Robotic manipulator capable of sorting moving objects alongside human workers using a budget-conscious control system

Adela Wee*, Christopher Willis, Victoria Coleman, Trevor Hooton, Andrew Bennett*

Intelligent Vehicles Laboratory
Franklin W. Olin College of Engineering
Needham, MA, United States
{Adela.Wee, Christopher.Willis, Victoria.Coleman, Trevor.Hooton} @students.olin.edu

Abstract*— Robots are already becoming commonplace for tasks which are dirty, dull, and dangerous. However, current technology limits tasks to be performed by either humans or robots in isolation. In the near future, tedious tasks will no longer be done solely by humans or robots, but will be completed by human-robot teams. In today's factories, significant resources have been invested into complex, strong and agile robotic arms; however, due to the lack of their spatial awareness and difficulty in programming them, they are not easily adaptable to new tasks. In addition, humans are prevented from moving into the work envelope while the machines are running, so robots are generally left to tackle the dangerous tasks on the factory floor with poor or no sensing capability. In order to revolutionize production with the least cost possible to the factory, these arms must be retrofitted with a fairly inexpensive and adaptable sensing and control system. We have developed a low cost control system that can be adapted to any arm which is based off of a commonly available 3D sensing system, such as the Microsoft *Kinect* (a consumer-grade RGB-D camera available for $100 USD that has a resolution of approximately 2mm at 1m to 2.5cm at 3m). Use of such a sensor would allow an automated industrial manipulator arm to not only grasp and articulate moving objects alongside human beings in any environment, but also enable rapid reprogramming for new spatially oriented tasks, regardless of lighting conditions. Because the depth camera collects diffracted infrared beam data and correlates it to each pixel in a color image it can be used in any indoor environment without the need for special equipment. Using simple image-processing techniques on the images from the depth camera, we have created a working proof-of-concept prototype that can recognize uniquely shaped objects moving on a 122cm by 36cm conveyor belt. The system demonstrated its ability to recognize, grasp, and manipulate pieces to play a game of Tetris, making sure to optimize the position and orientation of each piece as it was detected. We are confident this technology can be applied to other applications, such as sorting waste products, organizing nuts and bolts, and any other task involving sorting of unique objects even if moving.

Keywords: manipulation, sorting moving objects, Microsoft *Kinect*

*Corresponding author

I. Introduction

A. Why Do Robots Matter in Industrial Applications?

1) Unsafe Environment

In areas where humans are at significant risk, such as factory floors, integrating robots is especially useful. For welding of frames, robots are faster than humans and and keep them away from of hazardous tasks. However the robot arms used for this particular type of work are designed to work at ultra-fast speeds with heavy payloads and are a great danger to humans--they must be kept behind fences.

2) Precision and Speed on Repetitive, Tedious Tasks

As markets grow and quality standards improve, manufacturing processes must become faster and more precise to keep up. Robotics research pushes the envelope of automated processes, and industries are able to leverage more and more technology to increase throughput, reduce errors and free workers from dull, repetitive, or dangerous tasks. The industrial revolution brought about improvements like assembly lines that led to high volumes in manufacturing; sub-dividing tasks into modular, repeatable tasks for individual workers to mindlessly complete.[1] The robotics revolution takes that paradigm one step further, and employs technology to solve those repeatable problems faster and more perfectly than a human could.

[1] P. Backer, Industrialization of American Society http://www.engr.sjsu.edu/pabacker/industrial.htm

Humans and robots have very different skill sets. Generally speaking, robots excel in tasks focused on data manipulation, while humans excel at complex reasoning. In many factories, assembly line workers are faced with dull tasks like attaching two parts together for hours on end every day. That same manufacturing process could be performed by a robot, which could complete the task more quickly and less tediously, freeing human workers for more interesting tasks. Robotics research into characterizing and developing many more robust automated manufacturing techniques means that the set of automatable tasks is becoming more extensive and effective every year. This research is allowing the manufacturing industry to adapt to growing markets and increasing requirements for precision and speed.

