122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Seattle Making Value for Society
Paper ID #11932

Development of a Vision-based Sorting Operation Laboratory: A Student Driven Project

Dr. Arif Sirinterlikci, Robert Morris University

Arif Sirinterlikci is a University Professor of Industrial and Manufacturing Engineering and the Department Head of Engineering at Robert Morris University. He holds BS and MS degrees, both in Mechanical Engineering from Istanbul Technical University in Turkey and his Ph.D. is in Industrial and Systems Engineering from the Ohio State University. He has been actively involved in ASEE and SME organizations and conducted research in Rapid Prototyping and Reverse Engineering, Biomedical Device Design and Manufacturing, Automation and Robotics, and CAE in Manufacturing Processes fields.

Ms. Alexandra M Macek , Robert Morris University

December 2013 graduate of Robert Morris University currently employed by the steel industry.

Mr. Bruce Allen Barnes Jr, Robert Morris University

Projected completed as a senior undergraduate student at RMU.

Development of a Vision-based Sorting Operation Laboratory: A Student Driven Project

Introduction

There have been many examples of machine vision system implementation in engineering curricula. However, a few recent ones are noteworthy. Zhuang and Sudhakar developed an undergraduate machine vision laboratory under the sponsorship from National Science Foundation, Cognex Corporation, and Florida Atlantic University[1]. This laboratory supported a 3-credit senior-level machine vision course called, Introduction to Machine Vision. Most of the laboratory experiments were design-oriented and open-ended. A more recent study proposed a comprehensive program to introduce vision technologies to manufacturing and mechanical engineering technology students at Oregon Institute of Technology[2]. The study identified software and computer programming as the major barriers that keep manufacturing and mechanical engineering technology students from learning vision systems and their use in automated and robotic manufacturing applications. Thus, after collaborating with their computer science and software engineering counterparts, the project team took the approach of using introductory "canned" programs for providing basic functionality and tools while encouraging use of some libraries of "code" functionality found on manufacturer's web sites as well as user forums. Finally, pure development of applications was to be carried out with a variety of Applications Programing Interface (API) languages including Visual Basic, C++, C#, and others. Other notable vision projects focused on industrial applications such as "Implementing a Machine Vision System for Robotic Application" by Yeh and Hammond where Wayne State University completed a project for Applied Manufacturing Technologies Inc. [3]. Their paper described the details of an industrially sponsored student vision project at Wayne State for a robot to pick up car wheels from the conveyer line and place them accurately into the drop-off fixture. In a similar effort to generate interest within their own program and from local industries, The Department of Technology at Northern Illinois University responded to strength its curriculum by adding new relevant areas in its automation courses such as machine vision[4]. Within NIU's automation course, basics principles of vision are covered, including camera systems, basic optics, lighting, and image capturing and processing. A key component in this section of their automation course is the hands-on experience where student teams use and apply the vision systems hardware and software in an automated work-cell. In addition, the students are taught the principles of vision integration with other control devices, such as Programmable Logic Controllers (PLCs) and robots.

Industrial Robotics and Automation have long been one of the strengths of this engineering department, especially in its BS Manufacturing Engineering program. In addition, its Learning Factory has a comprehensive automated manufacturing cell including two HAAS CNC centers, two Fanuc robots, bagging and bar code printing machines - all integrated through a SLC 500 PLC and connected through a Siemens conveyor[5,6,7]. This equipment has been utilized along with two other stand-alone Fanuc robots in ENGR 4700 Robotics and Automation and other pertinent courses including ENGR 4950 Integrated Engineering.

Design, a senior capstone course. However, within the last few years, the Vision software Visloc has become outdated and the cell's Cognex camera has gone out of commission. To add a new vision assignment and a work-cell exercise to the curriculum, a new project was envisioned. The project was handled as open-ended and student-driven nature. This project was completed by the students to fulfill their university requirement for a three credit ENGR 4900 Engineering Practice course. A small team of two students worked on the design and development a robotic work-cell that performs a simple sorting operation for quality control, by utilizing multiple pieces of peripheral equipment. The work-cell was intended to be used as a part of the curriculum for future sections of the ENGR 4700 as a vision systems laboratory and an introduction to work-cell design. Further use in ENGR 4950 was also planned to incorporate additional features.

