning techniques are discussed to present the capabilities and limitations of each.

Chapter Four provides an overview of state-of-the-art mechanical, sensing, and control technologies that could be leveraged by an intelligent system in a rail yard. Sufficient detail on commercially available vision systems (2D & 3D) and robotic platforms, manipulators, and end-effector technology will be provided.

Chapter Five proposes several methods for autonomously identifying, recognizing, and tracking a moving cylindrical object (e.g. uncoupling lever). These techniques are prototyped in software and hardware to solve a domain specific task.

Chapter Six summarizes the design, fabrication, and testing of an autonomous robotic system for rail yard applications. This system utilizes methods outlined in Chapter Five to reliably identify, track, approach, grasp, and manipulate a standard railroad safety appliance.

Chapter Seven summarizes the contributions of this research, identifies other applications for this technology, and proposes topics for future work.

CHAPTER 2: RELATED WORK

Chapter 1 provided an overview of past and present initiatives to improve safety and efficiency in the freight rail industry. Chapter 1 also identified several monotonous and potentially hazardous tasks that are still performed manually in modern rail yards and established the motivation to augment rail yard operations with autonomous robotic technology to improve safety and efficiency. This chapter summarizes the use and functionality of vision & robotic systems already utilized on the railroad. Additionally, an overview of existing patents and conceptual designs for technology that aims to automate traditionally manual rail yard tasks (e.g. uncoupling) is provided to understand what has already been attempted and/or accomplished. The following review can be divided into three separate entities: existing wayside machine vision systems, existing robotic systems for rail applications, and patented technology for automating rail yard processes.

2.1 EXISTING MACHINE VISION SYSTEMS FOR RAIL APPLICATIONS

The following review is intended to examine machine vision applications that have been developed for rail-related purposes. This review will include both results from academic research projects and commercially available vision systems. The machine vision market has witnessed steady growth in the past decade and is projected to continue growing due to advancements in computer hardware (e.g. CPUs, GPUs), camera technology (e.g. increase in quality, decrease in price and required power), availability of open-source image processing and computer vision libraries (e.g. Open Source Computer Vision & Point Cloud Library), and an increase in the number of applications that require image processing. The rail industry has started to leverage this technology to monitor the health of its rolling stock to improve legacy inspection techniques and increase preventative maintenance.

TTCI and UIUC Research

The Transportation Technology Center Inc. (TTCI), an official subsidiary of the Association of American Railroads (AAR), and The University of Illinois at Urbana-Champaign (UIUC) have publicly been active in machine vision research for rail applications since 2004. One of the earliest works that came out of UIUC was a Master's thesis titled, "Improving the Efficiency and Effectiveness of Railcar Safety Appliance Inspections using Machine Vision Technology [Edwards, 2006]." In this work, the author summarized existing FRA regulations for all safety appliances found on rolling stock and evaluated numerous machine vision algorithms that could be used to detect and measure appliances to assess their condition [Edwards, 2006]. The results of this work appear to have influenced the rail industry to start integrating wayside vision systems in addition to other modes of sensing (e.g. wheel impact load detectors, truck performance detectors, etc.) to make the railroads safer and potentially reduce yard congestion [Edwards, 2006].



Figure 2-1: Extreme deformation to sill steps on rolling stock [Edwards, 2006]

TTCI and UIUC have continued to work on varying machine vision applications related to rail applications, including but not limited to, rail equipment undercarriage inspection using multi-spectral imaging [Ahuja et al., 2007], vision systems for track

inspection [Sawadisavi, et al., 2009], and vision analysis of the energy efficiency of intermodal freight trains [Lai et al., 2007].

Commercial Systems

Beena Vision Systems Inc. is a leading manufacturer of automatic wayside, track, and handheld inspection systems for the railroad industry [BeenaVision, 2013]. Many of these systems are similar to the ones that TTCI and UIUC prototyped in the early 2000s. Beena Vision has worked with Class I freight railroads (e.g. BNSF, UP, NS, CSX) and TTCI to install and test high quality imaging hardware coupled with machine vision algorithms to inspect rail equipment and track [BeenaVision, 2013]. A brief overview of Beena Vision's products begins with their wayside systems used to inspect components on moving freight cars as they pass at low or high speeds to assess their health against the latest FRA standards. These systems are installed under, along, or over railroad tracks depending on the location and illumination requirements of the object(s) targeted. These wayside inspection systems include [BeenaVision, 2013]:

- (1) ApplianceView: Automatic Car Safety Appliance Inspection
- (2) BrakeView-Pad: Automatic Brake Pad Inspection
- (3) BrakeView-Shoe: Automatic Brake Shoe Inspection
- (4) CouplerView-CK: Coupler Securement (draft key) Inspection
- (5) CouplerView-Pin: Coupler Securement (carrier plate) Inspection
- (6) CSCView: Car Undercarriage & Structural Components
- (7) WedgeView: Friction Wedge Rise and Truck Inspection
- (8) WheelView: Automatic Wheel Profile Measurement
- (9) TrainView: Full Train Imaging and Inspection



Figure 2-2: Beena Vision wayside inspection system [Nayebi et al., 2012]

The second product category offered by Beena Vision includes track inspection systems: these systems are used to inspect all track components, including the rail, fasteners, and ties. Specially equipped vehicular systems are used to generate rail profile measurements, calculate track gauge, and measure cross-level between two rails. The products [BeenaVision, 2013] offered in this category are:

- (1) SurfView-Rail: Vehicle Based Rail & Track Imaging
- (2) TrackView-Profile: Vehicle Based Rail Profile & Track Geometry Unit

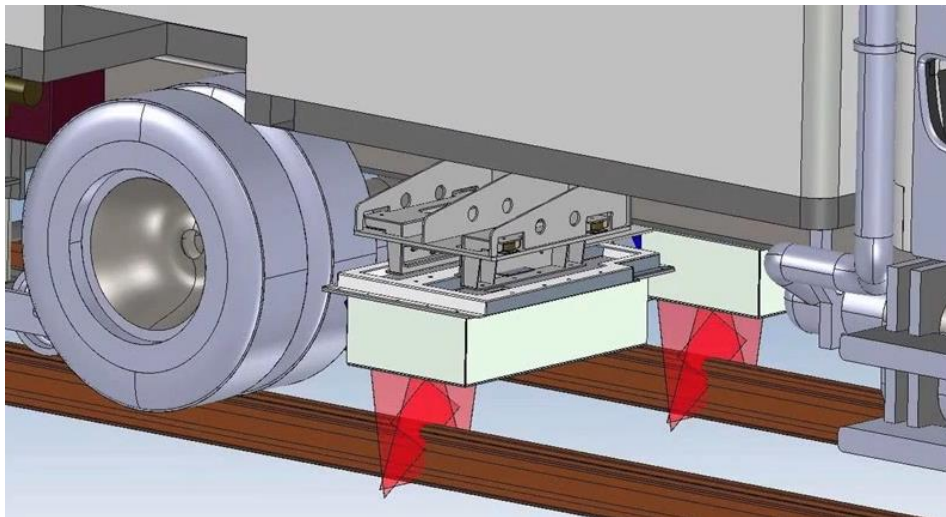


Figure 2-3: “TrackView-Profile” vehicular track inspection system [BeenaVision, 2013]

Lastly, Beena Vision offers several options for non-contact handheld and portable measurement systems to measure wheel profiles, wheel diameters, rail profiles, track parameters, track curvature, and axles (back-to-back). These products are:

- (1) DiamView: Beena Wheel Diameter Gauge
- (2) LazerView: Handheld Wheel Profile Measurement
- (3) TrackView-HH: Handheld Rail Profile & Track Parameter Measurement

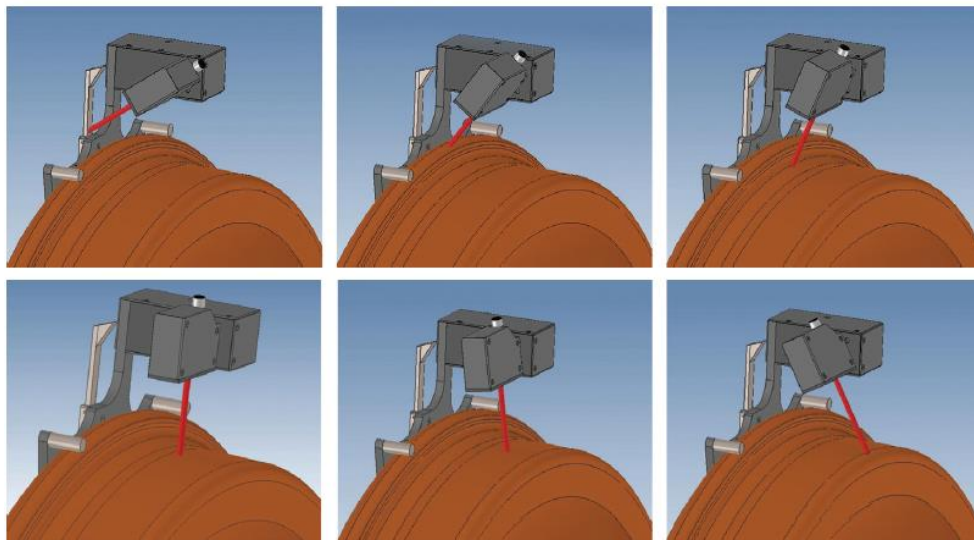


Figure 2-4: “LazerView” non-contact handheld laser wheel profile [BeenaVision, 2013]

Machine Vision Summary

While the machine vision research performed at TTCI and UIUC was applied successfully for certain rail-related tasks, it was of limited use for detecting uncoupling levers. The RRG was not able to use the same hardware as UIUC or Beena Vision due to the high equipment & calibration costs. Additionally, real-time processing of 3D data using a combination of image processing and machine learning techniques is required to control a robotic system. More detail on these techniques is provided in Chapters 5 and 6.

2.2 MODERN ROBOTIC SYSTEMS FOR RAIL APPLICATIONS

This section will provide an overview of robotic systems currently used in the freight rail sector for manufacturing, inspection, and maintenance purposes. Robotics technology has not yet had a large impact on the freight rail industry, but a lot of progress has been made over the last decade. Manufacturing of freight rail cars and components is currently the most prevalent example of robotic systems in the rail industry. Figure 2-5 below illustrates a schematic for a robotic manufacturing cell supplied to one of the largest wagon builders in Africa by Yaskawa Motoman for the welding and manufacturing of new coal wagon bellies [Motoman, 2007]. This system has two robotic manipulators connected to specialized welding systems and has a total of 20 synchronized servo axes to efficiently weld the seams of coal wagon bellies. These workcells aim to produce 500 new units annually with the robotic cell below [Motoman, 2007].

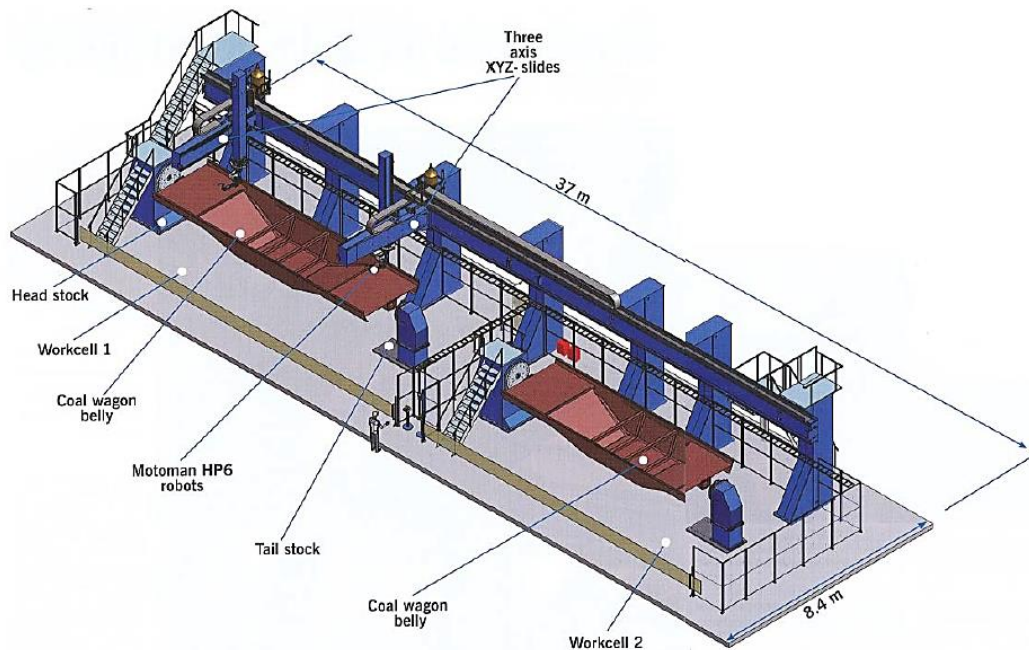


Figure 2-5: Motoman robotic gantry system for welding coal cars [Motoman, 2007]

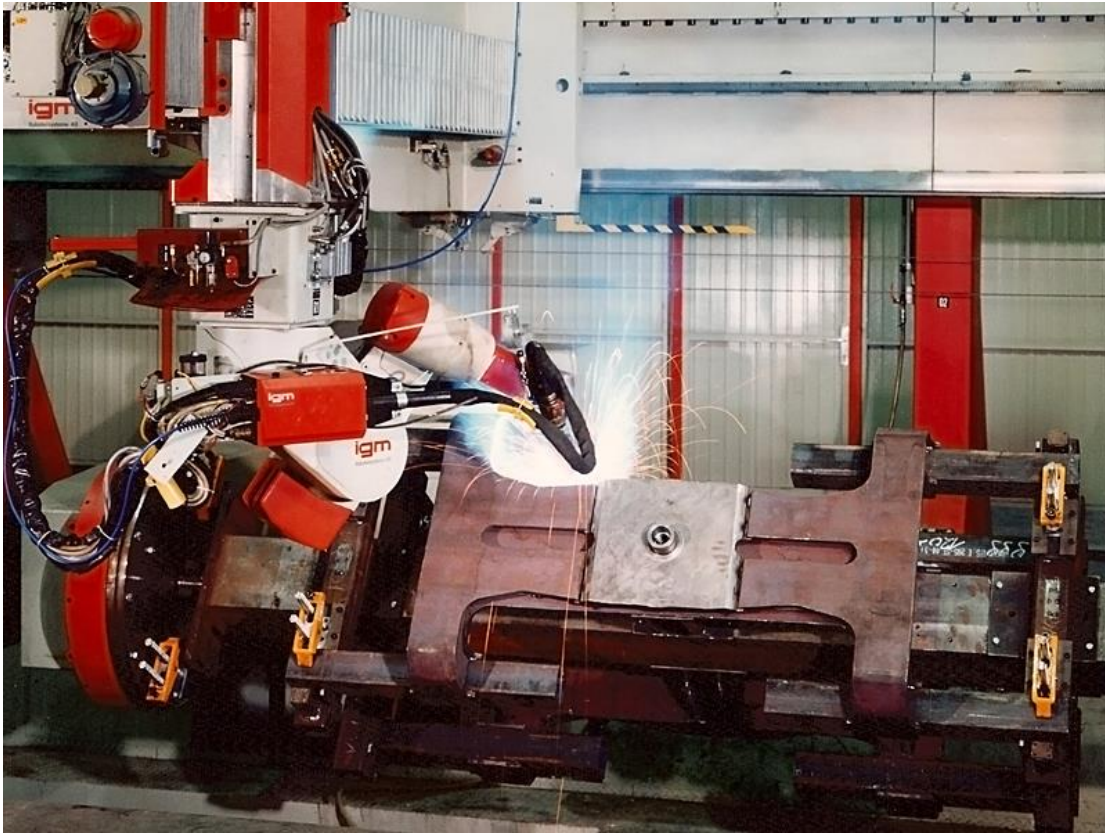


Figure 2-6: igm gantry robot welding a bogie cross frame [igm, 2013]

A similar system supplied by igm Robotic Systems AG is designed to manufacture bogie cross frames for rolling stock and is shown in Figure 2-6 above. This company specializes in custom welding robot systems for welding all types of components for rail vehicles, including motors, frames, pivoted bogies, side panels, roofs, front panels for cars, locomotives, subways, streetcars, and high-speed trains [igm, 2013]. Robotic workcells provide substantial value for the production of large-scale components, allowing manufacturers to cut costs and increase throughput of products that would otherwise require time-consuming methods and highly skilled labor.



Figure 2-7: Plasser & Theurer APT 1500 R rail welding robot [P&T, 2013]

Robotic systems are currently emerging in the track welding industry. Historically, rail track was manufactured in 60' lengths and connected in the field using bolted joint plates. Over time, lengths of track are subjected to a wide variety of stress patterns, the most complex of which occur at these bolted joints [P&T, 2013]. Consequently, these local stresses lead to rapid crack growth and failure. Therefore, the rail industry has transitioned to continuous welded track that is often produced in lengths up to 720m at the factory. These lengths of track are transported via rail to the site and joined using a process called “flash-butt welding” [P&T, 2013]. This technology provides the most uniform quality welds over the entire length of rail, minimizing track failure due to cracking [P&T, 2013]. Flash-butt welding is a mature technology, but until 2013, it had not been automated. Figure 2-7 shows a fully autonomous robotic welding system manufactured by Plasser & Theurer.

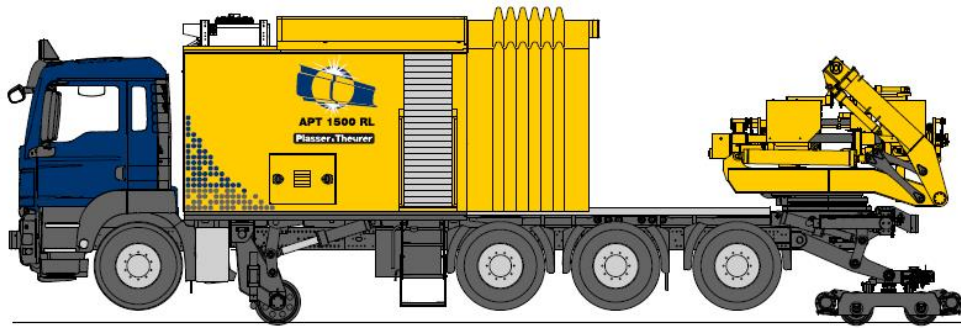


Figure 2-8: Integrated robot/lorry design (top) and resulting welds (bottom) [P&T, 2013]

In a recent news release, the manufacturer states: “the welding robot performs an automated welding sequence without manual interaction, thus achieving a high reproducible welding quality. The rails to be welded are lifted automatically into the welding head, placed in position with the help of automatic height centering and running edge alignment and the entire welding process is supervised constantly by a special measuring system. The welding gap between the rail ends is also produced automatically, if necessary, by pulling the rails together. The entire welding sequence is evaluated and documented in the machine” [P&T, 2013]. An example of the resulting weld quality produced by the robot is shown at the bottom of Figure 2-8. Lastly, the robot is mounted on a lorry vehicle shown at the top of Figure 2-8 that enables the robotic unit to travel quickly from weld to weld to decrease the time required to assemble and install a length

of track [P&T, 2013]. This is a perfect example of how modern robotic and sensing technology is being leveraged to increase safety and efficiency within the rail sector.

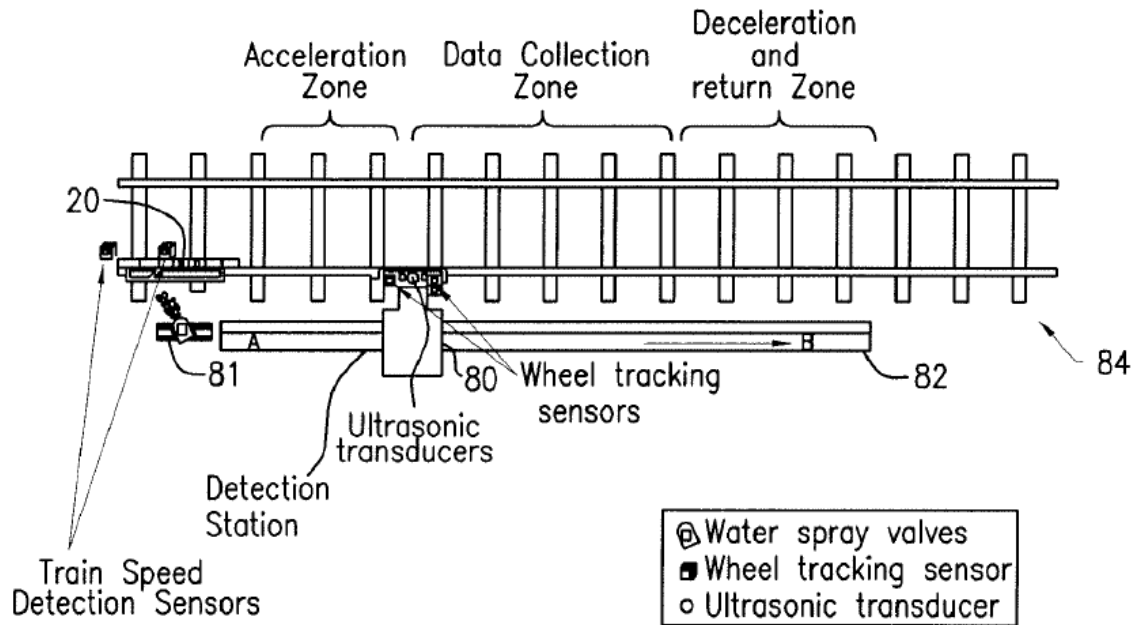


Figure 2-9: Schematic of ultrasonic wheel inspection stations [Pagano, 2009]

A wayside ultrasonic cracked wheel detector developed by Union Pacific Railroad and Dapco Industries Inc. also uses robotic technology to perform important inspection tasks. Damaged wheels due to internal cracks are of major concern to the rail industry from both a safety and economic perspective. A broken wheel can lead to a costly derailment or cause damage to several miles of track before it is noticed [Nordco, 2013]. The system shown in Figure 2-9 above uses ultrasonic sensing technology to dynamically inspect rail wheels at train speeds of up to 5mph. The ability to *dynamically* inspect wheels is critical to fulfilling Union Pacific’s desire to scan every wheel in its fleet (approx. 400,000) at 60 day intervals to eliminate all broken wheels and related derailments [McKeough, 2009].

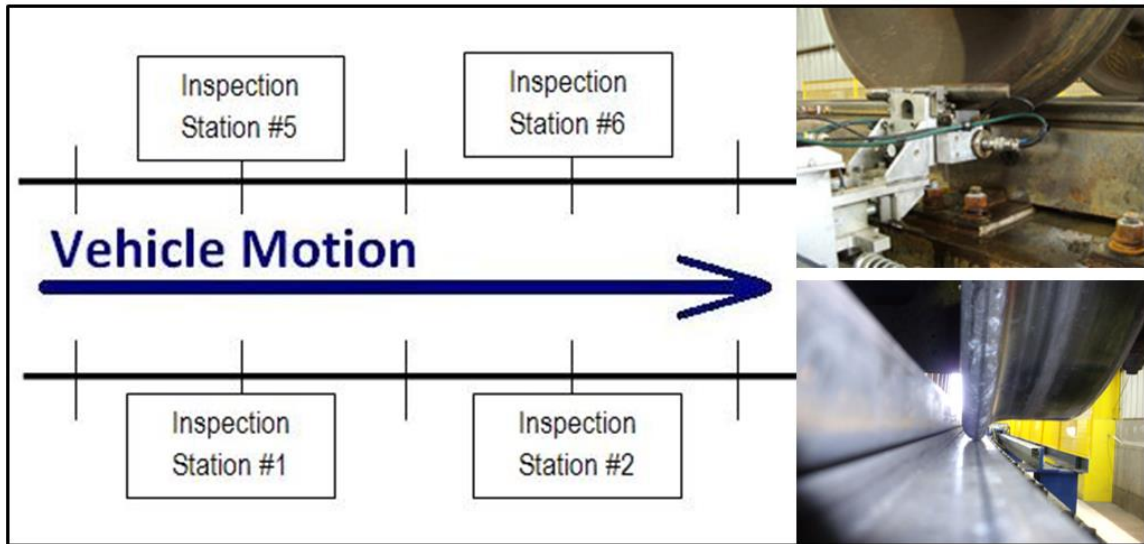


Figure 2-10: Schematic of ultrasonic wheel inspection stations [Nordco, 2013]

The overall system, shown in Figure 2-10 above, consists of four inspection stations per rail with up to eight ultrasonic transducers housed in each station. Each inspection station (see Figure 2-10, top-right) is mounted on a linear robotic axis with integrated cabling and is capable of inspecting one wheel out of every four that pass by. As the train passes, the system detects its velocity and synchronizes position with the assigned wheel. Upon synchronization, the station will raise its inspection elements directly beneath the exposed tread of the wheel to acoustically couple the pair using a fine mist of water. The inspection elements then iteratively inject pulses into the wheel to detect acoustic responses caused by any discontinuities in the wheel [Pagano, 2009]. During inspection, the system provides real-time feedback and reports critical information such as flaw type, size, and location in and across the tread and rim of the wheel [Nordco, 2013]. Once inspection is complete, the station lowers its elements and returns to its home position to prepare for the inspection of the next assigned wheel.

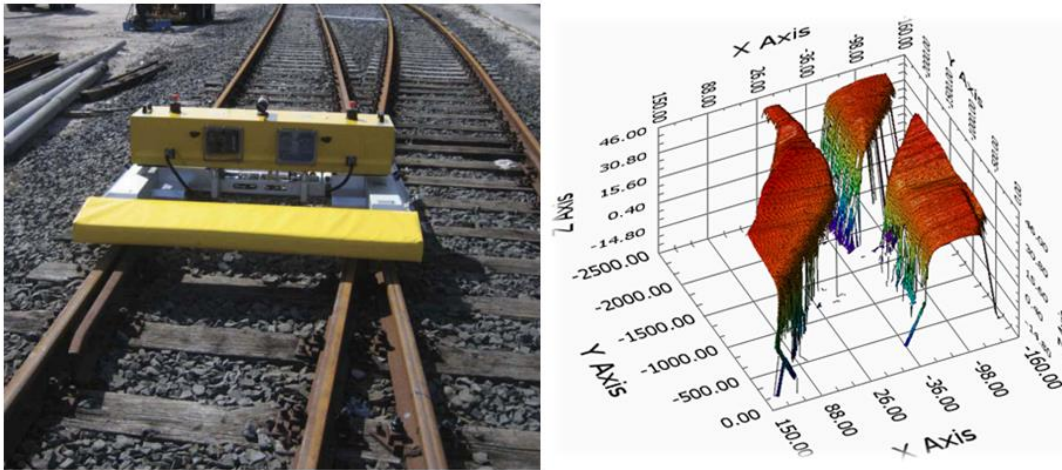


Figure 2-11: An autonomous mobile robot for remote switch inspection [Loccioni, 2013]

Felix, a switch inspection robot shown in Figure 2-11, utilizes machine vision technology to autonomously inspect the state of switches and measure the physical wear of critical components, such as track gauge and cross level. The robot has a resolution of 0.1 mm, continuous working life of 6 hours, and can operate in any weather condition. Additionally, the robot can be controlled remotely by a human operator with a PDA [Loccioni, 2013].

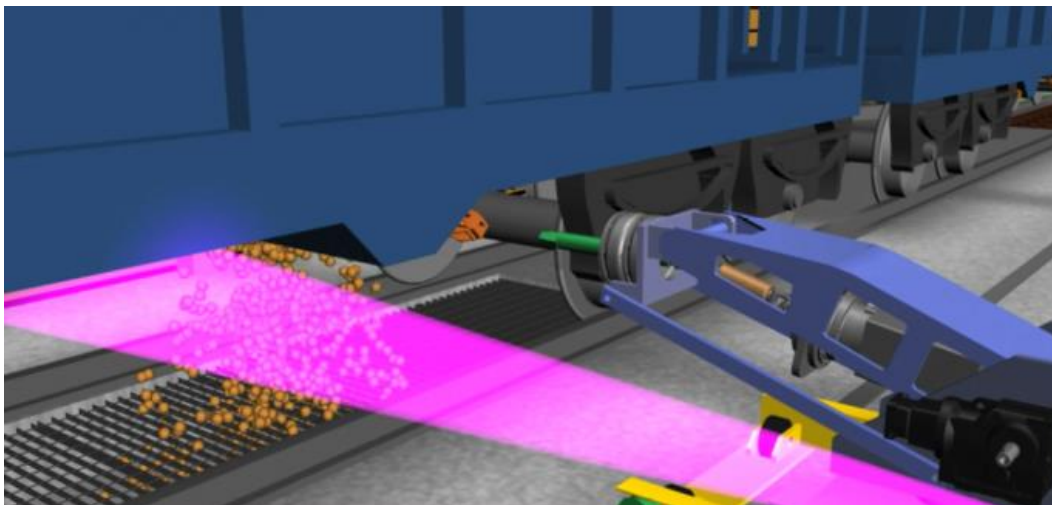


Figure 2-12: RailHawk, autonomous robot unloading rail cars [Concept, 2013]

RailHawk, an autonomous railcar unloading robot shown in Figure 2-12, utilizes machine vision technology to scan freight cars as they enter the pit to detect capstans [Concept, 2013]. Once identified, the capstan's position and orientation are sent to the robot, which uses this information to automatically align, insert, and rotate the capstan to open the gate underneath the rail car. The system is able to monitor material flow as it exits the rail car (see Figure 2-12) to detect when unloading is complete. When the car is empty, the robot is commanded to close the gate and return to its home position. The system is designed to work on stationary or moving, coupled or uncoupled freight cars [Concept, 2013]. RailHawk decreases the cycle time for the unloading process and eliminates the risk of injury to the operator.

Robotics Summary

Of the existing robotic rail-related applications outlined in this section, the ultrasonic cracked wheel detection and RailHawk systems have the most in common with the RRG's solution for an automated pin-pulling robotic system. These systems both have the capability to interact with a moving train in some capacity; a requirement also shared by uncoupling rail cars. More detail on the robotic uncoupling system designed by the RRG can be found in Chapter 6.

2.3 EXISTING PATENTS FOR AUTOMATING YARD PROCESSES

Uncoupling

The rail industry has incrementally automated hump yard processes to remove human operators from hazardous conditions and to achieve higher throughputs. The uncoupling process is the only task that is still performed manually in most modern hump operations. Over the last 70 years, 15+ patents have been filed for systems and

mechanisms that aim to automate the uncoupling process. The concepts detailed in these patents can be categorized into two fundamental approaches:

- (1) Local methods: augment existing rolling stock to remotely trigger uncoupling
- (2) Global methods: install custom wayside systems that uncouple moving freight cars with no changes to active rolling stock.

Both approaches have their respective advantages; adding local components to each freight car is a simple and reliable solution that does not require complex integration or path planning required by a wayside system. However, local methods are not cost-effective, as they require parts and labor to augment two coupler assemblies on every active freight car. Figure 2-13 includes schematics from numerous U.S. patents that propose to automate the uncoupling process by modifying/augmenting existing rolling stock hardware.

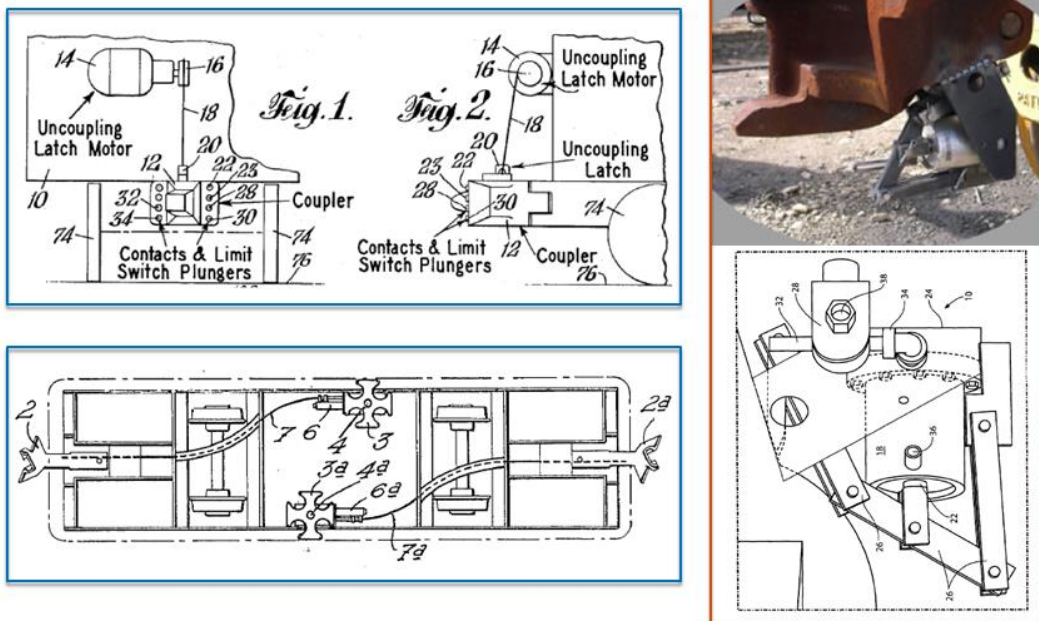


Figure 2-13: Patented concepts for local automated uncoupling systems (top-left [Heimaster, 1956] bottom-left [Sanyas, 1965] right [Kandoth, 2009])

The patent on the right side of Figure 2-13 proposes to use electrical and/or pneumatic actuators with custom mechanical assemblies to raise the lock pin inside the coupler to achieve uncoupling at the push of a button. The release button could be located in a central location next to the track, the hump engine, or locally on each car. Although the solution here is simple in design and extremely effective, the costs associated with retrofitting couplers with this technology is tremendous. There are four different types of AAR approved couplers (Types D, E, F, and H) that have operational features specific to the type of load the car typically carries. The geometry of each coupler type and its internal components also varies between coupler manufacturers. Additionally, a small portion of these couplers are top-operated, which adds another variable to the list of couplers that must be accounted for to perform a full-scale retrofit. Finally, depending on the age and condition of a coupler assembly, one-off solutions with much higher fabrication costs would be required to retrofit oddly shaped couplers. Considering these facts, it is virtually impossible that a single design for a local uncoupling mechanism could be used for the different types of couplers found on existing freight cars. Additionally, at a conservative estimate of \$350 per freight car, this retrofit would cost North American railroads and car suppliers more than \$500 million to outfit the entire fleet of 1.4 million cars. Furthermore, these assemblies would likely require periodic maintenance to guarantee reliable operation over a significant period of time.

It should also be noted that this retrofit requires several welds to attach. The cost of the skilled labor associated with welding the electro-pneumatic assembly on 2.8 million couplers would be extraordinary. Yet another problem lies in installing, routing, and/or maintaining a constant supply of energy to these coupler mounted systems. The system on the left side of Figure 2-13 uses a mechanical assembly with several Geneva cams to reset the state and remotely unlock the coupler. Again, this would be a costly

retrofit to augment every freight car to have this functionality. Most active freight cars and their components are 50 to 75 years and will likely last for another number of decades, which raises a final point regarding local uncoupling methods; why fix what isn't broken?

The second category of patents addresses the enormous costs associated with retrofitting existing coupler hardware by proposing to use a single wayside system per classification yard that is capable of uncoupling rolling stock as-is. These wayside methods can be separated into two categories: hard automation using mechanical systems and flexible robotic systems. Schematics extracted from three separate patents detailing mechanical wayside systems for autonomous uncoupling are shown in Figure 2-14.

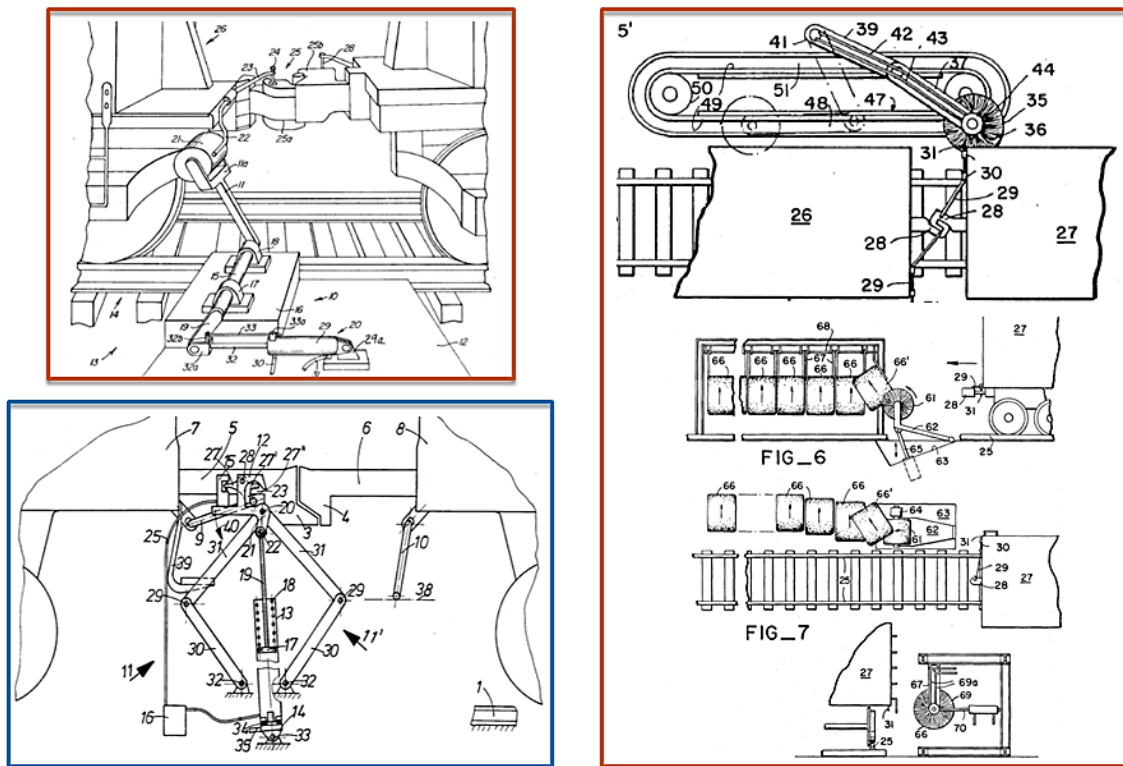


Figure 2-14: Patented concepts for wayside mechanical uncoupling systems (top-left [Peterson, 1972] bottom-left [Nell, 1977] right [Lindow, 1974])

The system in the upper-left corner of Figure 2-14 details a pivoted arm driven by a linear actuator that swings up to catch and pull the uncoupling lever as the train passes it. Similarly, the system in the lower-left corner of Figure 2-14 proposes to extend a ground mounted actuator to pull the lever as the train rolls by. Lastly, the system detailed on the right side of Figure 2-14 proposes the use of a cylindrical bristle brush rotating at a speed and direction that is capable of moving the uncoupling lever to the unlocked position. These methods and others are simple designs that could potentially rotate the uncoupling lever to the unlocked position on a moving car. However, there are a few problems with implementing such systems. The first is that many of the patents involve stationary arms that are rigidly mounted next to the railroad track. These stationary arms could pull the lever in that one position, but the cars may not uncouple based its location with respect to the hump. Ideally, the lever should be pulled well before the uncoupling point and held until car separation is confirmed. The ability to pull earlier increases the likelihood for successful uncoupling because the location at which a car separates from another depends on a variety of factors, including: the number of cars being separated at once, length of cars, coupler slack, car weight, and weight distribution. Therefore, wayside designs must have the capability to travel alongside the train with the same velocity to ensure separation over a wide range of uncoupling points. Finally, the amount of clearance present in the space around the handle varies tremendously, meaning any mechanism that swings in, around, behind, or between two cars to pull the handle must avoid obstacles, such as sill steps or other structures present near the lever. Due to the lack of well-defined standards and a wide variety of rail safety appliances, the likelihood of these methods working reliably for all car and lever combinations is very low.

Several methods using robots to uncouple cars in classification yards have also been proposed. The system portrayed on the right side of Figure 2-15 includes an

industrial manipulator using a vision system to reach in between two cars and pull the pin on a top-operated coupler with a custom end effector. The location of the locking pin and the overall operation of top and bottom operated couplers are very different. Top-operated couplers are usually only found on the front and rear of locomotives. In 2011, there were only 31,875 locomotives in service on North American railroads. From a volume standpoint, this solution is targeting less than 2.2% of all couplers in North America. Additionally, this concept is stationary, meaning it will have the same drawbacks as the mechanical approach in the previous section. The robotic arm on the left side of Figure 2-15 is mounted to a mobile platform, making it much more suitable for operation in a classification yard.

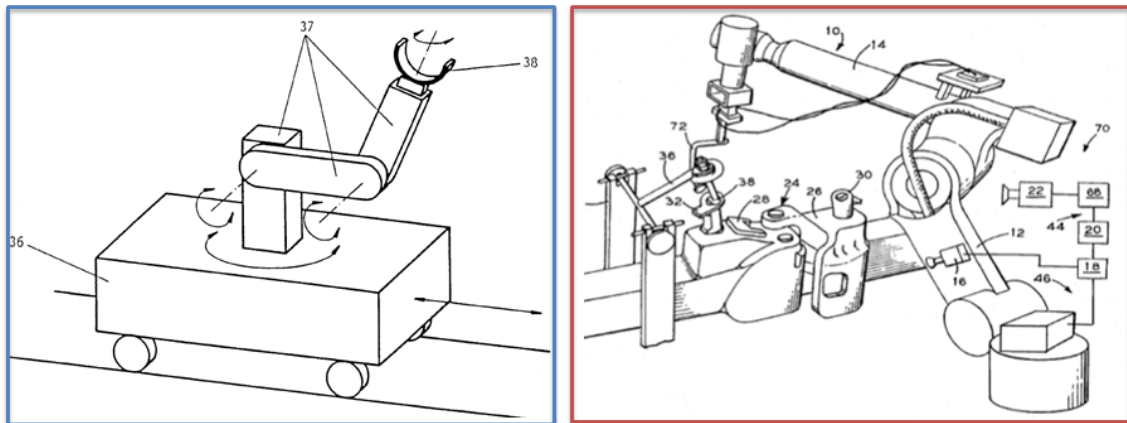


Figure 2-15: Patented concepts for wayside robotic uncoupling systems (left [Bruns, 1990] right [Cappelletti, 1996])

Summary of Automated Uncoupling Patents

These patents and others demonstrate the demand associated with automating the uncoupling process. Although many of the concepts are interesting and detailed, the majority of these designs have not addressed all of the complexities associated with uncoupling and do not appear to have been prototyped. The RRG is not aware of any

system that has been successfully tested in a laboratory environment or integrated into a classification yard. Therefore, Class I railroads are still searching for a solution to automate this process to increase safety and efficiency.