3) Human-Robot Interaction

Despite advances in automation, humans are still necessary for monitoring manufacturing lines in order for processes to be properly prepared and maintained. Industrial robots are typically designed for a specific task in their environment and to do the task as efficiency possible, which can involve moving heavy parts at a high speed. This results in a fairly dangerous environment that blocks humans from being within the work envelope of a robot, and does not enable factories to have robots working alongside humans. However, the workforce of the future will need robots to work safely alongside humans in order to sustain the same levels of productivity in low-cost production [1].

II. What is State of the Art?

A. Baxter

Baxter, made by Rethink Robotics, is an industrial robot made specifically to reduce burden on humans on the assembly line. Unlike other robotic arms, it requires no safety cages, can be programmed by an assembly line worker, and is portable. Due to its two 7 DOF arms with torso and head, Baxter moves at a much slower pace and is made for low volume, high mix manufacturing. Baxter, a $22,000 robot, can be programmed by the user either through the touchscreen interface (which also doubles as its face, Fig. 1) or by direct manipulation of its arms by an operator.

Figure 1. *The many faces of Baxter*

B. UBR-1

The UBR-1 is a mobile manipulation platform designed both for academic research as well as business automation by Unbounded Robotics, a spin-off of Willow Garage (Fig. 3). This robot can see where it's going, drive itself around, and manipulate objects using its arm. It has a PrimeSense RGB-D camera in its head and can accommodate custom robot grippers. It is controlled either using a PS3 controller or through the onboard software, which allows it to track and grasp an object within its field of view. This robot is available for $35,000.

C. ZenRobotics Recycler

The ZenRobotics Recycler is the first robotic waste sorting system in the world. It's currently designed for reclaiming the valuable raw materials from construction and demolition waste, such as metal, wood, and stone (Fig. 3). This robot uses visible spectrum cameras, near infrared cameras, 3D laser scanners, and haptic sensors to create an accurate real-time analysis of the waste stream currently being processed. The semi-mobile product version can be shipped in a standard shipping container, and its system which includes two robotic arms weighs 20 tons, and is 12 m long.

Figure 2. *UBR1 mobile manipulation platform used in academia.*

.

Figure 3. *ZenRobotics Recycler, the first robotic waste sorting system to reclaim raw materials*

D. Goal of the project

The ultimate goal for the project is to perform a task in a shared space with humans using cheap, portable sensors. The specific task is a physical implementation of Tetris, with pieces travelling down a conveyor belt (treadmill) at greater and greater speeds. The challenge is to safely operate the arm when people are within the manipulator's grasp, which may happen when placing pieces on the belt for the arm to manipulate. The Microsoft *Kinect* sensor is inexpensive and widely available, and is able to produce an image correlating the distance of an object from the camera to the color image that is produced. Calculations can be performed to compare a customizable software model for the robot actuator and its motion (in this case the position in 3-space of the R17 arm) and the depth map of the workspace to determine whether an action is safe to perform or not. For example, a human entering the space that the arm will travel through is not safe. The TETARM system can attempt a different movement strategy, or it can slow or stop the treadmill until the human is away from danger, and then continue with the sorting protocol. The specific safety implementation depends on the application at hand.

E. Enabling technologies

1) Kinect RGB-D Sensing

To sense the pieces, we are using the Microsoft Kinect for XBox 360 RGB-D camera as our low-cost vision sensor. Through the depth images produced by the Kinect, our system is mostly light-independent within an indoor environment and can determine more precisely where objects are in a 3-D environment. Due to the nature of the IR diffraction, the light can be easily faded out and does not operate well in brightly lit areas. The Kinect is mounted to a frame above the treadmill, and while this does result in some image occlusion by the arm covering the pieces underneath it, the pictures that are used to determine piece type, location, speed, and orientation are cropped to the area outside the arm's work envelope. If either of these limitations needed to be removed, the sensor could be mounted at a different location to allow for an unobstructed view of the work area. Due to the predictability of the the system, a closed-loop control to confirm the piece was picked up is not necessary.

