INVESTIGATION INTO A LOW COST ROBOTIC VISION SYSTEM

G.S. Sandhu, C. McGinn and K. Kelly

Department of Mechanical and Manufacturing Engineering, Trinity College Dublin.

ABSTRACT

Industrial robots are of great economic and technological importance within the manufacturing industry due to the wide range of benefits that they provide over their human counterparts. In recent times, technological developments driven by industry demands have led to the development of more autonomous robots. However, the increased strain on production cycles and the demand for higher quality products outlines the need for these robots to be more flexible and adaptable. Existing robot vision systems are generally very expensive (A top of the range industrial vision system from Bosch is priced at $120,000) and would be unaffordable to many small to medium sized businesses. This has created a further need for a more cost effective option which encompasses both the heightened productivity of an industrial robot with the adequate amount of dexterity required to complete its tasks. This paper presents an investigation into a low-cost solution for real-time pick & place operations. A modular approach was taken to complete this model in which it was broken down and its constituent parts – sensing/perceptual capabilities, 3D image processing/manipulation and robotic manipulators – were connected using the Robot Operating System (ROS), a comprehensive tool used to provide an extensible inter-process communication framework. The sensory capabilities were provided by a depth sensor, in the form of a Microsoft Kinect; which utilised the OpenNI drivers and Point Cloud Libraries to perform segmentation and clustering techniques. These techniques enabled the system to detect and isolate individual objects from 3D point clouds. Further to this, motion planning algorithms were developed within the ROS Industrial platform in order to drive an ABB IRB 120 robotic manipulator and a servo-electric end effector was incorporated within the system to provide the ability of grasping the objects to perform real time pick and place operations on the objects outlined by the Kinect.

KEYWORDS: Robotics, Real-Time Perception, Microsoft Kinect

1. INTRODUCTION AND BACKGROUND

The field of robot vision guidance is developing rapidly and robotic vision systems are now commonplace in many automated manufacturing and packaging processes. The benefits of sophisticated vision technology include savings, improved quality, reliability, safety and productivity. Robot vision is used for part identification, navigation and organising. Vision applications generally deal with finding a part and orienting it for robotic handling or inspection before an application is performed. Sometimes vision guided robots can replace multiple mechanical tools with a single robot station. A year-on-year increase of 24% in robot vision systems [1] undoubtedly suggests that companies within the manufacturing industry have started recognising the countless benefits of utilising this technology. A combination of vision algorithms, calibration, temperature software, and cameras provide the vision ability.

Existing vision systems incorporate a wide range of technologies such as cameras and image sensors, lasers, lighting sources, optical filters and so on. Current robotic vision systems available on the marketplace today can generally be divided into two categories: ‘Pure’ vision systems comprising solely of the sensors and software required and ‘complete’ vision systems which also contain an incorporated robot.

One of the main issues with many (not all) of the pure vision systems available on the market is the fact that the systems can typically only be integrated with robots which are produced by the same company. An example from the top end of the spectrum is the FANUC Integrated iRVision System which is advertised as the “only fully integrated vision system available in the robotics industry” [2]. Using an advanced 3D laser vision system incorporating both cameras and lasers it enables robots to not only perform ‘bin-picking’ operations on static objects but also identify, distinguish between and pick up different products off high speed moving lines, error proof parts/products and handle a wide range or object sizes ranging from small metal stampings to large metal sheets. While the features of this vision system are certainly impressive, the main downfalls are that it cannot be used on non-FANUC robots and also comes with a high price tag, potentially reducing its use by many SME's that may not be able to afford the system itself or may be using a robot manufactured by a different company. This problem holds true for pure vision systems on the cheaper end of the spectrum also, as highlighted by the Motoman 'MotoSight 3D Spatial Vision system' which is listed at $11,000 and provides the ability to pick and place 6-10 parts per minute [3] but again can only be incorporated with Motoman robots.