CHAPTER 3: LITERATURE REVIEW

The following review is intended to summarize standard methods that an autonomous robotic system uses to monitor, interact with, and adapt to its surroundings. The specific tasks investigated in this section include but are not limited to object detection & recognition, object tracking, and robotic path planning. Reliable object detection and recognition is the first requirement for an automated system that does not rely on user input or commands from a human operator. If the target is capable of motion, object tracking techniques must be used to “lock-on” the target to predict its future position and orientation. Finally, some form of path planning is required to safely navigate between a start and end goal once the level of system autonomy surpasses teleoperation. Additional functionalities may be needed depending on the requirements of a given application, but an awareness and understanding of the algorithms associated with the three tasks above provide a solid base for designing a robotic system to be used for rail yard operations.

3.1 OBJECT DETECTION & RECOGNITION

While object detection and recognition are closely related, an important difference exists: detection refers to the recognition of a learned object class while recognition is the identification and classification of an object within a class or subclass [Grauman, 2011]. The methods and hardware used for object detection or recognition is highly dependent on the requirements of the application. For highly constrained environments, a simple proximity sensor and accompanying bar code scanner can be used to respectively detect the presence of and identify an object. However, dynamic settings like an active rail yard require flexible solutions that can account for multiple part types, geometries, and conditions. Additionally, the system will likely need to adapt to changes in surrounding

environment including lighting, temperature, and humidity. Vision systems (2D or 3D) with accompanying algorithms can be used to detect and classify objects in these types of settings. The following sections will focus on common image processing, computer vision, and machine learning algorithms used for object detection. An overview of available hardware to gather data for processing can be found in Chapter 4.

3.1.1 Digital Imaging and Preprocessing

Imaging systems use varying configurations of digital cameras, lenses, and lighting to capture a picture of an object. An illumination source is typically used to project light on a scene where the reflected light is seen by an imager in the camera. The imager contains an array of photo-sensitive diodes (i.e. pixels) that convert photons into electrons. The number of electrons captured at each diode determines the magnitude of the strength of the reflected signal (i.e. intensity) at that pixel. These intensity values are quantized and recorded in matrix form to create digital images as shown in Figure 3-1.

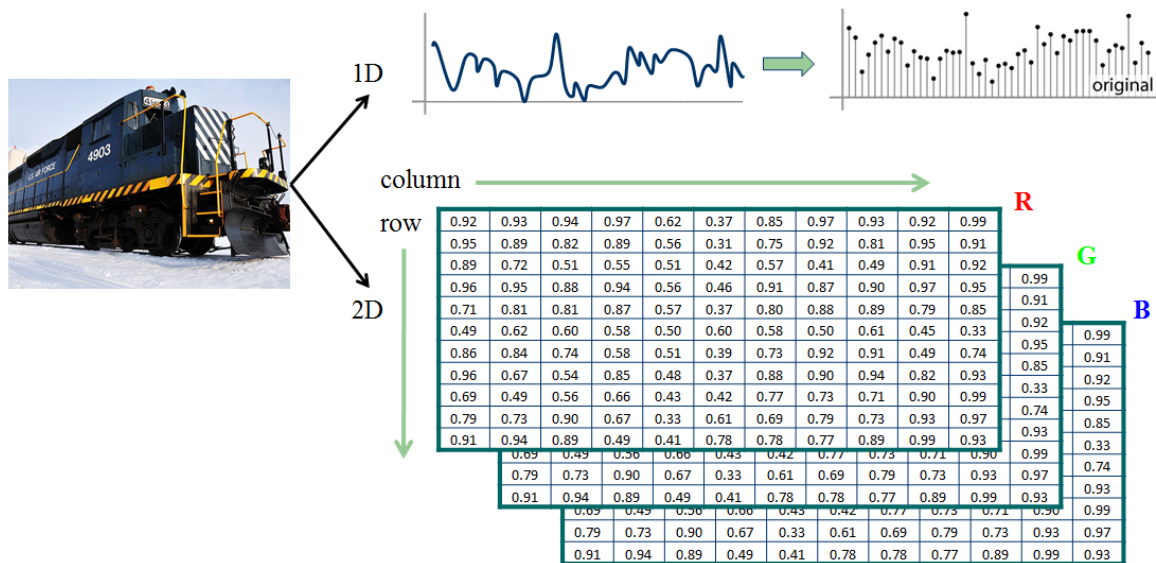


Figure 3-1: Mathematical representation of digital images [Grauman, 2011]

The bit depth, also known as the color depth, of an image defines the range of potential intensity values. Resolution increases proportionally to the bit depth and is critical to the number of unique colors or depths that a sensor can provide. More information on commercially available imaging hardware can be found in Chapter 4.

Image processing operations are often required to enhance, filter, and/or transform raw image data as it is acquired. These processes and others are referred to as “preprocessing” algorithms. The preprocessing algorithms covered in this review are used to solve common system problems, such as image noise, low dynamic range, out-of-focus optics, and the difference in color representation between input and output devices [MathWorks, 2013]. Histogram equalization is a nonlinear point operation that is used to modify the dynamic range and contrast of an image by uniformly distributing grayscale values over the entire available range and simultaneously [Bovik, 2009]. This “flattens” an image’s histogram, as shown in Figure 3-2, to boost the amount of visible detail and/or correct non-linear effects introduced by a digital imager [Bovik, 2009].

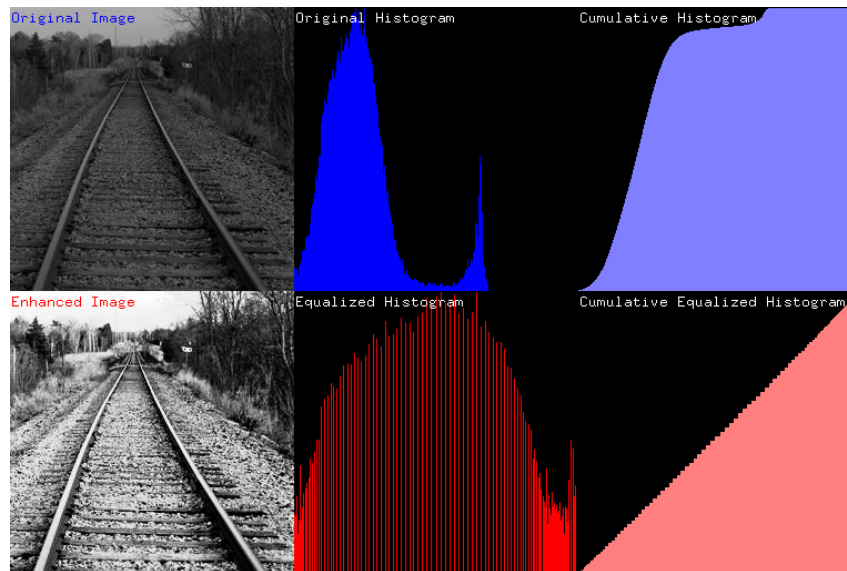


Figure 3-2: Histogram equalization [Ahn, 2006]

The original image of a railroad track in Figure 3-2 is underexposed. Due to low lighting conditions, a majority of the image's pixels are biased towards the darker end of the intensity gradient, a fact that is made evident by both the original and cumulative histogram. After the equalization procedure is performed, the distribution of pixel intensities becomes more uniform and image contrast is increased. The histogram equalization process is defined using Equations 3.1 – 3.3 below [Bovik, 2009].

$$p_a(i) = p(a = i) = n_i/n, \quad 0 \leq i \leq L \quad (3.1)$$

$$cdf_a(i) = \sum_{j=0}^i p_a(j), \quad cdf_b(i) = i * C \quad (3.2)$$

$$b_{norm} = cdf_a(k) \rightarrow b = b_{norm} * (\max(a) + \min(a)) + \min(a) \quad (3.3)$$

Consider two grayscale images, a & b , where a represents the original image and b the equalized result. The total number of pixels in the input image is represented by n , n_i denotes the number of occurrences of each gray level i , and L signifies the bit depth of the image. Equation 3.1 calculates the normalized histogram (0 to 1) of the image, which is also defined as the probability of an occurrence of a pixel with intensity i [Bovik, 2009]. Equation 3.2 (left) defines the cumulative distribution function (CDF), i.e. the cumulative histogram, of the image as shown in Figure 3-2 (top-right). Equation 3.2 (right) then defines the linearized CDF used to “flatten” the histogram (Figure 3-2, bottom-right) of the resulting image using some constant C [Bovik, 2009]. Finally, Equation 3.3 produces the normalized and scaled versions of the equalized image b .

Many other algorithms can be used to optimally prepare images for further processing. These operations may include geometric transformations to rotate, translate, crop, or register images. If the input data is large, data compression algorithms (lossy,

lossless) may be required to conserve disk memory for data logging applications or to boost processing speed for real-time object detection. Deblurring procedures, such as the blind deconvolution algorithm, can be used if the captured data is out of focus. Standard image transforms (e.g. Fast Fourier Transform) can be used to enhance, compress, and analyze images. Numerous filters (e.g. linear, median, adaptive) can also be leveraged to suppress or remove unwanted noise from digital images. The following section on edge detection will provide more background on these filters. For the sake of brevity, this review will only provide detailed background for the algorithms that were the most useful for prototyping autonomous systems for rail yard operations. More information regarding the aforementioned techniques can be found in [Bovik, 2009] & [MathWorks, 2013].

3.1.2 Edge Detection

Edge detection is a fundamental image processing technique that is frequently used in object recognition applications. Carefully derived mathematical models have been developed to identify and locate abrupt changes in pixel intensities, also known as discontinuities, in an image. These discontinuities typically correspond to physical variations in the illumination of the captured 3D objects or changes in the properties of the objects' surfaces (e.g. changes in reflectance, texture, object boundaries, depth or orientation irregularities) [Bovik, 2009]. Hence, edge detection algorithms are often combined with higher level vision tasks to analyze and characterize real-world object. Additionally, edge detection significantly reduces the amount of data to be processed by subsequent operations at a minimum cost to an image's structural properties. Three of the most common edges detected in rail yard environments include: (1) physical boundaries or objects, (2) variations in object material (color/texture), and (3) changes in lighting due to an illumination source. The edge types are shown below in Figure 3-3.

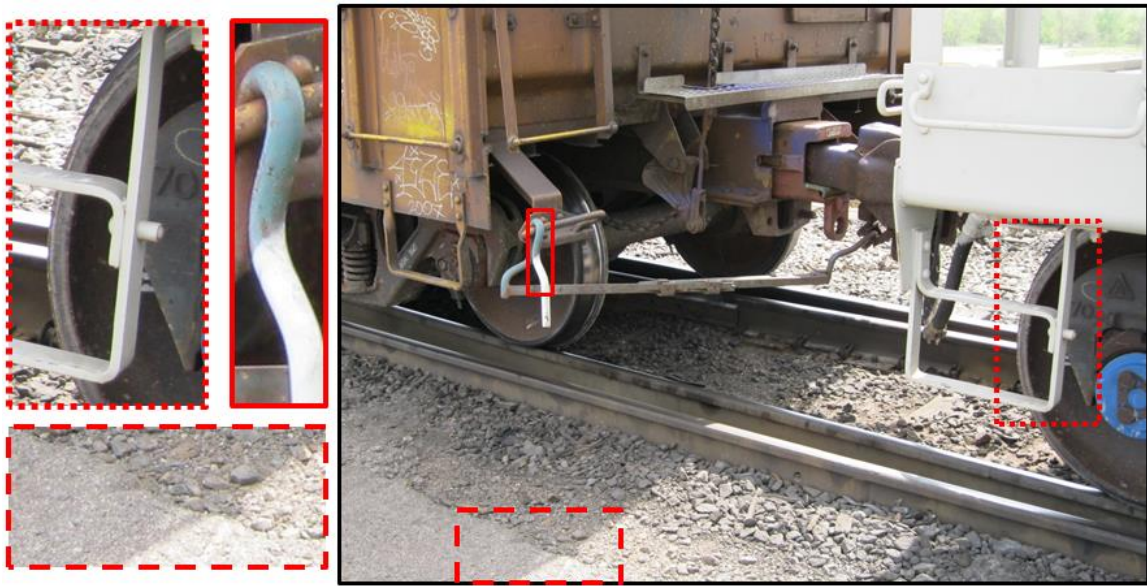


Figure 3-3: Typical intensity changes in rail yards (physical, material, & lighting)

The upper left image in Figure 3-3 is a zoomed-in view of a sill step in the foreground with a wheel present in the background. The contrast between the edge of the light and dark colored metallic components provides a good example the abrupt intensity change that represents a physical boundary between two objects. The slender image between the sill step and full image in Figure 3-3 depicts a portion of a two-toned uncoupling lever handle. A change in pixel intensity is present between the blue and white colored portions of the handle. This demonstrates the effect of varying material properties on the resulting intensity gradient of an image. Lastly, a change in illumination is depicted in the bottom-left corner of Figure 3-3, where a shadow causes a sudden change in pixel intensity. The effects of these occurrences will be evaluated post edge detection later in Figure 3-5. Several different types of edges exist and are classified by their shapes: (a) step, (b) ramp, (c) roof, and (d) real [Cattin, 2012]. These edge types with respect to their image gradients are shown in Figure 3-4 below.



Figure 3-4: Edges types (a) step, (b) ramp, (c) roof, and (d) smooth [Cattin, 2012]

Step edges (Figure 3-4a) are considered to be ideal edges where the entire intensity gradient is in one pixel width, but these types of edges never occur in real digital images [Cattin, 2012]. The intensity gradient for ramp edges (Figure 3-4b) is more gradual and is a much closer representation of actual edges in an image. These real edges can actually be thought of as step edges blurred with a Gaussian operator, and represent the closest available model to reality [Cattin, 2012]. The edge detection process can be divided into three main sub-processes described by [Bovik, 2009]:

1. Smoothing: image intensities are smoothed via filtering techniques to suppress noise and decompose edges at multiple scales.
2. Differentiation: amplifies the edges to create easily detectable patterns.
3. Decision: edges are detected as peaks in the magnitude of the first-order derivatives or zero-crossing in the second-order derivatives.

Many different edge detection methods that use varying smoothing, differentiation, and decision techniques exist. The first approach developed for edge detection used discrete approximations to an image's linear partial derivatives by convolving it with small edge-enhancing kernels (e.g. Prewitt, Sobel, & Kirsch) [Bovik, 2009]. These approximations are combined to produce a gradient magnitude in which peaks corresponding to edges could be found. These approaches do not perform well due to the amplification of high-frequency noise and inability to provide scale localization due to the convolution masks [Bovik, 2009]. A second group of edge detection methods involve smoothing an image using linear convolutions with a two-dimensional Gaussian

function $G(x, y)$ shown in Equation 3.4, where σ is the standard deviation of the distribution, to localize edges in both the space and frequency domains [Bovik, 2009]. Differentiation is then accomplished by taking the Laplacian $L(x, y)$ of an image with pixel intensity values $I(x, y)$ as shown in Equation 3.5. These two steps were combined using a Laplacian-of-Gaussian (LoG) kernel (defined in Equation 3.6) that is used to define image edges as the zero-crossings of an image's convolution [Bovik, 2009].

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.4)$$

$$L(x, y) = \nabla^2 I(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \quad (3.5)$$

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.6)$$

Although the LoG method is effective at finding edges, John Canny extended this approach to develop an optimal edge detection algorithm that is robust to image noise, has good edge localization, and provides only one response for a single edge to maintain uniqueness [Canny, 1986]. This algorithm first removes noise from the frame using the Gaussian filter shown in Equation 3.4. Next, the Sobel operator is used to calculate the intensity gradients in both directions (G_x & G_y) using Equation 3.7. The absolute magnitude of the gradient at each point (G) and orientation of each edge (θ) can then be found using Equations 3.8 & 3.9 [Canny, 1986].

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.7)$$

$$|G| = \sqrt{G_x^2 + G_y^2} \quad \text{or} \quad |G| = |G_x| + |G_y| \quad (\text{for speed}) \quad (3.8)$$

$$\theta = \tan^{-1}(|G_x|/|G_y|) \quad (3.9)$$

Non-maximum suppression is then used to remove pixels that are not believed to belong to an edge; leaving only thin lines that are thought to be real edges [Canny, 1986]. The final step in Canny's method uses a hysteresis operation that utilizes two thresholds (upper & lower) to detect strong and weak edges [Grauman, 2011]. The algorithm outputs strong edges in addition to weak edges physically connected to strong edges. The results from the standard Canny Edge algorithm using the process outlined above are shown in Figure 3-5 below.

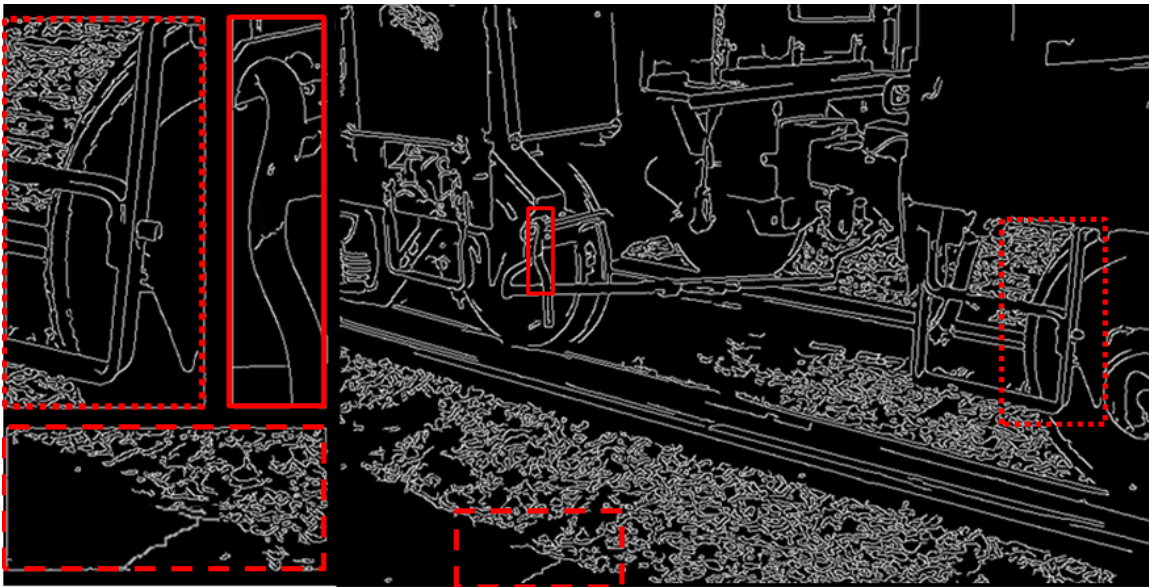


Figure 3-5: Edges detected using Canny (physical, material, or lighting)

The image shown in Figure 3-5 is the same as the one shown previously in Figure 3-3. As evident in Figure 3-5, the approach developed by [Canny, 1986] is capable of

detecting edges that represent object boundaries (physical), changes in texture/color (material), and changes in illumination (lighting). The last edge detection method covered in this section is the Shen-Castan Infinite Symmetric Exponential Filter (ISEF). The ISEF algorithm is similar to Canny's method in that it uses convolution with a smoothing kernel followed by a search for valid edge pixels, but the two functions differ on the optimal smoothing filter used [Pithadiya, 2011]. This method does have a better signal-to-noise ratio and more accurate localization than Canny, but does not address the occurrence of multiple edge responses [Pithadiya, 2011]. Thus, the Canny approach is the most widely used form of edge detection due to its strengths in all three categories. A comparison of all the covered edge detection methods is included in Table 3-1 below.

Table 3-1: Comparison of common edge detection methods (adapted from [Maini, 2009])

<u>Operator</u>	<u>Advantages</u>	<u>Disadvantages</u>
<i>Classical:</i> Sobel Prewitt Kirsch	<ul style="list-style-type: none"> • Simplicity • Detection of edges and their orientations 	<ul style="list-style-type: none"> • Sensitivity to noise • Less accurate
<i>Zero Crossing:</i> Laplacian Second Directional Derivative	<ul style="list-style-type: none"> • Detection of edges and their orientations • Having fixed characteristics in all directions 	<ul style="list-style-type: none"> • Tendency to detect same edge multiple times • Sensitivity to noise
<i>Laplacian of Gaussian:</i> LoG Marr-Hildreth	<ul style="list-style-type: none"> • Finding the correct locations of edges • Searches broader area around the pixel 	<ul style="list-style-type: none"> • Errors at locations where gray level intensity function varies (e.g. corners) • Inability to calculate edge orientation due to the Laplacian filter
<i>Gaussian:</i> Canny Shen-Castan	<ul style="list-style-type: none"> • Uses probabilistic methods to find error rate • Localization and response • Improved signal to noise ratio • Better detection rates in noisy images 	<ul style="list-style-type: none"> • Complexity • Computationally expensive • False zero crossing

3.1.3 Hough Transform

Once lines are found using an edge detection algorithm, the user often wants to collect more information about and/or classify lines that possibly represent an object in space. Given the points that make up a line, the following questions can be answered using a technique called the Hough Transform: (1) What does the line represent (2) How many lines are in the image and (3) Which points belong to which lines [Grauman, 2011]. To begin explaining the Hough Transform algorithm, the representation of a line in space must be fully understood. In the Cartesian coordinate system, a line is defined by two variables (m, b) using the slope-intercept equation. Lines can also be defined in the Polar coordinate system using two parameters (ρ, θ) , where ρ is the perpendicular distance from the line to the coordinate origin of the image frame (top-left corner) and θ is the angle the perpendicular makes with the x-axis in radians [Grauman, 2011]. The parametric description of a line in Cartesian space is shown on the left side of Figure 3-6 below. A line in the image space is represented as a point (ρ, θ) in the polar (Hough) space shown on the right side of Figure 3-6.

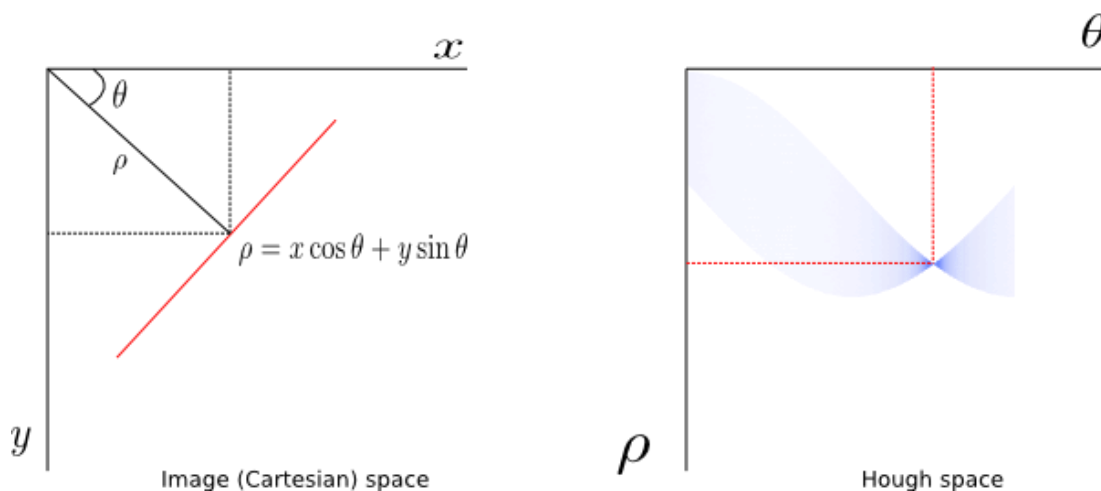


Figure 3-6: Representation of a line in Cartesian (left) & Hough (right) coordinates [Hough, 2013]

To find straight lines in an image using the Hough Transform, some type of edge detection algorithm must be applied to extract all points that belong to lines in the newly created binary image. A relationship between the lines in the Cartesian image space and points in the Hough parameter space must then be established using polar parameterization $\rho = x\cos(\theta) - y\sin(\theta)$ as shown in Figure 3-6. Once this is established, the basic Hough Transform algorithm uses the following steps [Grauman, 2011] to accumulate “votes” in a discrete set of bins in the Hough space to return a set of “straight” lines:

1. Initialize Hough space $H[\rho, \theta]$ to 0
2. For each point $I[x, y]$ in the image space and for each $\theta = [\theta_{min} - \theta_{max}]$:
 - a. $\rho = x\cos(\theta) - y\sin(\theta)$
 - b. $H[\rho, \theta] = 1$ (voting)
3. Find value(s) of (ρ, θ) where $H[\rho, \theta]$ is maximized
4. Detected lines in the image are given by: $\rho = x\cos(\theta) - y\sin(\theta)$

Now that an array of lines defined by (ρ, θ) has been calculated, the user can filter the results based on heuristics for the desired accuracy, length, orientation, and gap between line segments for a given application. The Open Source Computer Vision (OpenCV) library also provides a probabilistic adaptation of the standard Hough Transform that returns individual line segments rather than the entire line [OpenCV-2, 2013]. The Hough Transform can also be structured to search for circles in an image using the equation for a circle $(x - a)^2 + (y - b)^2 = r^2$ where (a, b) represent the center of the circle in the image coordinates and r represents the circle’s radius [Grauman, 2011]. A similar step-by-step approach to the one above can be used to calculate a and b to vote for bins by setting $H[a, b, r] = 1$ where the values are

maximized. An example of the “straight” line version of the Hough Transform on an image of an office building is shown below in Figure 3-7.

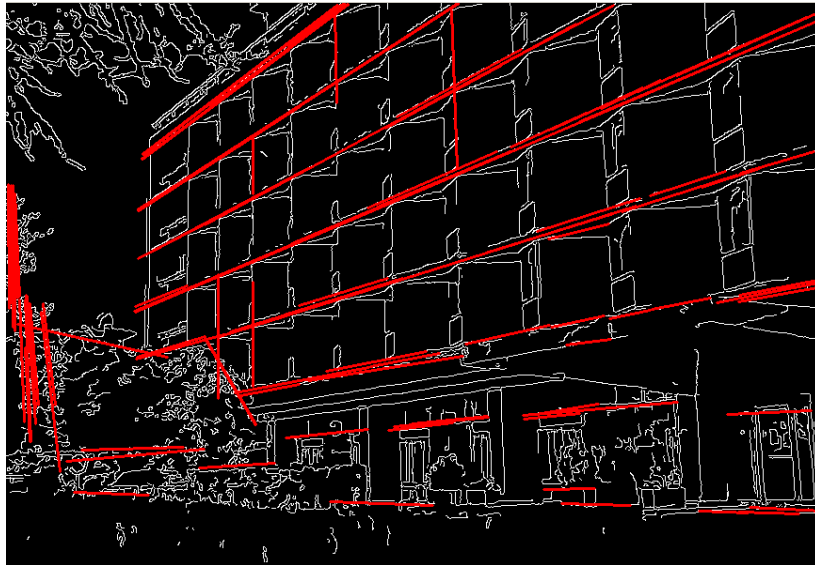


Figure 3-7: Straight lines detected by Hough transform algorithm [OpenCV-2, 2013]

3.1.4 Pattern Recognition & Machine Learning

Many different pattern recognition and machine learning techniques exist to analyze sets of data to search for particular configurations and establish relationships to ascertain meaning from that dataset. Some of the simpler pattern recognition techniques include Bayes Classifiers and K-Nearest Neighbors. A Normal Bayes Classifier is a simple classification model that assumes feature vectors from a set of given classes are normally distributed, where the entire distribution function is assumed to be a Gaussian mixture, with one component per class [OpenCV, 2013]. The Bayes algorithm can then estimate mean vectors and covariance matrices for every class using training data to calculate predictions corresponding to patterns in a set of data [OpenCV, 2013]. The K-Nearest Neighbors algorithm is a supervised scheme that combines the known classification of a user-defined number of nearest points (K) to determine the

classification of the current point [OpenCV, 2013]. This algorithm also uses training data to make predictions based on new data input to the system. More complex techniques can also be used to process visual information to find patterns, such as Neural Networks and Support Vector Machines. The following section will cover a specific adaptation of a Neural Network that is particularly generic and can be used to solve many different object detection problems. A thorough explanation of any of the aforementioned techniques can be found in the material at [OpenCV, 2013].

One of the most popular pattern recognition techniques is the Artificial Neural Network (ANN). An ANN is a machine learning algorithm based on a mathematical model inspired by the inner workings of the human brain. These networks use layers of “neurons” to define complex relationships between system inputs and outputs. A network can be designed to output a binary result for one (single-class) or multiple (multi-class) hypotheses. Multi-class classification using ANNs is especially powerful, as it can be used to search for multiple patterns simultaneously. For an object recognition application, a standard multi-layer perceptron network can be used to predict an output given clusters of connected pixels from digital images based on their shape, size, orientation, etc.

The Multilayer Perceptron (MLP) is one of the most basic ANN architectures. A MLP is a network of directionally linked neurons with an input layer, one or more hidden layers, and an output layer [OpenCV, 2013]. Each neuron has multiple input links that consist of weights and biases from previous neurons that are summed together before being transformed by a neuron’s activation function [OpenCV, 2013]. A neuron’s activation function defines the shape or behavior of the output(s) of a given neuron. Many different functions can be used to influence the output; sigmoid and hyperbolic tangent functions are two that are commonly used to achieve satisfactory results. The updated weights and biases then leave the activation function and are propagated forward

through the network, passing through layers of new neurons where the same process is repeated iteratively until the output layer is reached. A three layer perceptron network and a diagram for a single neuron are shown in Figure 3-8.

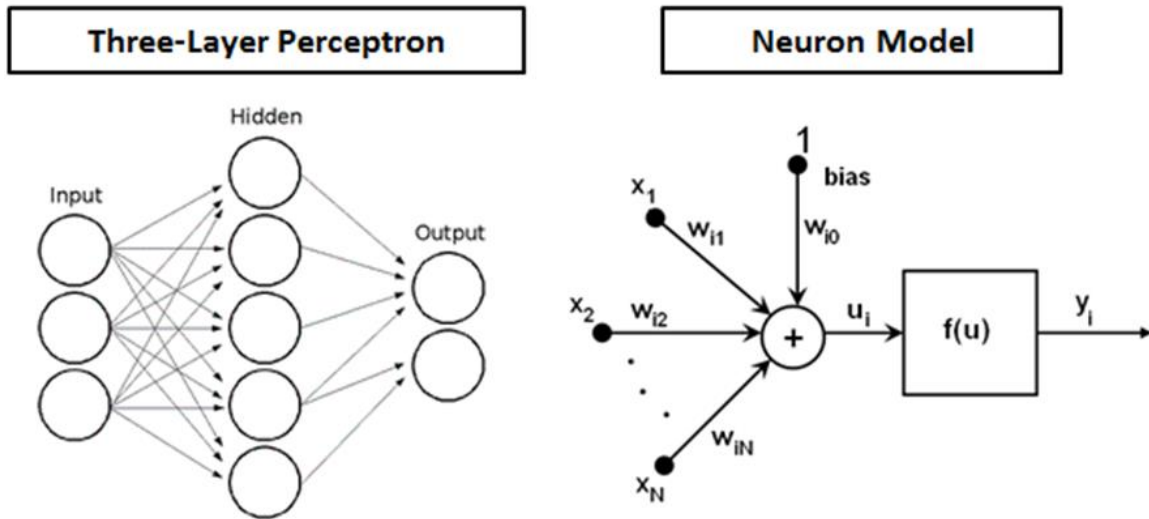


Figure 3-8: Three-layer perceptron (left) & input function to an individual neuron (right) [OpenCV, 2013]

Once the output layer is reached, the “forward propagation” step is complete. At this point, a cost function defined by the user is evaluated to determine the error present in the system. If the error in the system remains above an allowable threshold set by the network designer, the weights must be propagated back through the network toward the input layer to complete a process known “backpropagation”. To further expand on the ANN algorithm, the following list [OpenCV, 2013] provides a detailed set of items that must be coded to define, build, and train an ANN:

- 1) Define network topology and initialize all training parameters for the ANN
- 2) Define input vector size/type and target outputs for supervised learning

- 3) Feed input (feature) vector to the network and randomly initialize weights to begin forward propagation
- 4) Calculate the outputs of the first hidden layer using the weights and bias values from neurons in the input layer using the corresponding activation functions
- 5) Repeat Step 4 for all hidden layers until the output layer is reached. This completes forward propagation
- 6) Evaluate cost function to determine error present in the system (e.g. typical cost function - symmetrical sigmoid $f(x) = \beta * \frac{(1-e^{-\alpha x})}{(1+e^{-\alpha x})}$)
- 7) Iteratively calculate gradients for each layer to propagate weights back to the input layer to update all weights to enable the network to give the desired response to the provided input vectors. Many different backpropagation methods can be used
- 8) Iteratively repeat steps 4-6 until training terminates when max iteration set by the user is reached or when the error in the system falls below an acceptable threshold.
- 9) Save final weights to use for predicting/detecting objects

These steps define how a neural network is “trained”. There are multiple approaches to training a neural network. Three of the most popular training methods are: (1) supervised learning, (2) unsupervised (adaptive) learning, and (3) reinforcement learning. For supervised learning, the inputs and desired outputs are both provided by the network designer. The network takes the set(s) of known inputs and iteratively compares resulting outputs against a desired set of outputs using steps 1 - 9. Desired outputs are not

directly provided to the network for unsupervised and reinforcement learning. The network must independently determine what features it will use to classify the data given to the network.

3.2 ROBOTIC PATH PLANNING

Path planning involves the generation of a trajectory to move a robot's end-effector from an initial to a final position and orientation. Path planning is strictly kinematic and does not account for a system's dynamic properties. Robotic motion planning refers to the planned trajectories that account for both a system's kinematic and dynamic properties. Motion planning is very important to the control of a robot in order to reduce the impact of shocks, vibrations, backlash, and general wear and tear on a system's actuators [March, 2004]. The focus of this work is primarily on robotic path planning. Two classic forms of robotic trajectory generation include: (1) Joint Space Planning and (2) Cartesian Space Planning.

3.2.1 Traditional Path Planning Techniques

Joint Space Planning

Joint space trajectory planning involves the individual generation of trajectories for each degree of freedom of a robot to move its end-effector from one position to another. Essentially, each actuator is treated like a one degree-of-freedom (DOF) system and smooth trajectories are typically generated for each by ramping up smoothly to a maximum velocity at the start and an efficient ramp-down to zero velocity at the end of the trajectory [March, 2004]. Multiple trajectories can be blended together if a particular plan requires a series of different joint angles. This type of path planning has several advantages, including the simplicity of its mathematical formulation (forward kinematics) and the guarantee of smooth motion of all actuators. However, the end-effector of the

robot cannot adequately be controlled, making critical tasks, such as obstacle avoidance, nearly impossible. Therefore, another type of path planning that focuses on the path of the end-effector should be used for robotic control.

Cartesian Space Planning

Cartesian space planning requires the development of position and orientation trajectories for a robot's end-effector in six DOF. To accomplish this, end-effector positions must be converted to joint positions at discrete points in time using a complex set of kinematic equations formally known as inverse kinematics. This process is computationally more complex than the forward kinematic solution used for joint space planning and often determines the control interval by which commands can be sent to the robot's controller. One of the main problems associated with this type of path planning is that a smooth end-effector motion may result in undesirable motions at the joint level [March, 2008]. This problem is often addressed by approximating the Cartesian path of the end-effector with smooth joint trajectories. This is accomplished by dividing the path into a discrete number of "knot" points and calculating the required joint position(s) to meet the end-effector position and orientation at each knot point using inverse kinematics [March, 2008]. Many different processes for Cartesian space planning exist to generate these trajectories, including the use of cubic splines, B-splines, trigonometric splines, and blended path segments [March, 2008]. Although these methods have proven to be effective, the errors in the resulting approximation of the end-effector path may not meet the requirements of an application that requires high precision.

3.2.2 RRG Geometric-based Spatial Path Planning

This section provides an overview of a robotic path planning technique developed by the Robotics Research Group at the University of Texas at Austin that involves the

generation of end-effector trajectories based on the intrinsic physical properties of spatial curves: curvature and torsion, and their derivatives. This differs from the traditional spline-based methods in that those methods tend to be defined in terms of some independent parameter with physical meaning [March, 2008]. The primary motivation for this work is to provide an alternative approach to spatial trajectory planning by generating spatial curves based on the higher-order properties of curvature and torsion whose values correlate closely with the physical nature of a family of curves based on those values [Tesar et al., 2013].

This type of path planning primarily focuses on parametric descriptions of spatial curves as defined in Equation 3.10. In this equation, an independent parameter u is defined on some finite interval $[a, b]$ (often normalized from 0 to 1). Each value of u along this interval maps to $[x, y, z]$ coordinates defined in vector form as p . Here, the parameter u is strictly a geometric parameter and must be defined as a function of time to define a smooth motion [Tesar et al., 2013].

$$p(u) = [x(u), y(u), z(u)], u \in [a, b] \quad (3.10)$$

To fully understand the geometry of spatial curves, the Frenet Frame must be discussed. The Frenet Frame is a set of three orthogonal unit vectors attached to the geometry of the curve and is shown below in Figure 3-9. It consists of the Tangent (Equation 3.10), Normal (Equation 3.11) and the Bi-normal (Equation 3.12) [Tesar et al., 2013]. The unit Tangent is defined as the normalized first derivative with respect to u and represents the heading of the curve [Tesar et al., 2013]. The unit Normal is the derivative of the unit Tangent and represents the direction of bending in the curve. Finally, the unit Bi-normal can be calculated by taking the cross product of the Tangent and Normal [Tesar et al., 2013].

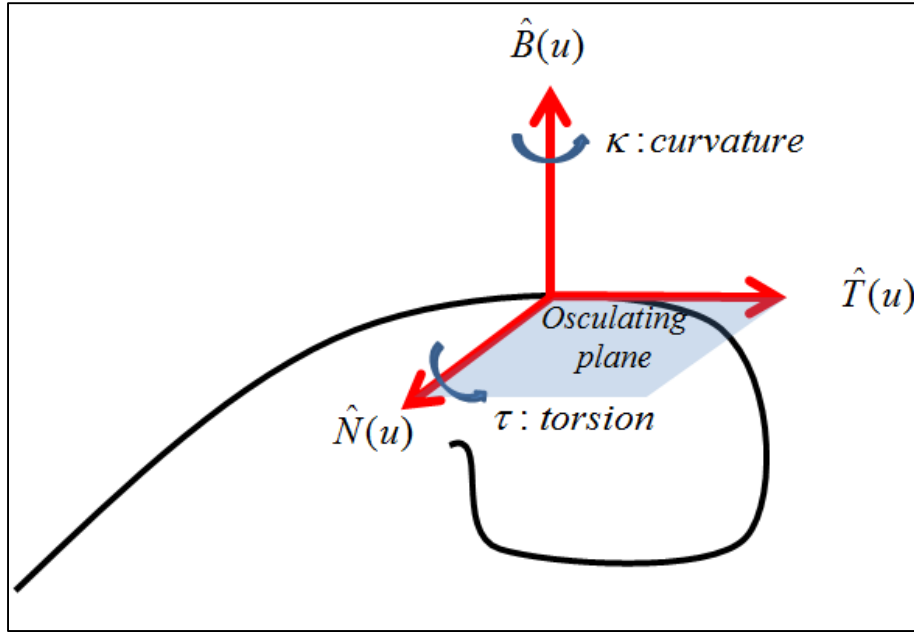


Figure 3-9: Frenet Frame on a spatial curve [Tesar et al., 2013]

$$\hat{T}(u) = \frac{\frac{dp}{du}}{\left\| \frac{dp}{du} \right\|} \quad (3.11)$$

$$\hat{N}(u) = \frac{\frac{d\hat{T}}{du}}{\left\| \frac{d\hat{T}}{du} \right\|} \quad (3.12)$$

$$\hat{B}(u) = \hat{T} \times \hat{N} \quad (3.13)$$

Curvature is a measure of the local radius of curvature around the Normal vector of a curve. The curvature of a straight line is equal to zero and is infinite for a discontinuity in the curve. For a spatial curve, the curvature will always be defined as positive and is a function of the first and second derivatives with respect to u [Tesar et al., 2013]. Using the parametric description of a curve in Equation 3.10, curvature can be calculated as a function of the independent parameter u as shown in Equation 3.14.

$$\kappa(u) = \frac{\sqrt{(y'z'' - y''z')^2 + (z'x'' - x'z'')^2 + (x'y'' - y'x'')^2}}{(x'^2 + y'^2 + z'^2)^{3/2}} \quad (3.14)$$

Torsion is a measurement of the movement in the Bi-normal direction of a curve (i.e. out of the osculating plane) [Tesar et al., 2013]. The magnitude of torsion of a planar curve is equal to zero and the curve will stay in the osculating plane. Unlike curvature, torsion is a signed value and can describe movement in both the positive and negative Bi-normal directions [Tesar et al., 2013]. Torsion is a function of the first, second, and third derivatives with respect to u and can be calculated using Equation 3.15 below.

$$\tau(u) = \frac{(y'z''x''' - y''z'x''') + (z'x''y''' - x'z''y''') + (x'y''z''' - y'x''z''')}{(y'z'' - y''z')^2 + (z'x'' - x'z'')^2 + (x'y'' - y'x'')^2} \quad (3.15)$$

In this section, the fundamental equations for using higher order geometric properties used for path planning have been provided. More detail on the derivations of these equations and their use in trapezoidal and polynomial curve generation techniques can be found in [Tesar et al., 2013] and [March, 2008]. The following section will demonstrate how these formulations are used to generate and influence trajectories using higher order geometric properties.

Path Generation Examples – Tangent, Curvature, & Torsion

In the previous section, a method for generating parametric constraints from geometric properties (i.e. curvature/torsion and their derivatives) was presented. Several examples of full trajectories will be presented here that highlight the effects and physical meanings of these parameters. The first example demonstrates the effect of the tangent scale on a two-dimensional curve. Coordinates and constraints for a spatial trajectory used to visualize the effects of geometric path planning are shown in Table 3-2. Three points (P_1, P_2, P_3) are listed with geometric constraints defined at the middle point P_3 .

Note that in this example these values are arbitrary and meant only to describe the theoretical process and underlying mathematic framework.

Table 3-2: Position and orientation at each point of defined trajectory [Tesar et al., 2013]

	P_1	P_2	P_3
x	0.0	2.0	1.0
y	0.0	1.0	3.0
z	0.0	1.0	-1.0
\hat{T}	[1, 0, 0]	[0, 0.71, -0.71]	[0, 1, 0]
\hat{N}	[0, 1, 0]	[-1, 0, 0]	[-1, 0, 0]
\hat{B}	[0, 1, 1]	[0, 0.71, -0.71]	[0, 0, 1]

The simplest constraint is the first-order tangent scale. The tangent scale $\|dp/du\|$ controls the amount of “bias” toward the tangent vector at the beginning or end of a trajectory segment. As shown in Figure 3-10, the tangent scale property biases the first order derivative in the y direction.