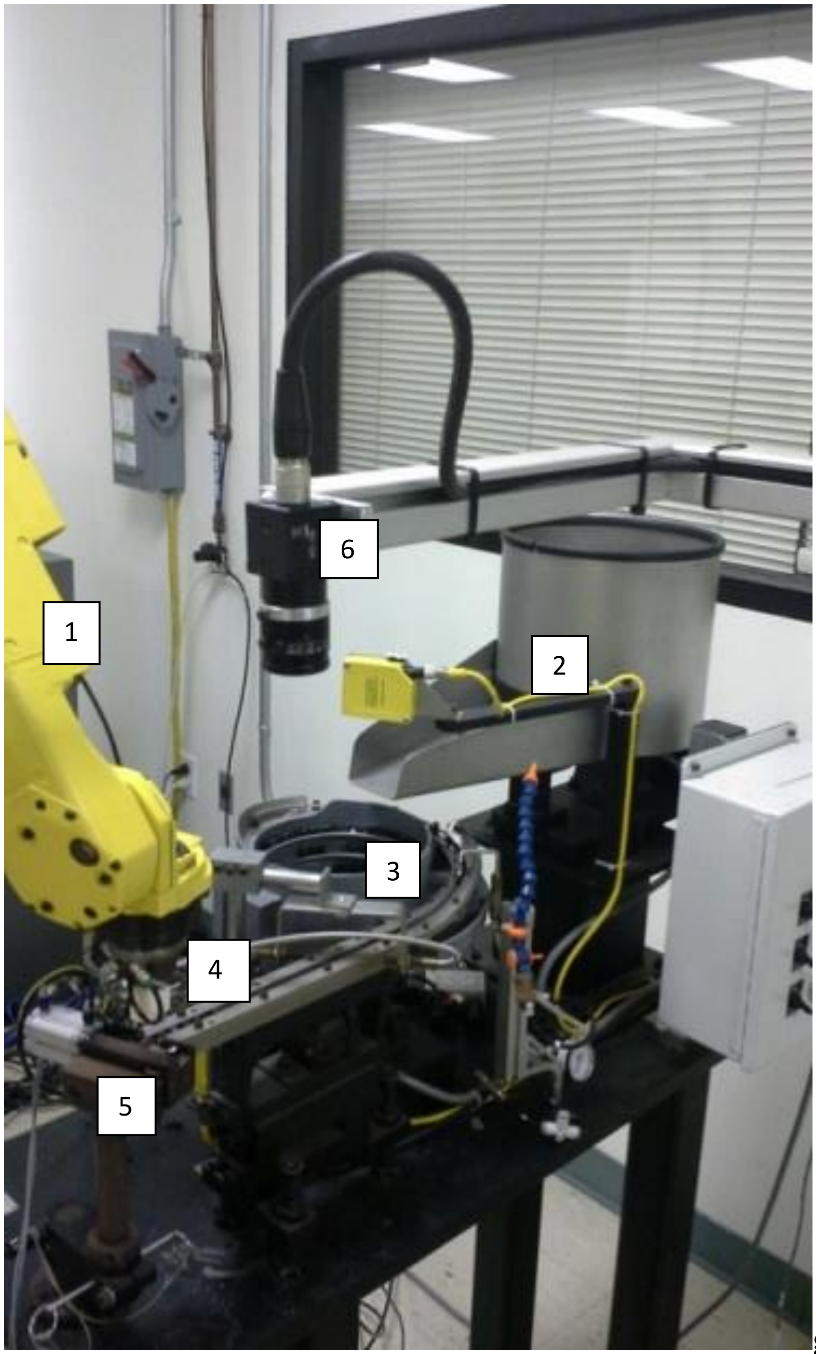
The overall goal of the project was to create a small work-cell, as illustrated in Figure 1, including a bowl feeder part loading system, a Fanuc M10iA robot, and a Sony XC-56 camera to complete a simple part sorting task. The robot picks up a screw from the feeder, presents it to the camera, determines the quality of the part, and deposits the part in the appropriate receptacle.

Project tasks included:

- Wiring of the proximity sensor to allow the robot to check for the presence of a part at the exit of the feeder.
- Wiring the output signal controlling the pneumatic actuator at the exit of the loader.
- Installing pneumatics for a linear actuator in the part loader.
- Connecting the camera into the robot controller.
- Installing modulation software to allow the camera and robot controller to communicate.
- Training the camera to identify "good" and "bad" parts coming from the feeder.
- Program the robot to look for, pick up, and move parts in coordination with other moving components.

The main learning objective for the project was to showcase mastery of past coursework and demonstrate competence to enter the field as an engineer. The practical experience of creating a functioning work-cell with multiple communicating parts is very broad-based, but can be broken into the following segments as illustrated in Table 1 below.

Students were also expected to keep safety a primary concern throughout the entire build process. This was accomplished by following a set of established guidelines outlined in Appendix D.



g

Figure 1. Work-cell with an articulate (1) FANUC robot, (2) parts and (3) bowl feeders, (4) a proximity sensor (located on the gripper), (5) pneumatic part presenter, and (6) a vision system

Project Component	**Associated Course Work**
Practical application of wiring theory learned	ENGR 4700 Robotics and Automation
Increase understanding of how robots communicate with other equipment to complete tasks	ENGR 4700 Robotics and Automation
Refine programming knowledge to include vision system integration	ENGR 4700 Robotics and Automation
Design custom gripper to allow sensor mounting that does not interfere with operation in close quarters	ENGR 4100 Machine Design
Create wiring diagrams and blueprints for replacement parts	ENGR 2160 Engineering Graphics
3D Printing replicas of damaged gripper parts	ENGR 4801 Rapid Prototyping and Reverse Engineering

Table 1. Project components and their association with past coursework

The Build Process

The project took approximately 12 weeks to complete with multiple work sessions per week. Each session was fairly in-depth and was often the fruit of several hours of research done outside of the laboratory. Students tackled the hardware deficiencies first and then covered the software challenges. First, the team ran the wires for the proximity sensor input (DI1), the output signal (DO1) and airline for the pneumatic actuator. The robot controller in the cell used negative common logic and details for the wiring routing and connections can be found in the diagram given in Figure 2. Figure 3 depicts the pneumatic circuits used in actuation for presenting a screw to the robot. The Laboratory Engineer was the primary contact during this phase. After the sensors and actuators were installed and functioning correctly, students wrote a simple program to check gripper function at the pick-up point. A few small adjustments had to be made to the gripper design at this time to allow for compatibility with the feeder equipment. The changes are reflected in the drawings included in Appendix A. The accuracy of the points taught was crucial at this step to ensure the proximity sensor would be close enough to find the screws. The sensor does not sense metal until it is 0.8mm from it, so the error margin during this segment of the program is almost nonexistent.