The Bosch delta robot with vision system is a 'complete' robot vision system and is priced at $120,000 [4]. This certain model is supplied with an advance 3D vision system and a 2-arm Delta Robot Cell capable of performing over 80 pick and place operations per minute along with object recognition/classification, motion tracking and also perform product quality inspections based on geometry and CAD drawings. The main problem associated with 'complete' vision systems such as the aforementioned, is the fact that they generally have a much higher price tag associated with them, primarily due to the fact that they are supplied with a robot itself. The robots in these systems also tend to be quite large in size with advanced specifications which could potentially be excessive in terms capacity/output for some SME's.

Hence, there is a need for a robotic vision system that is capable of being used with robots that a manufactured by a wide range of companies and also one that is cheaper in price than the industry average, making the technology more accessible to small-medium sized enterprises.

1.1. Perception

Sensory capabilities are vital if a robot is to function autonomously in any unknown/partially specified environments, if it is to carry out complex tasks and also if it is to interact with the world around it. Robots produce corresponding actions based on perceptual information. Among a variety of sensing methods is vision; the primary sensing modality. Visual perception in particular plays a key role in the behaviour of humans [5]. The hand-eye coordination ability gives us flexibility and robustness of movement that no machine can currently match. From locating and identifying to grasping and handling objects, we often rely heavily on our visual sense – over half of the human sensory cortex is attributed to seeing.

However, visual perception within robotics is a compellingly difficult problem due to the unavailability of vast amounts of information about surrounding environments for robots. Much research has been conducted on object recognition and environment scanning to date [5]–[8]**.** A promising recent development in this field is the integration of 3D visual information. Scanners/Sensors are used to provide robots with a platform capable of using 3D vision to recognise and react accordingly to their surrounding environment.

1.2. Depth Sensor

The perception aspect for this project ultimately is provided by the sensors that are integrated within a complete model for real time robotic perception. 3D data provides extra information to a robot, such as distance and shape, and enables different approaches to identifying objects in within the vicinity of the robot. The depth sensor will be responsible for analysing the environment, object detection & object recognition.

The Microsoft Kinect sensor was chosen as the depth sensor to be integrated with the system to provide perceptual capabilities. The Kinect sensor is used for analysing the environment around the ABB robotic arm and in particular for object detection and isolation. Microsoft launched the Kinect in November, 2010, in order to add a new and innovative breed of entertainment to its Xbox 360 gaming console. However, the Kinect's low price and impressive quality of the depth information produced by it immediately caught the attention of researchers and software developers alike, resulting in the development of open source drivers which facilitated its use for more diverse applications. By means of these drivers, researchers have integrated the Kinect with systems used in room mapping, surveillance, surgical application and more importantly; robotics. These drivers also allow the information produced by the Kinect to be represented as a cloud of points located in 3D space known as point clouds. A point cloud is simply a set of data points (X, Y, Z) represented in a three-dimensional coordinate system and is intended to represent the external surface of an object [9]. The innovative technology behind the Kinect is a combination of both hardware and software contained within the flat black casing. There is a trio of hardware innovations that work together within the sensor:

- Colour VGA video camera
    - This video camera aids in facial recognition and other detection features by detecting three colour components; red, green and blue. Also referred to as an RGB camera based on the colours it detects.
- Depth Sensor
    - Comprises of an infrared (IR) projector and a monochrome CMOS (complimentary metal-oxide semiconductor) that work to see the environment in 3D regardless of the lighting conditions.
- Multi-array microphone
    - An array of four microphones that can isolate the voices of the users from the background noise of the environment.

Most infrared projectors used in depth sensors paint a grid on to the scene with invincible markers identified by numbers. Same numbers on a grid represent that they are sitting on the same depth level: 1’s are on a certain level, 2’s on another level and so on. The Kinect on the other hand uses a speckle pattern instead of the number grid, where the local speckle pattern in any given area is unique. As the angles between the speckles of every one of these groups is known, it is possible to triangulate the different distances and hence act as a depth sensor [10].

The speckle pattern is projected onto a scene by means of the IR projector and detected by the IR camera. A reference pattern of the speckle pattern is hard coded into the Kinect at the manufacturing stage. An algorithm used by the Kinect chooses a particular dot in the reference pattern and then searches for that dot in the observed scene by also looking for its eight unique surrounding pixels. The disparity can then be determined and used in combination with the focal length of the IR camera used to detect the speckle patterns and the baseline between the projector and the camera in order to determine the depth of that given point in the scene [11].