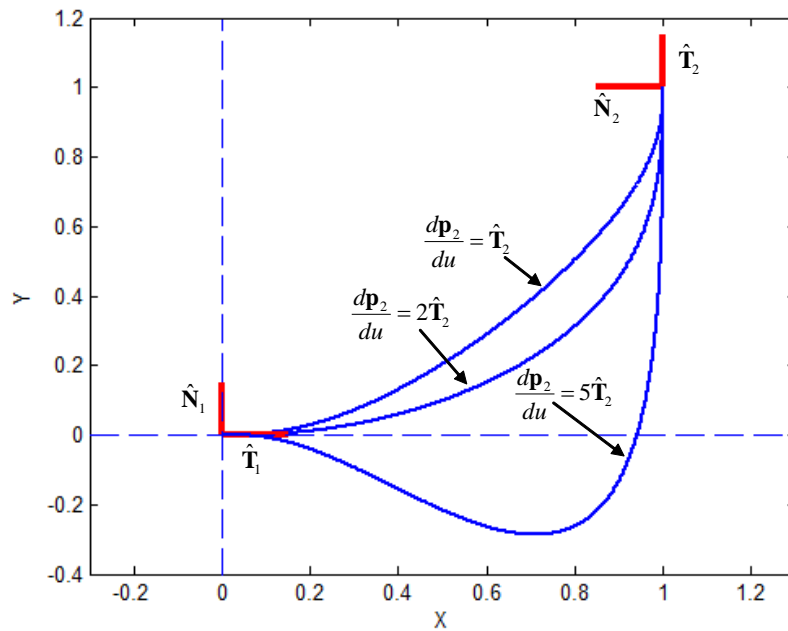


Figure 3-10: Effect of varying tangent scale on a planar curve [March, 2008]

Increasing values of the tangent scale at the end point cause the curve to follow the second tangent for a longer distance in Figure 3-10. Table 3-2 shows the calculated parametric properties for three different tangent scales, (0.5, 1.0, 2.0), for a spatial trajectory that better represents a motion plan for a robot. The effects of the tangent scale can be inferred for two and three dimensional trajectories in Figures 3-10 and 3-11 respectively.

Table 3-3: Parametric constraints using varying tangent scales [Tesar et al., 2013]

	Tan. Scale	dp_1/du	dp_2/du	dp_3/du
<i>Path A</i>	0.5	[0.5, 0.0, 0.0]	[0.0, 0.36, -0.36]	[0.0, 0.5, 0.0]
<i>Path B</i>	1.0	[1.0, 0.0, 0.0]	[0.0, 0.71, -0.71]	[0.0, 1.0, 0.0]
<i>Path C</i>	2.0	[2.0, 0.0, 0.0]	[0.0, 1.14, -1.14]	[0.0, 2.0, 0.0]

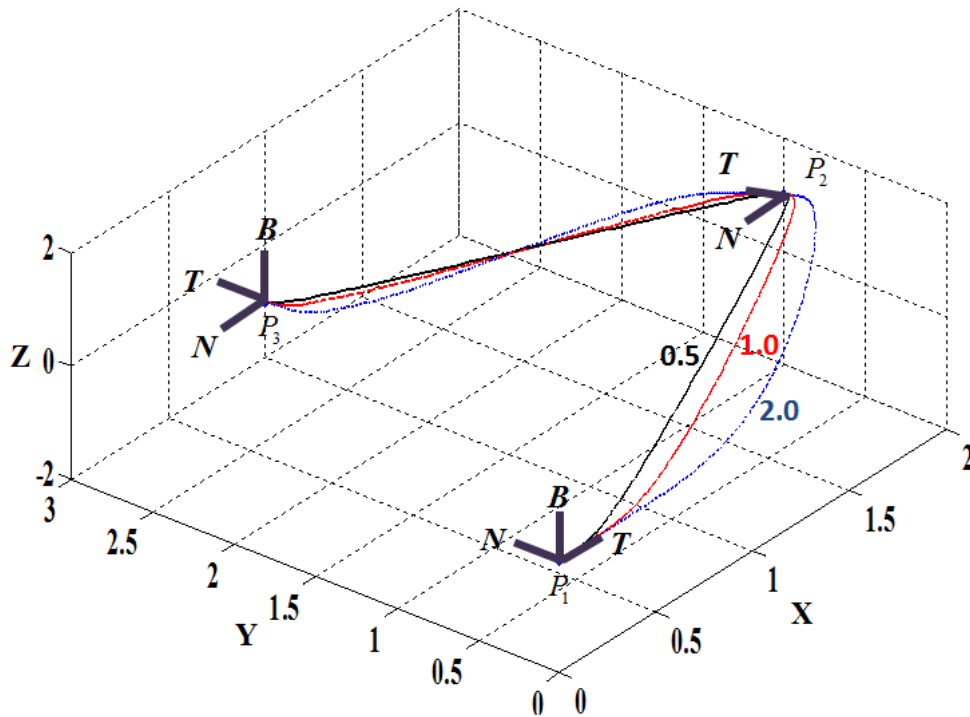


Figure 3-11: Spatial example of varying tangent scale [Tesar et al., 2013]

The second-order geometric constraint of curvature κ is now explored. As mentioned previously, curvature is a measure of the local bending in a spatial curve. Thus, an increase in magnitude of κ should cause a sharper or larger bend in the local geometry of the curve, as shown in the planar example in Figure 3-12.

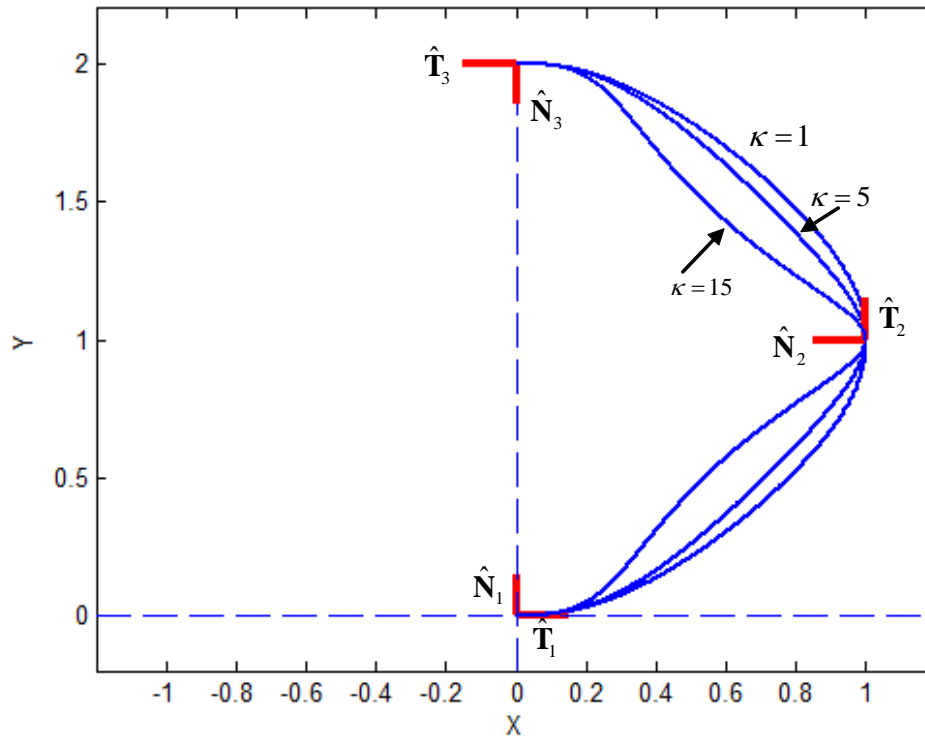


Figure 3-12: Planar example of varying curvature [March, 2008]

This same effect can be visualized in three dimensions. Suppose a spatial trajectory is defined using the three points shown earlier in Table 3-2 with five different values for curvature specified at the middle point: -10, -5, 0, 5, and 10. With a tangent scale of 1, the parametric constraints can be calculated as shown in Table 3-4. Here, it can be seen that curvature has an effect on the second-order parametric derivative.

Table 3-4: Parametric constraints using varying curvature scales [Tesar et al., 2013]

	κ	dp_2/du	d^2p_2/du^2
<i>Path A</i>	-10	[0.0, 0.71, -0.71]	[9.99, 0.0, 0.0]
<i>Path B</i>	-5	[0.0, 0.71, -0.71]	[4.99, 0.0, 0.0]
<i>Path C</i>	0	[0.0, 0.71, -0.71]	[0.0, 0.0, 0.0]
<i>Path D</i>	5	[0.0, 0.71, -0.71]	[-4.99, 0.0, 0.0]
<i>Path E</i>	10	[0.0, 0.71, -0.71]	[-9.99, 0.0, 0.0]

The effects of curvature on an example, three-dimensional robotic motion plan can be visualized in Figure 3-13. This clearly shows that increasing the magnitude of the curvature for the trajectory also increases the bend in the curve. Locally, this bending occurs within the osculating plane and around the unit normal vector. As the curvature increases, the curve appears to approach a spherical-type trajectory.

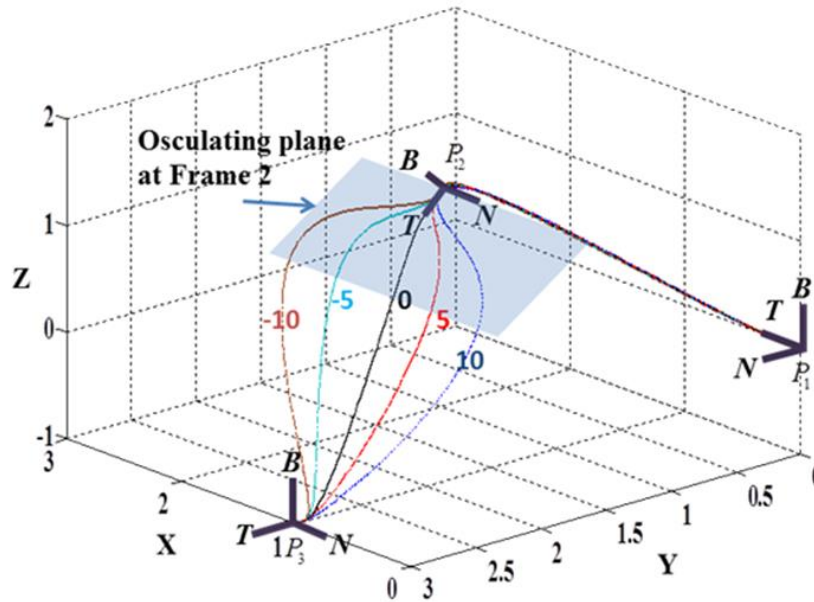


Figure 3-13: Example spatial trajectories with varying curvature [Tesar et al., 2013]

With first and second-order properties defined by the unit tangent vector, unit normal vector, tangential scale, and curvature, the third-order parametric constraints can now be defined. Two third-order properties are investigated in this section: torsion and

the derivative of curvature. Torsion defines the motion out of the plane (i.e. the bi-normal direction) as opposed to curvature which defines motion in the osculating plane. Using the same points defined in Table 3-1, the tangential scale and curvature are both set to 1 to define the first and second-order properties of the example trajectories. Five different values for torsion are specified at the middle point: -30, -10, 0, 10, and 30. The visual effects of changing the magnitude of torsion using the parametric constraints in Table 3-5 are shown in Figure 3-14.

Table 3-5: Parametric constraints using varying torsion scales [Tesar et al., 2013]

	κ, τ	$d^2 p_2 / du^2$	$d^3 p_2 / du^3$
<i>Path A</i>	1, -30	[-0.99, 0.0, 0.0]	[0.0, -21.20, -21.20]
<i>Path B</i>	1, -10	[-0.99, 0.0, 0.0]	[0.0, -7.07, -7.07]
<i>Path C</i>	1, 0	[-0.99, 0.0, 0.0]	[0.0, 0.0, 0.0]
<i>Path D</i>	1, 10	[-0.99, 0.0, 0.0]	[0.0, 7.07, 7.07]
<i>Path E</i>	1, 30	[-0.99, 0.0, 0.0]	[0.0, 21.20, 21.20]

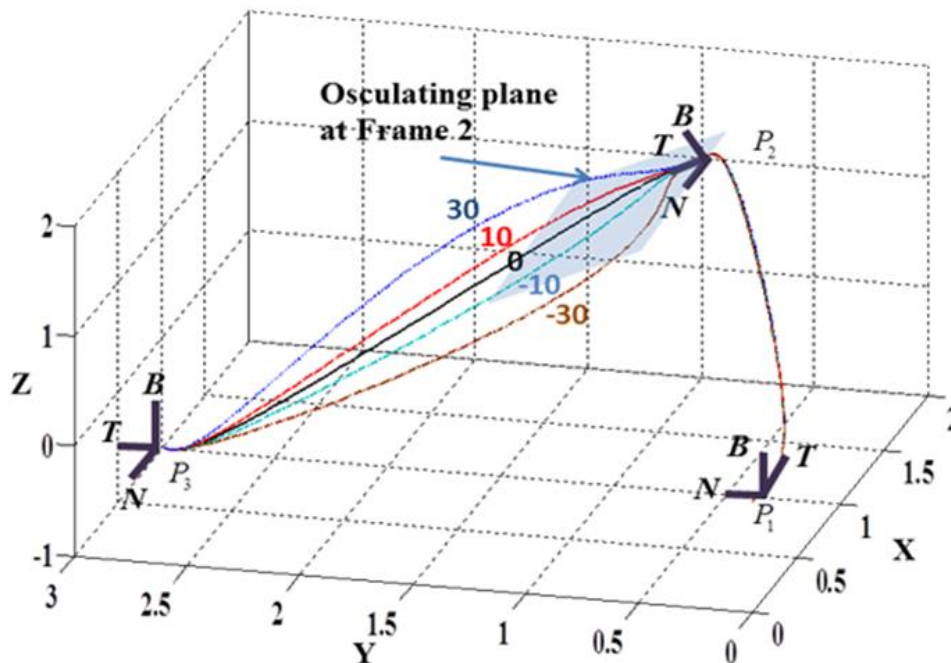


Figure 3-14: Example spatial trajectories with varying torsion [Tesar et al., 2013]

The positive values of torsion cause the curve to bend in the direction of the bi-normal vector in Figure 3-14 above. Conversely, the negative values for torsion cause the trajectory to bend in the opposite direction of the bi-normal vector. Increasing the value of torsion also increases the curvature of the resulting trajectory.

Another property that depends on the third-order parametric constraints is the derivative of curvature. The derivative of curvature represents the rate of change of the curvature parameter at a particular point. Thus, a large value signifies that the curvature is rapidly changing at the given point. The tangent scale at P_2 is set to 1 and curvature is set to 10 to specify the first and second-order properties. Five different values for the derivative of curvature are defined at the middle point: -60, -30, 0, 30, and 60. Table 3-6 shows the parametric constraints for this example.

Table 3-6: Parametric constraints using varying derivatives of curvature
[Tesar et al., 2013]

	$\kappa, d\kappa/du$	d^2p_2/du^2	d^3p_2/du^3
<i>Path A</i>	10, -60	[-9.99, 0.0, 0.0]	[59.98, 0.0, 0.0]
<i>Path B</i>	10, -30	[-9.99, 0.0, 0.0]	[29.99, 0.0, 0.0]
<i>Path C</i>	10, 0	[-9.99, 0.0, 0.0]	[0.0, 0.0, 0.0]
<i>Path D</i>	10, 30	[-9.99, 0.0, 0.0]	[-29.99, 0.0, 0.0]
<i>Path E</i>	10, 60	[-9.99, 0.0, 0.0]	[-59.98, 0.0, 0.0]

If a large positive value is provided, the magnitude of curvature changes quickly at the midpoint P_2 . Thus, it causes a large bend between the midpoint P_2 and the last point P_3 . If a large negative value is specified, the large bend is suppressed. Therefore, this parameter can be used to control the rate of change in curvature. It may also be useful to change the shape of the trajectory between two points while keeping the curvature constant at a particular point. The effects of changing the derivative of curvature can be visualized in Figure 3-15.

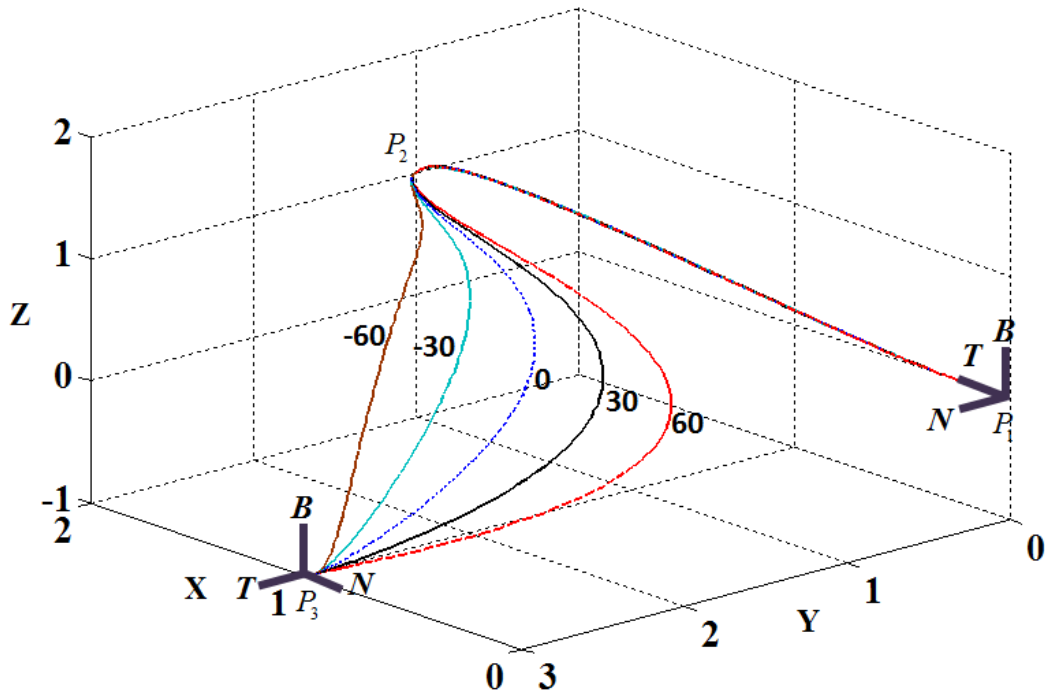


Figure 3-15: Example trajectories with varying rates of curvature [Tesar et al., 2013]

Finally, rate of change in torsion $d\tau/du$, a fourth-order property, can be examined. Equation 13 represents the derivative of torsion as a function of the first, second, third, and fourth-order parameters. However, the first through third-order parameters have already been defined. Thus, the fourth-order derivative must be treated. The derivative of torsion also represents how fast the torsion value is changing at a particular point. The tangential value and curvature is set to 1 and torsion is set to 30 for defining the first through third-order parameters. Again, five different values for the derivative of torsion are defined at the middle point in Table 3-7 below: -300, -100, 0, 100, and 300.

Table 3-7: Parametric constraints using varying rates of torsion [Tesar et al., 2013]

	$\kappa, d\tau/du$	d^3p_2/du^3	d^4p_2/du^4
<i>Path A</i>	30, -300	[0.0, 21.20, 21.20]	[0.0, -212.04, -212.04]
<i>Path B</i>	30, -100	[0.0, 21.20, 21.20]	[0.0, -70.68, -70.68]
<i>Path C</i>	30, 0	[0.0, 21.20, 21.20]	[0.0, 0.0, 0.0]
<i>Path D</i>	30, 100	[0.0, 21.20, 21.20]	[0.0, 70.68, 70.68]
<i>Path E</i>	30, 300	[0.0, 21.20, 21.20]	[0.0, 212.04, 212.04]

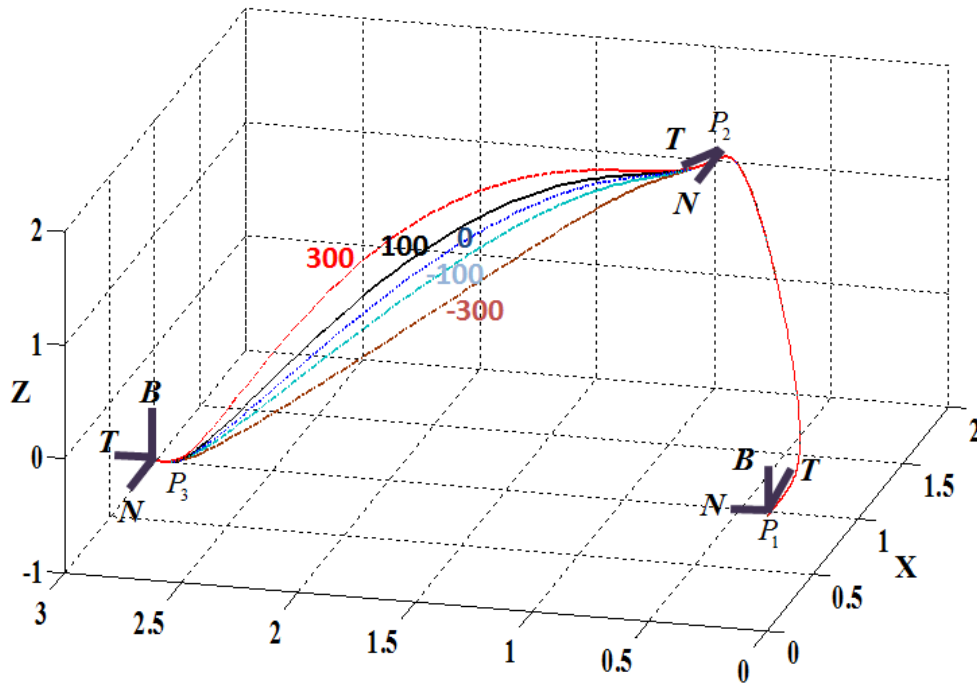


Figure 3-16: Example trajectories with varying rates of torsion [Tesar et al., 2013]

Figure 3-16 shows the effects of the changing the derivative of torsion. Similarly, a good way to understand the physical meaning of this parameter is to relate it to curvature. If a large positive value is provided, the magnitude of torsion changes quickly at the midpoint P_2 . Thus, it causes a large bend in the bi-normal direction between the midpoint and the end point of the trajectory. In the case of a large negative value, the bend in the curve is suppressed. Therefore, this parameter can be used to control the rate

of change in torsion. It may also be used to change a curve's shape in the bi-normal direction between two points while keeping torsion constant at a particular point.

Summary of path planning process

The following steps [Tesar et al., 2013] to create a trajectory using geometric path planning are presented in this section:

1. A set of spatial coordinate frames (positions and orientations) is defined. These come from a CAD environment in most real cases.
2. The desired scaled geometric properties at each frame are defined such as tangent, curvature, torsion, and derivatives of curvature and torsion.
3. The first-order parametric constraint (dp/du) is defined using the tangent vector and a desired tangential scale.
4. The second-order parametric constraint (d^2u/du^2) is defined using the unit normal vector and a desired curvature.
5. The third-order parametric constraint (d^3u/du^3) is defined using the desired torsion and derivative of curvature.
6. The fourth-order parametric constraint (d^4u/du^4) is defined using the derivative of torsion.
7. Once all parametric constraints have been defined, the x, y, and z trajectories are calculated independently using the curve generation techniques in Section 4.
8. The individual trajectories are combined to produce a parametric description of the desired spatial curve.

Evaluation of Generated Path

Now that a framework for path generation with associated physical meaning is established, a method for evaluating this technology over a robot's workspace is presented. All robotic arms, both in industrial and research settings, have physical limitations based on the size, configuration, mechanical bandwidth, etc. of the design. As shown previously, changing geometric parameters that control the tangent, curvature, or torsion of a path quickly change many characteristics of the trajectory. Therefore, robot programmers and operators must have access to information that quickly and clearly illustrates the impact of the choices for the geometric path parameters. This section will use one of the most basic parametric characteristics of a curve, arc length, to evaluate geometric path planning over an example workspace for a robot, similar to the one shown below in Figure 3-17.

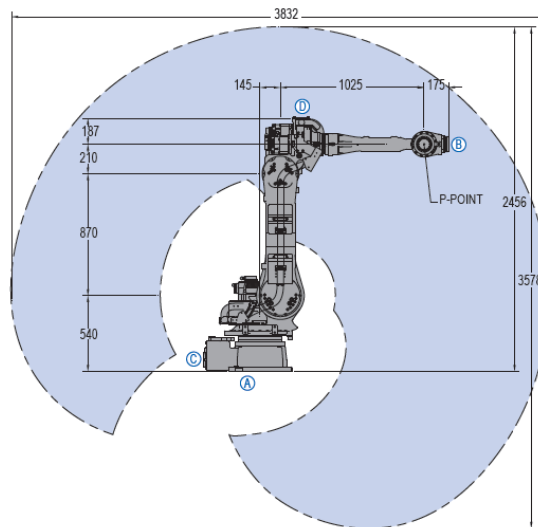


Figure 3-17: Typical workspace for an industrial manipulator [MH80, 2013]

The schematic in Figure 3-17 represents a typical workspace for a six axis industrial manipulator. All points in the shaded area can be achieved by the end effector

in at least one orientation. To adequately evaluate the impact of parameter choices on geometric-based spatial path planning, a robot's workspace must be defined. A 2 x 2 meter discretized grid of target locations, between which trajectories governed by geometric parameters can be generated. To begin the analysis, arc lengths of curves constrained by a constant first order tangent scale property are computed. The resulting paths governed by $\hat{T} = 10$ are shown in Figure 3-18. These paths are calculated using tangent values with positive or negative values corresponding to each target's position with respect to the robot to maintain symmetry over the workspace for path length efficiency purposes.

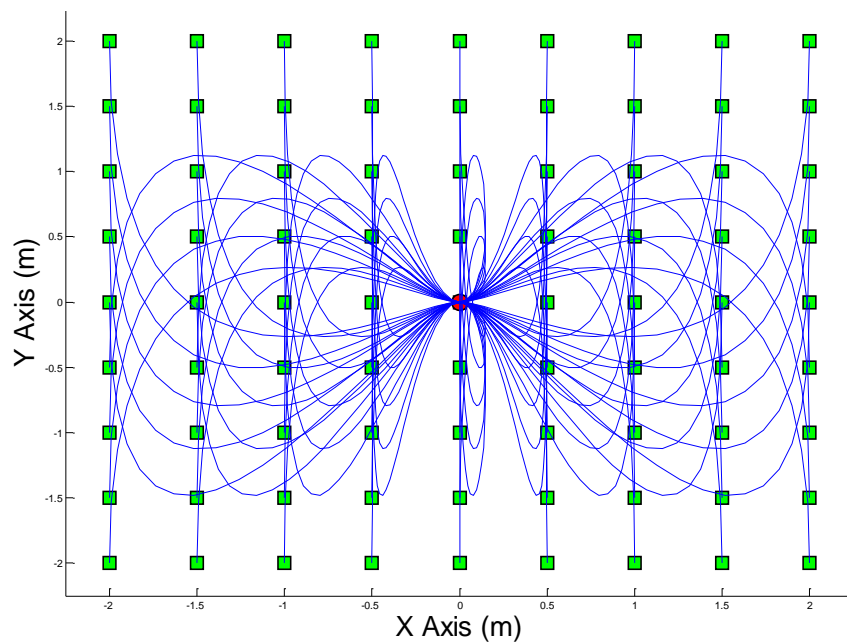


Figure 3-18: Path trajectories constrained by $\tau = 10$

Evaluation of arc length can be extremely important to many applications. First, if the length of the trajectory exceeds the workspace of the robot, that path cannot be achieved and should be ignored. In addition, if time is critical to a given operation, a

threshold for arc length must be set to accomplish the task successfully. To evaluate arc length over the workspace, a discrete number of trajectories between the robot's base and sample target positions must be calculated. A "performance map" can then be generated to give the operator invaluable insight into the effects of geometric path planning. Figure 3-19 represents the map for arc length of paths constrained by the tangent scale $\hat{T} = 10$ over a planar X-Y representation of the robot's workspace.

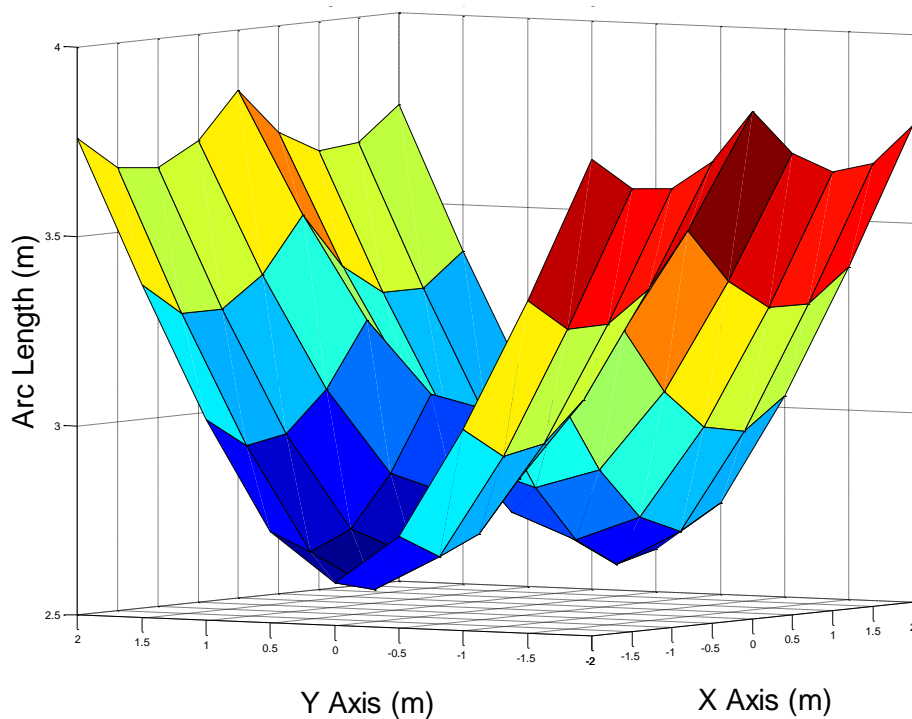


Figure 3-19: Arc lengths of paths defined by $\tau = 10$ over robot's workspace

As expected, Figure 3-19 shows that arc length increases as the target positions approach the boundaries of the workspace. Maps such as these can be used to determine thresholds for required path length globally or locally for a particular manipulator. For the same specifications, Figure 3-20 below represents the percent difference between arc

lengths of a straight line between the robot's base and each target position and the arc length of the path controlled by the first order tangent constraint.

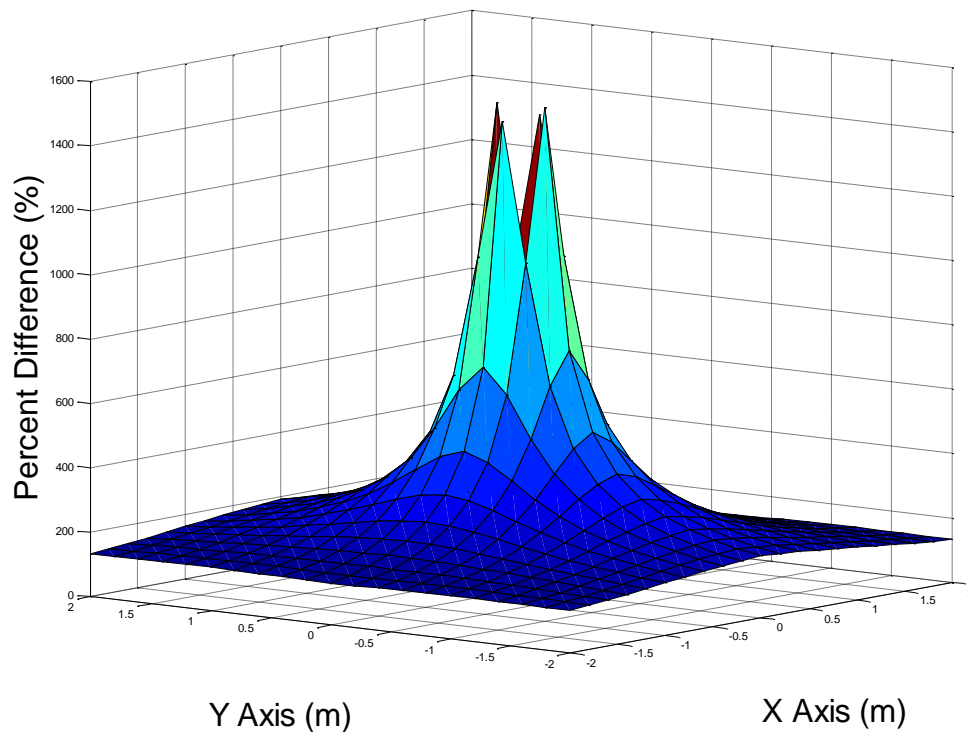


Figure 3-20: Percent difference b/t arc length of straight paths & paths defined by $\tau = 10$

This figure visually informs the user that the cost associated with using first order geometric constraints with respect to path length is exponentially higher for points closer to the base of the robot than those near the boundaries of the workspace. Depending on the location of obstacles in the workspace or required waypoints for a task, first order geometric path planning may still be beneficial, but the efficiency of all paths in the workspace is clear in Figure 3-20. Overall, this type of path planning could be very useful for commanding robots to quickly and efficiently perform geometrically complex maneuvers in a rail yard environment.

CHAPTER 4: REVIEW OF ROBOTIC & SENSOR HARDWARE

The previous chapter provided a summary of available image processing and object detection techniques than an intelligent robotic system can use to find and track moving objects. An overview of path planning techniques that a robot can use to grasp and manipulate an object of interest was also provided. This chapter focuses on readily available robotic hardware that meet the physical requirements of the aforementioned yard operations and the sensor technology required to make the system autonomous. The specifications and principles of operation for each device are covered in sufficient detail to emphasize their respective strengths and weaknesses. All hardware discussed in this chapter is separated into the following categories: vision systems (2D & 3D), robotic platforms & manipulators, and end-effector tooling/sensing.

4.1 VISION SYSTEMS

Vision systems are frequently used to augment industrial automation equipment with digital cameras and image processing techniques to create a “machine vision system.” Vision sensors constantly monitor the surrounding environment to provide a supervisory control system with the information it needs to make decisions. In industry, vision systems are often used for one of four applications [SICK, 2006]:

- 1) **Locate:** find an object and report its position and orientation
- 2) **Measure:** calculate the physical dimensions of an object
- 3) **Inspect:** validate features (e.g. presence/absence of screws in an assembly)
- 4) **Identify:** interpret codes, symbols, or characters to identify object

Using vision for these applications typically improves production yield, cycle times, and/or product quality. The flow of operation for a standard machine vision system

includes capturing an image, processing the image to obtain a desired result, sending the result to the control system, and waiting for the next frame while the system takes action based on the previous result [SICK, 2006]. A diagram for this process is shown below in Figure 4-1.

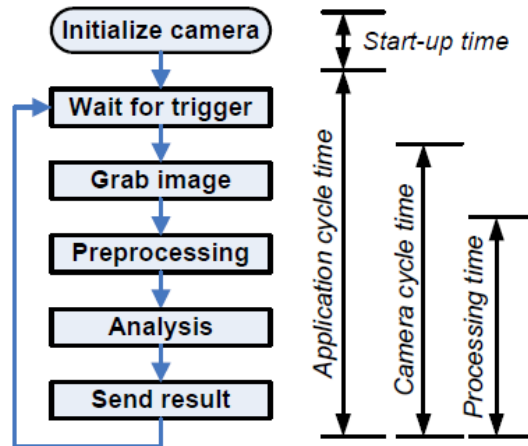


Figure 4-1: Operation flow and cycle times for machine vision systems [SICK, 2006]

Vision systems used to provide input to robotic systems need be fast and reliable. The performance of a vision system with regards to speed is defined by its cycle time. A system's cycle time can be divided into multiple subcategories as shown in Figure 4-1. The start-up time is measured from the power-on state of the system to the point where the camera is ready to begin acquiring frames. Start-up time can vary anywhere from 0.50 to 500 seconds, but is typically not critical to overall performance because vision systems often run hours or days at a time after a single initialization. The application cycle time is measured between two consecutive iterations of the image acquisition and analysis process [SICK, 2006]. This includes the cycle time of the camera, also known as frame rate, and the processing time. The frame rate of the camera refers to the frequency at which it can produce new images while the processing time denotes the elapsed time

between the start of analysis and the instant a result is calculated. This work focuses on vision systems with application cycle times of 10 - 40 milliseconds to control robots in near real-time.

4.1.1 Two-Dimensional Vision Systems

Most 2D vision systems use digital camera technology to monitor the surrounding environment. Digital cameras use a sensor chip with arrays of light sensitive pixels to “grab” a digital image [Spring, 2013]. The two main types of digital image sensors are Charge-Coupled Devices (CCD) and Complementary Metal Oxide Semiconductor (CMOS). Historically, the CCD sensor is known to create higher quality images with less noise than its CMOS counterpart since the inception of both in the 1970s [Teledyne, 2013]. Conversely, CCDs use ~100 times more power than the equivalent CMOS sensor [Teledyne, 2013]. For decades, lithography-related limitations prevented designers from being able to manufacture high performance CMOS sensors. However, with the recent boom in mobile devices, the semiconductor industry has invested an enormous amount of capital in the design and fabrication of compact, low power, and high performance CMOS imagers. In today’s high volume consumer market, CMOS imagers outperform CCDs in nearly every area [Teledyne, 2013].

Vision systems that use 2D digital cameras can be configured to meet virtually any requirement. Common specifications for digital cameras include lens types, focal length, field of view (FOV), aperture, depth of field, resolution, color/monochrome, illumination, and frame rate. Most operations in the rail yard involve physical interaction between the human and the freight cars. Consequently, a robotic system designed for these tasks will be in close proximity to the freight cars, which minimizes the required depth of field and eliminates the need for complex lenses that can zoom and adjust

aperture. Therefore, the most important features of a vision system for automating rail yard applications include the resolution, FOV, illumination, and frame rate.

Illumination will be very important for any rail yard application in which vision systems are used. An adequate amount of light must be reflected off an object's surface for the vision system to detect it (Figure 4-2, top right). Most yards operate 24/7 in outdoor environments with little to no shelter, meaning a standard 2D vision system will need an adaptive illumination source to work in low light settings. The FOV of a camera is also important for rail yard applications. If multiple objects in a scene need to be identified with one camera, they should appear simultaneously in the same frame. The field of view expands as the distance between the camera and the object of interest increases, as shown in Figure 4-2.

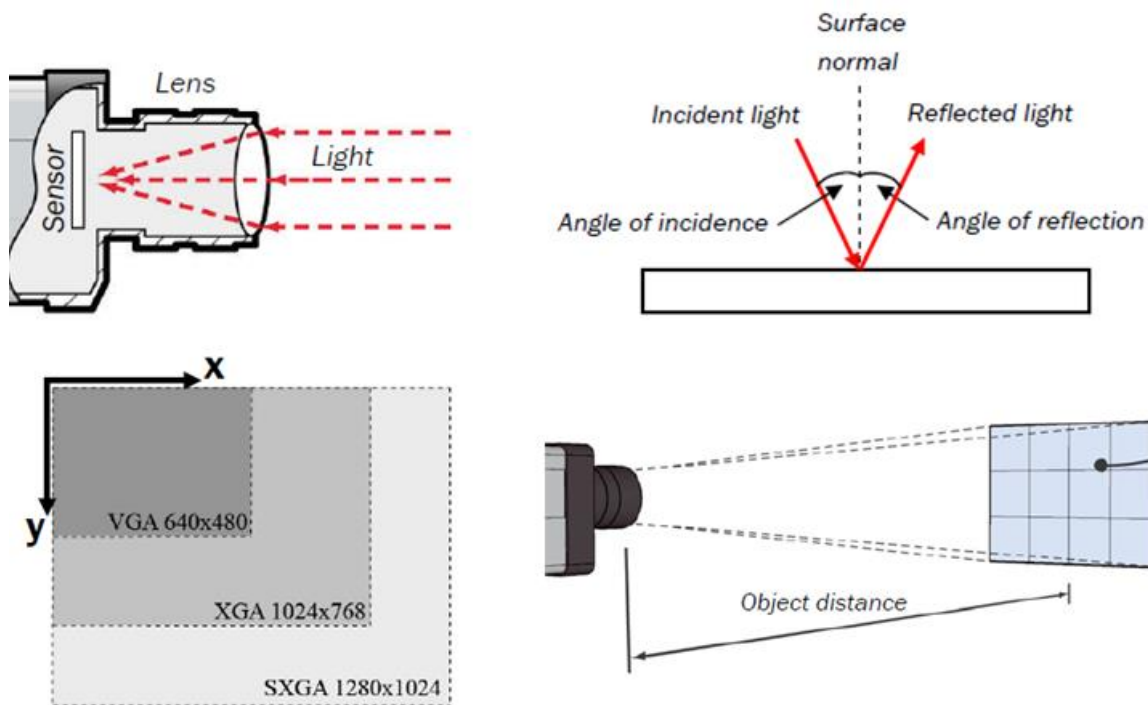


Figure 4-2: a) Illumination, b) reflection, c) resolution, & d) field of view [SICK, 2006]

Selecting the correct output resolution for digital cameras is also critical to the performance of a vision system. Three standard sensor resolutions used are: (1) VGA – 640 x 480 pixels, (2) XGA – 1024 x 768 pixels, and (3) SXGA – 1280 x 1024 pixels (see Figure 4-2c). The resolution of the desired target (object resolution) should first be calculated to determine the optimum camera resolution for a given application using Equation 4-1.

$$\frac{FOV_{width}}{Sensor_Resolution_{width}} = Object_{Resolution} [unit\ length/pixel] \quad (4.1)$$

The calculated object resolution should be compared to the physical size of the object(s) of interest in the approximate field of view to determine if the frame to target ratio is sufficient for processing. Finally, the frame rate of the hardware used in a vision system should be greater than or equal to the object frequency. Object frequency represents the number of objects that pass the camera per second and is calculated in Equation 4.2 below. For detecting objects on a moving train, the designer should ensure that multiple frames contain the target to adequate object detection using the methods outlined in Chapter 3.

$$\frac{1}{Application\ Cycle\ Time} = Object_Frequency [Hz] \quad (4.2)$$

Several 2D cameras are shown below in Figure 4-3 to demonstrate the wide range of form factors, resolutions, and frame rates. The camera on the left is an industrial smart camera manufactured by SICK with a CCD image sensor, 640 x 480 resolution, and a frame rate of 30 frames per second (FPS). The camera also has on-board processing capabilities and an IP-65 rated housing, which make it an ideal candidate for use in a deployed system subjected to the elements at a rail yard.



Figure 4-3: Digital 2D cameras a) SICK smart camera [SICK, 2006], b) Logitech Pro 9000 webcam [Logitech, 2013], & c) PlayStation Eye [Sony, 2013]

The camera shown in the middle of Figure 4-3 is a Logitech Pro 9000 web-cam. This device uses a CMOS sensor to capture frames at 30 FPS with up to 1600 x 1200 resolution. For high speed video capture, the affordable Sony PS Eye camera (Figure 4-3, right) is capable of recording data at 75 FPS with 640 x 480 resolution or 180 FPS with 320 x 240 resolution. Increased frame rates can be used to boost the speed of real-time object detection applications if the computational time required by the image processing algorithm is less than the time required to acquire frames. It should be noted that increased frame rates are typically achieved at the cost of the output resolution.