III. Implementation

A. Software

1) ROS

For our system, we use ROS (Robot Operating System) as a standardized communication framework. Initially created by Willow Garage, and currently maintained by the ROS consortium, ROS is a multi-platform communications protocol for coordinating and communicating between multiple parallel processes on different platforms. Additionally, the platform can be any sort of processor, as long as it is connected to the network and can speak the common language of the ROS protocol.

Using ROS, each platform has one or more nodes running simultaneously. Each node can "publish" and "subscribe" to "topics." Whenever a new packet is published to a topic, every node that subscribes to that topic runs it's individual callback function. This allows for effective asynchronous computation. ROS is also very useful for handling multiple data sources, as it uses new data as it comes in.

2) OpenNI and OpenCV

The OpenNI library for the *Kinect* is an open-source framework to standardize Natural Interaction (NI) devices, applications, and middleware which also has a large developer community behind it.[2] All the relevant information from the camera is bundled and packaged into ROS topics, and published to topics whenever new data comes in. The depth image is collected using the OpenNI library and is then processed through OpenCV. OpenCV is a library of programming functions aimed at real-time image processing systems that was initially developed by Intel but is now open source.[3] The image from the *Kinect* is cropped (Fig. 4) and thresholded for only the height of the pieces to create a black image with the pieces in white. To identify pieces, the templates of the different pieces are compared to the processed *Kinect* image to determine the location, type, and orientation of the piece. .

3) Python

The majority of processing in the TETARM system occurs in a structure of python nodes. Python is a widely used, relatively simple open-source programming language which works well with ROS using the rospy library. ROS can launch multiple python executables as independent nodes running simultaneously.

4) Arduino

Arduino is an open source hardware microcontroller. The software is free to use, and it has extensive example code and an active community. The Arduino Uno I/O controls the gripper servo and treadmill motor using PWM outputs, and measures the speed of the treadmill using a reed switch encoder. The Arduino can be set up as a ROS node to subscribe and publish to topics.

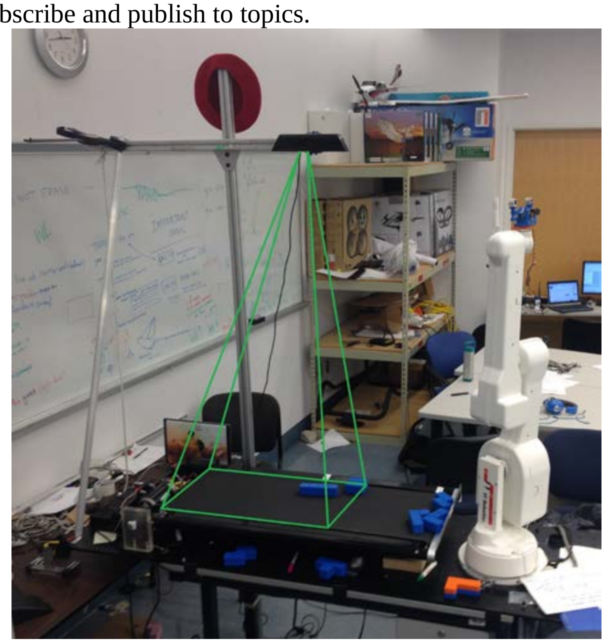

Figure 4. *Kinect's view of the treadmill to determine the presence and charactereistics of pieces. The region in which the Kinect looks for new tetris pieces is highlighted by the green lines*

[2] http://www.openni.org/about/

[3] http://en.wikipedia.org/wiki/OpenCV

IV. Logic Structure

The general code structure is loosely based off the "Olin Robot Brain," a biologically inspired structure that is a hybrid of reactive and subsumption paradigms often used in robotics at Olin College. It is organized in low-level (hindbrain), mid-level (midbrain), and high-level (forebrain) sections (Fig. 5, 6). In this way, the levels of processing are separated so reactive behaviors happen in a fast loop, the maintenance behaviors go slightly slower, and the overarching behaviors to accomplish high-level goals take in various pieces of data in order to determine what commands to send in order to carry out the mission.