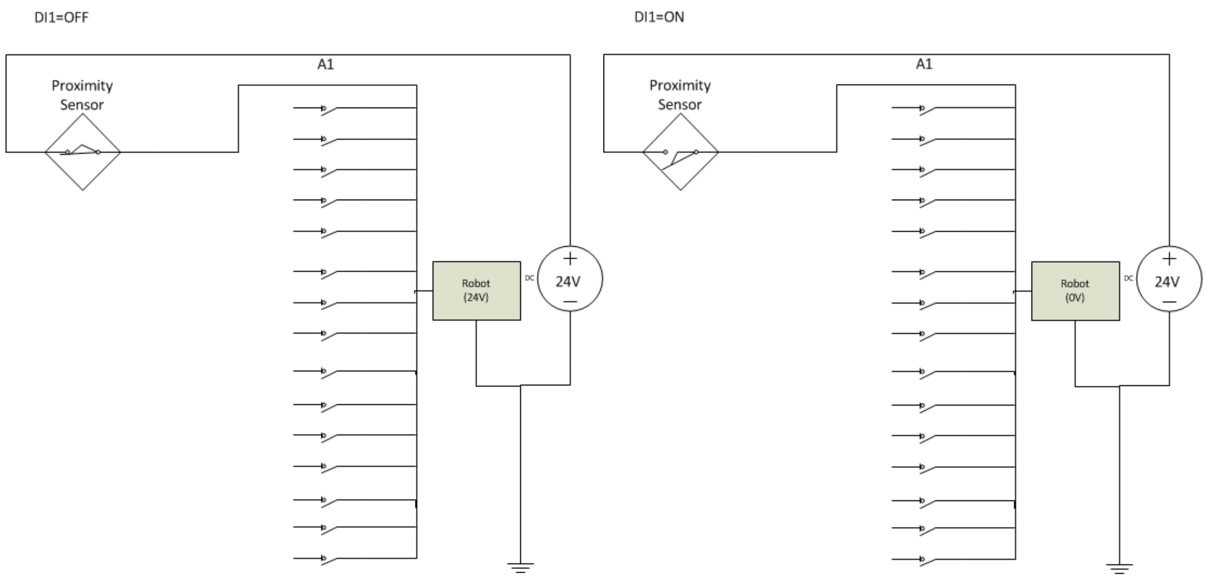


Figure 2. Wiring Diagram for the proximity switch (left) DI1=OFF – no screw is present, (right) DI1=ON – there is presence of a screw at the exit of the loader

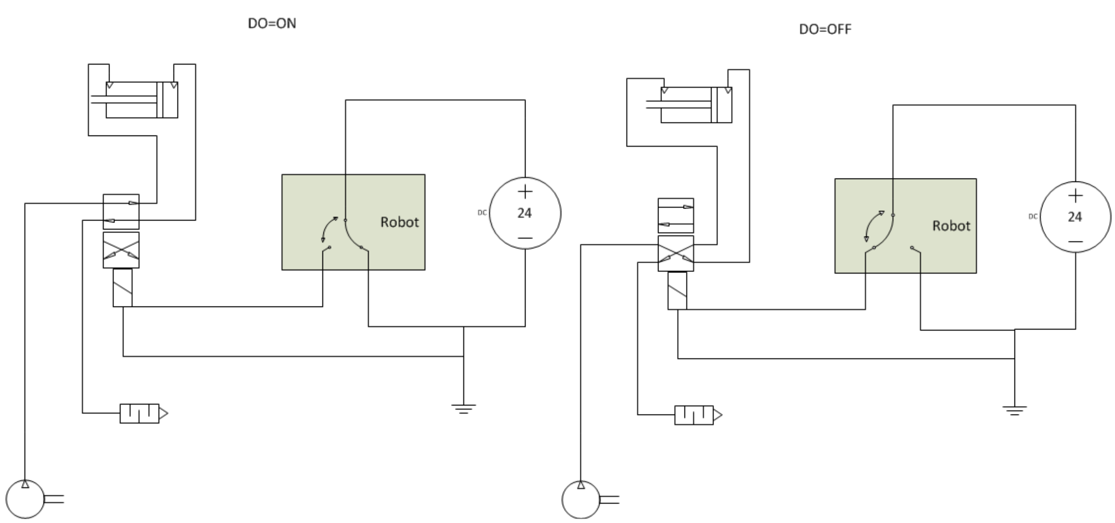


Figure 3. (left) Pneumatics for presenting a screw at the exit of the loader - DO1=ON (right) Pneumatics is inactive – DO1=OFF.

Once the sensor hardware of the cell was in order, students ensured the camera was properly wired and began the software portion of their task. Since only one camera was used, it could be plugged directly into the JRL6 port in the main board of the robot controller. To use more than one camera or a 3D laser, it is necessary to use a multiplexer. The camera is fixed, meaning that it always sees the same image at the same distance. This allows the vision software to function continuously while the robot completes other tasks. A wrist-mounted camera would allow the robot to take measurements at various locations on the work piece and therefore was not appropriate in this application. The Dual Inline Package (DIP) switches on the camera were left in the off position except for number 7 and 8, which were set to on as illustrated in Figure 4. The 75 ohm terminal was turned on as well. HD/VD signal was set to EXT**.** Details of the program and software versions can be found in the Programming and Software section below.