4.1.2 Three-Dimensional Vision Systems

Three-dimensional (3D) vision systems provide the physical distance from a sensor's reference frame to objects in the scene. Once the distance and direction to a point in the environment is known, the spatial location of objects in a scene with respect to any frame of reference can be calculated. Many different types of commercially available technologies can provide 3D data with varying degrees of accuracy and repeatability. The market for 3D sensors has grown tremendously since the release of Microsoft's Kinect in 2010 due its wide availability and low cost. Developers, engineers, and computer scientists have leveraged the technology to create new ways of interacting with machines, from assisting doctors in operating rooms to controlling robots with 3D

feedback [Zhang, 2012]. This increase in demand has forced manufacturers of similar but historically more expensive technologies to adapt to make their products available to a wider audience. The following section will provide an overview of existing 3D sensing technologies, including laser/line scan systems, laser rangefinders, stereo vision, time-of-flight, and structured light to characterize their performance in a variety of conditions.

3D Laser/Line-Scan Systems

Industrial 3D vision systems often use a combination of 2D cameras and lasers to create a 3D profile of an object. The 3D data is acquired by illuminating an object with a laser in one direction while simultaneously monitoring the scene with a camera. The laser beam is iteratively divided into a “laser line” by the object. The camera is mounted at an angle such that it can capture the object’s cross-sectional view to build a 3D map, as shown on the right side of Figure 4-4 below.

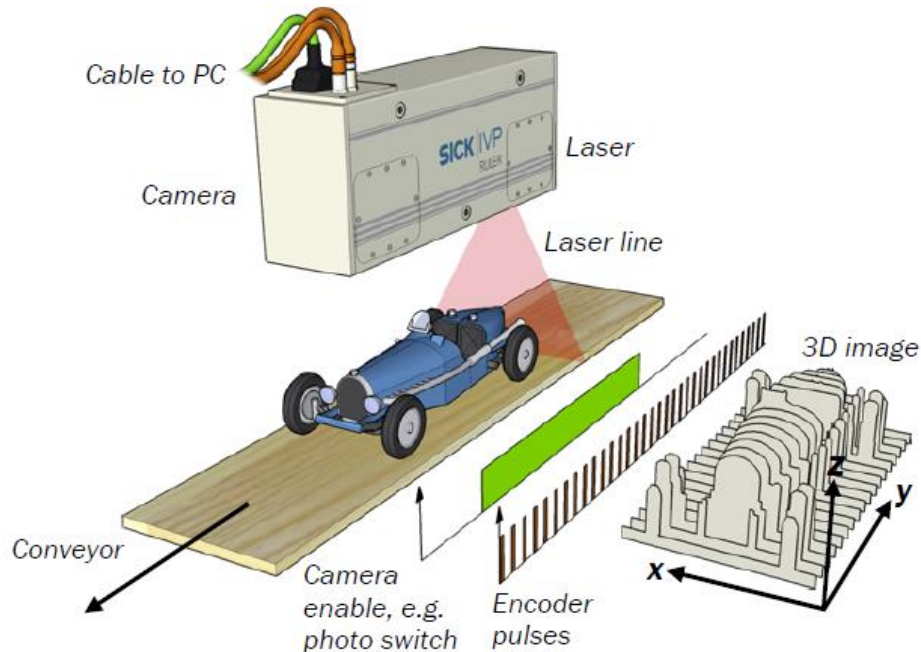


Figure 4-4: 3D SICK VP line scan camera [SICK, 2006]

Laser/line scan systems are often used in tandem with a conveyor system that transports parts down a manufacturing line; building 3D maps or inspecting parts as they pass the static camera. A potential disadvantage of this system for some applications is that the object of interest must be dynamic and actively pass the camera to build the 3D map. Line scan cameras are currently used to take continuous images of passing trains and to visually inspect railroad track using custom vehicles outfitted with a network of cameras [Shah, 2010]. Therefore, this type of system is recommended for rail applications if the object of interest is in constant motion.

Stereo vision

One of the most mature and well-known principles of 3D image generation and acquisition is the concept of stereo vision. Stereo vision is accomplished by taking multiple images of a scene simultaneously from different perspectives to passively measure distances. A schematic of two cameras with intersecting fields of view used for stereo vision is included below on the left side of Figure 4-5.

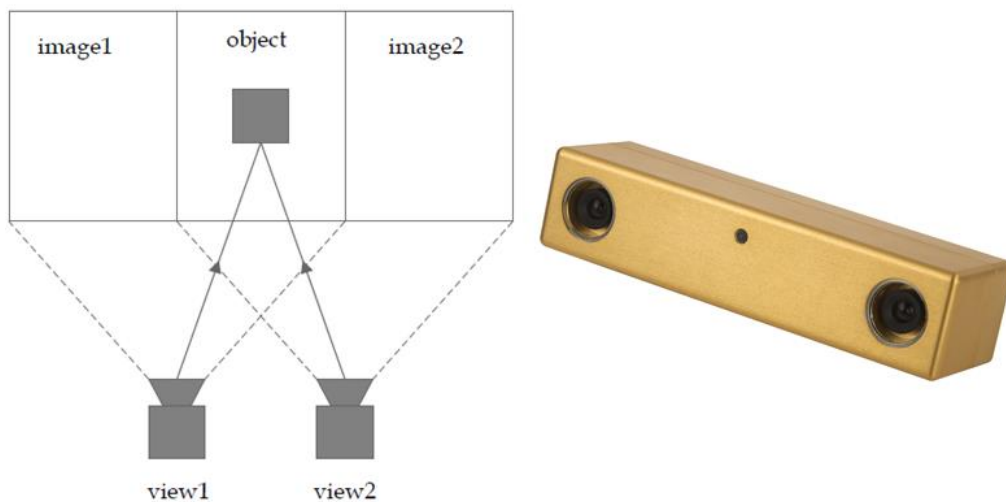


Figure 4-5: Stereo vision principle of operation (left) [PMD, 2008] & Point Grey Bumblebee stereo camera (right) [Point Grey, 2013]

The two cameras in the schematic shown in Figure 4-5 above are separated by a known distance; very similar to a human’s binocular vision with a pair of eyes. This configuration provides “stereopsis,” which is the impression of depth that is perceived when a scene is viewed from two different locations. Stereo vision replicates this phenomenon by mathematically comparing two images shifted on top of one another to find the areas that match. The resulting shift is known as “disparity,” which is the magnitude of the vector between the pixel location in the left and right images that correspond to the same object. A commercially available stereo camera is also shown on the right side of Figure 4-5 above. Most stereo cameras use a pair of CCD cameras separated by a known distance (baseline) along the same horizontal axis with parallel lines of sight. Several parameters affect the resolution and quality of the resulting depth map, including: (1) type of imager used (color/monochrome), (2) baseline, and (3) lens type. Larger baselines and lenses with a narrow field of view allow cameras to calculate depths of objects at greater distances while smaller baselines and lenses with a wide field of view excel at detecting object depth very near the camera. Therefore, the user must select a camera configuration specific to a given application to maximize sensitivity in detectable depth resolution and performance. An overview of a stereo vision system is shown below in Figure 4-6.

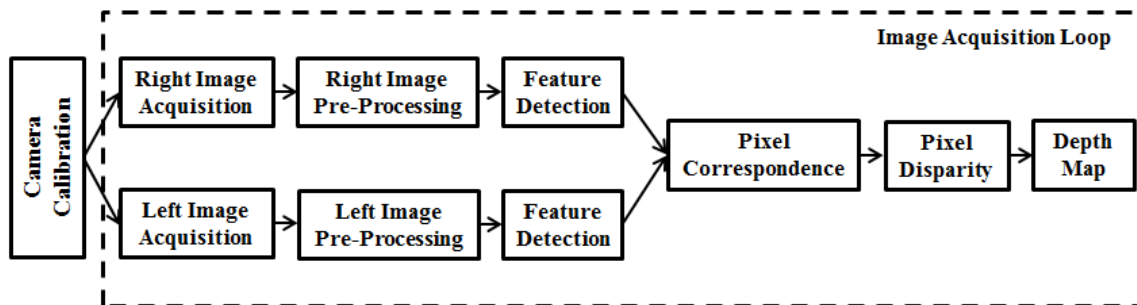


Figure 4-6: Stereo vision system operation

Stereo vision is a computationally expensive operation as evidenced in Figure 4-6. First, images from two separate cameras must be acquired simultaneously and then rectified to prepare them for further processing. Next, one or multiple feature detection processes must be performed to extract critical information from each image. These features are then compared to obtain pixel correspondence information critical to calculating disparity between the two images to produce a dense depth map. These steps must be performed iteratively to get 3D information from the system. Although the process above is taxing, stereo does have the advantage of being able to passively acquire 3D data without any moving parts or active emissions.

Time-of-Flight

Infrared time-of-flight (ToF) range cameras calculate distances to points in space by measuring the amount of time required for a pulse of light emitted by the camera to reach and reflect off objects in a scene. This is either accomplished by a direct measurement of the runtime of a travelled light pulse using cutting-edge hardware or by measuring the phase shift between a reference signal and the reflected signal. A visual of the phase shift method is shown below in Figure 4-7 [Piatti, 2012].

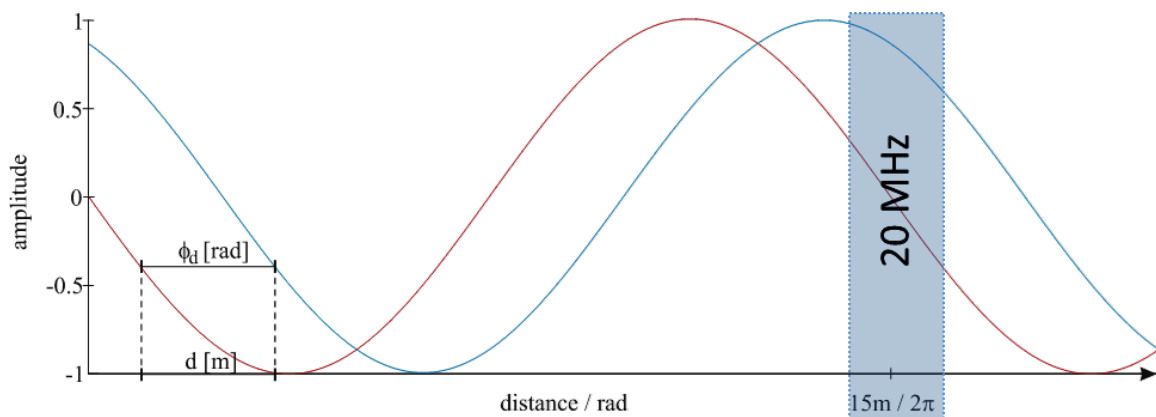


Figure 4-7: Time-of-flight principle of operation [Metrilus, 2013]

Cameras using the phase shift method will be the primary focus of this chapter. These cameras typically consist of three different components: (1) an active illumination unit, (2) a lens, and (3) an imaging sensor [Metrilus, 2013]. The active illumination unit emits intensity modulated light in the near-infrared range (non-visible) of the spectrum. The light travels with a constant speed in the surrounding medium and is reflected by the object's surface [Metrilus, 2013]. Distance (d) is then calculated using Equation 4.3 below:

$$d = \frac{c}{f_{mod}} * \frac{1}{2} * \frac{\phi_d}{2\pi} \quad (4.3)$$

where d is the distance travelled by the light (m), c the speed of light (m/s), f_{mod} the modulation frequency (MHz), and ϕ_d the phase shift (radians). In addition to distances, ToF cameras also provide intensity values that represent the amount of light sent back from a specific point [Metrilus, 2013]. It should be noted that ToF cameras have a non-ambiguous range of $d_n = c/2 * f_{mod}$ due to the periodicity of the cosine-shaped modulation signal. Currently available ToF cameras operate at an average modulation frequency of approximately 20 MHz, which would produce a range of non-ambiguous depth values up to 7.5m.

ToF technologies have several advantages over other 3D imaging technologies, including the direct, computationally inexpensive calculation of depth values for each pixel, small footprint, onboard illumination, increased frame rates, and no moving parts. These advantages make complex tasks, such as visual servoing with a robotic manipulator, entirely feasible and affordable. Some ToF cameras can be used in outdoor environments, which is a critical aspect of many rail yard applications. Three commercially available ToF sensors are shown in Figure 4-8 below.



Figure 4-8: Time-of-Flight cameras a) Mesa Imaging SwissRanger 4000 [Mesa, 2013] b) PMD CamBoard Nano [PMD, 2013] & c) Creative Sens3D [Creative, 2013]

MESA Imaging's SwissRanger 4000 is a popular industrial ToF camera in Figure 4-8a has a sensing range of 0 to 5m, depth resolution of 176 x 144, and a maximum frame rate of 50 fps. The absolute accuracy of the SR4000 is $\pm 10\text{mm}$ (0.4"). PMD Technologies provides the CamBoard Nano reference design (Figure 4-8b) for a ToF camera prototype that is currently best suited for gesture recognition. It has a sensing range of 0 to 1m, depth resolution of 160 x 120, and can provide frame rates as high as 90 fps. This sensor also uses PMD's Suppression of Background Illumination (SBI) algorithm that is suitable for both indoor and outdoor operation [Piatti, 2012]. The CamBoard Nano's depth accuracy was found to be approximately $\pm 20\text{mm}$ (0.8"). Finally, Creative Labs has leveraged SoftKinetic's technology to design a consumer ToF product named Sens3D (Figure 4.8c). This sensor has a range from 0.5 to 3.25ft, depth resolution of 320 x 240, and a frame rate of 30 fps.

Laser Rangefinder

A laser rangefinder is another type of 3D sensor that uses an infrared laser beam to measure time-of-flight instead of emitted light pulses [SICK, 2012]. This pulsed laser beam is generated by a transmitter that directs the beam to a rotating mirror. The beam is then redirected off the moving mirror, which sends it out into the environment where it is

reflected off objects in the scene. A receiver inside the sensor unit constantly receives the reflected beams to calculate distances using the time of flight principle in Equation 4-3. This produces a planar map of the surrounding environment that includes distances to objects in the scene that are at the same height as the rotating mirror. The SICK LMS series of laser range finders are the industry standard that can provide measurements with high accuracy (± 25 to 35mm) and repeatability [SICK, 2012]. One of SICK's laser rangefinder models is shown below in Figure 4-9.



Figure 4-9: SICK LMS 500 Pro laser rangefinder [SICK, 2012]

The sensor in Figure 4-9 above can provide distance measurements over a 180 degree area with various degrees of resolution (0.167° to 1°) at 100Hz [SICK, 2012]. This sensor is capable of measuring distances to objects up to 80 meters away. Indoor and outdoor versions of this sensor exist for use in either environment. Outdoor sensors have an IP67 enclosure rating, built-in heater for colder weather, and multi-echo technology for adverse environmental conditions. Multi-echo technology allows the sensor to receive multiple pulses that can be inadvertently produced by rain drops, fog, glass, etc. [SICK, 2012]. These conditions can reflect part of the energy and create smaller echoes but most

of the sending pulse energy will eventually reflect against the object of interest [SICK, 2012]. The sensor can be programmed to receive the first, last, or all of the reflected echoes depending on the application. A visualization of the multi-echo technology is included in Figure 4-10 below.

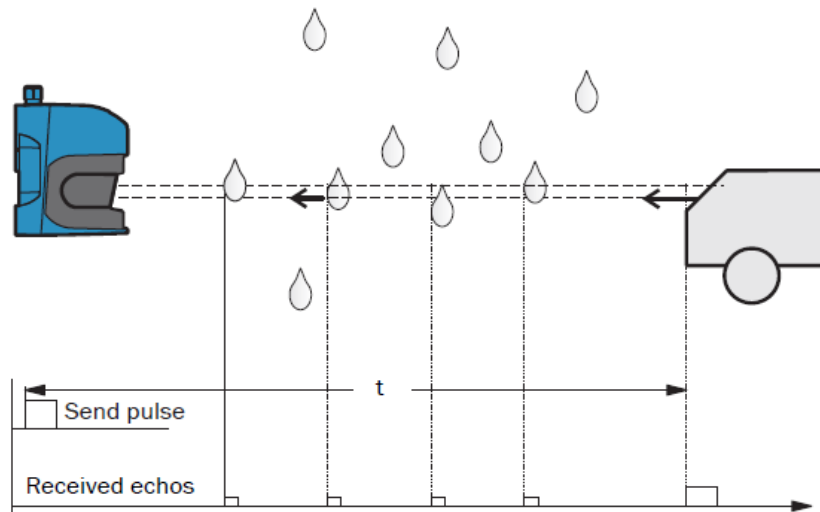


Figure 4-10: Laser rangefinder principle of operation [SICK, 2012]

One potential disadvantage of this sensor is the fact that it only takes a planar scan of the surrounding environment. If a dense 3D map of an entire scene is required (regardless of object height), the laser rangefinder can be mounted to a tilt axis and synchronize the reflected echoes with the encoder information corresponding to the amount of tilt at discrete instances in time to produce a spherical representation of points [SICK, 2012]. This has proven to be successful, but significantly reduces the rate at which data can be gathered. Another option to produce 3D maps is to use a sensor with multiple lasers. Velodyne Lidar offers sensors with up to 64 fixed-mounted lasers to measure the surrounding environment with a 360° field of view at up to 15Hz [Velodyne, 2012]. These lasers allow the Velodyne to produce 3D maps at rates of 1.3 million points per second and is one of the most important components of today's autonomous vehicles.

Structured Light

The last type of 3D technology covered in this section is structured light. In structured light imaging, a predefined pattern of light is projected onto objects in a scene and simultaneously observed by a camera [Metrilus, 2013]. The predefined light pattern can be gray codes, light stripes, sine waves, or speckle patterns [Metrilus, 2013]. The appearance of the projected light pattern on the scene will be distorted if three-dimensional objects are present. Distortion in local patches of the projected infrared image is compared with reference images of the predefined pattern projected onto a flat surface at known distances to compute the resulting depth maps [Metrilus, 2013]. A schematic for a structured light system with an infrared projector and camera is shown in Figure 4-11 below.

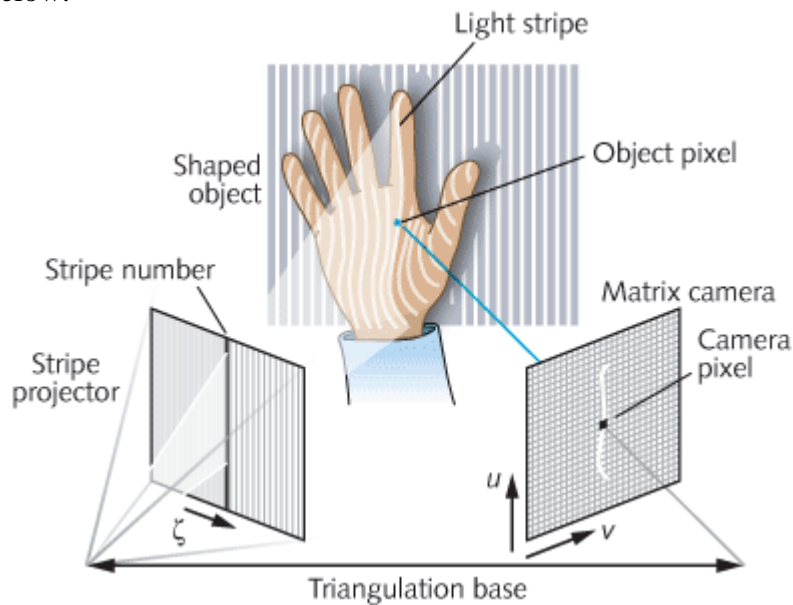


Figure 4-11: Structured light camera principle of operation [Hecht, 2011]

Traditionally, structured light projects rectangular grids or arrays of lines, but powerful lasers are needed to provide a high signal to noise ratio and the projection unit has to be mounted very accurately, which leads to increased costs [Hecht, 2011].

However, an Israeli company named PrimeSense has recently developed a proprietary technique called “light coding” in the optics it supplies to 3D camera manufacturers, including Microsoft and ASUS. The code they use is a unique speckle pattern that is “very rich in information, with almost zero repetition across the scene. It’s a code, not a grid, and this is what gives us reliability and replicability” [Hecht, 2011]. An example IR image that displays PrimeSense’s speckle pattern is shown below in Figure 4-12a.



Figure 4-12: (Left to Right) a) IR image of scene, b) Grayscale distance image, & c) Color coded point cloud [Metrilus, 2013]

The projected light pattern in Figure 4-12a is used to create grayscale distance images (Figure 4-12b) that can be converted into a three-dimensional color coded point cloud (Figure 4-12c) that provides instant feedback on the relative positions of objects in the scene. Microsoft's Kinect is a widely used, low-cost 3D camera that uses PrimeSense’s structured light technology. The Kinect’s depth sensor has a resolution of 320 x 240 and a typical range of 0.8m to 3m. The sensitivity in depth decreases as the distance from the camera increases (approx. resolution at 1m is 3mm). The Kinect provides an RGB color stream in addition to a raw depth stream at 30 FPS at 640 x 480 image resolution. The current cost for the Kinect sensor is \$150, much cheaper than all of the aforementioned competing technologies. ASUS also produces and sells a 3D camera named Xtion Pro Live that uses the same image sensor as the Kinect. Functionally, both

cameras are the same, but the Xtion Pro has a much smaller footprint, is lighter in weight, and can be purchased without a RGB stream for \$100. Additionally, the camera is powered solely by a standard 5V USB connection (compared to the Kinect's 12V supply), meaning custom cables from the manufacturer are not required for image acquisition. Both of these sensors are shown below in Figure 4-13.

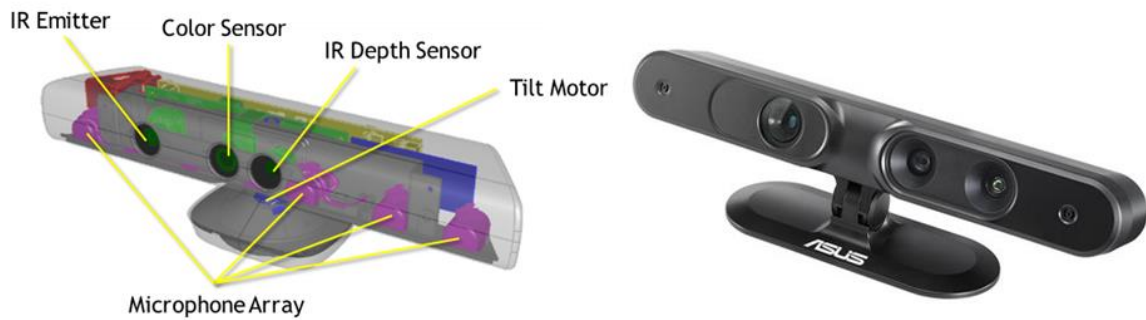


Figure 4-13: Microsoft Kinect schematic [Beyer, 2013] (left) & ASUS Xtion Pro Live (right) structured light sensors

Vision Summary

This section provided an overview commercially available two- and three-dimensional vision systems that can be used with robotic systems designed for rail yard applications. Two-dimensional systems can be used to monitor processes and gather information pertinent to potential object detection algorithms, such as color and changes in intensity. Three-dimensional vision systems can be used in tandem with these systems to provide the robotic system with the spatial coordinates needed to interact and manipulate objects of interest. Each of the aforementioned systems has its advantages and disadvantages with respect to accuracy, speed, sensing range, low light performance, outdoor performance, complexity and cost are listed below in Table 4-1.

Table 4-1: Summary of available 3D sensing methods [Branding, 2013]

	Stereoscopic Vision	Structured Light		Time of Flight
		Fixed Pattern	Programmable Pattern	
Depth Accuracy	mm to cm <i>Difficulty with smooth surface</i>	mm to cm	μm to mm <i>Variable patterns & different light sources improves accuracy</i>	mm to cm <i>Depends on resolution of sensor</i>
Scanning Speed	Medium <i>Limited by software complexity</i>	Fast <i>Limited by camera speed</i>	Fast/Medium <i>Limited by camera speed</i>	Fast <i>Limited by sensor speed</i>
Distance Range	Mid range	Very short to mid range <i>Depends on illumination power</i>	Very short to mid range <i>Depends on illumination power</i>	Short to long range <i>Depends on laser power & modulation</i>
Low Light Performance	Weak	Good	Good	Good
Outdoor Performance	Good	Weak/Fair <i>Depends on illumination power</i>	Weak/Fair <i>Depends on illumination power</i>	Fair <i>Depends on illumination power</i>
Software Complexity	High	Low/Middle	Middle/High	Low
Material Cost	Low	Middle	Middle/High	Middle

As evidenced by the information in Table 4-1, there is not a single solution that meets all possible criteria. For outdoor rail yard applications that occur exclusively in an adequately illuminated environment, a stereo vision system would be recommended due to its accuracy, range, and performance outdoors. However, if a task must be performed at night with no illumination, a ToF or structured light system would be recommended. Therefore, the information in this table should be used to select the right vision system for a given application after a careful and thorough review of an application's requirements. The information and hardware presented in this chapter will be discussed with respect to the design an applied system in the Chapter 5.

4.2 ROBOTIC MANIPULATORS & MOBILE PLATFORMS

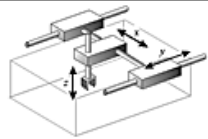
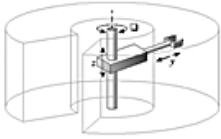
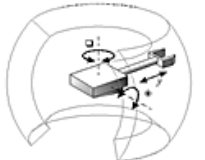
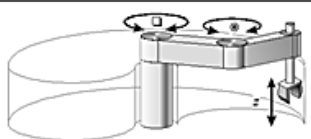
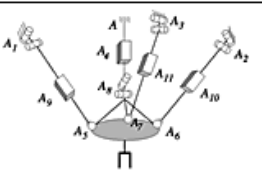
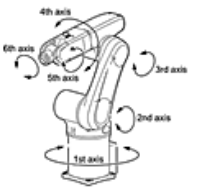
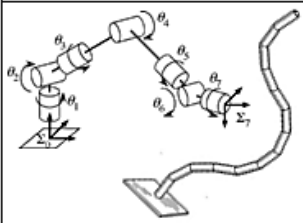
4.2.1 Robotic Manipulators

Industrial robotic manipulators are complex electromechanical assemblies that are capable of being programmed to autonomously perform a specific task. The base of a robotic arm can be rigidly mounted to the ground, ceiling, or wall in a work cell or to a mobile platform that is capable of travelling from one location to another. An overview of existing mobile robotic platforms is provided in the following section if one is needed for a particular application.

Robotic manipulators are classified according to the combination of joints used in their construction [Petriu, 2012]. Two basic types of joints are used to build robots: prismatic and revolute. Prismatic and revolute joints represent the opposite extremes of a universal screw: in a revolute joint, the screw pitch is zero, constraining the joint to pure rotation, whereas in a prismatic joint, the pitch is infinite, constraining the joint to pure sliding motion [Petriu, 2012]. Designers often prefer to use revolute joints over prismatic due to their strength, low friction, and increased reliability. Joints that allow a combination of translation and rotation (e.g. lead screws) are not normally used to connect links of robot arms [Petriu, 2012].

The types of robots covered briefly in this section include: Cartesian (also known as Gantry robots), Cylindrical, Spherical (Polar), Selective Compliance Assembly Robot Arm (SCARA), Parallel, Articulated, and Redundant robots. Each of these categories has its own set of advantages and disadvantages with regards to mobility, speed, and accuracy due to the number and configuration its degrees of freedom. This information, in addition to the standard uses for each, is included in the comparison of these manipulator types provided in Table 4-2 below.

Table 4-2: Geometric configurations for robotic arms (adapted from [Petriu, 2012])

Schematic	Type of Arm	DOF	Advantages	Disadvantages	Uses
	Cartesian (Gantry)	3 DOF 3 Prismatic Order: PPP	<ul style="list-style-type: none"> Simple kinematics Only linear motion Rigidity Inexpensive 	<ul style="list-style-type: none"> Workspace is smaller than robot volume Unable to reach areas under objects Guide surfaces of 	<ul style="list-style-type: none"> Pick & Place Assembly Sealant Application Arc Welding CNC Milling
	Cylindrical	3 DOF 2 Prismatic 1 Revolute Order: RPP	<ul style="list-style-type: none"> Simple kinematics Powerful w/ hydraulics Good access into cavities and openings 	<ul style="list-style-type: none"> Restricted workspace Guide surfaces of linear axes exposed Back of robot can overlap work volume 	<ul style="list-style-type: none"> Spot Welding Assembly Handling Tools
	Spherical (Polar)	3 DOF 1 Prismatic 2 Revolute Order: RRP	<ul style="list-style-type: none"> Covers large volume from central support Can bend down to pick up objects off floor 	<ul style="list-style-type: none"> Difficult to visualize Complex kinematics 	<ul style="list-style-type: none"> Diecasting Assembly Gas/Arc Welding Spot Welding Handling Tools
	SCARA (Selective Compliance Assembly Robot Arm)	3 DOF 1 Prismatic 2 Revolute Order: RRP	<ul style="list-style-type: none"> Rigid in z direction, compliant in x&y (good for inserting pins) Price to performance ratio Speed & footprint 	<ul style="list-style-type: none"> Difficult to visualize Complex kinematics Headroom intensive 	<ul style="list-style-type: none"> Pick & Place Assembly Sealant Application Handling Tools
	Parallel (Delta, Stewart Hexapod, etc.)	3-6 DOF Stewart: 6 Prismatic Delta: 3-6 Prismatic and/or Revolute	<ul style="list-style-type: none"> Strength Stiffness High Speed Repeatability 	<ul style="list-style-type: none"> Large footprint Small range of motion 	<ul style="list-style-type: none"> Stewart: Flight Simulators Automotive Testing Delta: Pick & Place Assembly
	Articulated	3-6 DOF 3-6 Revolute Order: RRRRRR	<ul style="list-style-type: none"> Maximum flexibility Sealed revolute joints Covers a large workspace relative to robot volume Satisfies most application requirements 	<ul style="list-style-type: none"> Complex kinematics Difficult to visualize Rigidity at full reach Control of linear motion is difficult 	<ul style="list-style-type: none"> Pick & Place Assembly Deburring Palletizing Bin Picking Spray Painting Cleaning Welding
	Redundant	7+ DOF 7+ Revolute & Prismatic	<ul style="list-style-type: none"> Collision avoidance Fault tolerance Speed & footprint Ability to minimize energy consumption 	<ul style="list-style-type: none"> Non-existence of an analytic solution to inverse kinematics Highly complex Robustness & accuracy of control More moving parts Higher costs 	<ul style="list-style-type: none"> Surgical Robots Pipe Inspection Robot Navigation in Complex Areas

An overview of commercially available robotic manipulators is not included in the scope of this work. Using the information above, a designer should reference the leading manufacturers' documentation (e.g. Yaskawa Motoman, Kuka, Fanuc, ABB, Denso, Adept etc.) to find the latest models that meet an application's requirements.

4.2.2 Mobile Robotic Platforms

Mobile platforms provide the ability to remotely or autonomously move a robotic arm from one location to another. A mobile platform should be considered as an open architecture system that can utilize a wide variety of steered/powered wheels, passive/active suspensions, and/or tracks for locomotion [Kulkarni, 2009]. Most rail yard applications require a mobile platform because tasks associated with these applications do not occur in a single location. Some tasks require interaction with moving trains and thus need a platform that can move in tandem. This section provides insight into the design and availability of wheeled and tracked mobile platforms, in addition to motorized linear axes that can expand the footprint of a robotic system. Table 4-3 below compares four common types of wheeled/tracked platforms, including: (1) fixed wheels (skid-steer), (2) active casters, (3) active centered wheels, and (4) omnidirectional wheels.

Table 4-3: Qualitative comparison of the four categories of planar mobile platforms [Kulkarni, 2009]

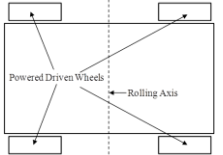
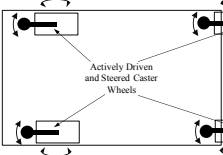
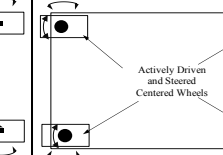
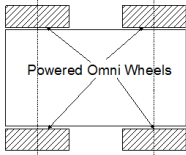
	Wheeled/Tracked Mobile Platform Configurations			
Attributes	Fixed Wheels/Tracks on Both Sides	Active Caster Wheels	Active Centered Wheels	Omnidirectional Wheels
Schematic				
Dexterity	Low dexterity	High dexterity	Good dexterity	High dexterity
Ruggedness	High ruggedness	Moderate ruggedness	Good ruggedness	Low ruggedness
Efficiency	Low efficiency	High efficiency	High efficiency	Least efficient
Sensitivity to Ground Quality	Low fault tolerance	High fault tolerance	No fault tolerance (when steering actuator fails)	Low fault tolerance (no fault tolerance up to the 4-wheel configurations)
# of Actuators per Wheel	1	2	2	1

Table 4-4: Commercially available mobile robotic platforms

<u>Platform</u>	<u>Make & Model</u>	<u>Type of Platform</u>	<u>Weight, Payload & Speed</u>	<u>Power & Runtime</u>	<u>Cost</u>
	Clearpath Robotics Platforms: 2 Shown: Husky	Wheeled Fixed (Skid-Steer) Qty: 4	Weight: 50kg Max. Payload: 75kg Max. Speed: 1.0m/s	Onboard Battery: Sealed Lead Acid Run Time: 8 hours Charge Time: 4 Hours	\$20 to \$25k
	Vetex Platforms: 3 Shown: Robomate 21	Wheeled Omnidirectional Qty: 4	Weight: 2500kg Max. Payload: 9091kg Max. Speed: 2.0m/s	Onboard Battery: AGM (Qty: 8) 48VDC	not available
	Segway Platforms: 7 Shown: RMP 400	Wheeled Fixed (Skid-Steer) Qty: 4	Weight: 120kg Max. Payload: 182kg Max. Speed: 8.0m/s	Onboard Battery: LiFePO4 Run Time: 20 hrs. Charge Time: 6-8 hrs.	\$46k
	Adept Platforms: 9 Shown: Seekur	Wheeled Active Centered Qty: 4	Weight: 300kg Max. Payload: 70kg Max. Speed: 1.8m/s	Onboard Battery: NiCad Run Time: 3-8 hrs. Charge Time: 8 hrs.	\$60k
	iRobot Platforms: 4 Shown: 710 Warrior	Tracks Adaptive Pitch Qty: 4	Weight: 165.5kg Max. Payload: 68kg Max. Speed: 3.58m/s	Onboard Battery: BB-2590/U (Qty:12) Run Time: 4-10 hrs.	200k+
	robosoft (FRA) Platforms: 4 Shown: robuROC 6	Wheeled Pods ± 30° Tilt Qty: 3x2 (PodsxWheels)	Weight: 160kg Max. Payload: 100kg Max. Speed: 3.61m/s	Onboard Battery: 72V Li-ion (Qty:3) Run Time: n/a	109k+
	Gudel Platforms: 25+ Shown: TMO	Linear Axis Rack & Pinion Qty: 1	Weight: 100-2000kg Max. Payload: Any Max. Speed: 7.0m/s	Tethered Flexible Cable Tray AC Power	Length Dependent

Table 4-3 provided an overview of each wheeled platform with respect to its dexterity, ruggedness, efficiency, sensitivity to terrain, and number of actuators per wheel. Overall, the platforms with active wheels tend to outperform the skid-steer and omnidirectional platforms, but require more actuators and have limited availability. This type of table should be referenced against the requirements of a particular application to pick the best wheeled platform. Planar wheeled or tracked mobile platforms with three or more DOF are a requirement if: (1) the location of a task cannot be predicted, (2) location of a task changes frequently, (3) task requires synchronized non-linear motion with another moving body, and/or (4) a single robot is responsible for multiple tasks in different locations. Further detail on the mathematical models and motion planning for each of type of wheeled and tracked mobile platforms can be found in [Kulkarni, 2009].

If a given task only requires repetitive motion along a single axis and can be performed in a single area, a motorized linear axis with one DOF can be used to rapidly move a robotic manipulator from one location to another. Commercially available examples of both planar mobile platforms and motorized linear axes are included in Table 4-4 above. This table is meant to provide an overview of existing mobile platforms and their capabilities with respect to platform weight, payload capacity, and speed. A designer should first choose a robotic arm with sufficient payload capacity and reach to accomplish a specific task before selecting a mobile platform. This should ensure that both the manipulator and mobile platform are properly suited for the task at hand to avoid overloading or damaging either component.

4.3 ROBOTIC END-EFFECTOR SENSING & TOOLING

4.3.1 Robotic Collision Sensor

A collision sensor is a device that is capable of mechanically detecting a crash before or during inadvertent impact between a robot's tool and an obstruction. The device "breaks-away" from its working geometry to absorb crash energy while simultaneously sending a signal to the robot's controller to stop all motion [ATI-1, 2013]. This functionality prevents costly damage to robotic manipulators and tooling resulting from an overload condition. Most commercially available collision sensors have an automatic reset capability, high repeatability, and carefully designed "allowable" moment/torsional rotations for a wide variety of applications. A schematic for a robotic collision sensor manufactured by ATI Industrial Automation is shown in Figure 4-14.

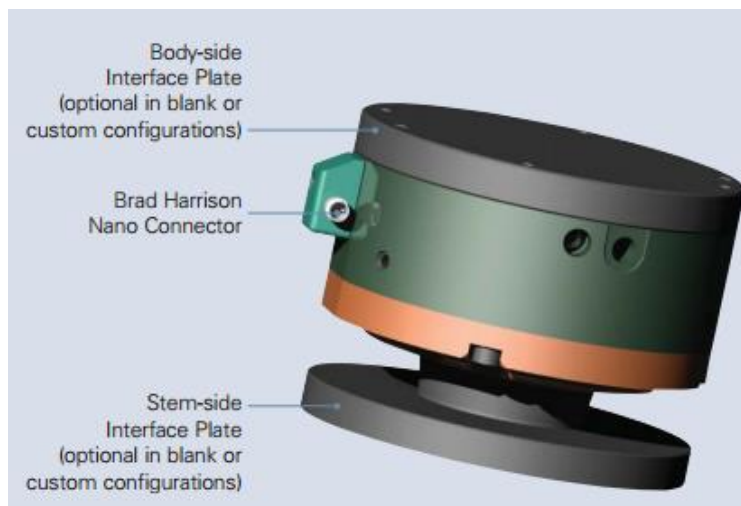


Figure 4-14: Robotic collision sensor diagram [ATI-1, 2013]

The collision sensor shown in Figure 4-14 consists of a main body and two interface plates (body-side & stem-side) that allow the device to be mounted to the robot and tool respectively using the required bolt patterns. The body typically houses a mechanical assembly that allows the device to elastically deform in the event of an

overload condition. In Figure 4-15, a “stem” protrudes a cover plate assembly that has an internal cam responsible for accurately and repeatedly positioning the stem assembly. The stem assembly is forced into position against the cam by a piston that is typically supported by a holding force proportional to a pneumatic input [ATI-1, 2013]. The nominal pneumatic input controlling the “break-away” point of the device should be determined by calculating the estimated loads due to static, dynamic, and working forces.

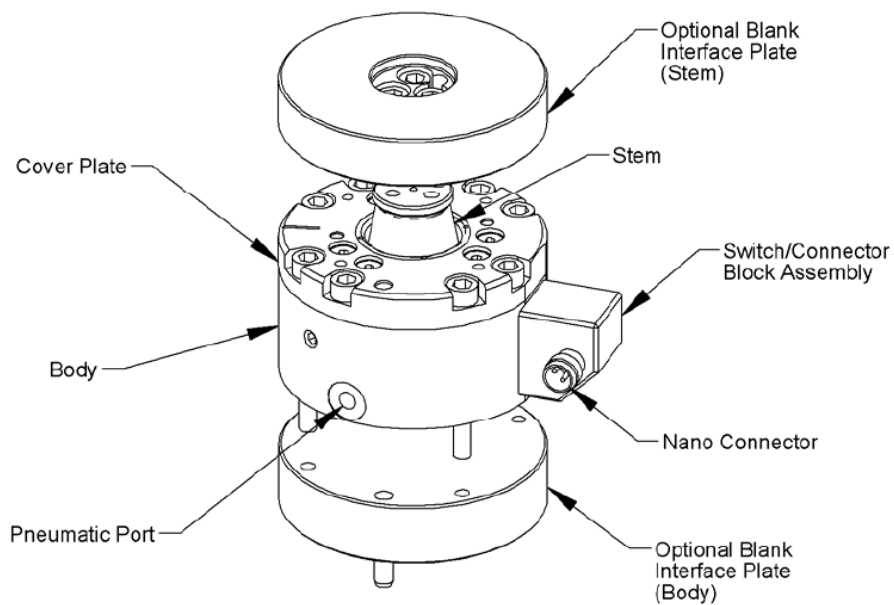


Figure 4-15: Robotic collision sensor diagram [ATI-1, 2013]

A collision sensing switch is positioned on the right side of the device’s body in Figure 4-15 above. This collision sensor is connected to the robotic control panel as a normally open PNP or NPN proximity switch. This output of the proximity switch changes state when the break-away force of the device is exceeded, allowing the stem assembly held by the cam and piston to move in order to absorb the crash energy [ATI-1, 2013]. More detail on the proper wiring diagram for the device and the internal components is shown in Figure 4-16 below.

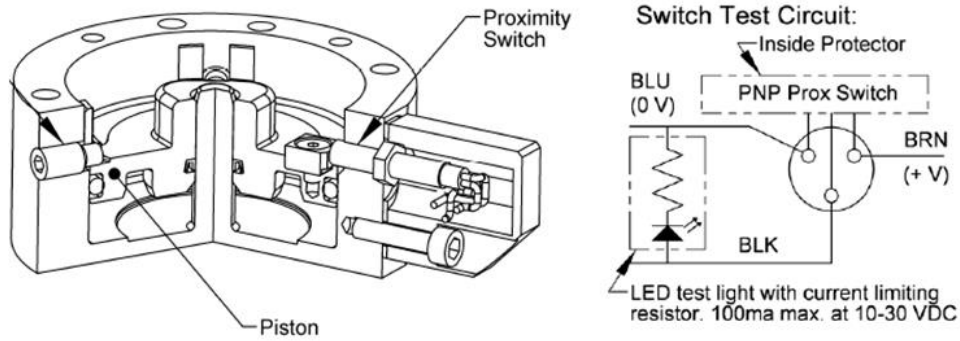


Figure 4-16: Internal components and circuitry for collision sensor [ATI-1, 2013]

The main function of a collision sensor is to protect industrial robots and tooling in the event of accidental impacts and unanticipated loads. The device is designed to respond to excessive torsional, moment, and compressive axial forces, or any combination of these by breaking-away from its working geometry. Collision sensors cannot respond to pure axial tension, which is not a common requirement of most robotic applications. Removal of the upsetting force(s) or moment(s) allows the device to automatically reset, eliminating the need for an operator to enter the robotic work cell to reset the system [ATI-1, 2013]. The crash responses provided by the device are shown in Figure 4-17.