The hindbrain interfaces with the sensors and actuators by executing the desired commands of the midlevel, monitoring the encoder and doing low-level processing, and updating the midbrain on the status of the hardware. The midbrain parses data, tracks the timing of the system, coordinates commands and information coming from the forebrain and hindbrain. This section does the bulk of the computation in the architecture. The high-level receives relevant information from the mid-level such as the type of piece available for placement and returns a high-level decision of what Tetris column the piece should be placed in and in what orientation. The midbrain then coordinates the actions to successfully collect the piece.

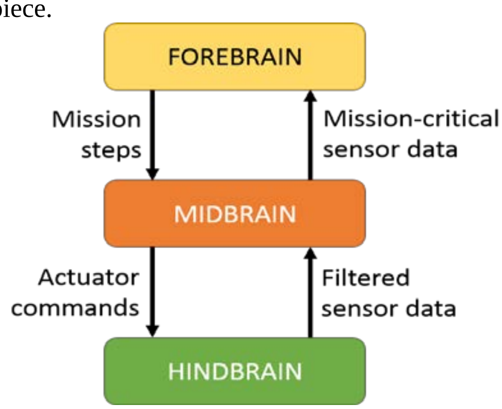

Figure 5. *The diagram of the structure of code distinguished by levels of processing*

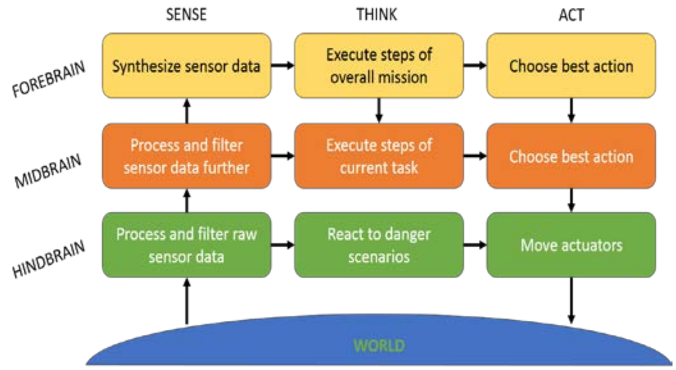

Figure 6. *More detailed description of code architecture and where types of processing occur in the code.*

Beyond reacting to the presence of pieces to accomplish the main goal of playing Tetris, the arm must also be safe around humans. Therefore, the hindbrain processes the depth image from the *Kinect* to tell if there are any unexpected obstacles in the arm's trajectory for the safety of itself and others. If so, it stops the arm and treadmill to prevent it from going near or through that foreign object and saves the state of the system.

A. Hardware

The TETARM project employs a number of different hardware systems: the ST Robotics *R17* Arm controller, an Arduino *Uno*, a Microsoft *Kinect*, a linux computer, and three mechanical systems: the R17 Arm itself, a servomotor gripper and a motorized treadmill.

The R17 Arm is a 5 degree of freedom arm comprised of revolute joints. The R17 serves as a close analog for much larger industrial robotic arms; its joints are kinematically very similar and it has no built-in compliance when in use. Its controller takes in RoboForth commands, but as previously mentioned, we are using a python wrapper for RoboForth. This provided many built-in functions such as position control in Cartesian coordinates that allowed us to focus our efforts on creating a human-safe work environment and object sorting.

The gripper was created using a stratasys 3D printer and a standard hobby servo. It is attached to the end of the arm using magnets, which allows it to be easily swapped out if a different mechanism is desired and to help prevent damage by falling off if the gripper collides with an unexpected object. The treadmill is a standard treadmill with the hand holds removed as to not interfere with the *Kinect*. Additionally we mounted a reed switch and magnets on one of the belt pulleys so that we can ensure that the treadmill is travelling at the desired rate.

The Arduino *Uno* is an inexpensive controller for the servo gripper and treadmill, both of which are controlled using a PWM signal.

We chose the Microsoft *Kinect* because it is fairly inexpensive and gives a fairly accurate depth image compared to other systems. The RGB camera feature of the *Kinect* allow simplifies debugging because it allows us to directly compare the depth and RGB images in real time.