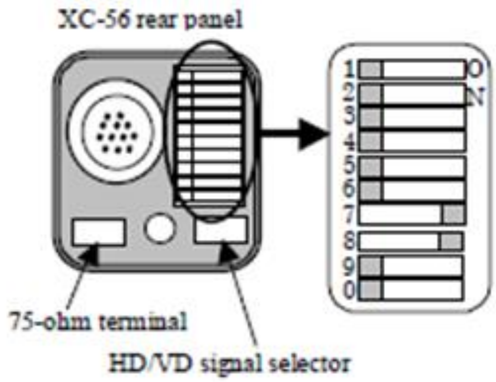


Figure 4. DIP switch settings for the Sony XC-56 camera[8]

Motion optimization and cycle time reduction are relatively simple tasks for this particular setup. Since the robot is carrying a small screw, the mass of the part is, for all intents and purposes, negligible when compared to the M-i10A's maximum payload of 10 kg. This allowed students to use very high acceleration and deceleration values and sharp motion termination types without fear of inertial damage to the robot or part. Consequently, students were able to achieve a massive reduction in cycle time from around 30 seconds to the times pictured below in Table 2.

Trial 1 Cycles	Seconds	Trial 2 Cycles	Seconds
1	8.617	1	7.653
2	7.763	2	8.012
3	8.820	3	8.105
4	8.198	4	7.775
5	8.180	5	7.891
6	8.498	6	7.852
7	8.373	7	7.720
8	8.247	8	8.078
9	8.172	9	7.893
10	8.437	10	7.962
AVG	8.321	AVG	7.894

Table 2: Cycle Times

In a real world manufacturing setting, cycle times are critically linked to productivity. Even minor changes in operating speed can create a massive ripple effect when multiplied by the thousands of parts that could be handled in a given day.

Student Outcomes and Challenges Faced

Students were expected to hone their skills while setting up a hardware system, but there were unexpected learning challenges that arose throughout the project. First, in order to install the iRVision software, students had to update the software on the robot controller. This was a multi-step process that took two days. It required remastering the robot to teach its "home" position and several other small steps that helped give students a wider knowledge of robot setups. It also gave students a better appreciation of the complexity of their software. Students were exposed to different types of sensors through their research as well as how to set up the digital input for (DI1) their chosen proximity switch.

There were a few minor setbacks encountered over the course of the project, but students were able to apply their troubleshooting skills to solve all problems very quickly. Students initially had their black and brown wires (signal and positive potential, respectively) installed in the wrong positions. Once swapped, things ran smoothly. Another challenge arose when one of the students had a collision with the gripper and a piece of loading equipment. The gripper was demolished. The student was able to use rudimentary machine shop knowledge and guidance from the Laboratory Engineer to fabricate a new part and later created a SolidWorks file to allow for 3D printing of replacement parts (Appendix A). It was a chance to learn more advanced machining techniques and also a lesson for the necessity of slow speeds while teaching new points.

The tight working space students had to program in at the screw presentation area was exceedingly difficult to work in due to the nature of robots. The expected motion of the robot and what the robot actually does (or does not do) tend to be two very different issues altogether. There were several close calls with near misses that frustrated students during programming. Again, the importance of slow speeds and knowing the function of each jog key were reiterated.

Software and Programming

The robot operating system (Fanuc's Handling Tool) version installed on the M10iA's R-30iA controller is V7.30P. Fanuc's iRVision was also used to interface with the XC-56 camera. Laura Evans of Fanuc Robotics helped the team with Handling Tool and iRVision set-up. To teach the camera, students temporarily added a laptop to the cell. Setup was completed using Windows 7 operating system, Internet Explorer 9, and Internet Protocol Version 4. The PC communicated with the controller via Ethernet by setting the PC's IP address to 192.169.0.2. The sample laboratory guidelines in the appendices of this paper can be referred in how to set the PC's IP address.

Students then taught the necessary masks to the camera so it would be able to identify the type of screw presented by the robot. Since the camera focuses on contrasts, it does an excellent job of identifying the black ("pass") screw head against both the previous aluminum and current ABS (3D printed) robot gripper. The white ("fail") screws do not present enough contrast to be passed. Screws that are partially painted appear to have an irregular geometry also fail, as required.

To access iRVision, students needed to open Microsoft Internet Explorer 9 and access the web address http://192.168.0.1, the robot controller’s IP address. Once accessed, the iRVision home screen appears as shown below in Figure 5. From the homepage, one then can access the *Camera Setup tool, Camera Calibration tools, Vision Process Tools, Application Setup Tools, Robot Ring, and Help*- listed on the left portion of Figure 5.