Figure 4-17: Collision sensor response modes [ATI-1, 2013]



Figure 4-18: Commercially available collision sensors (AGI, RAD, ATI, IPR)

Collision sensor models are designed to meet certain requirements, including but not limited to recommended payloads, axial rotations, angular and axial displacements, break-away forces and moments, operating environments, and pressure ranges if applicable. Table 4-5 below provides a concise overview of the available collision sensor models sold by the manufacturers above. This table can be used to select the proper sensor model for a given application.

Table 4-5: Overview of specifications for commercially available collision sensors

Product Feature	Collision Sensor Manufacturer			
	AGI Automation Components	ATI Industrial Automation	IPR GmbH	RAD Robotics & Automation Tooling
Pneumatic Operation	✓	✓	✓	✓
Pneumatic with Spring Assist Option		✓	✓	✓
Spring Loaded Only Operation		✓		✓
Harsh Environments		✓		✓
Pressure Range	14-90 psi	20-90 psi	30-90 psi	10-90 psi
Response Time	2 - 8 ms	-	-	2 - 10 ms
Repeatability (XYZ)	± 0.001 in (x,y) ± 0.001 in (z)	± 0.001 in (x,y) ± 0.001 in (z)	± 0.001 in (x,y,z)	± 0.002 in to ± 0.0005 in (x,y,z)
Axial Rotation Range	± 360°	± 360°	± 360°	± 11° to ± 28°
Rotational Repeatability	± 0.024° to ± 0.029°	-	-	± 0.017°
Angular Displacement	± 5°	± 8° to ± 13°	± 4° to ± 12°	± 5° to ± 15°
Axial Displacement (Compression)	0.054 to 0.366 in	0.220 to 0.630 in	0.370 to 0.980 in	0.040 to 1.060 in
Axial Break-Away Force	0.7 to 1372 lbf	200 to 3100 lbf	25 to 4026 lbf	18 to 2697 lbf
Break-Away Moment Range	3 to 2250 lbf-in	60 to 17,000 lbf-in	18 to 10,267 lbf-in	15 to 12,833 lbf-in
Weight	0.44 to 8.20 lbs	0.70 to 25.10 lbs	0.70 to 56.2 lbs	0.20 to 32.00 lbs
Diameter	1.58 to 6.58 in	2.36 to 8.66 in	2.36 to 8.66 in	1.95 to 10.93 in
Maximum Depth	1.07 to 3.21 in	1.58 to 6.58 in	2.05 to 5.04 in	1.12 to 5.61 in

4.3.2 Force/Torque Sensors

Multi-axis force/torque transducers typically consist of three major parts: (1) mechanical components, (2) electrical components, and (3) a software interface. The sensor body itself is a monolithic structure designed to react to applied forces and torques according to Newton’s third law of motion: “For every action, there is always an opposed or equal reaction; or, the mutual action of two bodies upon each other are always equal, and directed to contrary parts” [ATI-2, 2013]. Forces applied to the transducer activate

three symmetrically placed beams in the sensor's body that are governed by Hooke's Law in Equation 4.4 below [ATI-2, 2013]:

$$\sigma = E * \varepsilon \quad (4.4)$$

where σ is the stress applied to the beam, E is the elasticity modulus of the beam, and ε is the corresponding strain applied to the beam. The resistance due to the applied strain can then be measured using high-end strain gauges attached to the beams according to Equation 4.5 below [ATI-2, 2013]:

$$\Delta R = S_a * R_o * \varepsilon \quad (4.5)$$

where ΔR represents the change in resistance of the strain gauge, S_a is the gauge factor of strain gauge, R_o is the steady state resistance of the strain gauge, and ε is again the strain applied to the gauge.

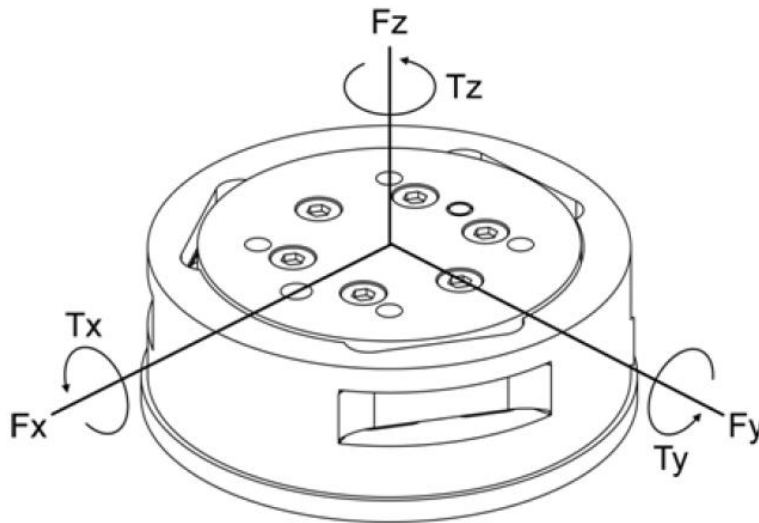


Figure 4-19: Force and torque vectors of Omega160 transducer [ATI-2, 2013]

The signals from the strain gauges in the Omega160 are connected to an ATI power supply box. This box requires a steady 5V DC power supply to amplify the raw voltages from the strain gauges. The 12 outputs from the transducer are shielded, twisted

pair wires. To further reduce noise and increase performance, these signals are wired differentially to a National Instruments (NI) data acquisition system. A diagram displaying a generic schematic for interfacing with an ATI F/T transducer is included in Figure 4-20.

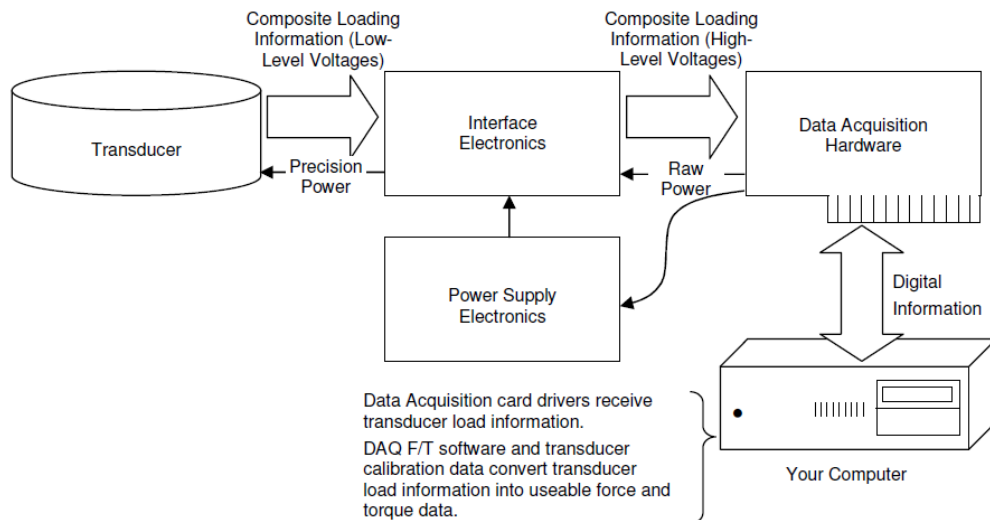


Figure 4-20: Electronics and software used to process F/T data [ATI-2, 2013]

ATI offers many different ways to communicate with their force/torque products, including Net protocols (Ethernet, DeviceNet, CAN bus), Wireless, DAQ, Controller (RS-232) and TWE (Transducer without Electronics). The RRG purchased the Omega 160 F/T sensor with a DAQ interface to take advantage of its superior data acquisition rates. The RRG selected National Instrument's CompactRIO 9082 (cRIO) real-time controller with a LX150 FPGA is used in conjunction with a NI 9205 32-channel analog input module to receive and process the raw voltages output by the Omega160 using LabVIEW Real-Time software. Using the cRIO 9082, analog sampling rates up to 40 kHz can be achieved. Rates of this magnitude allow the robot to sense and react appropriately to abrupt changes in F/T signals experienced at the end effector.

A custom application was developed in LabVIEW to deterministically collect and record F/T data at rates up to 1000Hz using NI's internal real-time protocols. This application also performs all of the operations required to obtain meaningful F/T values, including but not limited to: 1) biasing the raw voltages to remove the gravitational effects of hardware mounted to the end-effector, 2) multiplying the raw voltages by the working calibration matrix for the transducer to obtain meaningful values for force and torque, 3) ability to filter and/or implement statistical operations to manipulate the data, and 4) transform the local axis of the F/T sensor to match the required orientation of the gripper. The front panel of the custom data acquisition program developed for the Omega160 F/T sensor that includes the capabilities above is shown in Figure 4-21.

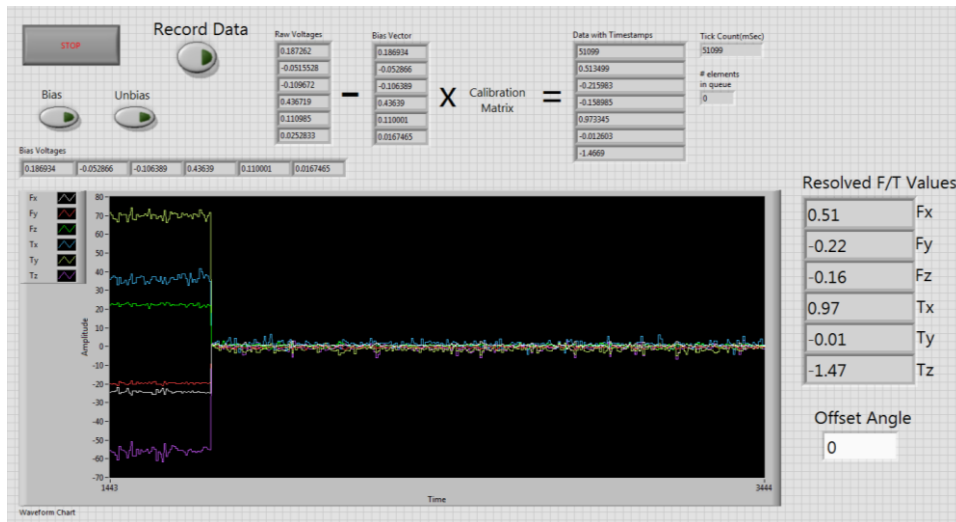


Figure 4-21: LabVIEW interface to view and log F/T data

This particular user interface transforms raw voltages into meaningful force and torque data and displays them on an interactive waveform chart on the front panel for real-time feedback. All aspects of the data acquisition program can be altered at any time to suit a user's preferences.

Overall, collision and F/T sensors are both end-effector components that can be extremely useful in the design of any robotic system tasked with automating a rail yard operation. Both can provide protection against critical failures to expensive robotic equipment. Additionally, the data gathered by a F/T sensor can be used to perform complex, coordinated maneuvers between two or more components in contact. More information on how these devices are used within a robotic system designed for pin-pulling will be provided in Chapter 5.

CHAPTER 5: APPLIED OBJECT DETECTION & RECOGNITION

As discussed in Chapter Two, machine vision is currently being used to identify and monitor the condition of numerous freight car components. These systems often capture frames of fast moving trains and save the data for post processing. The methods in this work aim to extend the capabilities of existing technology to accurately identify objects on slow moving freight cars in real time. One object that has not yet been reliably identified on freight cars is the uncoupling lever (Figure 5-1, left). The methods presented in this chapter were implemented to reliably detect multiple types of uncoupling levers. Many of the following techniques can be adapted to identify a multitude of different objects found in rail yards. Several critical freight car components are shown below in Figure 5-1.



Figure 5-1: Freight car components & safety appliances

Figure 5-1 displays images of an uncoupling lever handle and a standard sill step (left), a pair of Type E couplers and accompanying brake hoses (middle), and a handbrake wheel mounted on the end of a freight car (right). In a 2006 study by Edwards, the uncoupling lever “proved to be the most challenging of all safety devices to identify using machine vision” [Edwards, 2006]. The author states, “The primary difficulty in identifying the lever is the amount of “clutter” in the background which produces many

more edges than the uncoupling lever. Additionally, there is no angle from which the uncoupling lever can be viewed that allows it to appear parallel or perpendicular to any reference point in the image. Another difficulty in recognizing the uncoupling lever is the wide variety of uncoupling lever designs. While all of these uncoupling lever designs conform to CFR Part 231 the slight differences provide challenges to a machine vision system” [Edwards, 2006]. Although many of the concerns stated by the author are true, the RRG believes that this problem can be solved using currently available machine vision hardware and software. To begin tackling this problem, a review of existing uncoupling lever hardware was performed. Sample images of common lever handles are shown in Figure 5-2 below.



Figure 5-2: Uncoupling lever handle types

Although there are hundreds of different part numbers of uncoupling levers in the inventory systems of major railroads, the levers can be separated into two distinct categories: linear and curvilinear. The most common geometry for an uncoupling lever is the linear handle shown in the leftmost image in Figure 5-2. The geometry of the lever assembly may vary between different part numbers, but the handle to be grasped is simply a 12 to 16” piece of 1” diameter steel bar stock. The curvilinear handles can be further classified into “R-shaped” (Figure 5-2, middle-right) and “U-shaped” (Figure 5-2,

middle-left and rightmost) handles. While the geometry of these three handle types are different, each handle has a linear portion that is 6” or longer and is constructed out of 1” steel bar stock. In addition to these similarities, the uncoupling lever detection process is loosely constrained by the following parameters:

- Consistent location on each freight car (height, proximity to car edge, etc.)
- Standard material & size (1” diameter steel bar stock)
- Significant portion of each lever handle is linear
- 250+ lever types can be grouped into three categories (straight, R-shape, U-shape)
- Low operating speed (0.1 - 4.0 mph), direction often constant
- No standard color

Using the knowledge above, a generic lever detection application can be designed using standard image processing and machine learning techniques. If necessary, application-specific heuristics can be added to increase speed, reliability, and/or performance. To begin development, digital videos capturing hundreds of moving freight cars were acquired from classification yards using different cameras, sensors, and image formats. Data was acquired throughout the morning, afternoon, and evening to ensure a broad range of lighting conditions were accounted for. The following sections will detail the hardware and software used to develop an object detection system for freight car appliances.

5.1 CAMERA PERSPECTIVE & DIGITAL IMAGE ACQUISITION

Many different perspectives were tested to determine the best angle to find the uncoupling lever using machine vision. The relative position and orientation of the camera with respect to the freight car is critical to maximizing contrast between the lever handle and its background while minimizing noise in order to consistently segment the

lever from the image. A similar analysis for finding the maximum number of safety appliances with a minimum set of cameras was performed by [Edwards, 2006]. Two different perspectives proved to provide the best results: angled side view (approx. 45° relative to the track) and perpendicular side view (approx. 90° relative to the track). These perspectives are shown below in Figure 5-3.



Figure 5-3: Camera perspectives, perpendicular (left) & angled (right)

While the angled side view shows the entire lever assembly in Figure 5-3 above, the optimum angle for finding the lever handle using 2D and 3D vision systems was eventually found to be the perpendicular side view. From this perspective, the lever handle appears in a larger number of frames with less clutter in background, thus providing the required amount of contrast needed to successfully and reliably detect the lever handle. Additionally, the depth of the lever handle from a camera mounted parallel to the rail remains constant, whereas depth would steadily change (decrease) as a handle approaches a camera mounted at an angle to the rail.

Several off-the-shelf, inexpensive digital cameras listed in Section 4.1.1, including the Logitech Pro 9000 and Sony PS Eye webcams, were initially used to capture data perpendicular to the rail in a classification yard located in Houston, TX. These digital cameras provide color images, where each pixel contains a combination of red, green, and blue components that define its color. Visual features in color images, such as pixel intensity values, gradient of intensity, and contour geometries, can be used to identify objects of interest, e.g. uncoupling lever handle.

Object detection can be performed on-line using live data streams or off-line using saved images. The methods detailed in this chapter target on-line processing and recognition in real-time. Therefore, cameras with increased frame rates and reduced resolutions will be used to boost processing speed to ensure real-time detection of lever handles as they pass by the camera(s). The following section will detail a custom object detection application developed by the RRG to detect uncoupling lever handles.

5.2 OBJECT DETECTION USING DIGITAL IMAGES

Digital color images were initially used to detect the uncoupling lever handle to take advantage of existing data from rail yards, available standard image formats and compression techniques for storage, increased information flow, and low-cost acquisition hardware. Images and video streams were first acquired from several perspectives at varying times throughout the day to develop a generic application to detect handles in a wide range of scenarios. A majority of the data was collected during the day with lots of ambient sunlight while other videos were recorded at dawn and dusk. To minimize the effects of changes in the illumination, a histogram equalization process (Equations 3.1-3.3) was first performed on the raw data to evenly distribute the image's pixel intensities. This flattened the histogram of each image frame to enhance contrast and thereby

accentuate sharp changes in pixel intensity values in preparation for edge detection. Once equalized, the frames are converted to grayscale images to process the frame using the standard Canny Edge detection algorithm outlined in Section 3.1.2. Canny's method was implemented in both OpenCV and MATLAB to find edges corresponding to the lever handle similar to Figure 5-4 below.



Figure 5-4: Original image (left) & detected edges using Canny (right)

After testing a range of different sigma values, a magnitude of 1.1 was chosen for the Gaussian smoothing operation, as it provided consistently acceptable edges in both 2D and 3D images. Upper to lower ratios between 2:1 and 3:1 for the hysteresis operation eliminated extraneous edges and contours, thereby providing an acceptable output for subsequent operations. These ratios are consistent with recommendations found in [Canny, 1986]. To further improve the connectivity of the resulting edge map, an inexpensive morphological operation was used to bridge the gap between pixels in close proximity (one pixel width). Once this is complete, the edge map is ready to be analyzed to find edges corresponding to the object of interest, which in this example, is the uncoupling lever handle.

To successfully grasp and manipulate the uncoupling lever, the linear portion of the lever handle must be identified. This linear portion of any standard lever handle can be characterized as a cylindrical metallic object with a 1” diameter. In a 2D edge map, cylindrical objects are represented by pairs of straight, parallel lines having approximately the same length. Therefore, the Hough transform algorithm can be implemented to find straight lines in the edge map. Using the steps outlined in Section 3.1.3, the Hough transform algorithm accumulates “votes” in a discrete set of bins in the Hough space, and returns values (ρ, θ) where $Hough(\rho, \theta)$ exceeds a user-defined threshold. The straight lines detected (shown in red) in Figure 5-5 are then defined by the polar description of a line: $\rho = x\cos(\theta) - y\sin(\theta)$.

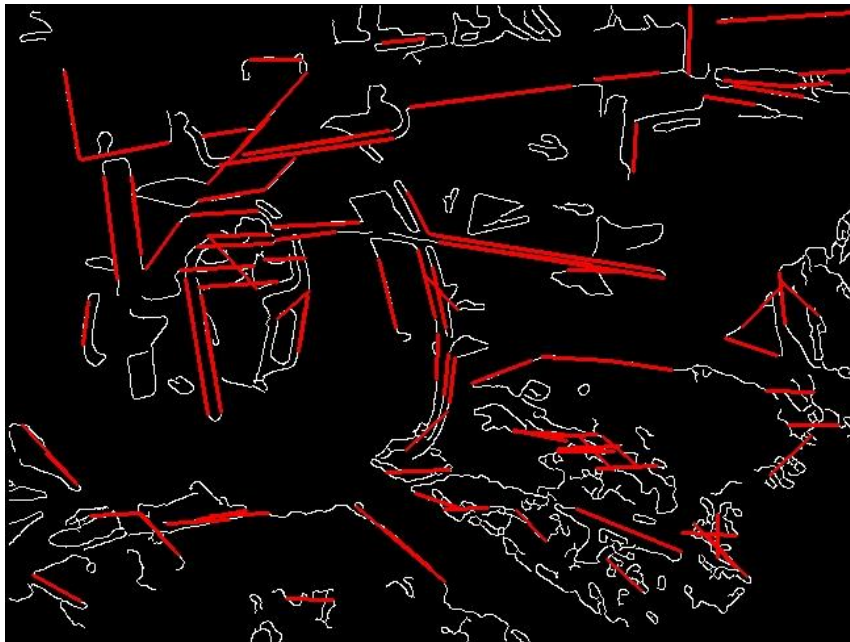


Figure 5-5: Straight lines detected by Hough transform

The straight edges returned by the Hough transform can be filtered based on their length, orientation, location in the image frame, and error detected in the approximation

of the contours. If the detection task is constrained to a finite set of objects with known geometries (e.g. finding uncoupling levers) and the relative size of an object is near constant, carefully calculated thresholds for desired edge lengths and/or orientations can be defined. Additionally, OpenCV has a function named *approxPolyDp()* that approximates contours using a number of points specified by the user. More points will result in lines that more accurately follow contours returned by the edge detection algorithm while fewer points will result in straight lines with greater lengths and less accuracy when compared to the actual shape contour. The edges corresponding to the linear portion of the lever handle must now be extracted from the vector of lines returned by the Hough transform. To accomplish this, the orientation of each line is calculated to cluster edges with similar orientations that are also within close proximity (10-20 pix) to one another to find pairs of parallel lines that could represent a lever handle.

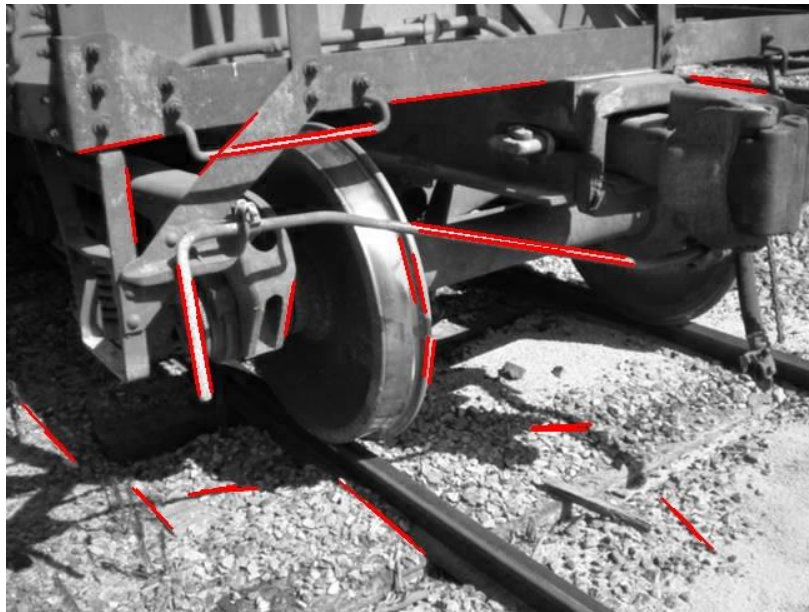


Figure 5-6: Pairs of parallel lines detected by Hough transform

An ideal object detection application should only return one set of parallel edges that belong to the uncoupling lever handle. However, the probability of multiple edge pairs meeting the required heuristics is quite high due to the number of edges detected in the image frame(s). As expected, Figure 5-6 above depicts multiple pairs of parallel edges. A threshold for minimum length was set previously to eliminate smaller, spurious edges caused by the ground or other discontinuities, but no heuristics were set for orientation. Lever handles typically hang in a near vertical orientation while cars are coupled. Therefore, the returned pairs shown in Figure 5-6 were again filtered by their orientations, looking for an angle of $90^\circ \pm 5^\circ$. The results from this operation are shown below on the left side of Figure 5-7.

Blob Creation & Tracking

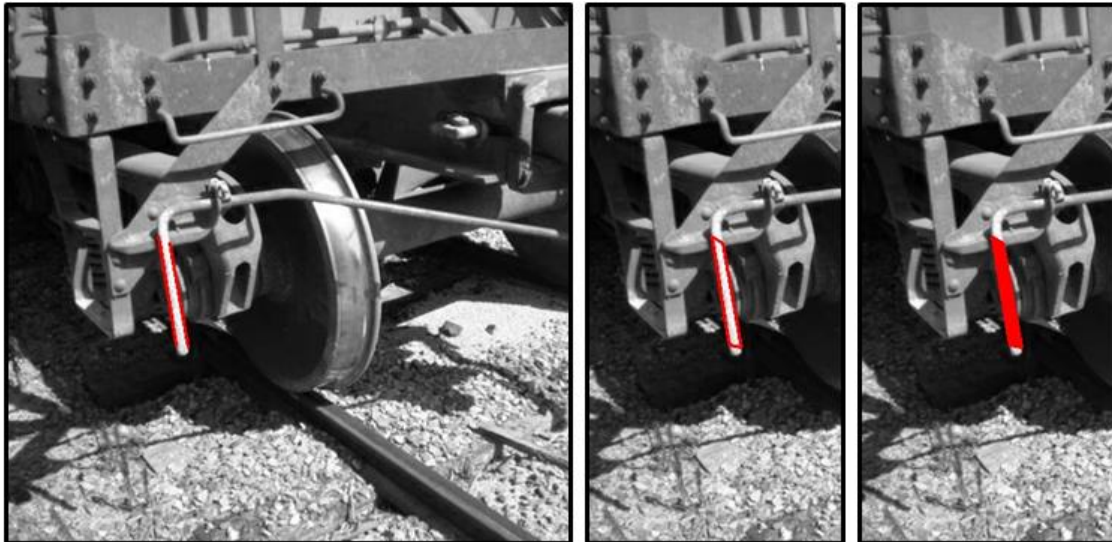


Figure 5-7: Filter line pairs for vertical orientation, enclose boundaries, and fill

The lone pair of lines returned by the orientation represents the edges of the uncoupling lever. To give the returned lines further meaning, “blobs” were created by connecting the end points of the pair of parallel lines to create a single, enclosed

boundary. This step is shown in the center of Figure 5-7 above. The resulting boundary is then filled with a uniform intensity to create a blob. This creates a binary image with “blobs” of interest. In the field of computer vision, blobs are used to detect and/or track regions of interest in digital images. Blobs are created and used here in an attempt to make lever detection consistent between video frames. Many standard libraries exist that include useful functions for characterizing (e.g. blob statistics, centroid calculation) and processing (e.g. geometric-based filters) blobs. The user can also decide whether or not to create blobs in each frame via the process outlined above, or “fit” blobs to detected regions in the image. Since all handles are one inch in diameter and the camera position remains fixed in this example, the same blob geometry can be used to approximate the movement of the handle across the frame, thereby reducing the influence of changes or discontinuities in edges. This increases the robustness of the algorithm to detect multiple handle types on any freight car in varying lighting conditions. The entire process for detecting levers using lines and blobs is visually summarized in Figure 5-8.

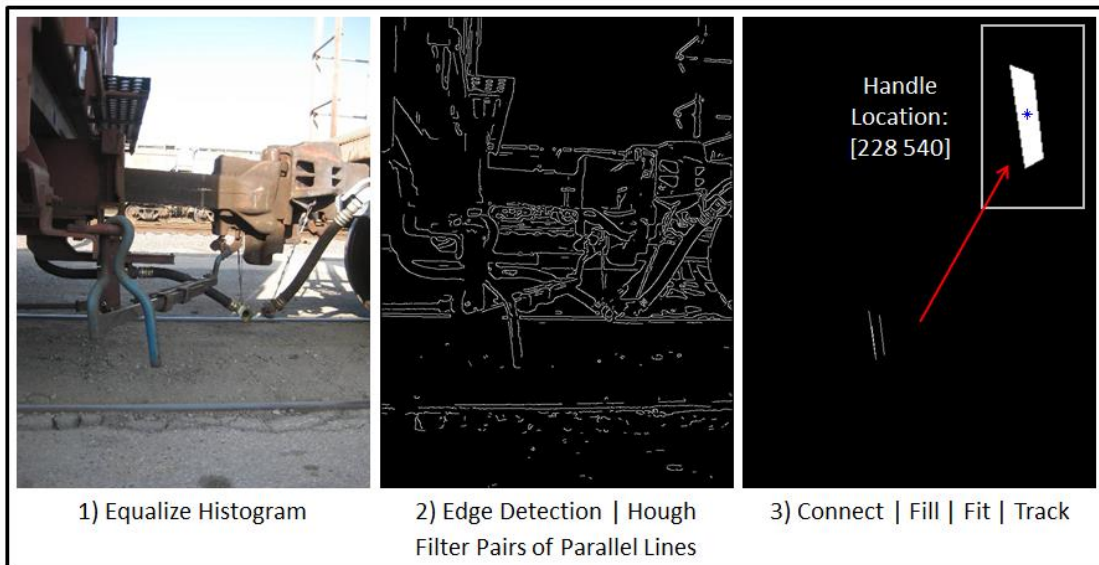


Figure 5-8: Uncoupling lever handle detection – blob creation and tracking

The last stage of the uncoupling lever detection process is to calculate the centroid of the resulting blob. This is accomplished by finding the center of mass for all connected components, which in this case, are the white intensities of the blob representing the handle. The resulting centroid, $I(i, j)$, is the image coordinate equivalent of the center of the physical handle. Since this is merely a digital image, the spatial coordinates of the handle are not known. To calculate the spatial location of the handle, the digital camera used to find $I(i, j)$ would have to be accurately calibrated with a range camera or a second digital camera looking at the same scene from a different perspective to triangulate the image using stereo vision. The use of 3D cameras for detecting uncoupling levers will be covered in the following section.

Uncoupling Lever Assembly Detection

In an attempt to make the handle detection algorithm more robust, the Canny Edge detection and Hough transform algorithms were applied to the entire uncoupling lever assembly. Many items found on freight cars are linear, and thus can lead to false positives when searching for the linear portion of a lever handle. However, the overall shape of the entire uncoupling lever assembly (handle, rod, mounting bracket) is very unique. Thus, if the top endpoint of a line representing the linear portion of the handle is found in close proximity to an endpoint of a line representing the uncoupling lever rod, the overall confidence in the detection of the handle increases. Straight handles typically form a 90° angle with the rod that attaches the uncoupling lever assembly to the coupler. This bend usually occurs near where the handle is hinged in the uncoupling lever bracket. As a result, the orientations for each of the straight lines output by the Hough transform were compared, looking for the proximity of the endpoints of a vertical line ($\sim 90^\circ$) and a horizontal line ($\sim 0^\circ$). If lines at these two orientations are detected and they either

intersect or their endpoints are within an acceptable proximity to one another, then the uncoupling lever assembly is detected with a similar result to that shown in Figure 5-9.



Figure 5-9: Uncoupling lever assembly detection – geometric fitting

The resulting detection is accentuated using lines and endpoints that correspond to the geometry of the lever handle. This method was able to recognize the lever assembly in a number of streaming video frames, but the rate of false positives was greater than the blob detection/tracking method. More filters were added in an attempt to fit the entire lever assembly; from the bottom tip of the lever handle all the way down the lever rod to the coupler. This however did not help improve the detection rate. Overall, this approach did not produce reliable results. The geometry of uncoupling lever assemblies (excluding the handle) vary much more than the handle type itself, thereby rendering the heuristics used to fit all line segments useless. Selecting thresholds for distances between the endpoints was also not an easy task, resulting in either too many false positives or false negatives. These methods could be potentially be used as statistical checks in future implementations of the algorithm, but should not be relied upon to find the lever.

Contour Extraction and Filtering

A contour is a sequence of points that represent a curve and differs from edges/edge detection in that it provides more insight into what an edge might physically represent [Bradski, 2008]. To calculate the contours in an image frame, a binary image must be created. This can be done using a standard intensity threshold, an adaptive

threshold that attempts to respond accordingly to changes in an image's histogram, or using an edge detection method. Contour extraction algorithms then find bounded, continuous sequences of points, storing each contour into an array [Bradski, 2008]. Several contours representing uncoupling lever handles and/or assemblies are displayed in Figure 5-10 below.

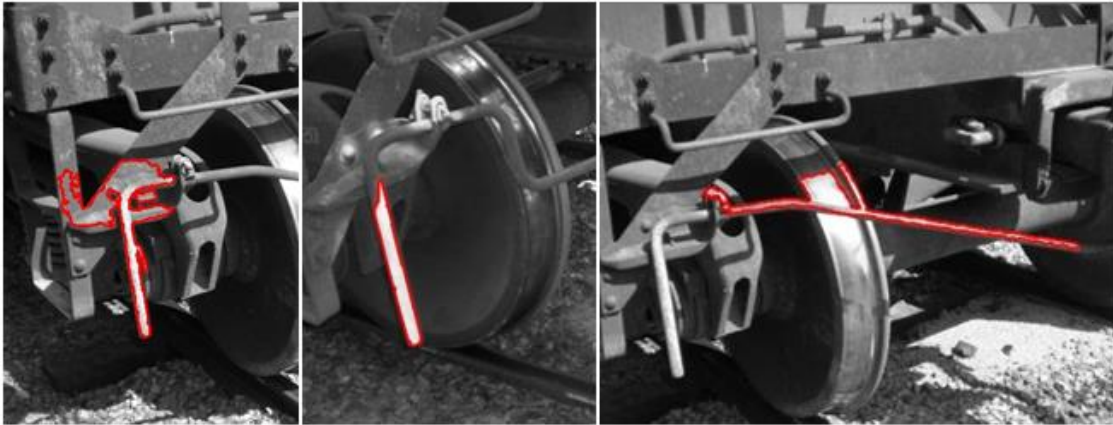


Figure 5-10: Contour extraction from digital images

Once contours are calculated and stored, they must be filtered similarly to the blobs created using the method describe earlier in this section. This can be a tough task, considering that many of the contours that include the geometry also include larger contour sequences that happen to intersect the object of interest when the binary image was created. Therefore, all contours must be processed to find ones that contain a portion with similar geometry to that of a handle type. Contour search methods are also computationally expensive and are not optimized for real-time applications. Contour detection has not yet proved to be a reliable method to detect the lever handles from frame to frame in a live video stream. Many sequences of points that represented uncoupling lever assemblies did not form a complete, closed structure due to

discontinuities in the edge detection step, and thus were not detected. Matching contours using “Hu Moments” or utilizing pairwise geometrical histograms could increase the effectiveness of this algorithm [Bradski, 2008].

Increasing Speed

To boost the speed of a detection algorithm, a binary mask can be used to ignore pixels that do not need to be processed. Uncoupling levers are typically hinged approximately 36” from the ground. Therefore, if the camera position is fixed, one could assume that all levers will appear within a certain linear “zone” in the video frame. A binary mask can be created by setting pixel intensities inside the zone to 1 and all others to 0. An example mask is shown in the top, left image in Figure 5-11 below.

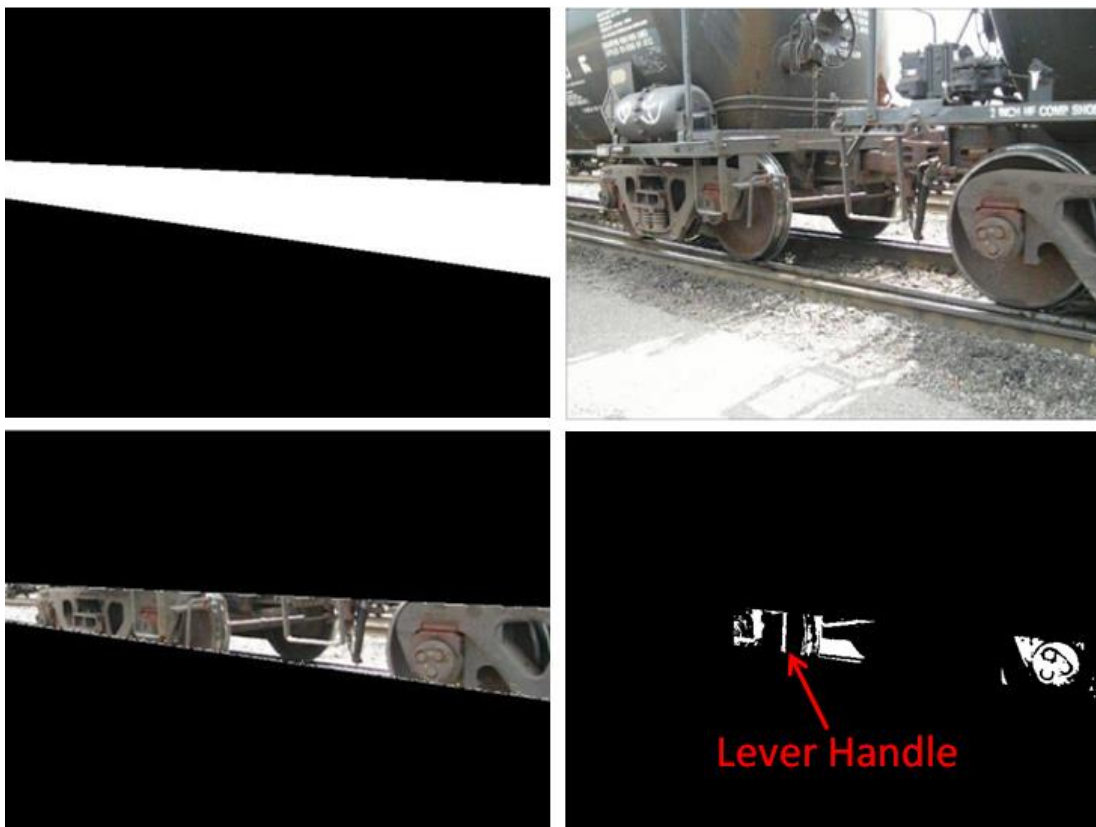


Figure 5-11: Lever detection using binary masks & contour extraction algorithms

When this mask is applied to the original frame (Figure 5-11, top-right), the only area that remains is the zone where the handle always appears (Figure 5-11, bottom-left). This reduction in data to be processed helps decrease the cycle time for the object detection application tremendously. However, the user must design the mask carefully to ensure that it doesn't have the potential to exclude the object of interest in any frame or instance. These types of masks are typically only used if the task is well defined and tightly constrained (e.g. detecting uncoupling lever).

Summary of Digital Image Processing Techniques

In summary, many of these methods worked well to detect uncoupling lever handles in single image frames. The blob creation and tracking method was tested on a video stream containing 30 different freight cars, each containing one of three handle types, in an actual classification yard. In this analysis, the algorithm correctly detected all 30 handles, but only maintained a 60% detection rate from frame to frame when the object of interest was in view. False positives were also common, caused by inconsistent edge maps (see Figure 5-12) due to cluttered backgrounds and constant changes in illumination. Higher quality vision systems with adjustable illumination sources would alleviate most problems associated with these two-dimensional digital imaging methods.

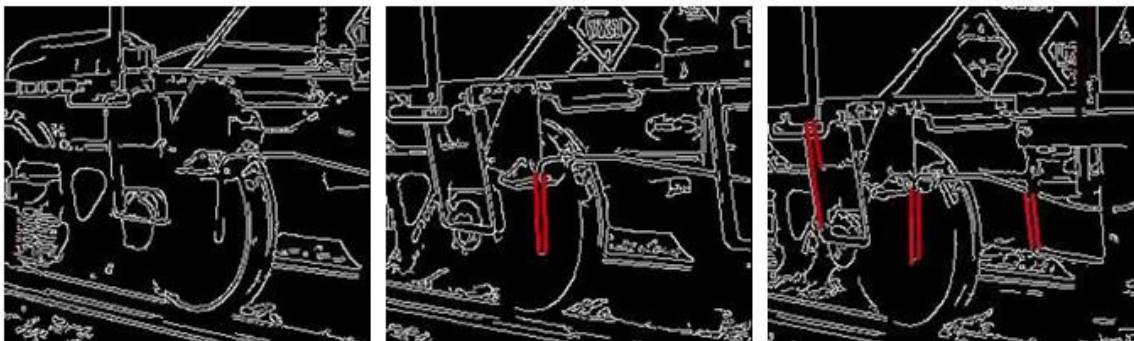


Figure 5-12: Inconsistent edge detection results using digital images

5.3 OBJECT DETECTION USING 3D DATA

An alternative approach to using a digital camera is to use range cameras. These technologies provide the direction and range to a discrete number of points in the environment from the sensors frame of reference. Many different types of commercially available technologies, including time-of-flight, structured light, stereo vision, and laser rangefinding, can provide three-dimensional data with varying degrees of accuracy and repeatability (see Chapter 3 further detail). Three-dimensional data can be represented in many different forms; two of the most common are point clouds and depth maps. A point cloud is a set of data points in a defined coordinate system. In this chapter, point clouds represent surfaces of real world objects in the Cartesian coordinate system. An example point cloud of standard rail appliances is shown below in Figure 5-13.

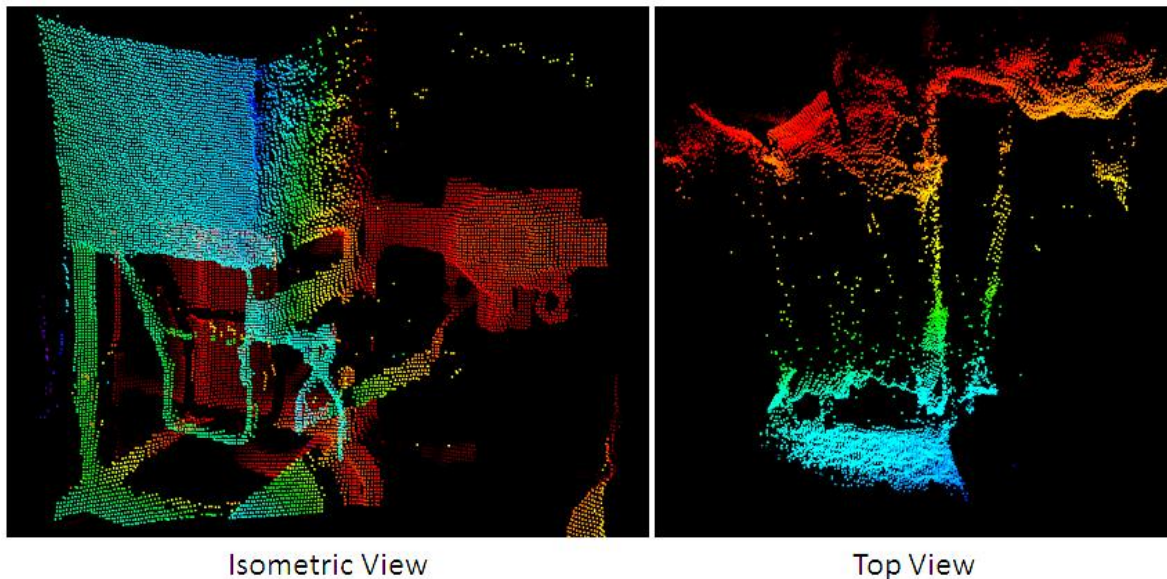


Figure 5-13: Three-dimensional point cloud acquired using SwissRanger ToF camera

The point cloud in Figure 5-13 is viewed from two different perspectives; an isometric and top view respectively. Point clouds are generally converted to triangular

meshes to construct a surface, as most 3D applications cannot process point clouds in their raw form as shown in the Figure above. Point Cloud Library (PCL) is a standalone, large scale project for 2D/3D image and point cloud processing that has in-built filtering, segmentation, surface reconstruction, and model fitting functionalities [PCL, 2013].