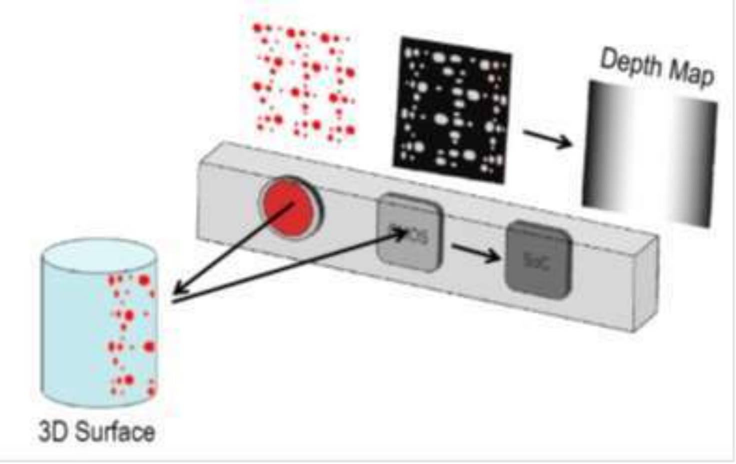


Figure 1: Operational schematic of a Kinect sensor

1.3. Point Cloud Library

Point Cloud Library (PCL) is a free, BSD licensed (BSD licenses are a family of permissive free software licenses, imposing minimal restrictions on the redistribution of covered software), comprehensive library presenting an advanced and extensive approach to the subject of 3D perception, providing support for common 3D building blocks that applications need. The library contains algorithms for filtering, segmentation, reconstruction and feature estimation and is supported by an international community of robotics and perception researchers [12]. The PCL is separated into a number of modular libraries that each develop their own algorithms that contribute to the entire PCL framework. PCL has been fully integrated with ROS and has been used by an increasing number of researchers in a wide range of projects in the field of robotics [13]; Rusu demonstrated the down-sampling of a point cloud by using a 3D voxel grid filter through PCL. This was accomplished by creating a 3D voxel grid (voxel grid can be seen as a set of tiny 3D boxes in space)[14] over the input cloud data. Then, all the points present in each voxel were down-sampled with their centroid. Rusu‘s work made use of the plane edge detection algorithms developed by Choi [15], a PhD student at Georgia Institute of Technology. These algorithms found points bordering on those whose coordinates are set to NaN value, hence allowing him to compute the absolute boundaries of the plane.

1.4. ABB IRB 120

ABB Robotics is a leading supplier of industrial robots and modular manufacturing systems, having installed over 200,000 robots worldwide [16]. The ABB IRB 120 is ABB‘s smallest multipurpose industrial robot. The 6-axis robot weighs 25kg and can handle a payload of 3kg with a reach of 580mm. Designed with a light aluminium structure; the powerful compact motors ensure that the robot is enabled with fast and accurate acceleration in any application [17]. The ABB IRB 120 comes with the IRC5 Compact controller which takes the capabilities of the extremely powerful IRC5 controller and presents them in a compact format, bringing a level of accuracy and motion control that previously was exclusive to much larger robots.