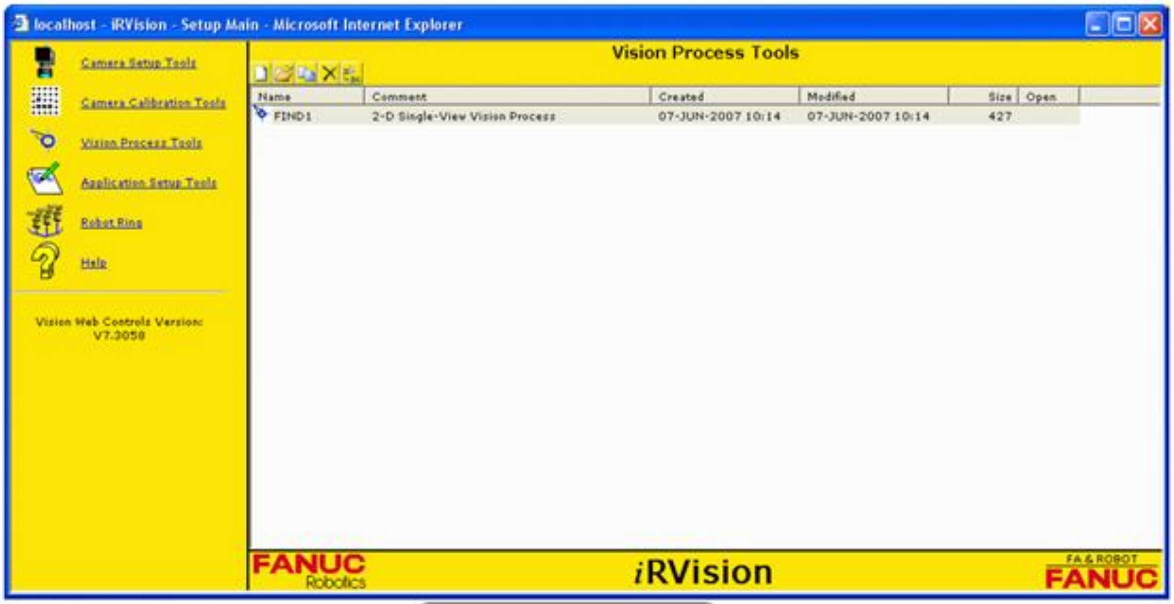


Figure 5. iRVision Software interface[9]

- The first option, Camera Setup Tool, is an important first step whenever setting up a work-cell utilizing vision software. Given the current setup of the cell, there should be no need to vary any parameters herein except the exposure time. The iRVision is set to work with the Sony XC-56 Progressive Scan Camera that uses a fixed mounting configuration and has a 33.333 ms exposure time. The exposure time is set to the factory default exposure time.
- Once the camera was properly set up, it was calibrated. The calibration was completed using the Camera Calibration Tools. To calibrate the camera a 10 mm center to center grid was used. Students selected this grid because it was the nearest in size to that of the screws to be checked. The calibration grid was centered in the camera at a distance of 12 inches from the bottom of the lens. Once centered, the software was configured to match the real-world settings. An image of the grid was then snapped and saved. The image is stored so the software can properly size any object in its field of vision at the given calibration distance.
- Having calibrated the camera, the students were then able to select an appropriate vision process. The following processes are supported by iRVision: 2D single view vision process, 2D multi-view vision process, depalletizing vision process, 3DL single view vision process, 3DL multi-view vision process, single view vision tracking, and error proofing. Since the purpose of the cell was to sort screws by color,

the error proofing vision process was chosen. The error proofing vision process searches an image for deviations from previously taught reference geometry. When the deviations exceed a specified threshold, the image fails and the robot discards the object. It should be noted that because this vision process operates on pixel variations and not on relative part position, the grid calibration step is negated. Although the robot was calibrated, the calibration does not affect the program's functionality. Further, due to the error proofing vision process students had no need for the Application Setup Tools and the Robot Ring tool. For further information regarding types of vision processes, please see Appendix B.

The actual Teach Pendant Programming (TPP) code for this project is saved on the controller as "SCREWSORT". The syntax was fairly simple and utilized if-then logic and registers. For the complete code, please refer to the annotated code in Appendix C.

Future Use of the Work-Cell

This cell was created primarily to serve as a learning medium for future sections of ENGR 4700. The simple input and output wiring systems in place can be easily followed and replicated by other students learning the fundamentals of robotics and device control. Students recommended the creation of three larger laboratory assignments:

- The first laboratory would correspond with the unit on input and output wiring diagrams. The terminal ends of the input and output would be removed from the robot controller by the instructor prior to the lab. Given instruction on the operation of digital signals, students should be able to provide the predicted voltage at each terminal in both the on and off condition of the circuit. Students would then be charged with finding the correct setup as described in the wiring diagram in Figure 2.
- The second exercise would be included in the programming unit. KAREL and TPP are both covered in the course, and the program written to run this cell is a fundamental example of what can be accomplished with TPP. Recreating the entire program could account for a higher-level lab. Flowcharting the logic could serve as a lower-level lab or homework assignment. Ideally, students would be asked to storyboard, flowchart, pseudo code, and then TPP code the program as a several-week lab exercise or project. Students could also be asked to storyboard or flowchart the logic behind the sensors in the loader cell logic for an introductory exercise in programming.
- The third laboratory would be one in which students created a vision process. The error proofing vision process is the recommended process for use by first time users due to its relatively simple programming. Students performing the lab would be required to access the iRVision software through a PC and then teach the required geometry or geometries for passing parts. Students would then have to use the teach pendant to create a simple program where the robot presents objects to a camera and then runs the appropriate vision process to sort the objects. An example may be to present the camera with two different pencils and have it sort them by passing the one that matches the taught geometry and failing the other. A sample laboratory guideline,

iRVision Process Laboratoty, originally envisioned as being included in Appendix F will be distributed to the participants of the session.

One unexpected benefit of the cell is its use in departmental tours and as an interest piece for the department. The open laboratory design makes it a focal point during a walk-through. During work sessions, students often stopped their task to explain what they were working on to groups of prospective students, curious faculty, and even their peers. Presenting the Learning Factory and demonstrating that the engineering department's student body is actively honing their practical skills is a huge selling point. Students hope they have encouraged more than a few prospective students to apply to or seriously consider this engineering department.

By engaging current students in discussion about the project, more interest was generated in building practical skills in addition to the theoretical knowledge gained through lecture. Other students offered suggestions to the project team and expressed interest in pursuing other large robotics projects in the Learning Factory.

Discussion and Conclusions

One of the hallmarks of a successful engineer is an awareness of what facets of a project could be improved. While students seized many of the learning opportunities available and made a very successful cell, they wanted to mention some areas that could benefit from further adjustment.