Although point clouds provide a wealth of information, this work primarily focuses on depth maps, also referred to as range images. A depth map is simply a two-dimensional representation of three-dimensional data. The physical depth of each pixel in a 2D depth map is stored as an intensity value. These intensities can be mapped to spatial distances using intrinsic properties of the camera. A sample depth map and its digital counterpart are shown in Figure 5-14 below.

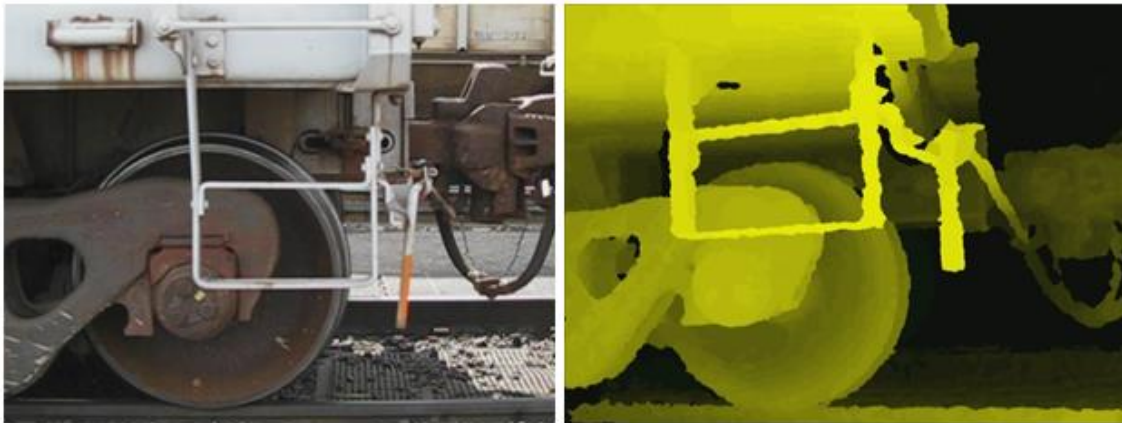


Figure 5-14: Digital image (left) and depth map (right) of a passing train

The image on the right of Figure 5-14 (right) was captured by a Microsoft Kinect sensor. Depth information is captured at 30 FPS that can be used to create depth maps. Lighter colored intensities in the depth map represent pixels that are closer to the camera while darker intensities characterize pixels that are further away. The resolution of a depth camera defines the smallest change in distance that it can sense, and is driven by its bit-depth. The Kinect uses a monochrome sensor with 11-bit depth, which provides 2048

levels of sensitivity [Zhang, 2012]. In Figure 5-14, the lever handle is approximately 1 meter from the camera with $\pm 1\text{cm}$ accuracy.

Depth Filtering

One of the best characteristics of the 3D data compared to the 2D digital images in section 5.2 is the ability to filter pixels based on their distance from the camera. If the motion of an object of interest is limited to a specific depth in the camera's sensing range, a clean 2D map can be obtained by filtering out the pixels at depths where the target cannot exist. This technique is especially useful for railroad applications because the rail constrains the freight cars from moving closer or further away from the vision system. An example of a filtered depth map can be seen in Figure 5-15 (right).

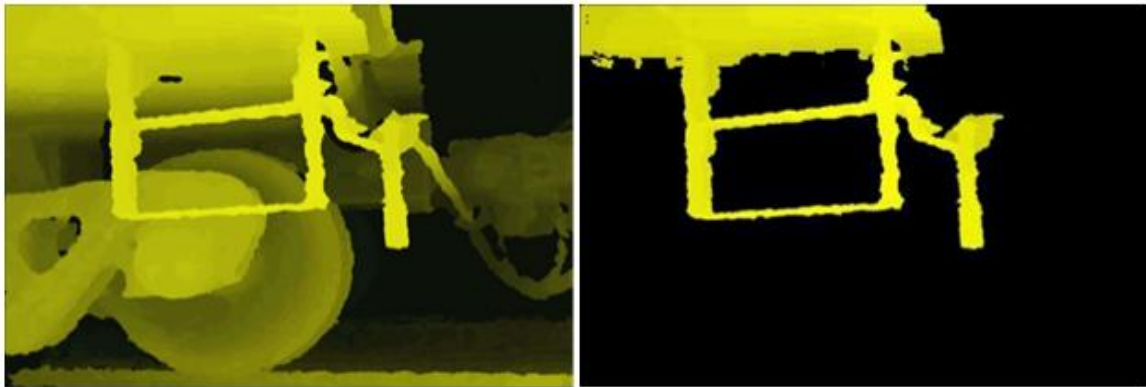


Figure 5-15: Full depth map (left) and filtered depth map (right) of a passing train

5.3.1 Lever Detection using Image Processing

The filtered depth map in Figure 5-15 provides an excellent image to begin searching for the linear portion of the uncoupling lever. The primary objects that are typically present in the filtered map are the uncoupling lever handle, lever mounting bracket, sill step, and side sill of the freight car. Occasionally parts of the wheel, truck, or lever will appear in the frame, but for the most part, the background clutter has been

removed. The primary task going forward will be to differentiate the linear portion of the lever handle from the vertical sides of the sill step, as they are geometrically very similar. One way to quickly avoid false positives corresponding to the sill step is to perform a simple morphological operation to “fill” the inner portion of ladder enclosed by the step’s outer edges shown in Figure 5-16a. This produces a new image with a solid cluster of pixels shown in Figure 5-16b, on which an edge detection process can be performed.

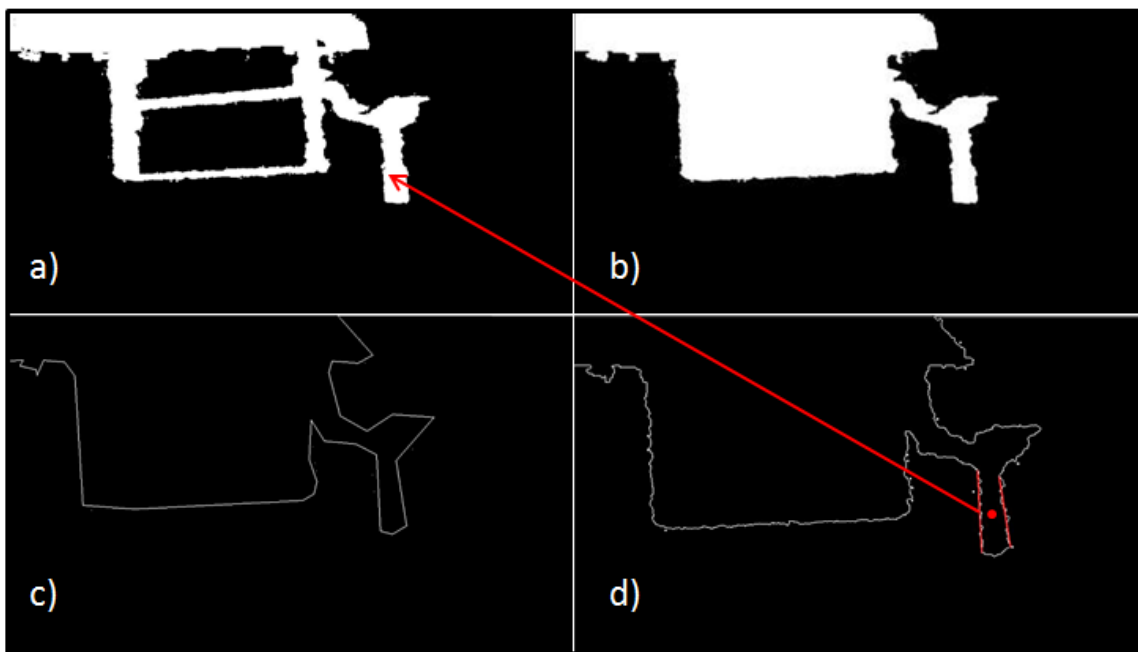


Figure 5-16: Lever detection process using depth maps a) original map, b) fill enclosed areas, c) polygon approximation, and d) find parallel lines & centroid

The Canny edge detection process was used in Figure 5-16c to extract the borders of filtered depth map. Canny was chosen due to its maturity, consistency, and optimized implementation in OpenCV. The ISEF algorithm described in Section 3.1.2 was also tested, but did not prove to be as fast or accurate as OpenCV’s *Canny()*. The polygon approximation algorithm *approxPolyDp()* was again used to smooth the resulting edges

to reduce uncertainty and increase speed. The Hough transform algorithm is again used to search for “straight” lines in the image and a similar process to that outlined in the previous section is used to search for sets of parallel lines corresponding to the handle. Heuristics are then used to filter out pairs of lines that do not meet loosely constrained length, orientation, and width requirements.

This algorithm works quite well on a live depth stream from a 3D camera and is sufficiently fast, but does have the potential to produce false positives (e.g. vertical parts of sill step). This occurs when the outer portion of the sill step does not form a complete contour, which causes the fill operation to fail. Additionally, many other objects have the potential to appear similar to a cylinder in the video. Therefore, post-processing techniques were used to guarantee success for each set of frames with a handle. With a static camera mounted parallel to the rail monitoring a moving train as it passes by, the detections shown in Figure 5-17 were detected.

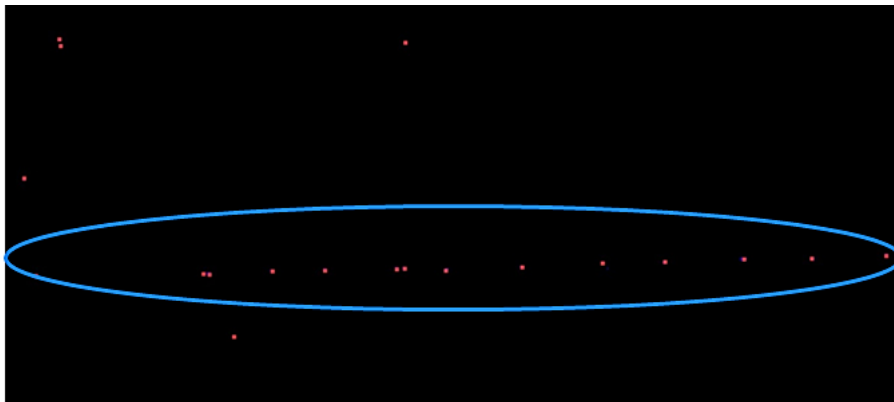


Figure 5-17: Handle detections over 20 video frames (true detections circled in blue)

The red circles in Figure 5-17 represent the centroid of the “detected handle” in a series of consecutive frames. As evidenced by the blue oval, a majority of the detections occurred at the same height in the image. Since the location of the uncoupling lever

handle does not change height as the train passes the camera, these points can be clustered to eliminate all the outliers after the data has been gathered. This ensures success, but cannot be used for the real-time control of a robot. This type of algorithm could potentially be deployed upstream to give the robotic system an initial estimate as to the location of the handle, but other methods must be explored for reliable real-time detection and control.

5.3.2 Lever Detection using Machine Learning

Artificial Neural Network (ANN) Training

Machine learning techniques are used in many fields, including machine vision, to quickly analyze large sets of data to identify patterns. The first pattern recognition technique tested for the detection of uncoupling lever handles was the Artificial Neural Network (ANN) that was first introduced in Chapter 3. In order to validate the effectiveness of an ANN for detecting uncoupling lever handles, test data was required to properly train the network. Therefore, hundreds of depth images were acquired of multiple uncoupling lever types using the ASUS Xtion sensor detailed in Chapter 4.



Figure 5-18: Filtered depth map (pixels shown are 0 to 1.5 m from camera)

Training images can then be generated from the filtered depth map to train the network using supervised learning. For supervised learning, true and false training images must be generated and input to the ANN to determine the final weights used to classify future images. Over 300 true images (true lever handle) and 475 false images (anything in the scene that is not a handle) were collected and input to the ANN for this application. A few samples of these training images are included below in Figure 5-19.



Figure 5-19. Three true images (left) & three false images (right) used for ANN training

The time required to adequately train an ANN often constrains the number of network sizes, network configurations, and training parameters that could be tested to optimize a network for a deadline. Training an ANN is typically considered the most expensive computational operation in the detection pipeline. The actual prediction (forward propagation) is only a series of matrix multiplications and can thus be done in real time. Therefore, if the time required for training can be significantly reduced using object oriented code provided by OpenCV, more training parameters can be tested to find the “optimal” performance of the network. The original training images collected for training were 80 x 120 pixels (9,600 total) in size. This size did not have a significant impact on the offline training of the ANN, but was later found to significantly reduce the application cycle time to the point where the prediction step could not be calculated within the frame rate of the Xtion camera (30 milliseconds). Therefore, the training

images were scaled down to 20 x 30 pixels (600 total, maintains aspect ratio) to reduce the size of the input to the network by 16. These training images are then fed to the ANN in the form of a row vector to iteratively perform backpropagation. To provide the network with 700+ training images, one large binary “training matrix” was created. This matrix contains 700 unique 1 x 600 arrays that each represents one 20 x 30 true or false image that has been “unrolled” into a row vector. A separate column vector is also defined to store the correct label (true = 1 or false = -1) for each of the 700 vectors. This column vector signifies that the ANN is learning will learn through “supervision”. A visual representation of the “training matrix” is shown in Figure 5-20.

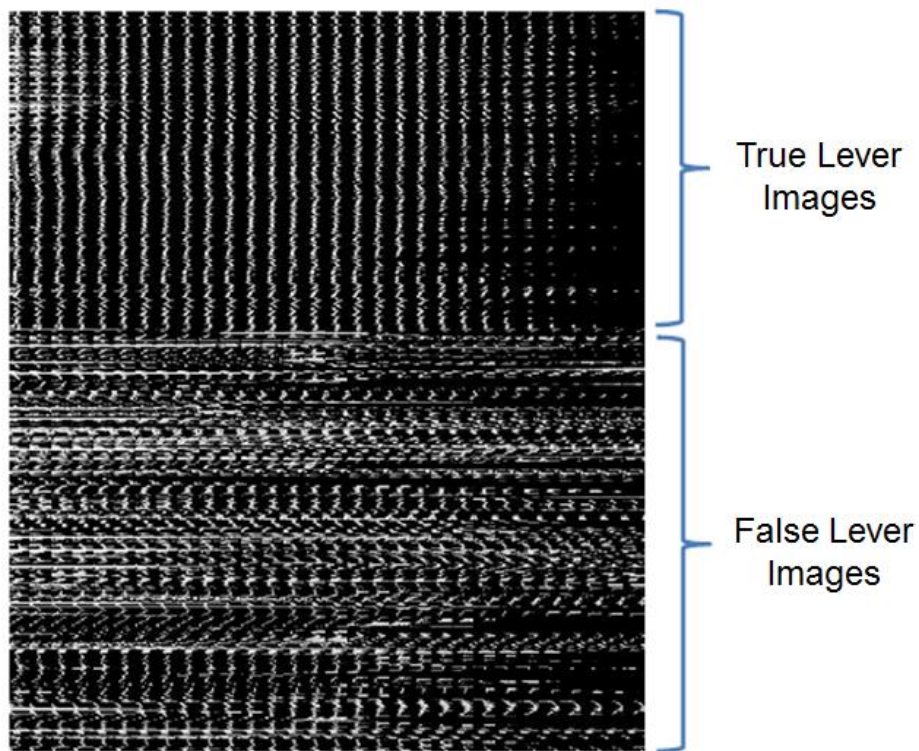


Figure 5-20: “Training” image, 300 true, 400 false images (row vectors = 600 pixels)

The first 300 rows of the “training matrix” contain true lever images and the bottom 400 rows contain false lever images. One interesting result to notice in Figure 5-

20 is the difference between the true and false sections due to the principal vertical component present in a majority of the true lever images. The difference between true and false samples is a good indicator that the ANN will work well for this application, but this difference not always evident to the human eye. The network is now ready to be trained. Training in OpenCV uses either classical random sequential back-propagation or batch RPROP (a direct adaptive method for faster learning). Both of these methods used to evaluate performance, in addition to an array of varying learning rates, momentum terms, and network topology variables, to find the optimal ANN topology. The varying results using each will be covered in the following sections.

After training, OpenCV conveniently has the ability to save the network topology and all pertinent weights to a file to use at for predicting future outcomes. The weights are used to again implement forward propagation to “predict” whether or not future images that the network has not yet seen are levers. Since the training step is one of the most time consuming steps of the implementation of an ANN, a checkpoint was implemented after training where the topology and weights of the tested network are saved to a file. If this file exists upon the execution of the program, the user is immediately prompted to choose between using the recovered trained network to predict future outputs and re-training the network entirely. A second checkpoint was also desired within the training function itself to save the weights of all layers after N iterations. However, the standard implementation of the training function in OpenCV unfortunately does not allow the user to access to any of its inputs during training.

To identify the lever in a full image frame from the camera (640 x 480), a “sliding widow” algorithm was created to scan the entirety of consecutive frames in search of the object of interest. As the window passes over the image, the trained ANN model predicts the output for each window. To further increase the speed of the algorithm for real time

use, frames from recorded video are scaled down by a factor of four to 160 x 120 and regions of interest are implemented to reduce the area over which the sliding window must evaluate. Naturally, the size of the sliding window is also set to be 20 x 30 to match the size of the scaled training images. Step sizes of five were used to horizontally and vertically to scan the region of interest in incoming images at high rates. A visual example of the sliding window algorithm is shown in Figure 5-21.

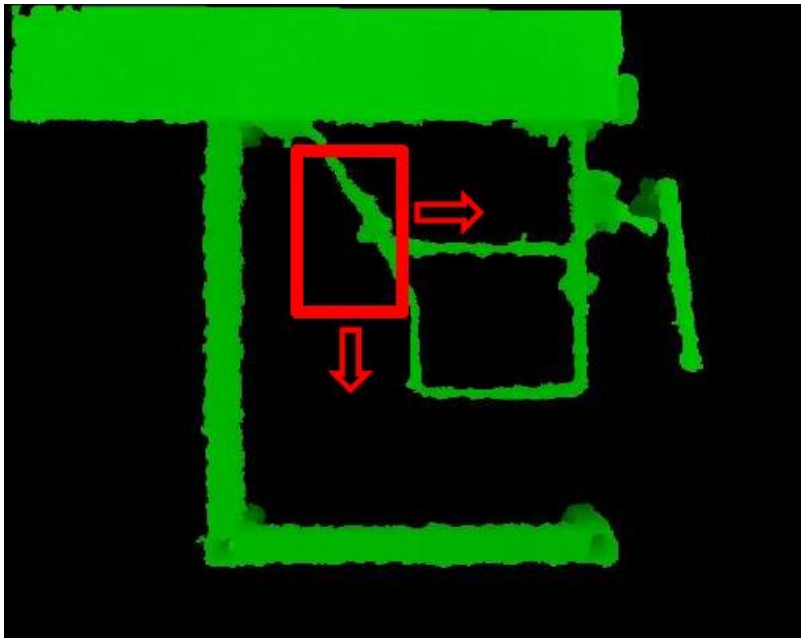


Figure 5-21: Visual representation of the sliding window algorithm

ANN Training Results

In an attempt to optimize the ANN for uncoupling lever detection, many different parameters were varied to achieve the best possible configuration using “test images” of the lever. The same set of 21 true and 21 false “test levers” were used to form an unbiased comparison across many different network topologies. Variables such as training time, predicted outputs, and expected outputs are printed to a console window and saved to evaluate the impact of each. The algorithm used for training was also tested

on both a local desktop PC and a supercomputer designed for visualization named “Longhorn” at The University of Texas at Austin to compare the difference at which each could successfully train the network. An example of the numerical results recorded after a training simulation is provided in Figure 5-22.

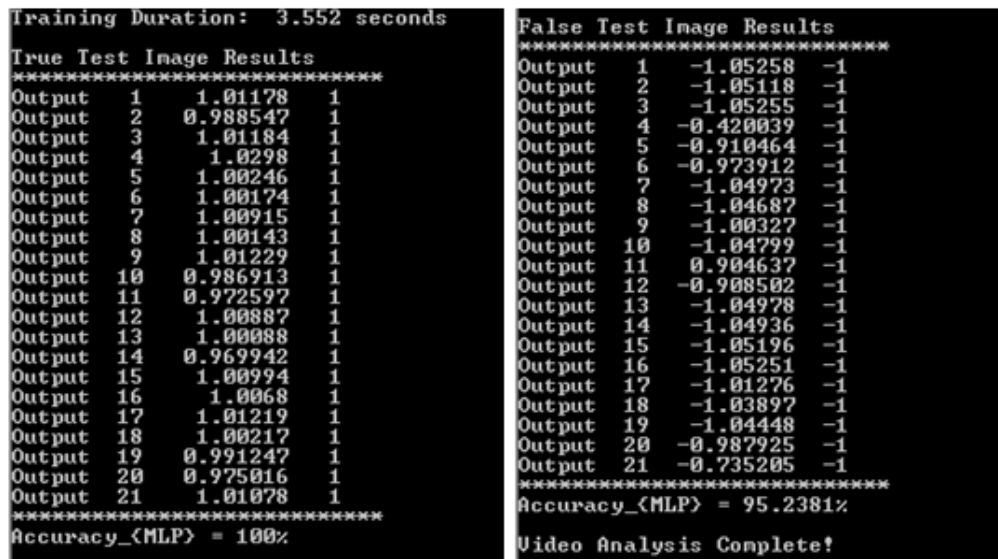


Figure 5-22: Predicted vs. actual outputs for true (left) and false (right) test images

In Figure 5-22, all of the true test images were detected correctly (i.e. all of their outputs are very close to 1). All of the false test images (with the exception of Output 11) were also detected correctly. Although the results are satisfactory, the magnitude of the prediction for Output 11 is very close to 1 (a strong positive) makes that particular instance tough to filter out using standard heuristics. A magnitude that large on the opposite side of the origin often indicates that the sample tested was not adequately represented in the training dataset. Therefore, more data corresponding to regions similar to that particular false positive should be added to the training set to eliminate future false positives.

To gain further insight into performance of every predictions, the ANN algorithm was applied to recorded videos to observe its predictions in real-time. A true detection is portrayed by red rectangular region overlaid onto the video. This provides the user with instant visual feedback into how the network is performing Two sample results yielded on separate frames extracted from the video analysis are shown in Figure 5-23.

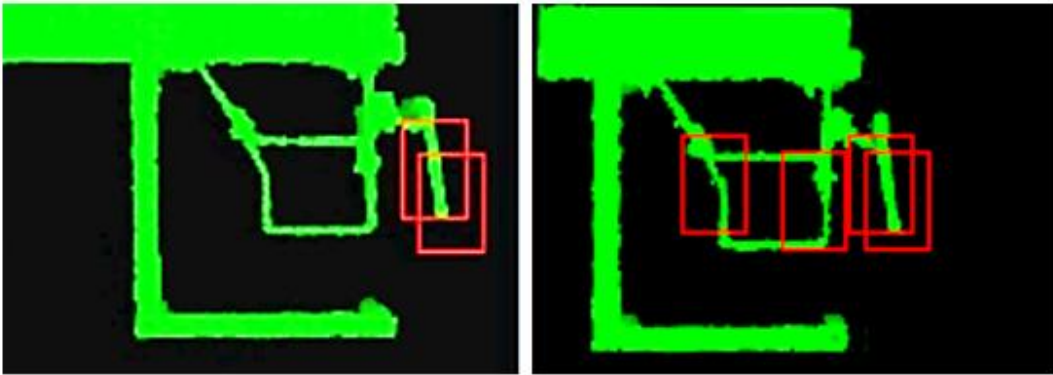


Figure 5-23: Bounding box over prediction: good (left) and false positives (right)

The left image in Figure 5-23 correctly predicted the location of the uncoupling lever handle, as indicated by the red bounding boxes in the image. However, the right image contains two false positives (also indicated by red bounding boxes) generated by the sill step ladder. This “on-the-fly” analysis tool was critical to fine tuning the network for correctly detecting rail yard appliances.

The performance metrics (Figure 5-22) are tabulated in the 2nd and 3rd columns of Table 5-1 below in addition to visual observations from the live video analysis (Figure 5-23) in the 6th column were used to tabulate the performance of the ANN. First, network configurations with one, two, and three hidden layers were tested. Within these network topologies, the number of neurons in each layer and the order of larger and smaller layers were varied.

Table 5-1: Tabulation of prediction performance using different network configurations

Num Hidden Layers (nodes)	True %	False %	Time (LH)	Time (PC)	Video
1(3)	100.00%	95.24%	0.51	1.911	Good/Great
1(5)	100.00%	95.24%	0.46	2.084	Good
1(9)	100.00%	95.24%	0.68	2.933	Good/Great
1 (15)	100.00%	95.24%	0.8	3.52	Great
1(25)	100.00%	95.24%	1.8	7.565	Good
1(50)	100.00%	95.24%	4.97	19.951	Excellent
1(75)	100.00%	90.47%	5.09	19	Bad
1(100)	100.00%	85.71%	5.93	23.448	Terrible
1(200)	100.00%	66.66%	5.84	40.072	Terrible
2(2,5)	28.57%	95.24%	0.42	-	Bad
2(5,2)	100.00%	80.95%	0.65	-	Bad
2(10,15)	100.00%	95.24%	0.9	-	Good
2(15,10)	100.00%	95.24%	2.58	-	Good
2 (15,20)	100.00%	95.24%	0.64	-	Good
2(20,15)	100.00%	95.24%	2.14	-	Excellent
2(30,50)	100.00%	95.24%	1.43	-	Good
2(50,30)	100.00%	95.24%	2.49	-	Good
2(100,150)	100.00%	95.24%	6.32	-	OK
2(150,100)	100.00%	95.24%	9.3	-	OK
3(150,100,150)	100.00%	95.24%	12.85	-	Bad
3(15,10,15)	100.00%	95.24%	0.6	-	Good/Great
3(10,15,10)	100.00%	95.24%	0.53	-	Good/Great
3(25,50,25)	100.00%	95.24%	1.26	-	Great
3(50,25,50)	100.00%	95.24%	1.12	-	Great

At first glance, the changes did not appear to have a very large effect on the predicted true and false states (Table 5-1). Although most of the percentages in 2nd and 3rd columns remained the same, the observations from the live video analysis would typically show that one configuration yielded far more false positives (Figure 5-23, right) than another. It should be noted that the percentages for true and false test images only represent a very small portion of the training dataset. Therefore, the visual observations from the live video analysis were weighted higher than the other criteria. Using all the results included in Table 5-1, a network with one hidden layer, size equal to 50 neurons,

outperformed all the other configurations. There were no false positives over the entire extent of all recorded videos and the one input that it did not correctly predict was much closer to zero than with others, meaning it could easily be rejected using simple heuristics.

In addition to the aforementioned analysis, several ANN configurations were tested on both a local desktop PC and a supercomputer to evaluate the difference in training time (Table 5-1, 4th & 5th columns). The supercomputer typically trained at a rate four to six times faster than the local PC. The reduction in training time and the increased speed at which the ANN algorithm can predict an output definitely accentuate the added computational power and parallel processing capability of the supercomputer over a standard PC. Depending on the complexity and real time requirements of a rail yard application, this type of added power might be required.

Real-time Lever Detection using ANN

Now that the ANN is adequately trained to successfully distinguish an uncoupling lever handle from its surroundings, the algorithm can be applied to a live video stream to detect the spatial location of the handle in real time. Once detected, the algorithm can track the moving handle by iteratively detecting it in successive frames. If an estimate for the target's velocity is known, the algorithm could even predict the next location, making the overall detection algorithm more robust. During tracking, the centroid of the detected handle inside the bounding box signifying a true positive is calculated frame (captured every 33ms) using methods similar to those described in Section 5.2. The real world coordinates (x,y,z) of the handle's centroid then become the target location for the robotic system to plan a trajectory to approach and grasp. The detected uncoupling lever handle on a live video stream during a successful laboratory test is shown below in Figure 5-24.

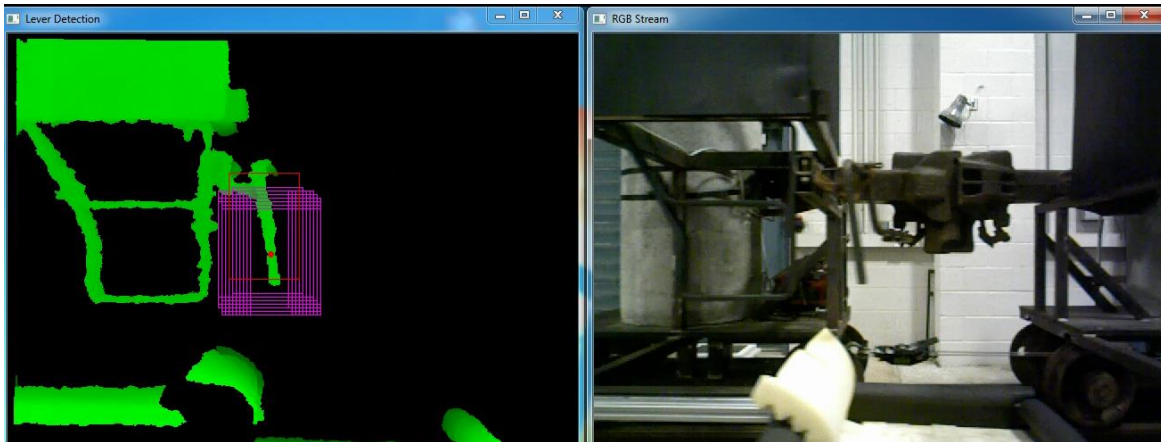


Figure 5-24: Uncoupling lever detected on UT Austin test bed during successful uncoupling procedure

Object Detection & Recognition Summary

The primary goal of this chapter was to develop a framework for a flexible, object detection algorithm to find railroad safety appliances. Image processing and machine learning techniques are tested and applied specifically to find uncoupling levers on freight cars. Geometric-based methods were successful in finding the lever in both 2D and 3D images, but struggled with unacceptable numbers of false positives. Corrections to alleviate these false detections were implemented offline, which could be a useful tool for rail yard applications with static objects of interest, but does not meet the real-time requirement for the control of an autonomous robotic system tasked with uncoupling freight cars. Therefore an object oriented Artificial Neural Network was implemented to find patterns in live video of passing freight trains to successfully find the moving lever in less than 35ms. The ANN has proven to be an acceptable method for autonomously detecting uncoupling levers on freight cars in operational classification yard environments and laboratory tests. Based on the ANN's success for this particular application, the RRG believes that it could also be used to detect pneumatic brake hoses and hand brakes to further increase yard safety through the automation of these tasks.

CHAPTER 6: ROBOTIC SYSTEM DESIGN & IMPLEMENTATION

Chapter One identified several laborious and hazardous tasks that are currently performed manually in rail yards throughout the country. Chapter Two surveyed machine vision and robotic systems currently used for railroad applications. It also examined the strengths and weaknesses of patents for conceptual designs of automated systems for rail yard applications. Chapter Three discussed standard and custom techniques for automated object detection using vision and robotic path planning. Chapter Four provided an overview of commercially available vision, robotic manipulator, and end-effector hardware that could be used in a variety of systems for rail yard applications. Chapter 5 summarized the results from the application of the object detection techniques covered in Chapter Three to gather and process data required by a robotic system tasked to operate in a rail yard. This chapter proposes multiple designs for an Autonomous Pin-Pulling System (APPS) to uncouple moving freight cars in classification yards. APPS has been prototyped and rigorously tested in a lab environment with full scale hardware. Although APPS is designed to solve a specific task, its subcomponents can be used to design automated solutions for other rail yard tasks, such as the ones covered in Chapter 1. The details of APPS and its subcomponents are provided in the following sections.

6.1 UNIVERSITY OF TEXAS RAIL TEST BED

To build and demonstrate a robotic system capable of pulling uncoupling levers autonomously, a rolling stock test bed was required. In an effort to conserve space and weight in the laboratory, wheelsets with a 24" gauge, 14" outer diameter, and 17" flange diameter from old mining carts were acquired. The RRG successfully installed two 80' lengths of AREA 112 rail with a 24" gauge using custom clips and bolts to securely mount the track to the lab floor. Next, the RRG successfully designed, fabricated, and

installed a pair of chassis' to mate with the small wheelsets. Each chassis must support one full scale Type E coupler, a large concrete counterweight for the coupler assembly, two walls representing the corner of a freight car, one uncoupling lever, and one side sill ladder. Together these items weigh two tons, which brings the total weight of each "freight car" to two and a half tons. The final configuration for the custom designed rolling stock test bed can be seen in Figure 6-1 below.

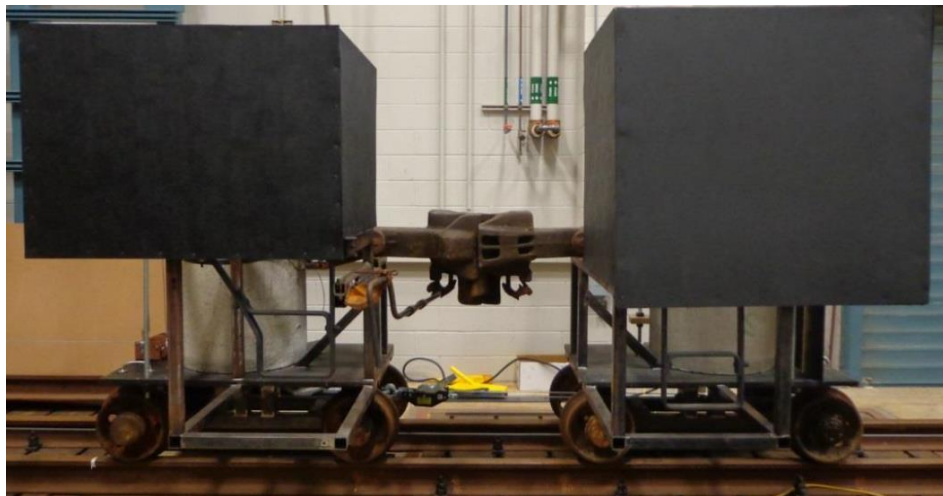


Figure 6-1: Rolling stock test bed for railroad robotics research

Three main requirements for the design of the test stock were: (1) Safe and reliable operation, (2) Ability to quickly swap the uncoupling lever and sill step types to test multiple configurations, and (3) Maintain realistic relative positions of the uncoupling lever and sill step. Only two bolts must be loosened and retightened to change either the uncoupling lever or sill step in Figure 6-1. This swap can typically be performed in less than three minutes. The relative position of the two safety appliances is important because portions of the sill step often closely resemble the uncoupling lever handle and thus can produce false positives that must be handled appropriately. Therefore, to develop a robust vision algorithm in a laboratory environment that will also

work in the field, a realistic test bed is required. Although the test stock in Figure 6-1 do not closely resemble real freight cars, the uncoupling lever assembly and metal components in its vicinity look nearly identical to that found in an actual classification yard. This is shown in the resulting depth maps corresponding to a freight car and a classification yard the test stock in the lab can be seen in Figure 6-2 below.

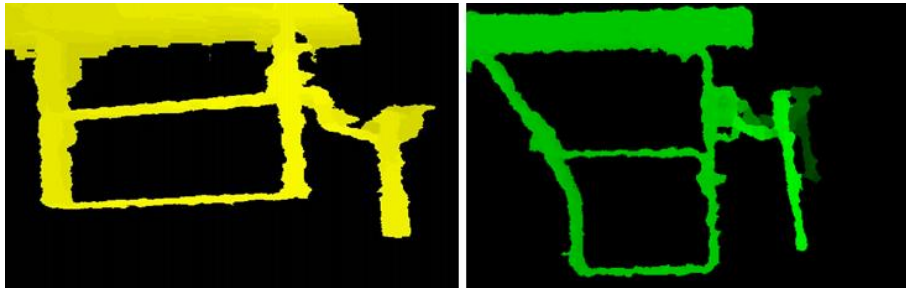


Figure 6-2: Comparison of 3D vision data from rail yard (left) and lab (right)

The rolling stock test bed is actuated using an electric winch with a variable frequency drive shown below in Figure 6-3. This winch can safely move the one or both pieces of rolling stock at any speed between 0.10 and 2.25 mph using a voltage reference provided by the operator. The voltage reference to the system can be provided manually using a potentiometer located on the control panel or via software using custom applications developed using National Instrument's LabVIEW software.

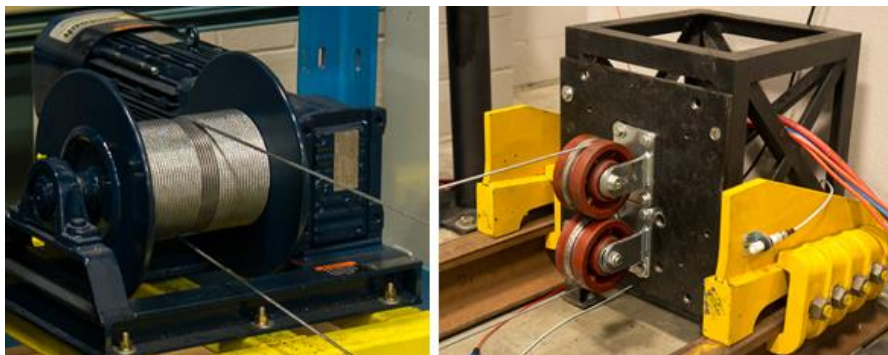


Figure 6-3: Electric winch (left) and wheel stops (right)

Custom wheel stops designed and fabricated to fit the outer diameter of the small wheelsets are installed at both ends of the rail to prevent the test stock from derailing due to over-travel. One of the carts in the rolling stock test bed is connected to the winch using galvanized 3/16" diameter wire rope with a breaking strength of 4,200 pounds. Two separate strands of wire rope are wrapped around the drum of the winch. One strand makes a direct connection from the top of the drum to the rear of the cart. The other strand is routed underneath the cart and through a pair of roller fairleads in series (see Figure 6-3, right) to connect to a ratchet puller on the coupler side of the cart. The ratchet puller is used to properly tension the wire rope to prepare the test bed for actuation. A basic schematic for the routing of the wire cable to achieve the push-pull winch configuration is shown below in Figure 6-4.

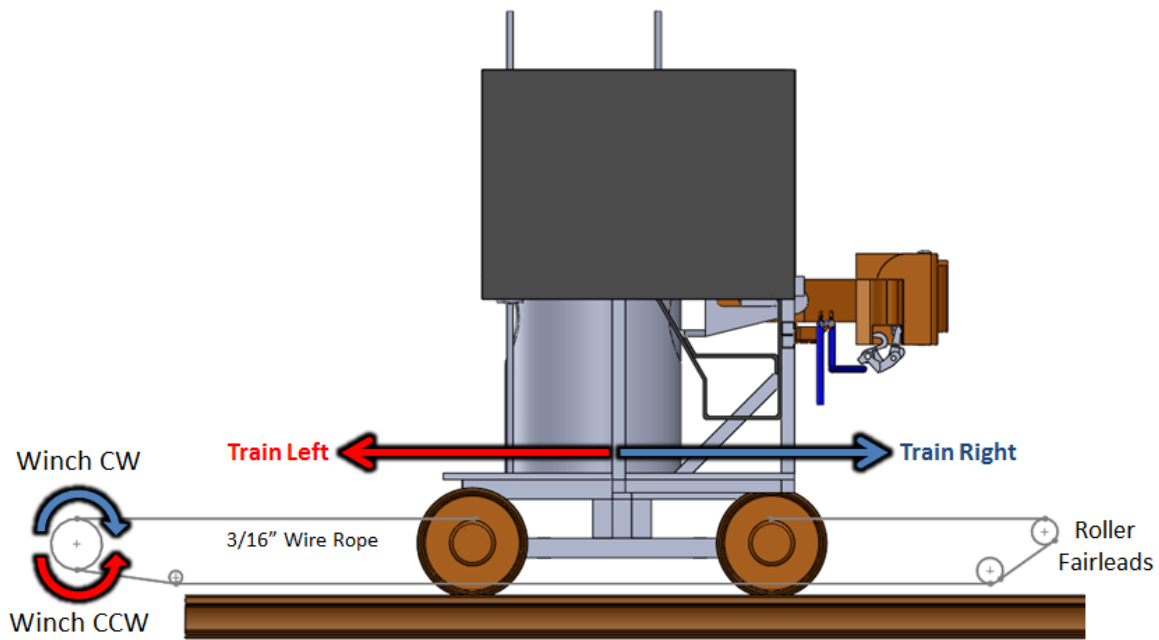


Figure 6-4: Custom push-pull configuration for winch to actuate rail test bed

The schematic in Figure 6.4 illustrates that if the winch is commanded to turn in the clockwise direction, the train will go to the left. Vice versa, if the drum of the winch turns counter-clockwise, the train will move right. The final feature of the rail test bed that should be mentioned is the installation and operation of a constant drag brake (Figure 6-5) applied to one of the wheels on the non-actuated cart (Figure 6-6, cart on right).

To adequately test an autonomous uncoupling system, the couplers need to be able to separate. In a hump yard, car separation is achieved when the center of gravity of the lead car passes over the hump crest and a pair of couplers transitions from compression to tension while the uncoupling lever is pulled. However, the flatness of the lab environment, only a “kicking” type scenario (described in Chapter 1) would cause the cars to separate. Due to limited lab space and the hazards associated with “kicking,” it was decided that the spring loaded brake shown in Figure 6-5 could provide enough drag to leave the passive cart behind once the uncoupling lever is pulled. This brake was carefully designed by a member of the RRG to achieve coupler separation.

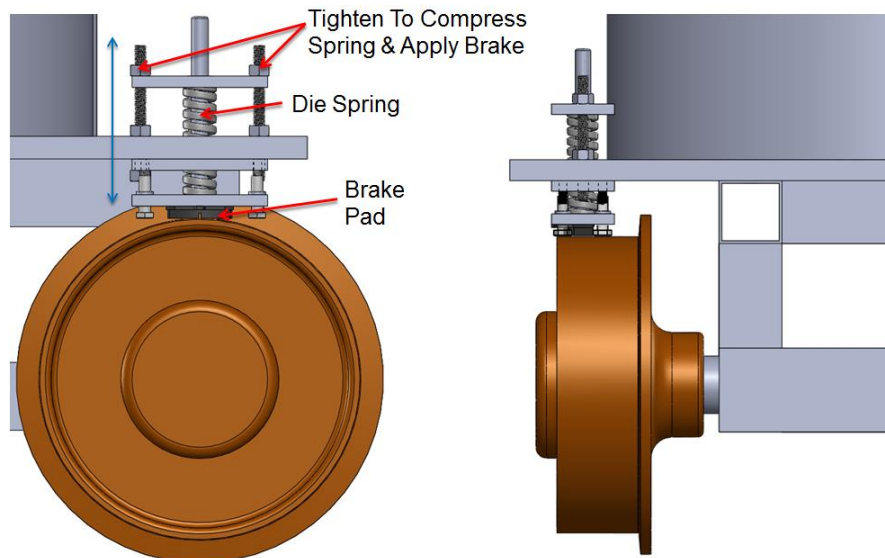


Figure 6-5: Custom brake on non-actuated cart

6.2 UNCOUPLING DEMONSTRATION

To successfully demonstrate uncoupling with an autonomous robotic system, a winch attached to the left cart in Figure 6-6 will move the coupled pair from the right end of the track to the left. The rolling stock ramp up to a near constant velocity, maintain that velocity for a set length, and then ramp down to a stop. The blue uncoupling lever handle is hanging vertically at the start of the test signifying that the carts are coupled.