1.5. Robot Operating System (ROS)

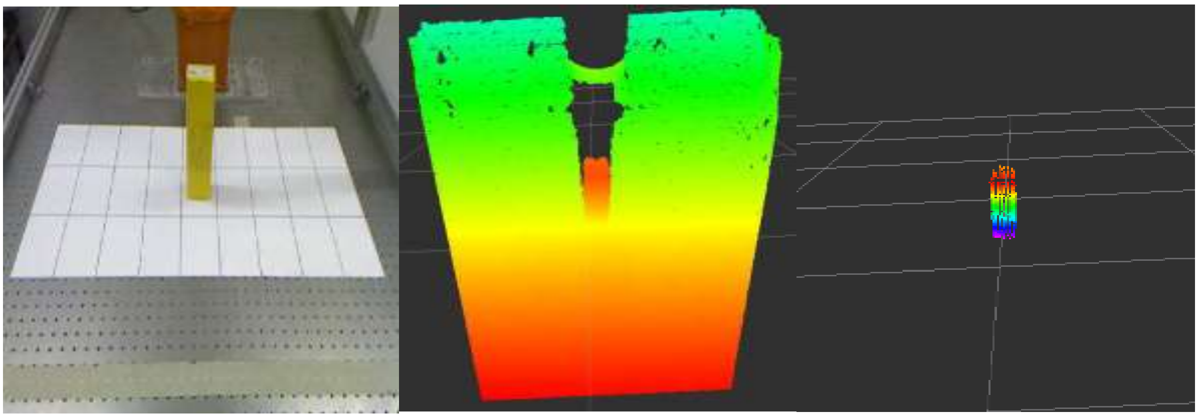
The Robotic Operating System (ROS) was used to develop the framework for the presented model. ROS is the comprehensive tool that was used for everything in this project, from developing a simulation of an industrial robotic arm for testing, controlling an actual robotic manipulator, to integrating sensor feedback to allow for object detection and ultimately allowing real-time pick and place of desired objects. ROS is a collection of software frameworks for writing and developing robot software. It is a collection of tools, libraries and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms [18]**.** ROS was developed and built from the ground up to encourage collaborative robotics software development because from a robots perspective, problems that seem trivial to humans often vary greatly between instances of tasks and environments and it is dealing with these variations that is so hard that no single individual, laboratory or institution can hope to do it on their own [19]. The building blocks of any ROS application are nodes. Nodes are essentially independent programs that can run simultaneously and must be able to communicate with one another. This communication is facilitated by the ROS master. Nodes communicate with one another in the ROS environment through the use of data structures called messages; which can consist of multiple types of data such as strings, integers and many more. Nodes send information by publishing messages to a topic which allows other nodes to subscribe/listen to these messages and use the information accordingly.

2. SYSTEM INTEGRATION

The presented model is divided into two separate nodes: a perception node; tasked with detecting and isolating objects using the point cloud data provided by the Kinect and, the motion planning node which is tasked with driving the robotic manipulator in order to perform pick & place operations.

The raw point cloud data provided by the Kinect sensor has over 300,000 data points per frame [20]. Naturally, processing this data comes at a high computational cost. Therefore, three different filtering techniques were implemented to achieve the processing speeds that would be required for real time situation assessment. These filters work on the principle of downsizing/down-sampling the information when maximum resolution is not required such that fewer data points remain but the important information of the point cloud is preserved. The filtration techniques remove any unwanted points from the point cloud, leaving the table and any objects that have been placed on top of it. Following the filtration steps, comes the planar segmentation which performs a plane fit to the points to find which points comprise the table and then subtracts this from the point cloud itself. This results with only the objects on top of the table remaining in the point cloud.

The final stage in detecting objects in the point cloud is done by using the Euclidean cluster extraction technique, which works by separating the points into groups where each member of a group is within a specified distance of at least one other member in the group. Each cluster/group of points represents an object in the point cloud. Once the objects have been detected and isolated, the position of each object in relation to the Kinect sensor must be found. Since the position of the Kinect is different to the position of the robot, the locations of the objects relative to the robot itself must be computed. Euler's rotational angles are used to carry out the required transformations between the different coordinate systems of the Kinect and the robot. Finally, the object coordinates are published to the motion planning node which drives the ABB IRB 120 robotic manipulator to perform pick and place tasks in a specified sequence. A servo-electric parallel gripper was used as the end effector to grasp the objects and was controlled manually using the ABB robot's *Flexpendant* controller.