The robot code was fairly straightforward, but could have been improved by a few simple additions. Firstly, the application of the failed and passed parts counting register would be useful in a real world setting. The register increments for every failed part, and if the register exceeded a predetermined percentage of a known lot size, the process could be aborted and restarted pending inspection by a human operator. Instead of aborting, the robot may instead turn on an output which could light up or sound a siren to alert quality assurance personnel if there is an exceptionally high fail rate. Once the passing screw counter increments to a certain number, the pass register could be used to allow the cell to communicate with packaging equipment that the bin was full and ready to be shipped.

Also, students believe there is room for improvement in the loader section of the cell. The in-line parts vibrator is excessively loud and would benefit from a redesign or some adjustment to reduce the noise it generates. The section of screw handling track that connects the inline vibrator to the bowl feeder is also slightly misaligned, which can cause screws to bind up as they travel. The air nozzle in the cell could be hooked up to blow caught screws down the line, or the track could be readjusted for a better connection.

The XC-56 camera is capable of panning and tracking moving parts and making incredibly precise measurements with high repeatability in a limited time frame. Given the camera's capabilities, students recognized that it could be used for more complicated applications such as in-line tracking - locating a part on a moving conveyor. Replacing the camera with a more simplistic model and installing the XC-56 in more advanced capacities in the future would allow for greater utilization of university resources.

Improving the lighting conditions of the work-cell would likely further increase the cameras ability to capture accurate images. Currently, the error proofing process is set to operate with maximum tolerances. Since the failed screws have a largely different geometry than the passing screws, the large tolerances do not hinder the ability for the camera to fail incorrect screws. When the tolerances were tightened, screws that should have passed began to fail. The issue may be largely due to lighting. It is possible that because the lighting being used is fluorescent the images are inconsistent and therefore not accurate. By incorporating a light source that does not fluctuate in intensity over time, the images would be more consistent. Another option would be to replace the current fluorescent light ballasts with ballasts that have a higher pulse frequency. This would cause the lights to flash enough faster than the camera exposure time to creature an average lighting effect. Both cases may provide greater image accuracy and thereby allow the camera to work with tighter tolerances.

Finally, cosmetic adjustments could be made to give the cell a more polished appearance. Due to the nature of the wiring harness, students were forced to run the wires for various components outside the silver loom on the robot frame. While completely safe and functional, the wires are not the most aesthetically pleasing feature of the cell and future work may include adapting the harness loom to cover them.

Ultimately, students feel that they have demonstrated mastery of the essential tenets of automation and design. They have showcased skills gained in their project during their job interviews. Both are employed as engineers at the US Steel Corporation and a FANUC Equipment Integrator.

References

[1] Zhuang, H., Sudhakar, R., Development of a Machine Vision Laboratory, 2003 ASEE Conference, Nashville, TN.

[2] Floyd, M., Kim, H., Culler, D.E., A Proposal to Implement a Course on Vision Systems with Applications in Robotics at the Oregon Institute of Technology. 2013 ASEE Conference, Atlanta, GA.

[3] Yeh, C., Hammond, H., An Industry Based Student Project: Implementing A Machine Vision Systems For Robotic Application, 2002 ASEE Conference, Montreal, Canada.

[4] Otieno, A.W., Mirman, C. R., Machine Vision Applications within a Manufacturing Engineering Technology Program, 2002 ASEE Conference, Montreal, Canada.

[5] Sirinterlikci, A., Development of a Comprehensive Industrial Controls Course in a Manufacturing Engineering Program, 2006 ASEE Conference, Chicago, IL.

[6] Sirinterlikci, A., Utilizing Robotics in Teaching Microcontroller Programming to Manufacturing Engineering Students, 2009 ASEE Conference, Austin, TX.

[7] Sirinterlikci, A., Practical Hands-on Industrial Robotics Laboratory Development, 2012 ASEE Conference, San Antonio, TX.

[8] Accessed on December 27, 2013 Sony XC-56 Camera Manual http://ccddirect.com/index.php?dispatch=attachments.getfile&attachment_id=100

[9] FANUC iRVision ONLINE Tool and Documentation.