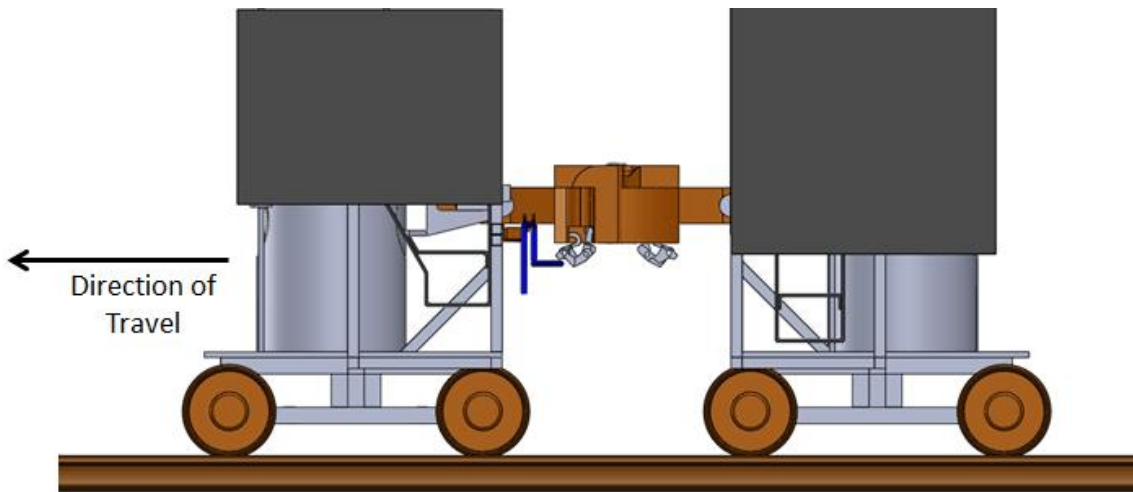


Figure 6-6: Rail test bed – uncoupling lever handle vertical - coupled

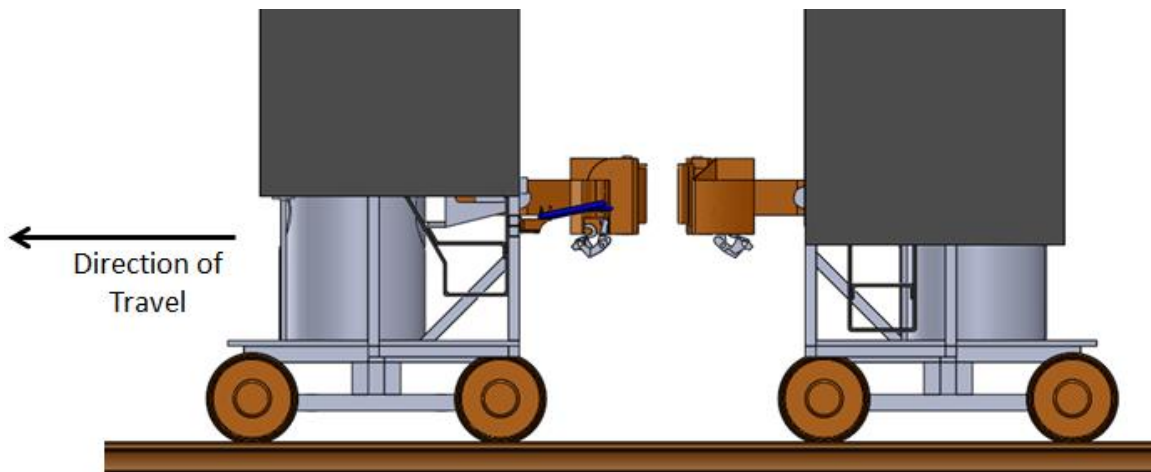


Figure 6-7: Rail test bed – uncoupling lever handle horizontal - uncoupled

At some point during the constant velocity phase of the demonstration, the autonomous robotic uncoupling system must enter the area between the two moving cars to lift the lever handle. Figure 6-7 illustrates the desired result of a successful test where the uncoupling lever has been lifted and the driven cart has left the passive cart behind.

6.3 ROBOTIC TEST BED

The robotic design for APPS consists of a six degree of freedom (DOF) industrial manipulator attached to the carriage of a floor mounted linear axis. The industrial arm is manufactured by Yaskawa Motoman (Model: MH80) with a payload capacity of 80kg (180lbs) and a horizontal reach slightly greater than 2m (6.75ft) [MH80, 2013]. The linear axis is manufactured by Güdel (Model: TMF-52) with a rack and pinion drive and a stroke of nearly 12m (40ft). The robot and linear axis were integrated by Motoman and delivered to the RRG. The integrated, seven DOF system is shown below in Figure 6-8.

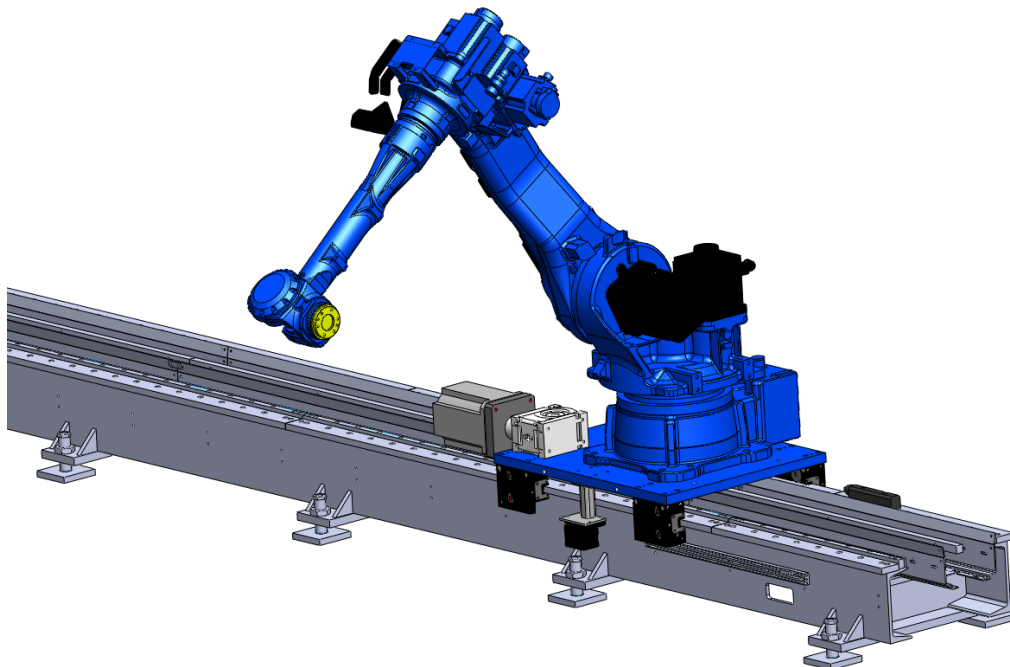


Figure 6-8: Initial platform | Motoman MH80 on a Güdel floor mounted linear axis

6.3.1 Robotic Arm

The MH80 was selected based on its payload capacity, degrees of freedom, working volume, and repeatability. To determine the required payload for uncoupling, hundreds of test pulls were measured using a handheld digital force gauge that logs data at 100Hz. The average force required for all handle pulls sampled was just over 60 pounds-force. However, maximum forces up to 400N (90lbf) were not uncommon. The weight of the end-effector described later in this chapter was also accounted for. These end-effector configurations with sensors, tooling, and grippers were projected to weigh roughly 20kg (44lbs). Combining the maximum pull force needed with the projected weight of the end-effector comes to ~625N (140lbf). Thus, the 80kg payload of the MH80 provides the required force needed to pull uncoupling lever handles with some buffer. The working volume for the MH80 (left) and samples of the uncoupling lever pull forces (right) are shown in Figure 6-9.

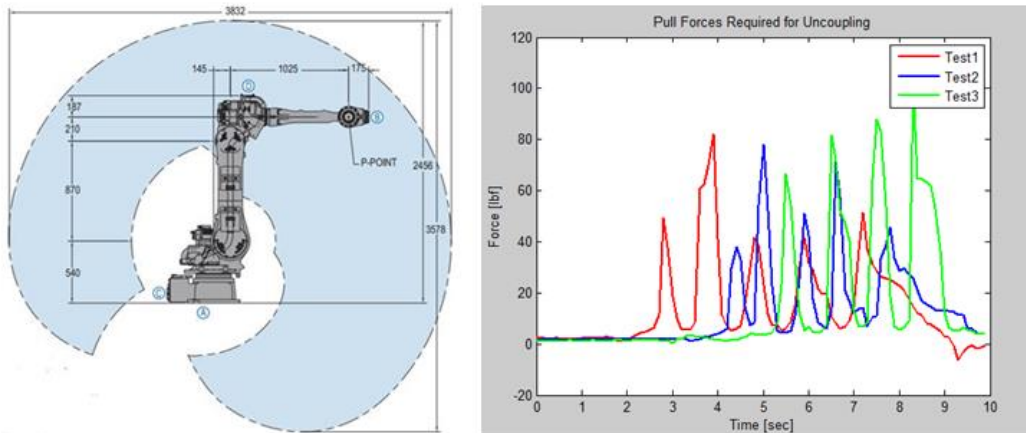


Figure 6-9: Working volume for MH80 manipulator and sample lever pull forces

The working volume of the MH80 was also attractive for automatic uncoupling. With just over a 2m (6.7ft) horizontal reach, the robot's base can be mounted a sufficient distance from the rolling stock test bed to give proper clearance between the two systems.

This clearance is important to the both the mobility/dexterity of the robot during operation and preserving the capability to uncouple cars manually during periods of maintenance or downtime.

6.3.2 Motorized Linear Axis

The principle mechanical components of the Güdel TMF-52 floor mounted linear axis (shown in Figure 6-10) are a base frame and carriage. The base frame is constructed from a structural steel C-channel and steel plate weldment with guideway flat bar rails and independent racks [Güdel, 2012]. The carriage rides along the base frame on the guideway rails via four roller blocks mounted to the carriage and the load is transferred along the base frame using a rack and pinion drive system [Güdel, 2012]. The high precision of this particular system allows for strokes up to 13.67 meters with velocities up to 1.4m/s, accelerations up to 1.4m/s^2 and a repeatability of no more than $\pm 0.15\text{mm}$. Standard rubber shock absorbers are used at both ends of the track to prevent the carriage from over travel. The unit weighs $\sim 2000\text{kg}$ and has a maximum payload of 690kg (robot weighs 555kg). The carriage is driven by an AC servo in series with a gearbox that has a 5:1 reduction ratio.



Figure 6-10: Güdel floor mounted linear axis

6.3.3 RRG Modifications & Additions

In order to transform the off-the-shelf tracked robot into an autonomous pin pulling system, an extension to the carriage was required to accommodate sensors, data acquisition modules, power supplies, and onboard computing. Two additional roller blocks were installed and connected to the original carriage plate via an extension plate with accompanying joint bars. A custom structure was then designed and assembled using extruded aluminum to mount vision sensors and data acquisition components. This structure has a reconfigurable mount for the cameras and can quickly be adjusted along the three components of a Cartesian axis. The SICK laser rangefinder is mounted directly beneath the waist axis of the robot and can be used for collision avoidance to position the arm reliably between two freight cars. These components and features of the carriage extension are shown below in Figure 6-11.

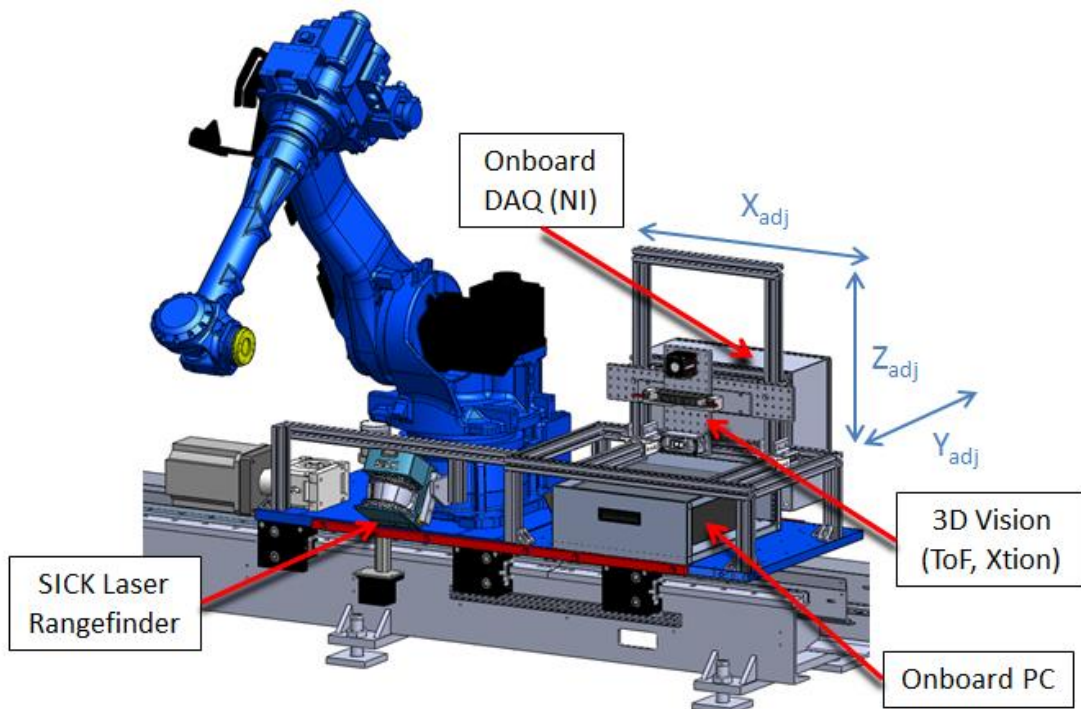


Figure 6-11: Carriage plate extension and sensor mounting

6.3.4 Robot Controllers

Industrial robots typically come with traditional robot controllers that are extremely robust, mature, and safe. Typically, operators interface with these controllers using a teach pendant to program robots by jogging to and saving points in space to form the required trajectories. This method of programming work well for highly repetitive manufacturing tasks in controlled environments, but cannot be used to program robots where the task, objects of interest, and/or environment are not predefined. These types of task require powerful C++ libraries and auxiliary sensing that cannot be connected directly to a standard industrial controller. Methods that are available often do not have the required level of determinism or control rates required by cutting-edge research tasks.

In order to successfully build an autonomous robotic system that can safely and reliably interact with another large moving body (e.g. uncoupling freight cars), cutting-edge sensors and computer vision systems must be intelligently integrated with an industrial robotic manipulator using a fast, deterministic, open-architecture controller. Therefore, a custom 3rd party controller from Agile Planet Inc. capable of communicating with a standard desktop PC was acquired. This controller utilizes high performance Yaskawa Servo Packs to control each axis of the robotic system at rates up to 1000Hz. In addition to high performance motion control, these servo packs have digital inputs and outputs that can be used to control safety appliances, sensors, and actuators. The control panel communicates over EtherCAT with a control application deployed on desktop PC with Windows CE real-time operating system installed. The standard DX100 controller provided by Motoman and the Agile Planet AX-I3 controller are both shown in Figure 6-12 below.

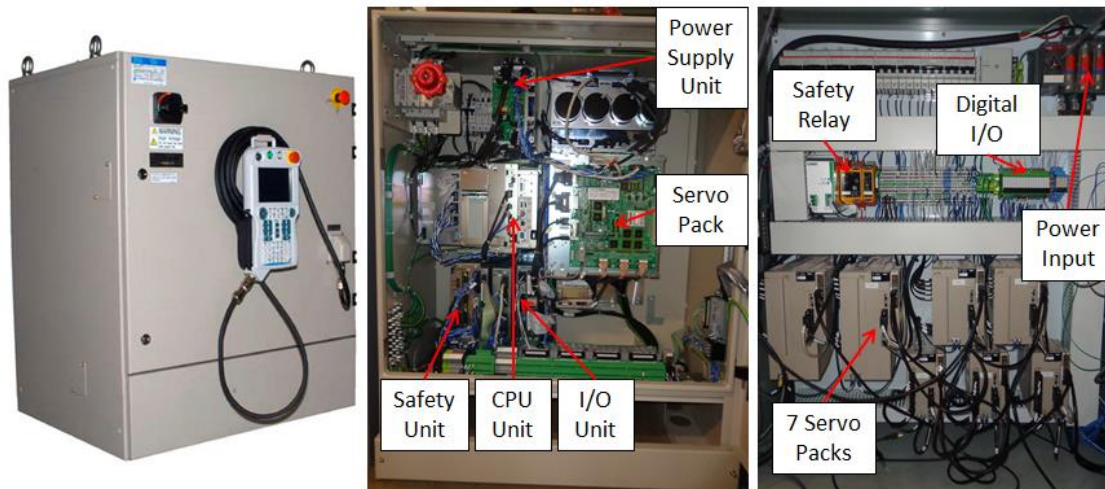


Figure 6-12: Motoman DX100 with teach pendant (left), internal components of DX100 (middle), & internal components of Agile Planet AX-I3 controller (right)

Externally, the two controllers in Figure 6-12 do not appear to be very different, with the exception of DX100’s teach pendant. Both are sturdy metal enclosures with an ON/OFF switch, E-Stop button, and servo packs for motion control. However, the inside of Motoman’s DX100 controller has many more components than the AX-I3, including a Safety, CPU, I/O, CPS, and Power Supply Unit as evident above in Figure 6-12 above. Many of these units have embedded circuit boards responsible for processing data, making the DX100 a standalone unit, whereas the AX-I3 controller does not contain any processing capacity for robotic motion planning.

6.3.5 Software

Robotic Software

To generate trajectories and send motion commands to the AX-I3 controller, Agile Planet provides a C++ software library named AX-SDK that utilizes Kinematix technology initially developed in the RRG as the Operational Software Components for Advanced Robotics (OSCAR) [Kapoor, 1998]. The commercially available AX-SDK

provides robust kinematic and dynamic control capabilities for a wide variety of robot geometries. Applications are developed using Visual Studio C++ in Windows 7 and deployed on Windows CE real-time target. CeWin, a former Kuka product, allows Windows CE to be installed together with Windows on the same machine, is used to manage the two operating systems while maintaining Windows CE's determinism. The control application communicates with the AX-I3 controller over EtherCAT, analog & digital sensors connected to an NI CompactRIO over TCP/IP, and vision systems connected to an on-board PC over TCP/IP to gather data critical to the uncoupling operation.

Computer Vision Software

Computer vision software and hardware was explored to autonomously detect uncoupling levers on slow moving freight cars. Geometric image processing algorithms and machine learning techniques were tested using MATLAB's Image Processing and Computer Vision libraries to develop algorithms capable of accurately detecting three types of uncoupling levers in both laboratory and yard environments. Successful methods were then optimized using the aforementioned OpenCV C++ library for increased reliability and speed. The final algorithm built around an artificial neural network (described in the Chapter 5) is capable of autonomously detecting and tracking a moving uncoupling lever at 30 FPS using the ASUS Xtion Pro Live 3D camera.

6.3.6 End-Effector Development

APPS End Effector 1: J-Hook Design

Multiple end-effector designs and configurations were developed for autonomously pulling the uncoupling lever. The first design utilizes a collision sensor for passive compliance, six-axis load cell for monitoring forces and torques about the

Cartesian axis, modified tool changer as a “fall-off” mechanism, and custom gripper designed and fabricated by the RRG and associates to grasp and pull the uncoupling lever handle. This particular end-effector configuration is shown below in Figure 6-13.

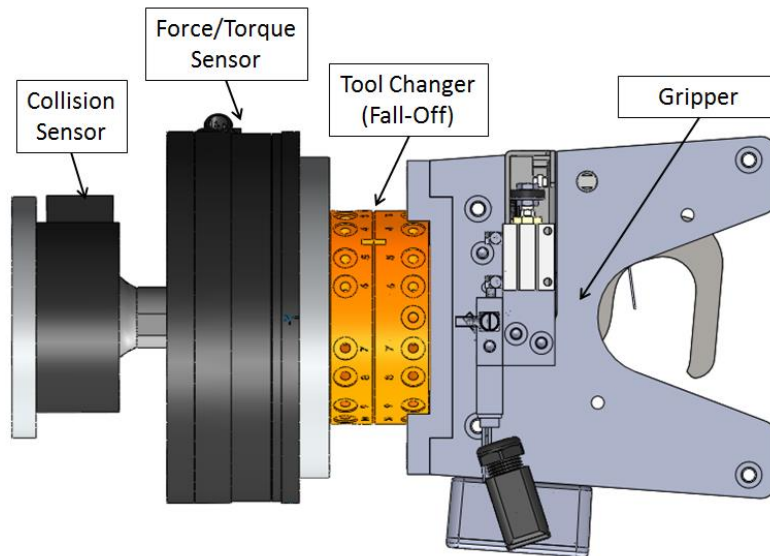


Figure 6-13: Custom J-Hook gripper and end-effector designed to pull lever

The robot connects to the end-effector via an interface plate on the left side of Figure 6-13. The interface plate is bolted to a collision sensor that is primarily used as a compliance device. Coordinating the motion of two moving bodies is a difficult task and the sensor data controlling the system will definitely contain some level of error and uncertainty. Therefore, the compliance provided by the collision sensor will allow the end effector to *passively deform* in the presence of an unexpected disturbance, or if significant position error exists between the commanded position of the robotic system and the instantaneous position of the train. The collision sensor is directly connected to a large ATI Omega 160 force/torque sensor responsible for monitoring forces up to 600lbf and moments as large as 3600lbf-in applied to the end-effector at rates up to 1000Hz. This

data can be used to retract the robot safely if the disturbances exceed a user-defined threshold or for *active compliance*. This process will be described in further detail below. The force/torque sensor is then connected to a robotic tool changer. Robotic tool changers are composed of two parts: a master plate and a tool plate. A schematic for an example tool changer is shown below in Figure 6-14.

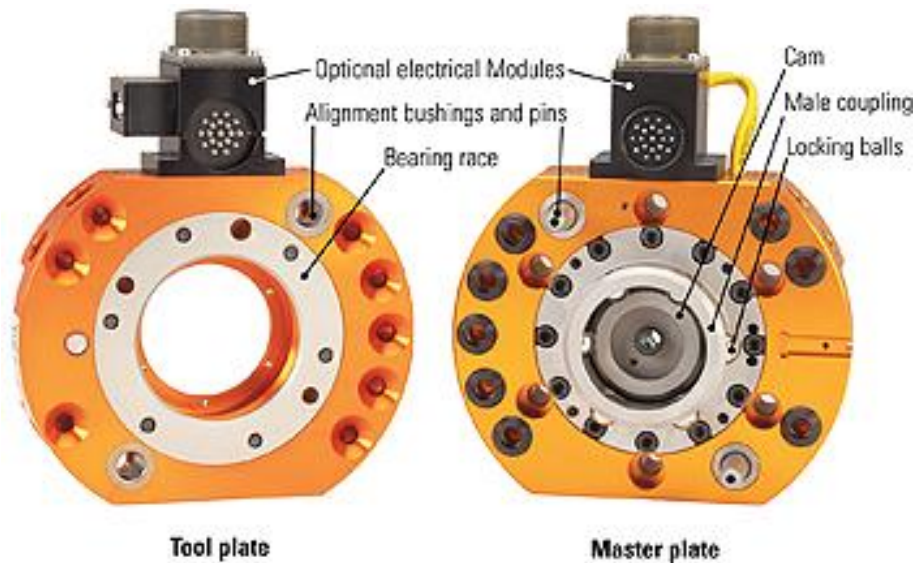


Figure 6-14: Robotic tool changer schematic [ATI-2, 2013]

The two plates shown in Figure 6-14 are pneumatically joined using a commanded pressurized input to unit to enable a rigid connection for changing tools autonomously. Each plate has pneumatic and electrical “pass-throughs” that connect and sever automatically during assembly and disassembly respectively. The bearing race and cam of an off-the-shelf tool changer were altered to create a “fall-off” mechanism, allowing the end-effector to passively, mechanically separate in the event of a large unexpected disturbance. This would allow the gripper to disconnect itself the end effector in order to preserve the robotic arm, the most expensive component in the system. This evasive maneuver can either be commanded via the firing of a digital output signal to a

pneumatic solenoid valve or occur passively due the removal and modification of the standard internal locking feature.

Finally, a custom gripper was designed to grasp and pull the uncoupling lever handle. The gripper has a hinged internal hook that is designed to passively rotate as it makes contact with a cylindrical object until it is secured in an off-the-shelf latch designed into the gripper body. During the pin pulling operation, the hook is designed to make contact with the handle in the open position (Figure 6-15, left) and passively rotate clockwise to the closed position (Figure 6-15, right). This process can be seen in Figure 6-15 below where the gray circle represents the 1” diameter of the uncoupling lever handle.

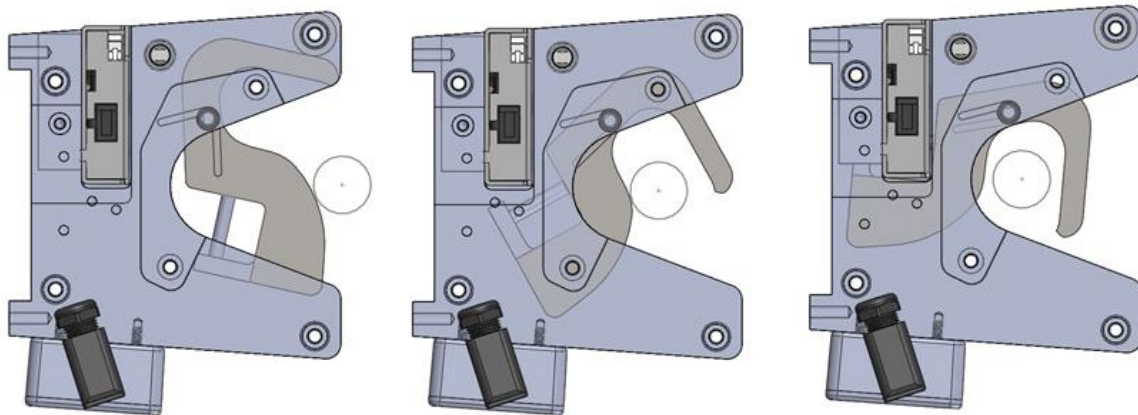


Figure 6-15: Operation and passive actuation of J-Hook gripper as it contacts handle

As evident in Figure 6-15 above, the closed position of the gripper is not designed to rigidly clutch the cylindrical handle. A tight grasp is not required to successfully pull the lever handle and the clearance shown above helps account for error inherent to the calculated position of the object of interest due to sensor inaccuracies. The U-Shape cutout in the housing also provides relief if the robot is slightly misaligned during its approach sequence (covered in detail later in this section). Finally, this gripper was

designed to release in less than 30ms using high performance pneumatic actuators and solenoid valves to quickly disengage in the event of a detected problem or failure.

APPS End-Effector 2: Rotary Design

During testing with J-Hook gripper design, it was discovered that a perfect circle could indeed pull the handle due to the compliance in the uncoupling lever assembly. The motion of the handle is not circular, but the “play” in the physical structure and a loose grasp, a circular trajectory would accomplish the same end result: successful uncoupling. Therefore, a second end-effector was designed by the RRG to uncouple freight cars autonomously using a fast and efficient circular trajectory. This design uses a combination of a pneumatic linear and rotary actuator to enter the area between two freight cars and operate the uncoupling lever. A schematic for this end-effector is shown below in Figure 6-16.

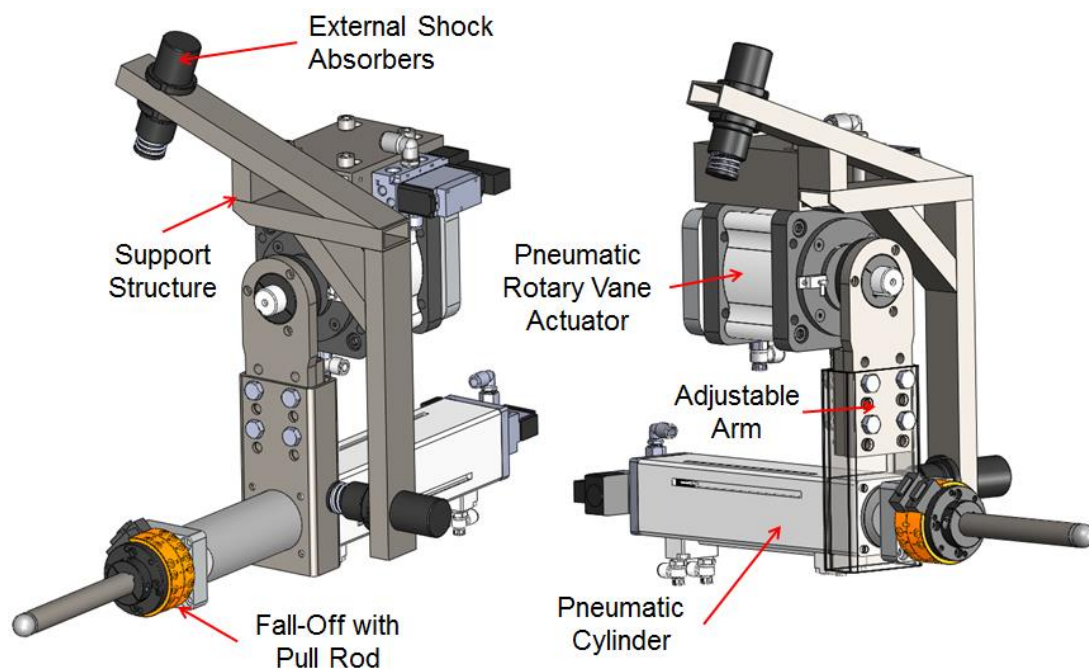


Figure 6-16: Custom rotary end-effector designed to pull lever

An interface plate is used to connect an off-the-shelf pneumatic rotary-vane actuator to the robot. This rotary actuator has a 110° stroke and is driven by a pneumatic solenoid valve. A custom, adjustable arm was designed and mounted to the output shaft of the rotary vane actuator. A large, robust pneumatic cylinder designed to resist substantial moments is attached to the end of the adjustable arm. This cylinder has a 6in stroke and is controlled by another pneumatic solenoid valve. The end of the cylinder is connected to the same “fall-off” mechanism (adapted from a robotic tool changer) described in the previous section. Finally, the tool plate of the “fall-off” is connected to a steel rod with a rounded end. This completes the design of the moving parts of the end-effector designed to rotate the uncoupling lever handle, the operation of which is shown below in Figure 6-17.

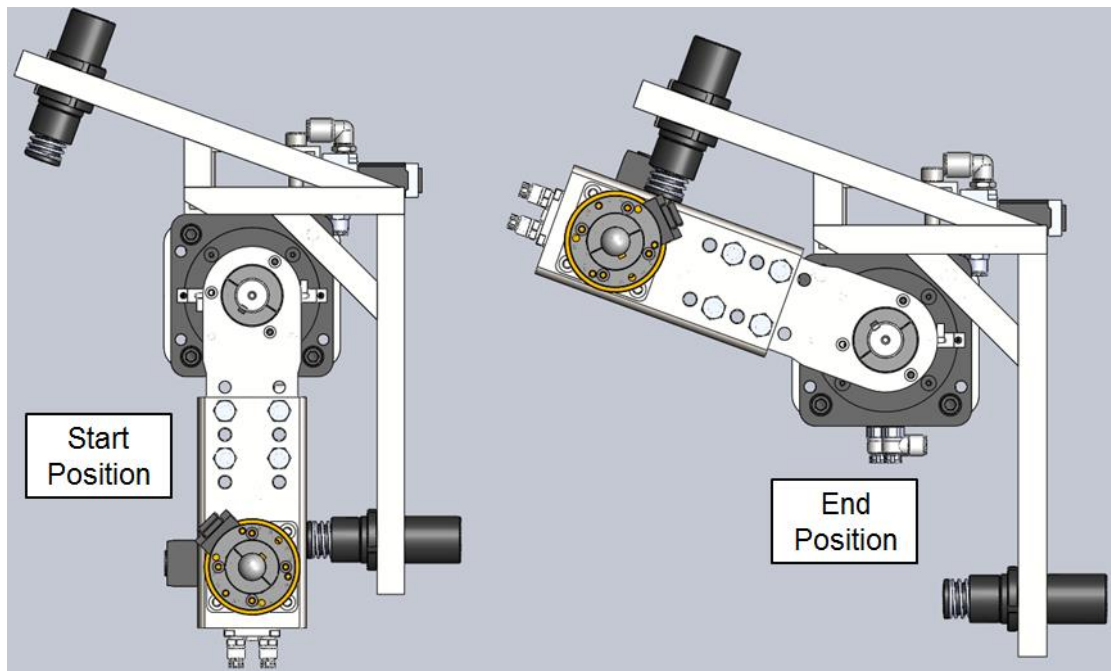


Figure 6-17: Start and end positions of the custom rotary end-effector

This rotary end-effector is designed to make contact with the lever handle using the steel rod shown at the end of the cylinder in Figure 6-16. First, the robot will align the end-effector with the moving lever handle and command the pneumatic cylinder to extend. Once contact between the rod and handle is achieved, the system will command the rotary actuator to rotate from its start to its end position, as shown in Figure 6-17, with sufficient force and momentum to quickly pull the uncoupling lever. The auxiliary L-shaped metal structure also shown in Figure 6-17 was designed with heavy-duty shock absorbers to dampen the load of the swinging arm during operation to protect the robot and its accessories.

6.4 APPS OPERATION

This section will describe the operation of APPS for both end-effector configurations. Since uncoupling can only be performed while the train is in motion, the first task that APPS must complete is position and velocity synchronization with the moving train. As the train begins to pass the static robot, a wheel detector (a railroad specific proximity sensor) mounted on the rail that monitors the instantaneous position and velocity of wheels on freight cars is used to trigger the robot tracking algorithm. The robot tracking algorithm initializes the motorized linear track axis to accelerate up to the same velocity that the train is running at. Subsequent wheel detectors readings will be used to provide the robot with the train's most recent velocity. Six wheel detectors mounted equidistantly along the RRG's rolling stock test bed provide a new velocity reading approximately every 1.5 seconds if the train is traveling at 1.5 mph. These instantaneous measurements of velocity are very accurate, but since they do not provide a continuous stream of data, they cannot be the sole source of information used to reliably sense the train's velocity in real-time.

6.4.1 Robot/Train Synchronization with Vision

To synchronize the moving robot with the moving train, an onboard vision system is used to keep track of the train position between consecutive wheel sensor readings. A depth map is created using the ASUS Xtion Pro Live 3D sensor using the same methods discussed previously in Chapter 5. Prominent geometric features on the moving freight car are then identified using image processing techniques e.g. Canny Edge detection. The same feature is then continuously detected and tracked in every subsequent frame to update the position of the train with respect to the robot at a rate of 30Hz. Using the error in position between the robot and the train, a “correction velocity” can be calculated and added to the baseline velocity provided by the wheel detector to either command the robot’s motorized linear axis to speed up or slow down to maintain position tracking. A visual for position tracking process during a laboratory test is shown in Figure 6-18.

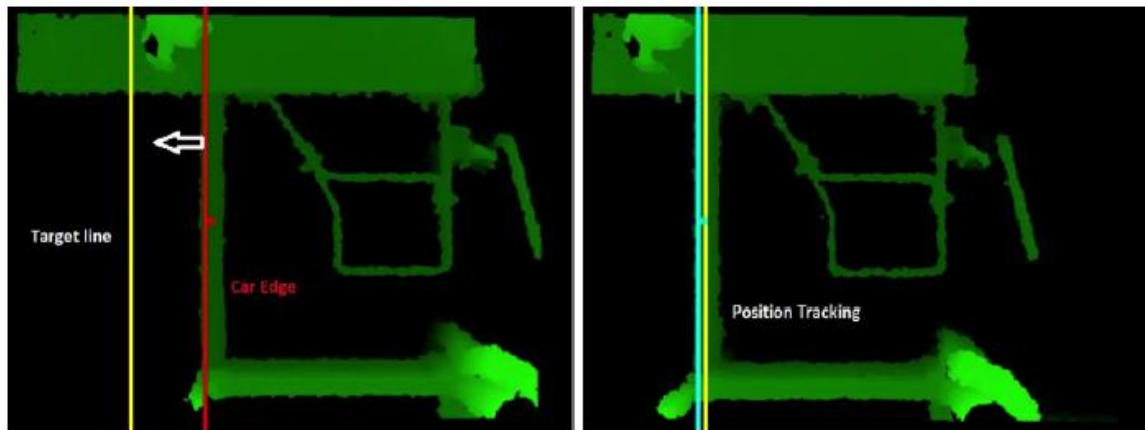


Figure 6-18: Position tracking using vision: offset error (left) and position tracking (right)

The red line in the left image of Figure 6-18 represents the prominent feature on the freight car that was found by the vision algorithm and the yellow line denotes the target position of the robot in image coordinates. Ideally, these two lines should overlap to synchronize properly, as seen in the right image of Figure 6-18. In this image, the

position of the robot and prominent feature on the freight car are correctly aligned to maintain synchronization throughout the uncoupling procedure. This combination of feedback from both wheel detectors and vision systems has proven to work reliably over hundreds of tests and eliminates the need of continuous measurements by an expensive, wayside Doppler radar system. Additionally, the use of vision for robot-train allows APPS to be a self-contained unit that can easily be deployed in a rail yard.

6.4.2 Sequence of Events for APPS Operation

Once the train and robotic system are properly synchronized, APPS can begin the lever pull sequence to uncouple the freight cars autonomously. An event-driven control application commands the robot to approach, grasp, and pull the uncoupling lever handle based on data received from wheel detectors, 3D cameras, and sensors on the end effector. This application is also capable of handling many different fault scenarios. The following sequence of events is executed for each pull of an uncoupling lever:

1. The robotic system waits for a wheel detector trigger to identify the next freight car to uncouple in the moving train. The wheel detector trigger initializes velocity and position synchronization between train and the robotic platform.
2. The track axis of the robotic platform starts moving with the instantaneous velocity recorded at the first of six wheel detectors. As the robot begins to move with approximately the same speed as the train, the vision system on the robotic platform identifies and tracks the most consistent feature present on a given freight car (e.g. prominent vertical edge). The algorithm “locks-on” and uses the spatial coordinates of the prominent feature for position and velocity synchronization. Given the position error between the robot and the rolling

stock, the control algorithm adds or subtracts velocity from future jog commands for the motorized linear axis to minimize the relative distance between the two objects. This method provides accurate position tracking to 15mm (typ.) using feedback from the ASUS vision system.

3. Next, the vision system identifies the uncoupling lever in the frames captured by the 3D camera using image processing and machine learning techniques (see Section 5.3.2). Once detected, the vision application sends the spatial position and orientation of the detected lever handle to the robot control program.
4. Upon receiving the spatial location of the lever, the robot is commanded to align the end-effector with the lever handle. Once the robotic arm has properly aligned the end-effector, an approach sequence is initiated and executed until the end-effector makes contact with the lever handle. **Step 4 varies for the two configurations for APPS, see Sections 6.4.3 (J-Hook) and 6.4.4 (Rotary)
5. Once contact with the lever is achieved, APPS pulls the lever handle about its center of rotation. APPS uses sensor data to determine whether or not the pull attempt successfully uncouples a pair freight cars. If the cars are not uncoupled, APPS will attempt to repeat the pull until uncoupling is achieved. **Step 5 varies for the two configurations for APPS, see Sections 6.4.3 (J-Hook) and 6.4.4 (Rotary)
6. Upon successful completion of the lever pull, robot releases the handle and is commanded to retract towards its base. The robot/train synchronization operation is terminated, and the robotic platform moves quickly back to its “home” position to prepare for the next lever pull.

6.4.3 Align & Pull Sequence: J-Hook Design

To properly align the J-Hook gripper with the lever handle, the robot must reach in between two moving freight cars to align the center of the gripper's U-shaped channel with the center of the lever handle. APPS uses 3D vision to find the approximate center of the gap between the two cars to safely reach in between without hitting any obstacles. The align sequence using the J-Hook gripper is shown below in Figure 6-19.

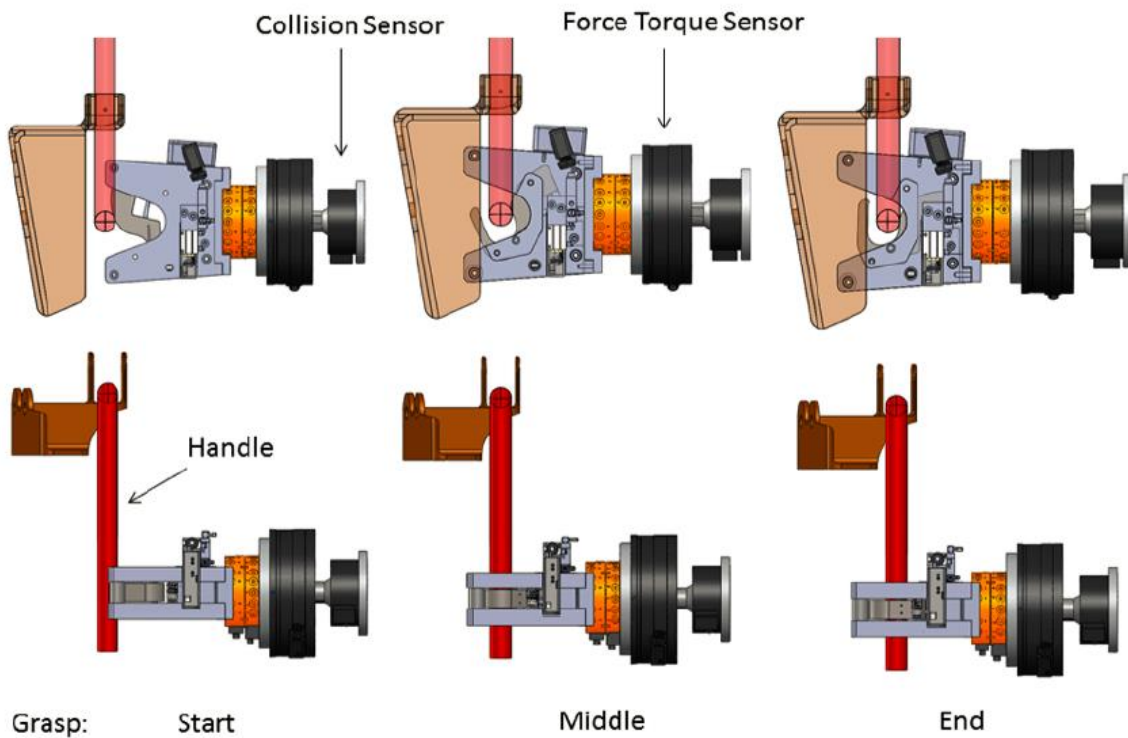


Figure 6-19: Align sequence with J-Hook gripper

Once the robot has aligned with the approximate position of the lever handle, the off-the-shelf latch assembly in the J-Hook gripper is released (allowing the hook to swing to the open position) and the robot linearly approaches the lever handle at 100mm/s using the six axes of the arm. As the gripper makes contact with the lever handle (see Figure 6-19), it passively latches and sends a signal to the robot control application. The approach

sequence is terminated upon receiving the gripper latch sensor signal. The collision sensor provides passive compliance during the grasp sequence, and is used to trigger an evasive maneuver if the amount of passive compliance in the stem exceeds a set threshold.