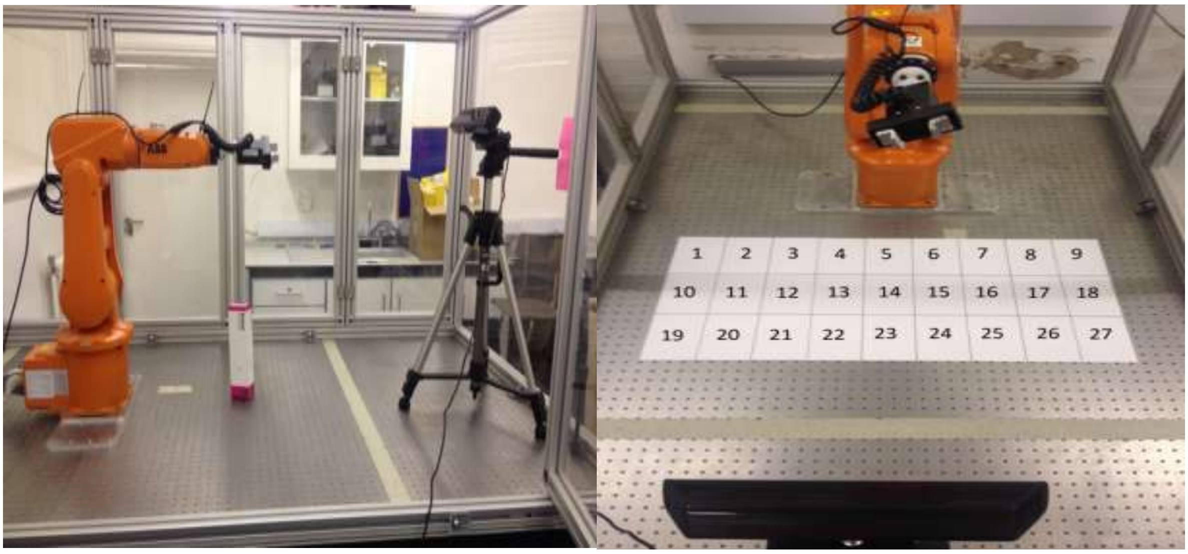


(a) (b) (c)

Figure 2: (a) Raw image from Kinect sensor, (b) Depth Image and (c) Isolated object post Euclidean clustering.

3. PRELIMINARY TESTING

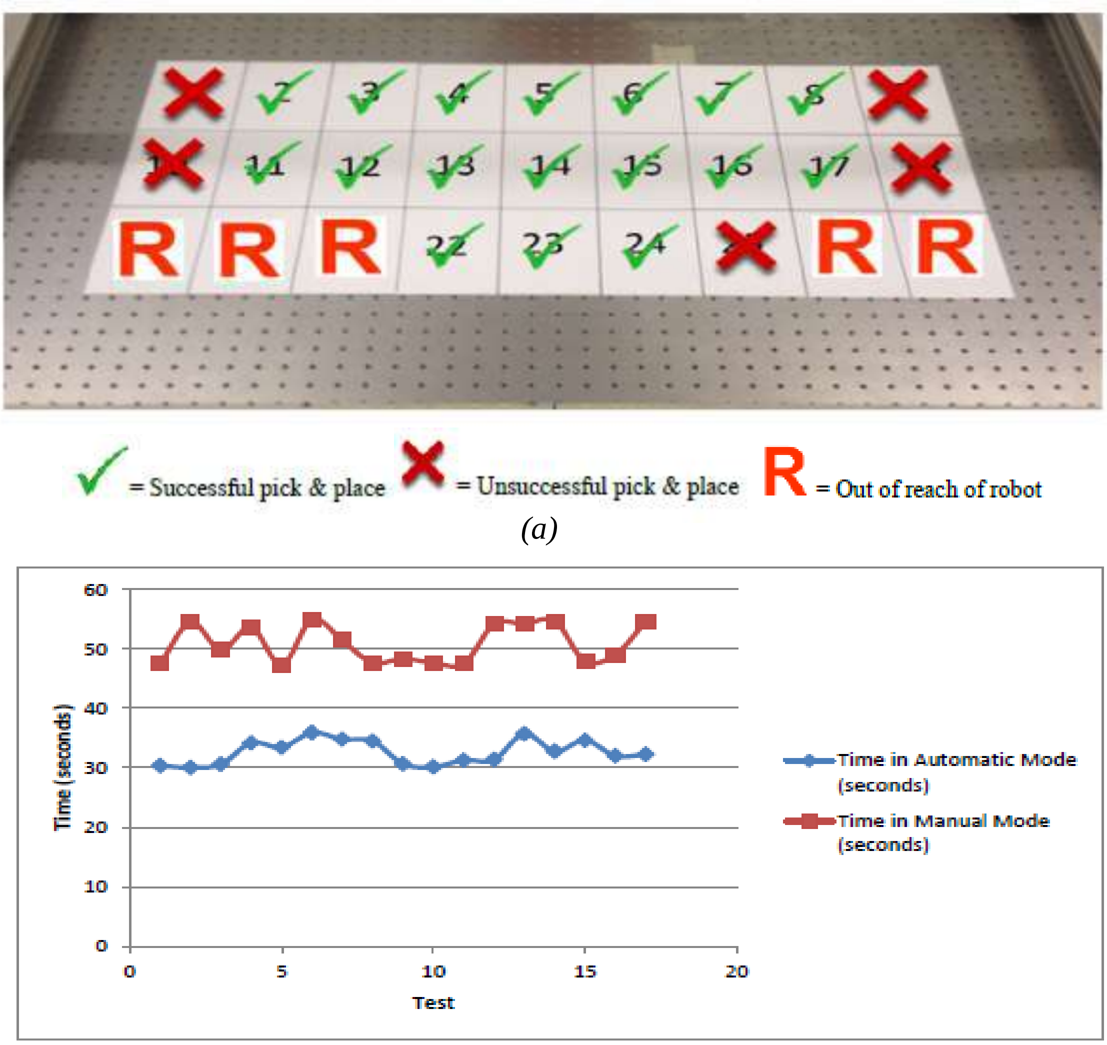
Preliminary tests were completed to assess the overall accuracy and robustness of the integrated system. A simple grid measuring 0.42m x 0.9m was laid out in the workspace directly in front of the ABB IRB 120 robot. This grid was further broken down into 27 sections, each measuring 0.14m x 0.1m. An object is placed in each of the 27 sections and a pick & place operation is attempted at each location. The time taken to complete each successful pick & place operation with the robot running in both automatic (full speed) and manual mode (reduced speed) was also noted and analysed.