Appendix A: Gripper Design

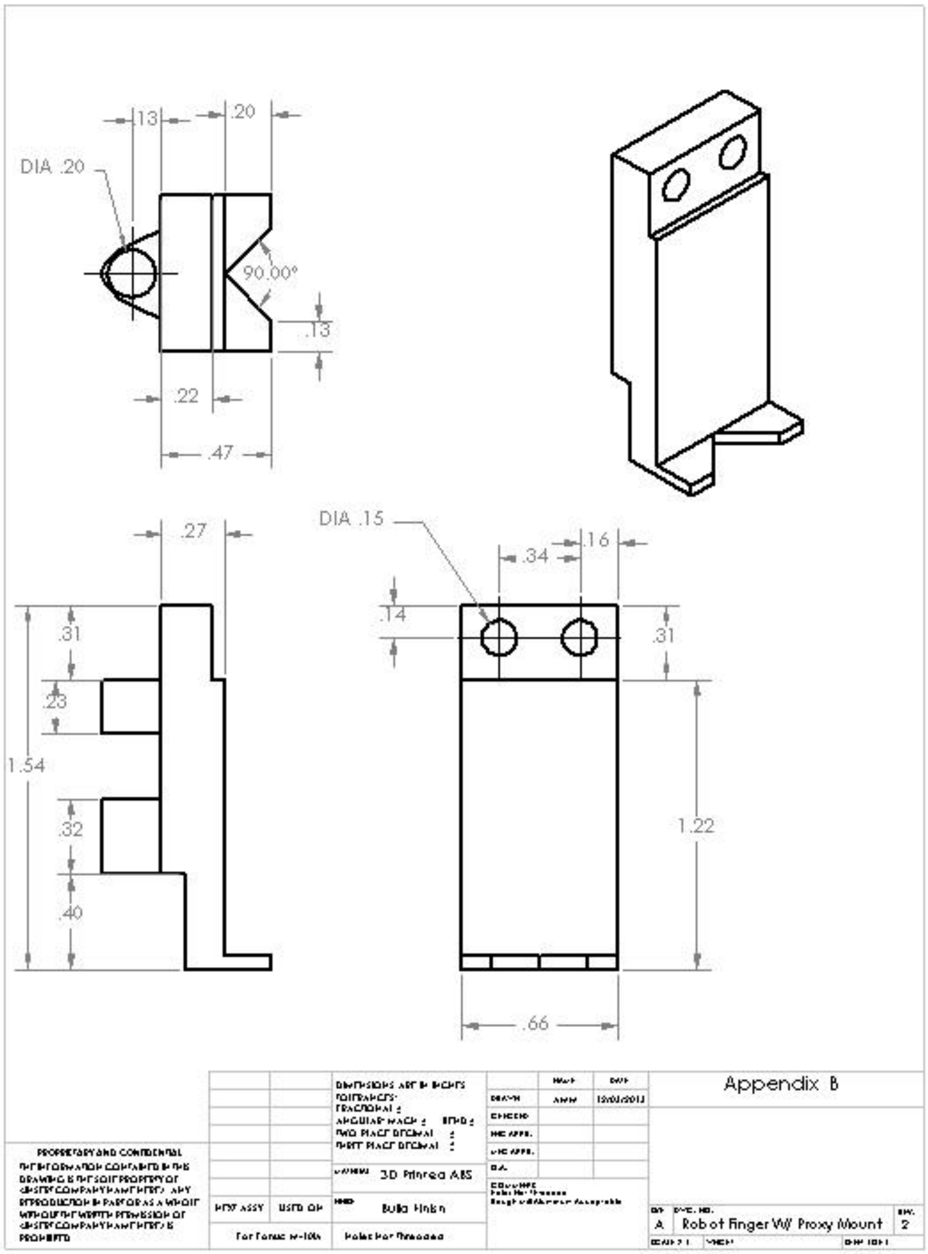


Figure A1. Solid model of the gripper

Appendix B: Vision Processes

2D single view vision process

iRVision detects the position of a work-piece in two dimensions and then offsets that position relative to a known robot position so that a robot can find the given work-piece.

2D multi-view vision process

iRVision detects the position of a work-piece in two dimensions through the use of multiple cameras and then offsets that position relative to a known robot position so that a robot can find the given work-piece. Typically this process is used when a work-piece is too large to fit within the view of a single camera. The process overlays separate images to generate an image of the whole part.

Depalletizing vision process

This process allows for a vertical direction offset in addition to the normal two-dimensional offsets. In this scenario, the camera is typically mounted to the robot. As the robot moves closer to a part, the camera determines how close the robot is by measuring the apparent size of the part see Figure .

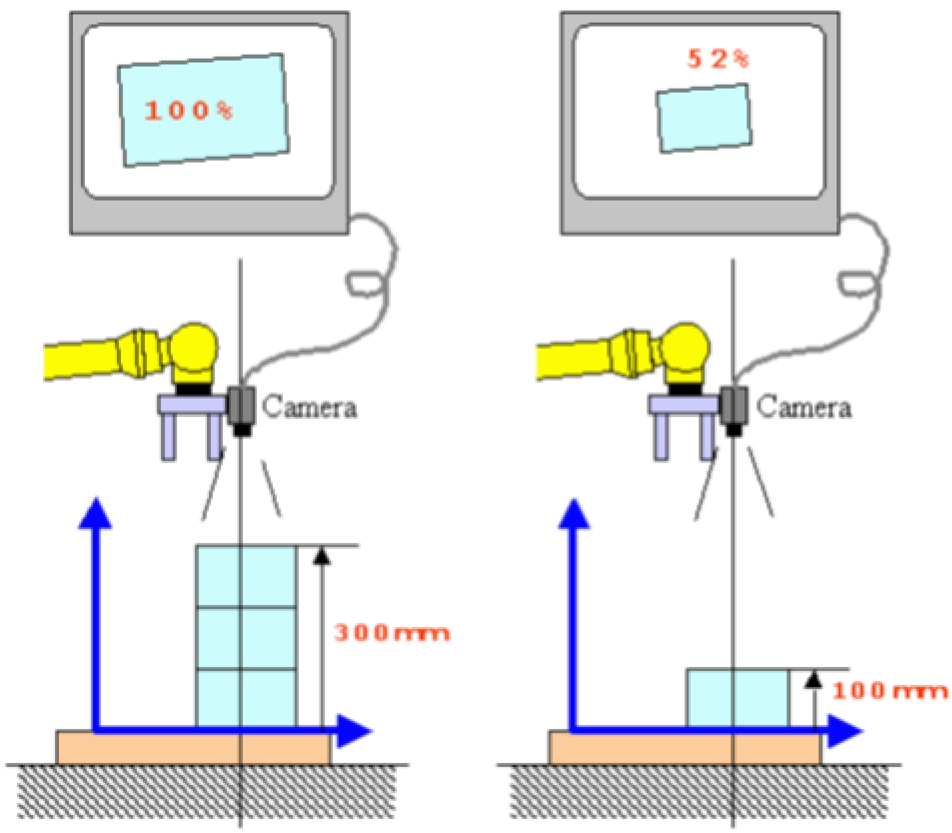


Figure A2. Depalletizing vision process[9]

3DL single view vision process

This vision process measures the position and orientation of a work-piece in three dimensions so that the handling of the work-piece by the robot can be appropriately adjusted.