V. PRELIMINARY EXPERIMENTS

With the treadmill running at about 1.3 m/s, the system can identify, pick-up, and place pieces moving on the treadmill to play a reasonable game of tetris. Occlusion caused by the arm does not affect the behavior of the system, and a human avoidance algorithm detects if is if any foreign object comes into the field of view of the depth camera within the range from the *Kinect* to above the pieces. If an object is detected before a command is sent, the arm waits and the system pauses until the object no longer in its trajectory. Due to the constant movement of the treadmill under the pieces, the pieces at the bottom tend to shift and rotate, which disturbs the placement of other pieces. An issue with the system is the lack of validation of its manipulation of the environment. Because it is not a closed-loop system, it has no way of knowing whether it misses a piece or if the places pieces shift. Without this knowledge, the system might assume an inaccurate state of the world, so future pieces are placed inappropriately. A possible solution to this is mounting another *Kinect* or a similar sensor such that it can see the bottom of the treadmill, which is in the occlusion zone of the existing *Kinect*.

VI. CONCLUSION

We successfully created robust platform which can be easily modified to perform a wide variety of assembly line tasks. Using only inexpensive equipment we modified a robotic arm to be able to detect objects in its environment; our system was able to successfully identify, manipulate, and sort Tetris pieces. The system is also separately able to detect and react to unexpected objects in its path by stopping all movement until the object exits the area of the arm's trajectory. The major limitations for this system are its inefficiency in the presence of a humans and the lack of a closed-loop system to confirm that it actually grabbed a piece and the piece went to the proper position.

The system we created can be quickly adapted for practical use due to the high level of modularity built in to it. Simply by changing out the high level node the function and the vision identification specifics and possibly adding sensors, the system could be changed to perform any of a variety of tasks, such as sorting recyclables or packing boxes. This allows it to be dropped into an existing factory floor for very little cost compared to a new full system, and will allow for increased safety and productivity if the task being performed by the robot's changes.

VII. FUTURE DEVELOPMENTS

The current arm requires some more modifications in order to make it functional for an assembly line workspace. Due to current limitations of the arm (eg. commands are string-based so it cannot pause in the middle of an executed command), both a fail-safe relay system that would trigger the emergency stop as well as our human-detection and avoidance algorithm that does not allow the command to be executed until the area is safe is necessary. Since the arm ceases operation when an object is present in its original trajectory due to the mentioned limitation, it's not very efficient to pause an entire system until a human leaves the working envelope. Ideally, the arm should be able to recognize the presence of an object and recalculate a trajectory to still complete its mission without hitting the object. Beyond that, a predictive element to analyze manipulator trajectory and possible human arm/body trajectory would improve the effectiveness of recalculating valid paths. Furthermore, a more robust and redundant sensor suite for tracking the pieces would enable the treadmill to move faster, but it would require a motor controller with a greater power capacity. Additionally, playing tetris is a very specific implementation of a pick-and-place system. A different gripper system (e.g. the iRobot Universal Gripper) would allow a greater variety of objects that can be manipulated. Additionally, as we were running our system, it became evident that it is not as robust as we had designed it to be. There is still the potential for better algorithms to be integrated into our system.

In order to validate our improvements to the system, we will run a series of tests to determine the efficacy of our system along several parameters. These include reaction time to an unexpected object in the workspace, percent of time obstacle is not hit, percent of pieces grabbed, and percent of successful placed pieced. The independent variables are treadmill speed, piece type, and piece orientation. This will give us a good set of data points describing how the arm behaves under various conditions.

ACKNOWLEDGEMENTS

Support from Intelligent Vehicles Lab, especially Garland O'Connell and David Barrett and the Olin Bio-Inspiration Robotics Lab (ORB).

REFERENCES

[1] R. Brooks, "Robots at Work: Toward a Smarter Factory," *The Futurist* World Future Soc. Maryland, vol 47, no.3 http://www.wfs.org/futurist/2013-issues-futurist/may-june-2013-vol-47-no-3/robots-work-toward-smarter-factory