(a) (b)

Figure 3: (a) Side view of rig setup and (b) Top view of rig setup

4. RESULTS & DISCUSSION



(a)

(b)

Figure 4: (a) Accuracy testing results and (b) Average times for pick & place operations

Some preliminarily sample results are shown in Figure 4 (a) and (b). The results for the pick and place test can be seen in Figure 4(a). Out of the 27 pick & place tests attempted, 17 were successful as the robot was able to move to the location defined by the Kinect, pick up the object and place it at a predefined location. 5 tests were found to be unsuccessful as the robot was unable to perform pick & place operations. This was mainly due to the fact that the locations of these tests lay on the periphery of the field of view of the Kinect; causing a distortion in the point clouds being produced which in turn moved the end effector to an incorrect location, resulting in the object not being picked up. 5 of the remaining tests were deemed to be void as they did not fail due to a fault in the system, but rather due to the limitations of the robotic manipulator itself (it would be impossible for the robot to configure its joint angles in the required alignment to facilitate picking the object up).

Overall, the results of the tests conducted with the robot operating in automatic mode replicated those of the manual mode tests. This is not surprising because the testing in automatic mode was not conducted in order to analyse and compare the accuracy of the results with the manual mode testing. The automatic tests were essentially conducted in order to compare the difference in overall time taken to perform a pick and place operation between the two operating modes of the robot. Speed is the only variable that changes when the robot is switched between the two modes – In manual mode, the robot can only be operated via the teach pendant and not by any external equipment while also having a limited maximum speed at which it can move. However, in automatic mode, the robot can be controlled externally by ROS without the use of the teach pendant and is capable of moving at full speed.

The speed at which the pick and place operations can be performed is an important characteristic since the speed of the system will determine its suitability to be integrated into an actual production/packaging line. The speed testing demonstrated that the average time to perform a pick and place operation while the robot was running in manual mode was 51.4 seconds. This was reduced to an average time of 32.61seconds while in automatic mode. These average times include the time taken to complete the sequence of each pick and place (approach, descend, pick, retreat, place and return to origin) as well as ‘wait’ times added to the system. It is necessary to note at this point however, that 10 seconds have been added into the motion planning node during the actual movement sequence of the robot. Seven of these are a direct result of accommodating for the manual opening and closing of the servo-electric gripper: Following some configuration problems between the available gripper and ROS, it was found that it was not possible to automatically control the gripper through ROS. This resulted in the gripper having to be manually opened and closed to pick up objects and hence, wait times were added in at certain points of the movement sequence to accommodate this. The remaining three seconds were added as a ‘buffer’ between each loop of the pick and place sequence for when the system must deal with multiple objects in the scene.