3DL multi view vision process

This vision process measures the position and orientation of a work-piece in three dimensions using multiple sensors so that the handling of the work-piece by the robot can be appropriately adjusted. It is most useful when the work-piece is too large to be viewed by a single sensor or when the work-piece in is oriented at an inconvenient angle.

Single view visual (in-line) tracking

This vision process uses a single camera to locate the position of a given work-piece on a conveyor so that a robot can pick up the work-piece without stopping the conveyor.

Error proofing

This is the only vision process that is not used for locating the position of a work-piece. This means does not require any position offsets and therefore does not require the calibration of a robot using the calibration grids. The process captures an image and searches it for deviations from the expected image mask created by the user. If the deviations are too large, the part fails.

Appendix C: Annotated Teach Pendant Program

SCREWSORT	COMMENTS
R[3:VISPASSFAIL]=0	Initialize vision process register
R[4:FAILCNT]=0	Initialize failed screw counter register
R[5:PASSCNT]=0	Initialize passed screw counter register
R[1:ARRCOUNT]=0	Initialize failed screw arrival counter
LBL[1]	
L P[2] max_speed CNT50	Safe point
LBL[3]	Sensor operation
IF R[1:ARRCOUNT]=3, JMP LBL[5]	If the screw failed to arrive after three attempts, then jump to end of program
J P[3] 75% CNT75	Sensor process approach point (AP-1)
DO[1]=ON	Actuate Sensor to retrieve screw
WAIT 1.0 sec	Wait for the actuator to reach its initial position
DO[1]=OFF	Actuate Sensor to move to pick up location
WAIT 1.00 sec	Wait for the actuator to reach its final position
L P[9] 100 mm/sec FINE	Move down to determine if screw is present
R[1:ARRCOUNT]=R[1:ARRCOUNT]+1	Increment screw arrival counter
IF DI[1]=OFF JMP LBL[3]	Is screw present? (if not go to sensor operation label)
IF DI[1]=ON JMP LBL[2]	Is screw present?
LBL [2]	
R[1:ARRCOUNT]=0	Reset screw arrival counter (next screw to come will now get three attempts to make it)
J P[3] 75% CNT100	Move to AP-1
J P[5] 75% FINE	Move to pick up approach point (AP-2)
RO[1]=ON	Open fingers
L P[6] 100 mm/sec FINE	Move to pick up location
RO[1]=OFF	Close fingers
L P[5] 250 mm/sec FINE	Move to AP-2
J P[7] 100% CNT100	Move to arbitrary midpoint to avoid any collisions
J P[8] 75% FINE	Move to point beneath camera and rotate wrist to present screw to camera
VISION RUN_FIND 'SCREWERROR'	Find and run the iRVision program SCREWERROR
VISION GET_PASSFAIL 'SCREWERROR' R[3]	After the program runs it will return a 0 for pass, a 1 for fail and a 2 for unable to determine and store it in register 3
IF R[3:VISPASSFAIL]=0,JMP LBL[8]	If the screw passes the vision test jump to label 8
IF R[3:VISPASSFAIL]=1,JMP LBL[7]	If the screw fails the vision test jump to label 7

IF R[3:VISPASSFAIL]=2,JMP LBL[5]	If the test resulted in an error and could not be determined if passed or failed jump to label 5
LBL[8]	
R[4:FAILCNT]=R[4:FAILCNT]+1	Increment counter for number of failed screws
J P[11] 100% FINE	Move to point above failed screw cup
WAIT 0.05 sec	Wait 0.05 seconds before dropping screw
RO[1]=ON	Open fingers to drop screw
WAIT .50 sec	Wait 0.5 seconds for fingers to open
RO[1]=OFF	Close fingers
JMP LBL[1]	Repeat process
LBL[7]	
R[5:PASSCNT]=R[5:PASSCNT]+1	Increment counter for number of passed screws
J P[10] 100% FINE	Rotate wrist so the screw can fall while moving to the screw pass cup
WAIT 0.05 sec	Wait 0.05 seconds before dropping screw
RO[1]=ON	Open fingers to drop screw
WAIT 0.5 sec	Wait for fingers to fully open and drop screw
RO[1]=OFF	Close fingers
JMP LBL[1]	Repeat entire process
LBL[5]	Terminate program
L P[3] 100 mm/sec FINE	Move to AP-1 to avoid collisions
L P[2] 100 mm/sec FINE	Return to safe position

Appendix D: Robot Safety Notes

- All operators and work cell observers must wear safety goggles
- Remain outside the black and yellow tape while robot controller is on
  - Robot may move without warning
  - Black and yellow floor tape marks the limits of the robot work volume
- Never operate the robot alone
- Always assume the robot will move at full speed even if programmed otherwise
- Never assume robot will move as anticipated
- While manually jogging the robot use an appropriate speed to manage the robot (first time: >10% recommended away from other objects and <10% near the objects within the cell)
- Do not manually jog robot without knowing what each jog key will do
- While using any pneumatic, hydraulic, and/or electric actuators, allow sufficient time delays in program for actuators to finish their motion before robot moves due to collision avoidance.

Appendix E: Troubleshooting

- **Robot will not move**
  - Reset any faults
  - Set frame to jog
- **Camera will not properly sort screws**
  - Make sure camera is focused
    - Line up the white marks on the camera lens
  - Avoid shadowing the imaging area
  - Check that dip switches on camera still match the image in Figure 4
  - Ensure the robot gripper is accurately presenting screws