Active Compliance During Align & Grasp

Additionally, the force/torque sensor at the end effector is also used to actively adjust the position of the gripper to compensate for errors present in the sensor data using the following equations:

$$\bar{v} = v_{magnitude} \left(\frac{F_x}{F} \hat{i} + \frac{F_y}{F} \hat{j} + \frac{F_z}{F} \hat{k} \right) \quad (6.1)$$

$$F = \sqrt{F_x^2 + F_y^2 + F_z^2} \quad (6.2)$$

Adjustments are *actively* made to the commanded trajectory in the lateral direction during the handle grasp if the gripper is misaligned. This misalignment will be present in the F/T data due to contact between the lever handle and the angled U-shape face of the J-Hook gripper. In this case, additional lateral component adjustments to the robot's commanded velocity (perpendicular to the approach direction) is added in the jog direction vector. The magnitude of the lateral commanded velocity can be constant or proportional to the force experienced (equations 6.1 & 6.2). In addition to active compliance, the axial force (i.e. into the F/T sensor) is constantly monitored during the grasp and the approach is stopped if the force exceeds a user-defined threshold. An overhead schematic of the active compliance procedure can be seen below in Figure 6-20.

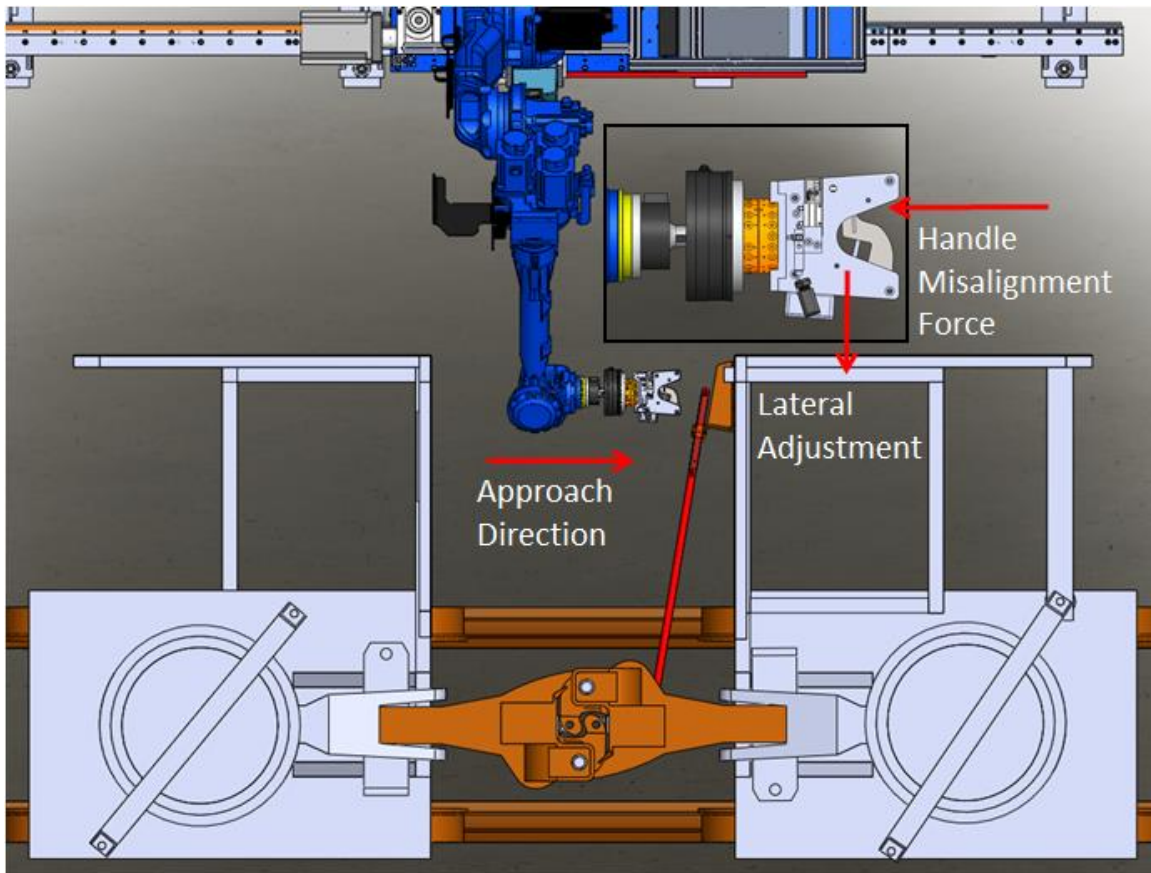


Figure 6-20: Gripper align with active compliance enabled

Once the gripper is latched around the handle, the robot is commanded to perform a circular trajectory about the handles center of rotation to pull the lever in order to uncouple the pair of freight cars. At any point during operation, the gripper can be commanded to release the handle within 25ms to allow the robot to retract safely from between the cars. The pull sequence consists of small incremental motion commands to the robot to achieve a circular trajectory. During the pull motion, the end effector forces are continuously monitored. Higher forces indicate increased friction or wedging between the lock pin and the inner surfaces of the coupler. The robot is commanded to retrace 5-10 degrees and re-attempt the pull if the force exceeds a user-defined threshold, meaning

the pull has bottomed out due to internal friction. The system also monitors the angle of rotation reached by the pull to determine if the uncoupling process was successful. The repetitive rotary motion of the robot during the pull is a direct replicate of the motion of that brakeman currently execute in a manual pin pulling operation. A CAD schematic for the circular pull and screenshots from a successful laboratory test are shown below in Figure 6-21.

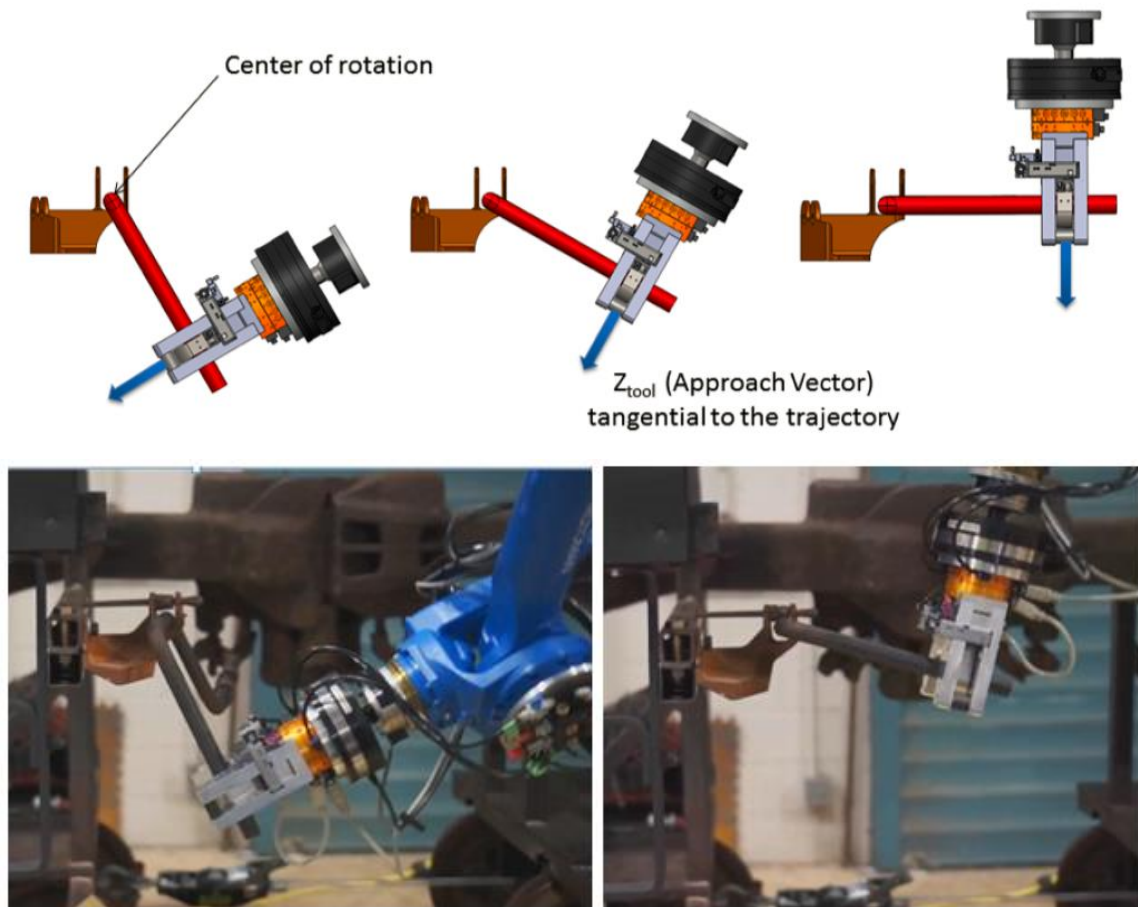


Figure 6-21: Circular lever pull with J-Hook gripper

6.4.4 Align & Pull Sequence: Rotary Design

To achieve a pull with the rotary design, APPS must first align its end effector with the position of the moving uncoupling lever handle while the robot and train are synchronized. The primary goal for alignment with the rotary design is to position the steel rod attached to the end effector behind the uncoupling lever handle (without making contact with its surroundings e.g. sill step). The pneumatic linear actuator is initially in the retracted position so it does not inadvertently make contact with an obstacle during the robots align sequence. Once the robot positions the rotary end-effector assembly behind the lever handle, the system actuates the pneumatic cylinder, which extends 6in into the area between the freight cars. This operation is shown in Figure 6-22 below.

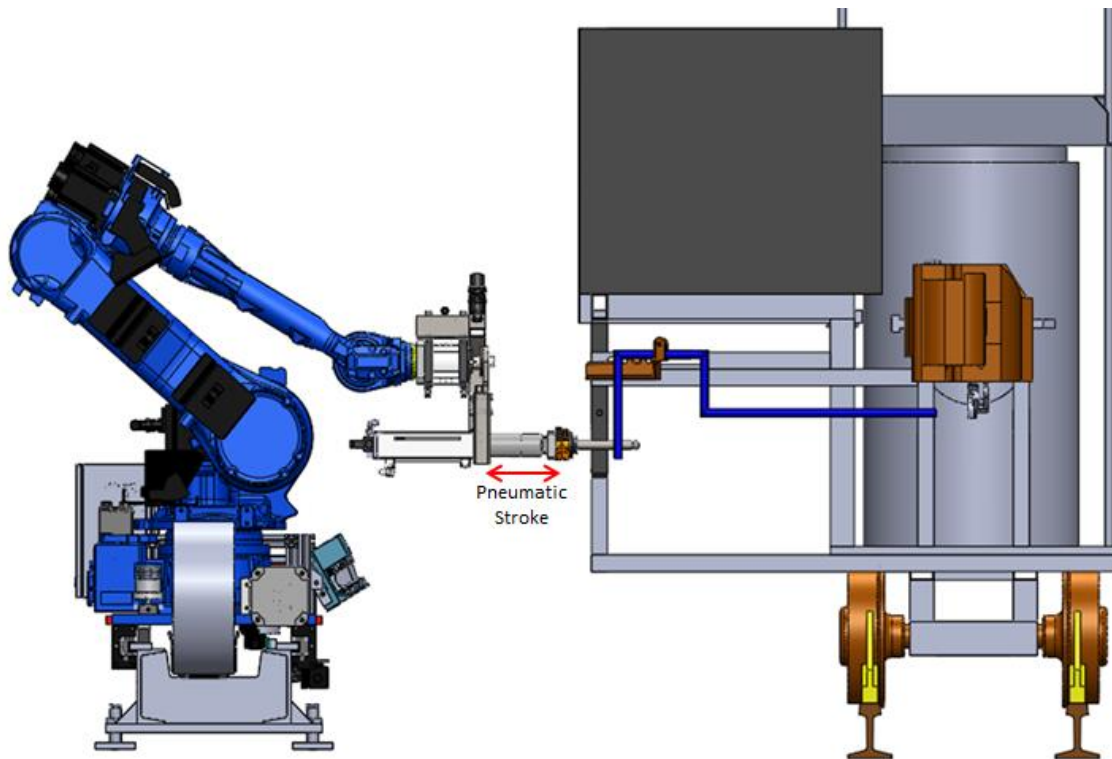


Figure 6-22: Alignment and extension of pneumatic cylinder for handle pull

Now that the steel rod connected to the pneumatic cylinder in Figure 6-22 is in the red zone and in contact with the lever handle, APPS can actuate the rotary-vane actuator to rotate the lever with a force proportional to the pressure of its pneumatic input. The speed of rotation for the pull can also be controlled using flow control valves on the outlet in addition to the pull force. This end-effector is capable of faster and stronger lever pulls in comparison to the J-Hook gripper in the previous section. Faster speeds help overcome the friction inside the coupler, resulting in fewer pull attempts for every successful pull. Faster operation also decreases the cycle time for the uncoupling process. More comparisons between the two end effector designs will be provided in the following section. Simulated and laboratory test schematics of the lever pull operation using the rotary design can be seen in below in Figure 6-23.

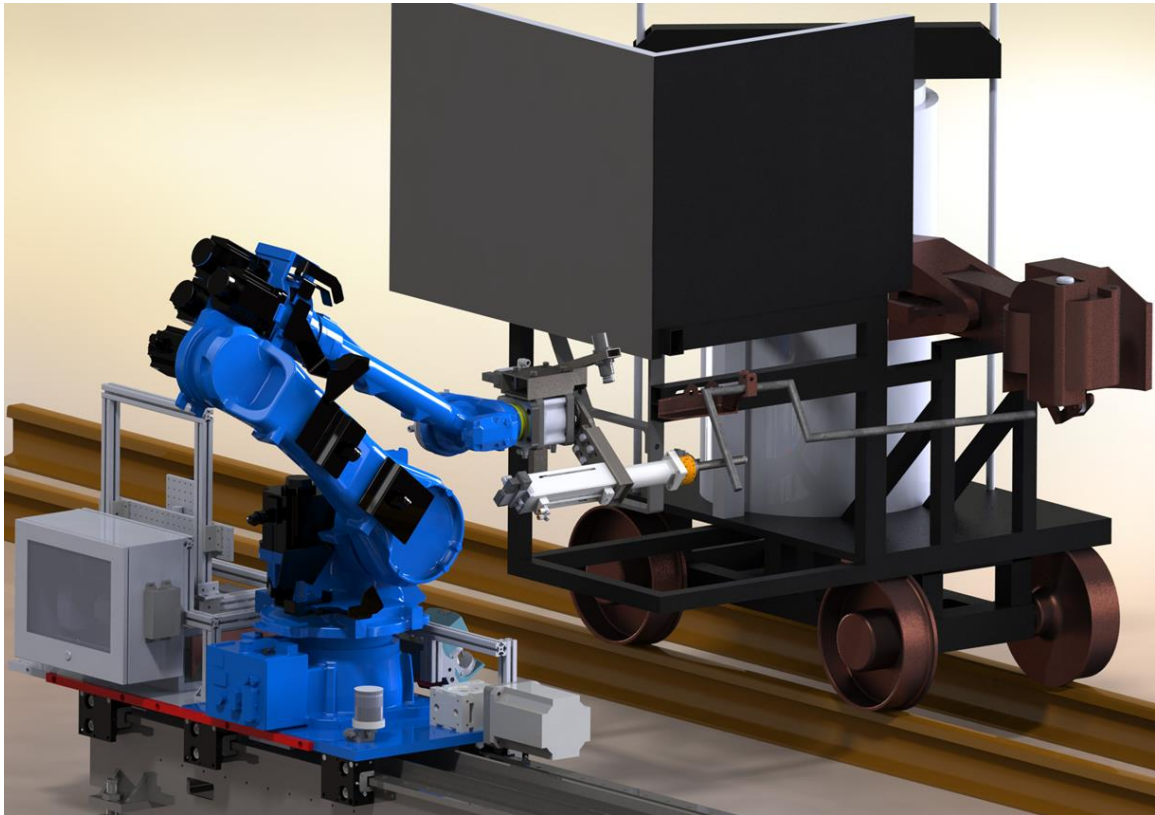




Figure 6-23: Lever pull with rotary end-effector design, simulated (top), lab test (bottom)

6.4.5 Summary of APPS End-Effector Designs

The J-Hook design described in Section 6.4.3 provides more options for control due to its utilization of seven DOF to perform the uncoupling task. These seven DOF can be used to pull the lever handle in any plane necessary (not only in plane parallel to the rail) and can be used to actively adjust the robots trajectory on the fly to compensate for errors due to sensor inaccuracies or uncertainty. However, this design currently requires the robotic arm, the most expensive component in the entire system, to enter the *red zone* between the two moving freight cars. The red zone is defined by the FRA as any area within 4 feet from the outside rail on each side of the track. Brakemen must enter the red zone briefly to grasp and pull the uncoupling lever on a moving train in a classification yard. The red zone as defined in the laboratory is shown in Figure 6-24.

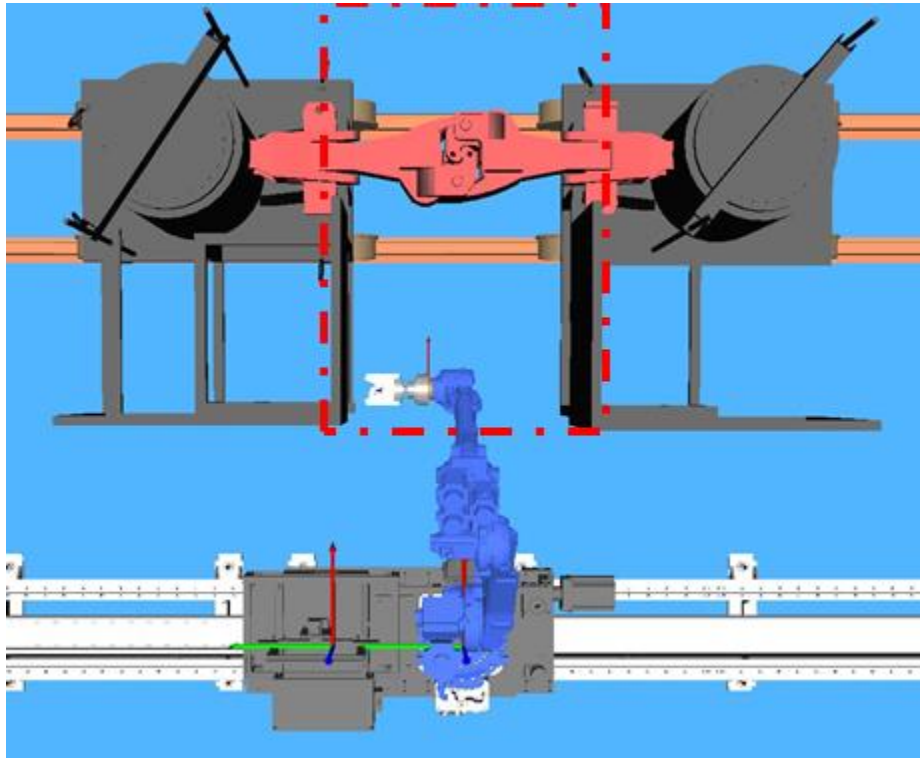


Figure 6-24: Robot in “red-zone” between rolling stock

In Figure 6-24 above, the robotic arm has entered the red zone and is preparing to grasp the lever mounted on the lead car. The robot is constantly monitoring the environment while in the red zone and is programmed to take evasive action in the event of a failure. However, the potential still exists for the expensive robotic equipment to be permanently damaged if a sudden, severe error occurs. Therefore, the ideal design for an autonomous pin pulling system would not require costly equipment to enter the red zone. This was accomplished using the rotary end-effector design.

The current design for the rotary end-effector never requires the robotic arm to enter the red-zone. The only piece of equipment that does enter the area between two moving freight cars is a simple steel rod with a rounded end. In the event of a failure where the robot stops while the pneumatic cylinder is extended between the cars, the train

will hit the steel rod and it will mechanically separate from the end-effector due to the fall off mechanism. If damaged, this component could be replaced for less than \$25, as opposed to \$2000-5000 for the fabrication of another custom J-Hook gripper. Additionally, the rotary and linear actuators on the rotary end effector are both pneumatic, meaning they are inherently backdrivable. If they inadvertently strike an object or bottom out during actuation, the actuators themselves will not be harmed. Finally, the speed and force advantage mentioned previously can save significant amounts of time and money.

6.5 CONCLUSION

In review, the automated pin pulling system (APPS) was designed using an off-the-shelf six DOF manipulator mounted on a motorized linear track axis. The track platform is controlled independently from the robotic arm and is used for velocity and position synchronization with the rolling stock test bed in the laboratory. Wheel detectors and a 3D vision system are used to calculate small incremental motion commands to match the velocity of the train based on sensor data.

Multiple end-effector concepts were designed and fabricated, each with different advantages. Upon the completion of both end-effector designs and significant computer vision application development, lever pulls with static rolling stock were tested. Robotic motion plans were generated, simulated, and tested to approach, grasp, and pull the lever. Static testing allowed the RRG to debug and validate all components of the system before transitioning to the dynamic testing phase described in section 6.4.2. Hundreds of successful runs have now been completed in the laboratory using both end-effectors. Based on the results of these tests, the RRG can confidently claim that it has developed an autonomous system capable of reliably pulling three different uncoupling levers on a

realistic test bed traveling at 0.1 to 2.0 mph to separate freight cars. Over time, the rotary end-effector design has proven to be the better option for uncoupling freight cars autonomously due to its robustness, reliability, simplicity, and speed. However, the J-Hook design is also a viable option, giving the end user a distinct set of options for future implementation. Further development is required to integrate the current system into an operational classification yard.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

The last decade has witnessed rapid development in the fields of machine vision and autonomous robotics, from the boom of affordable 3D vision enabled by the release of Microsoft's Kinect in 2010 to the widespread adoption and use of unmanned land and air vehicles by the U.S. military. All signs are pointing to that trend continuing, with the hope that more of these technologies will cross the valley currently dividing research labs and industry. The next decade will go a long way to defining whether this expansion will translate into economic success, with DARPA commissioning one of most demanding robotics endeavors yet with their third Challenge targeting humanoid robots for disaster response scenarios, Google's simultaneous acquisition of seven leading robotics companies [Markoff, 2013], and Apple's \$350 million purchase of PrimeSense [Israel, 2013], the 3D sensing technology behind the Kinect.

U.S. Class I railroads have also made it clear over the last decade that they are willing invest in cutting edge technology and infrastructure to make their operations safer and boost efficiency. Thus far it has worked, as safety record lows and performance record highs are currently being set on an annual basis. Therefore, the focus of this work is to identify areas and applications within the freight rail industry where modern advancements in robotics and automation can be leveraged to make operations safer and more efficient.

This chapter summarizes the material presented in the previous six chapters and suggests several themes for further work that would expand upon the success of demonstrated by APPS, the Automated Pin Pulling System developed by RRG at The University of Texas at Austin.

7.1 SUMMARY

Chapter One delved into the history of railroad operations to provide an understanding for the evolution of the industry over the past 200 years. Previous and current railroad initiatives regarding safety and efficiency were analyzed to gain an understanding of the industry's priorities. Several reports and studies regarding the safety of railroad employees were summarized to identify the areas where robotics and vision could be leveraged. In summary, several laborious and hazardous tasks that are currently performed manually in rail yards throughout North America were identified and a detailed solution to one, manual uncoupling, has been proposed. Using similar techniques, robotic solutions for air brake hose assembly and handbrake operation could be achieved.

Chapter Two surveyed existing systems and patented concepts for machine vision and robotic systems currently used for railroad applications. The uses of each system were outlined and briefly compared to the requirements for uncoupling, air brake hose operation, and hand brake manipulation. The strengths and weaknesses of numerous patents for conceptual designs of automated uncoupling systems for rail yard applications were analyzed to present a case for the design and benefits provided by APPS; the first wayside system prototype proven to uncouple cars autonomously.

Chapter Three discussed standard and custom techniques used for machine vision and image processing applications in all industries. Background information on core vision algorithms and techniques were provided to sufficiently prepare the reader for the material in following chapters. Classing and novel methods for robotic path planning, including the RRG's geometric-based path planning, were also covered to elucidate the benefits associated with the higher order properties of spatial curves and its potential impact on robotic applications for railroads.

Chapter Four provided an overview of commercially available 2D and 3D vision systems, robotic platform and manipulator technology, and critical end-effector hardware that could be used in a variety of systems for rail yard applications. Select items from this section were chosen for the development of APPS. The hardware presented in this chapter that was not used for APPS could potentially be used for a future version of the system to be deployed in an operational rail yard, where increased ruggedness and reliability would be required.

Chapter Five provided an overview of the object detection and recognition techniques developed by the RRG to detect, track, and grasp the moving uncoupling lever. Both geometric image processing techniques and cutting edge machine learning algorithms were used to perform these tasks. These algorithms could also be leveraged by railroads for inspection, maintenance, or other demanding railroad tasks.

Chapter Six provided detail on the RRG's rolling stock test bed that was designed and built specifically to test a robotic uncoupling system. This chapter then detailed multiple designs for an Autonomous Pin-Pulling System (APPS) to uncouple moving freight cars in classification yards. APPS has been prototyped and rigorously tested in RRG's laboratory environment with full scale hardware. Although APPS is designed to solve a specific task, its subcomponents can be used to design automated solutions for other rail yard tasks. Sufficient detail regarding the operation of APPS and its subcomponents were provided to give the reader full scope of the development of an intelligent robotic system designed for railroad operations.

7.2 RECOMMENDATIONS FOR FUTURE WORK

This work primarily focuses on the design, fabrication, and testing of an automated pin pulling system used for railroad applications. The design of this system

and subsystems were intentionally built to be modular to solve other problems existing in rail yards. Future tasks for work include:

- **Combination of vision algorithms for handle detection** – five methods for detecting uncoupling lever handles were designed, coded, tested, and documented. All of these methods could be intelligently fused to boost confidence in the detection algorithm’s conclusions and provide an unprecedented level of reliability.
- **Fusion of velocity tracking methods** – three different sensors and two algorithms were tested independently for the synchronization of the train and the robot. The current method for synchronization is adequate, but changes could be made to increase its robustness. These methods could also be combined to improve system performance and add further layers of fault tolerance.
- **Preparation for transition to yard** – several sensors and robotic-end effector components used for APPS are not currently suited for operation in hazardous environments. Ruggedized housings and coatings would be required for this system to operate in outdoor environments.
- **Optimize pneumatic and electrical wiring** – most of the wiring for APPS is external to the robotic arm. Robotic arms, end-effector components, and grippers can be purchased and/or fabricated with through holes to accommodate clean, reliable solutions for routing cabling.
- **Develop/modify APPS for automatic air brake hose coupling** – acquire and install air brake hose assembly components on the rolling stock test bed in the laboratory and develop a custom end-effector to autonomously assemble a pair of air hoses. The subcomponents of the system (vision systems, robot, end-effector

technology) are currently in place. Changes can be made to accommodate to demonstrate autonomous air brake hose assembly.

- **Develop/modify system for automated handbrake operation** – acquire and install handbrake components on the rolling stock test bed in the laboratory and develop a custom end-effector to autonomously rotate the wheel (similar to the manual valves used in the DARPA humanoid challenge). The subcomponents of the system (vision systems, robot, end-effector technology) are currently in place. Changes can be made to accommodate to demonstrate autonomous handbrake operation.

7.3 CONCLUDING REMARKS

The research presented in this work led to the development of an autonomous robotic system capable of reliably and efficiently performing a task that, despite 75 years of effort, has not yet been solved. The major contributions of this work are:

- **Real-time robust uncoupling lever detection** – several research institutions and commercial machine vision companies have attempted detecting uncoupling levers without sustained success. These existing methods used digital image data prone to noise and background clutter that affected the performance of the algorithm. Additionally, these techniques were not implemented in real-time, as post-processing techniques were used to identify handles from logged data after a train had passed. However, the design and implementation of the algorithms covered in this work are fast, robust, and flexible. The machine learning technique can be re-trained in a matter of seconds to adapt to other rail safety components or appliances. Finally, a combination of these methods can be used for redundancy to successfully detect multiple handle types in varying environmental settings.

- **Augmented industrial manipulator with intelligence** – this research combined multiple, real-time deterministic operating systems to route digital and analog signals from a network of various sensors to the robot’s control system to enable real time decision making. An onboard 64-bit PC and accompanying vision system give the robot the necessary feedback to perform complex tasks using computer vision software. The level of flexibility provided by third party hardware and software combined with the reliability of a mature industrial manipulator has resulted in a system that is unique to the industry.
- **Multiple unique end-effectors** – two custom end-effectors were developed to enable the coordination and interaction between moving multi-ton rigid bodies to perform a hazardous task. One design is passively actuated using affordable off-the-shelf components to safely grasp and pull an uncoupling lever. This design also has numerous layers of fault tolerance to protect high cost equipment in the event of a failure. The second design is especially promising, as it enables automated uncoupling at high speeds without entering the area between two freight cars through the use of simple, robust backdrivable actuators to quickly and efficiently pull pins.

References

- (2013). *Cartesian space vs hough space*. Available: <http://tex.stackexchange.com/questions/110012/tikz-code-cartesian-space-vs-hough-space>
- S. H. Ahn. (2006). *Histogram*. Available: <http://www.songho.ca/dsp/histogram/histogram.html>
- N. Ahuja and C. P. L. Barkan, "Machine Vision for Railroad Equipment Undercarriage Inspection Using Multi-Spectral Imaging," University of Illinois at Urbana-Champaign, Urbana, IL 2007.
- Association of American Railroads, *Educational sequences : AAR standard type E railroad car coupler : design, inspection, testing ; production ; history ; operation*, 1965.
- Association of American Railroads. (2013, Class I Railroad Statistics. Available: <https://www.aar.org/StatisticsAndPublications/Documents/AAR-Stats-2013-07-09.pdf>
- Association of American Railroads. (2013). *Positive Train Control*. Available: <https://www.aar.org/safety/Pages/Positive-Train-Control>
- Association of American Railroads. (2013, November). *Railroad Safety and Security*. Available: <https://www.aar.org/safety>
- ATI Industrial Automation. (2013). *Multi-Axis Force/Torque Sensors*. Available: <http://www.ati-ia.com/Products/ft/sensors.aspx>
- ATI Industrial Automation. (2013). *Robotic Collision Sensors*. Available: http://www.ati-ia.com/Products/collision_sensor/robot_collision_sensor.aspx
- Beena Vision. (2013, November). *Beena Vision Product Datasheets*. Available: <http://www.beenavision.com/products>

G. Beyer. (2013). *Technical aspects of a Kinect device*. Available: <http://programmingwithkinect.wordpress.com/2013/03/31/technical-aspects-of-a-kinect-device/>

A. C. Bovik and e. al., *The Essential Guide to Image Processing*: Elsevier Inc., 2009.

G. R. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st ed.: O'Reilly Media, 2008.

M. Branding, K. Salsman, M. Somayaji, B. Dipert, T. Droz, D. V. Nieuwenhove, *et al.* (2013). *3-D Sensors Bring Depth Discernment to Embedded Vision Designs*. Available: <http://www.embedded-vision.com/platinum-members/embedded-vision-alliance/embedded-vision-training/documents/pages/3d-sensors-depth-discernment>

M. D. I. Bruns, M. Enning, T. Kimpel, H. Rake, W. Schwanhaeusser, and W. Klaus, "Einrichtung zum vollautomatischen trennen von gueterwagen mit automatischen kupplugen," 4015878, November 21, 1990.

J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, 1986.

J. Cappelletti and S. Zakaria, "Automated Decoupler for Rail Cars," United States Patent 5531337, July 2, 1996.

P. Cattin. (2012). *Image Segmentation: Introduction to Signal and Image Processing* [Presentation]. Available: <http://miac.unibas.ch/SIP/07-Segmentation.html>

Concept Systems Inc. (2013, October). *RailHawk*. Available: <http://www.conceptsystemsinc.com/products/railhawk.php>

Creative Labs Inc. (2013). *Creative Senz3D*. Available: <http://us.creative.com/p/web-cameras/creative-senz3d>

J. R. Dirnberger, "Development and Application of Lean Railroading to Improve Classification Terminal Performance," University of Illinois at Urbana-Champaign, Master's Thesis, 2006.

D. Drudi, "Railroad-related work injury fatalities," *Monthly Labor Review*, 2007.

J. R. Edwards, "Improving the Efficiency and Effectiveness of Railcar Safety Appliance Inspections Using Machine Vision Technology," University of Illinois at Urbana-Champaign, Urbana, IL, Master's Thesis, 2006.

Federal Railroad Administration, "An Examination of Railroad Yard Worker Safety," Federal Railroad Administration, Springfield, VA2001.

Federal Railroad Administration, "Safety of Remote Control Locomotive (RCL) Operations," Final Report, March 2006.

Federal Railroad Administration. (2012, November). *Freight Rail Today*. Available: <http://www.fra.dot.gov/Page/P0362>

K. Grauman. (2011, CS 376 Computer Vision Spring 2011 Lectures. [Lectures]. Available: www.cs.utexas.edu/~grauman/courses/spring2011

Güdel AG, "Installation and Maintenance Manual," ed, 2012.

J. Hecht. (2011, PHOTONIC FRONTIERS: GESTURE RECOGNITION: Lasers bring gesture recognition to the home. *Laser Focus World*. Available: <http://www.laserfocusworld.com/articles/2011/01/lasers-bring-gesture-recognition-to-the-home.html>

J. W. Heimaster and R. R. Cosner, "Remotely Controlled Car Uncoupling Systems," United States Patent 2746615, 1956.

igm Robotic Systems Inc. (2013). *igm Products*. Available: <http://www.igmusa.com/us>

S. Israel. (2013, November 27). Why Apple Bought PrimeSense. *Forbes*. Available: <http://www.forbes.com/sites/shelIsrael/2013/11/25/why-would-apple-buy-primense/>

H. K. Kandoth-Kannoth, A. R. Vithani, D. C. Brabb, and R. S. Trent, "Automatic Cut Lever Apparatus," United States Patent 7562781, July 21, 2009.

C. Kapoor, "A Reusable Operational Software Architecture for Advanced Robotics," PhD, Mechanical Engineering, The University of Texas at Austin, 1998.

A. V. Kulkarni, "Instant Center Based Kinematics and Dynamics for Planar Mobile Platforms," PhD, Mechanical Engineering, The University of Texas at Austin, 2009.

Y.-C. Lai, C. P. L. Barkan, J. Drapa, N. Ahuja, J. M. Hart, P. J. Narayanan, *et al.*, "Machine vision analysis of the energy efficiency of intermodal freight trains," *Proceedings of IMechE* vol. Part F: Rail and Rapid Transit, 2007.

C. E. Lindow, "Brush Uncoupler for Rail Cars," United States Patent 3805968, April 23, 1974.

Loccioni. (2013, November). *Felix, the first mobile robot for automatic railroad switch inspection*. Available: <http://research.loccioni.com/robotics/felix>

Logitech. (2013). *Logitech Pro 9000 Webcam*. Available: <http://www.logitech.com/en-us/support/3056?osid=14&bit=64>

R. Maini and H. Aggarwal, "Study and Comparison of Various Image Edge Detection Techniques," *International Journal of Image Processing (IJIP)*, vol. 3, 2009.

P. S. March, "Criteria Based Motion Planning," Master of Science in Engineering, Mechanical Engineering, The University of Texas at Austin, 2004.

P. S. March, "Geometric-based Spatial Path Planning," Doctor of Philosophy Dissertation, Mechanical Engineering, The University of Texas at Austin, 2008.

J. Markoff, "Google Puts Money on Robots, Using the Man Behind Android," in *The New York Times*, ed, 2013.

MathWorks. (2013). *Image Processing Toolbox*. Available: <http://www.mathworks.com/products/image/>

McConway & Torley. (2013). *Type E Couplers & Parts*. Available: <http://www.mcconway.com/>

C. W. McDonald, "The Federal Railroad Safety Program: 100 Years of Safer Railroads," ed. Washington, DC: Federal Railroad Administration, 1993, p. 38.

T. McKeough. (2009, July 1) Robots Help Trains Stay on the Rails. *Fast Company*.

R. P. Merges, "The Uninvited Guest: Patents on Wall Street," University of California, Berkeley - School of Law, Berkeley, California, May 27 2003.

Mesa Imaging. (2013). *Swissranger SR4000*. Available: <http://www.mesa-imaging.ch/swissranger4000.php>

Metrilus. (2013). *Range Imaging Technology*. Available: <http://www.metrilus.de/range-imaging/>

K. Nayebi and S. Williams, "Application of Machine Vision in the Railroad Industry," in *International Rail Safety & Incident Investigation Seminar*, Orlando, FL, 2012.

K. Nell, "Uncoupling Device for Central Buffer Couplings on Railroad Vehicles," United States Patent 4008811, February 22, 1977.

Nordco. (2013, November). *Wayside Wheel Inspection*. Available: <http://www.nordco.com/products-catalog/inspection-technologies/wheel-inspection-systems/wayside-wheel-inspection.htm>

Open Perception. (2013). *Point Cloud Library (PCL)*. Available: <http://pointclouds.org/>

OpenCV. (2013). *Feature Detection*. Available: http://docs.opencv.org/modules/imgproc/doc/feature_detection.html

OpenCV. (2013). *Neural Networks*. Available: http://docs.opencv.org/modules/ml/doc/neural_networks.html

Operation Lifesaver Inc. (2013). *Rail Safety*. Available: <http://oli.org/rail-safety>

D. A. Pagano and D. W. Giragosian, "In-Track Wheel Inspection," United States Patent 7,555,954, 2009.

J. C. Peters, "Positive Train Control (PTC): Overview and Policy Issues," Congressional Research Service, 2012.

T. D. Peterson and R. W. Crosser Jr., "Apparatus for Uncoupling Railroad Cars," United States Patent 3682325, August 8, 1972.

E. M. Petriu, "Robotics-Introduction," ed, 2012.

D. Piatti and F. Rinaudo, "SR-400 and CamCube 3.0 Time of Flight (ToF) Cameras: Tests and Comparison," *Remote Sensing*, vol. 4, pp. 1069-1089, 2012.

K. J. Pithadiya, C. K. Modi, and J. D. Chauhan, "Selecting the Most Favourable Edge Detection Technique for Liquid Level Inspection in Bottles," *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, vol. 3, pp. 034-044, 2011.

Plasser & Theurer. (2013) Mobile rail treatment: focus on the rail. *Today*.

PlayStation. (2013). *PlayStation®Eye Camera*. Available: <http://us.playstation.com/ps3/accessories/playstation-eye-camera-ps3.html>

PMD Technologies. (2013). *CamBoard Nano*. Available: www.cayim.com

PMD Technologies GmbH. (2008, A performance review of 3D TOF vision systems in comparison to stereo vision systems.

Point Grey. (2013). *Stereo Vision Products*. Available: <http://www.ptgrey.com/products/stereo.asp>

M. Sanyas, "Mechanism for Automatically Uncocking and Re-cocking Automatic Couplings for Railway Vehicles," United States Patent 3224601, 1965.

S. Sawadisavi, J. R. Edwards, E. Resendiz, J. M. Hart, C. P. L. Barkan, and N. Ahuja, "Machine-Vision Inspection of Railroad Track," 2009.

M. Shah, "Automated Visual Inspection/Detection of Railroad Track," University of Central Florida July 2010.

SICK, "Machine Vision Introduction," 2006.

SICK, "Laser Measurement Sensors of the LMS5xx Product Family," ed, 2012.

K. R. Spring and M. W. Davidson. (2013, Fundamentals of Digital Imaging. *MicroscopyU*. Available:
<http://www.microscopyu.com/articles/digitalimaging/digitalintro.html>

Teledyne Dalsa. (2013). *CCD vs. CMOS*. Available:
<http://www.teledynedalsa.com/imaging/knowledge-center/appnotes/ccd-vs-cmos/>

D. Tesar, P. S. March, A. S. Boddiford, and M.-K. Park, "Geometric-Based Spatial Path Planning: Part I Framework and Application," unpublished|.

The Railway Educational Bureau, *Guide to Freight Car Couplers and Draft Gear Systems*. Omaha, NE: REB, 2000.

The Railway Educational Bureau, *Basic Training Manual for Brakemen and Switchmen*. Omaha, Nebraska: Railway Educational Bureau, 2004.

Transportation Research Board, *Compensating Injured Railroad Workers Under the Federal Employers' Liability Act*. Washington DC: National Academy Press, 1994.

Velodyne Inc. (2012). *HDL-64E LiDAR*. Available:
<http://velodynelidar.com/lidar/hdlproducts/hdl64e.aspx>

Wabtec Corporation, *Freight Car Brake Rigging Arrangements*, 2004.

P. J. Wong, M. Sakasita, W. A. Stock, C. V. Elliot, and M. A. Hackworth, "Railroad Classification Yard Technology Manual - Volume I: Yard Design Methods," Federal Railroad Administration, Springfield, VA1981.

Yaskawa Motoman, "Record robot gantry system to weld coal wagons," ed, 2007, p. 2.

Yaskawa Motoman. (2013). *MH80 Datasheet*. Available:
<http://www.motoman.com/datasheets/MH80.pdf>

Z. Zhang, "Microsoft Kinect Sensor and Its Effect," *IEEE Multimedia*, 2012.

Vita

Andrew Shropshire Boddiford was born in Sylvania, GA in 1987 to his parents David and Deborah Boddiford. After graduation from Screven County High School in 2005, he enrolled at Clemson University in Clemson, SC. During his time at Clemson, Andrew completed two engineering internships and a full co-op rotation with companies in South Carolina, Germany, and Georgia respectively. He received a Bachelor of Science in Mechanical Engineering with a German Language minor in May, 2010. Andrew entered the graduate program at The University of Texas at Austin in the fall of 2010 where he accepted a Graduate Research Assistantship in the Robotics Research Group to design and build an autonomous robotic system.

Email: aboddif@gmail.com

This thesis was typed by the author.