Thus, with the addition of a ROS compatible gripper, the overall time to perform pick and place operations in both, automatic and manual modes, can potentially be reduced by up to ten seconds. This results in average times of 25.61seconds and 41.4seconds for automatic and manual modes, respectively. The system operating at full speed would, therefore, be capable of performing just less than three pick and place operations every minute.

5. CONCLUSION

A low-cost vision system for real time robotic perception, capable of performing pick & place operations, has been developed. Preliminary testing has successfully demonstrated the functionality of the overall system. However, certain issues would need to be addressed along with further testing and adjusting of system parameters, in order for the model’s full potential to be utilised in industry.

REFERENCES

[1] J. Carroll, “North American machine vision market off to record start in 2015 - Vision Systems Design,” 2015. [Online]. Available: http://www.vision-systems.com/articles/2015/06/north-american-machine-vision-market-off-to-record-start-in-2015.html. [Accessed: 03-Jul-2015].

[2] “12 Reasons to Select FANUC’s Integrated iRVision System - Motion Controls Robotics,” 2014. [Online]. Available: http://motioncontrolsrobotics.com/12-reasons-to-select-fanucs-integrated-irvision-system/. [Accessed: 03-Jul-2015].

[3] M. Robotics, “MotoSight 3D Spatal Vision,” 2011. [Online]. Available: http://www.motoman.com/datasheets/MotoSight 3D_Spatial Vision.pdf. [Accessed: 08-Apr-2015].

[4] B. Packaging, “Packaging Line Automation using Delta Robots.” [Online]. Available: http://www.boschpackaging.com/doboy/eng/pdf/bosch-delta-robot-white-paper1.pdf. [Accessed: 08-Apr-2015].

[5] D. Kragic and M. Vincze, “Vision for Robotics,” *Found. Trends Robot.*, vol. 1, no. 1, pp. 1–78, 2009.

[6] B. Steder, “Feature-based 3D perception for mobile robots,” Ph.D dissertation, Faculty of Engineering, University of Freiburg, 2013.

[7] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, “NARF: 3D range image features for object recognition,” *Work. Defin. Solving Realis. Percept. Probl. Pers. Robot. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2010.

[8] Paul Malmsten, “Object Discovery With a Microsoft Kinect,” B.S. thesis, Faculty of the Worcester Polytechnic Institute, 2012.

[9] S. Johnson, “What Is a Point Cloud?,” 2015. [Online]. Available: http://www.wisegeek.com/what-is-a-point-cloud.htm. [Accessed: 08-Apr-2015].

[10] CuriousInventor, “How the Kinect Depth Sensor Works.” [Online]. Available: https://www.youtube.com/watch?v=uq9SEJxZiUg. [Accessed: 08-Apr-2015].

[11] Depthbiomechanics.co.uk, “How The Kinect Works.” [Online]. Available: http://www.depthbiomechanics.co.uk/?p=100. [Accessed: 08-Apr-2015].

[12] W. Garage, “Point Cloud Library.” [Online]. Available: https://www.willowgarage.com/pages/software/pcl. [Accessed: 08-Apr-2015].

[13] R. B. Rusu and S. Cousins, “3D is here: point cloud library,” *IEEE Int. Conf. Robot. Autom.*, pp. 1–4, 2011.

[14] P. C. Library, “Downsampling a point cloud using a voxel grid filter.” [Online]. Available: http://pointclouds.org/documentation/tutorials/voxel_grid.php. [Accessed: 08-Apr-2015].

[15] C. Choi, “PCL Developers Blog,” 2012. [Online]. Available: http://pointclouds.org/blog/gsoc12/cchoi/. [Accessed: 08-Apr-2015].

[16] ABB, “ABB Robotics.” [Online]. Available: http://www.abb.ie/robotics. [Accessed: 08-Apr-2015].

[17] ABB, “IRB 120 Industrial Robot.” [Online]. Available: https://library.e.abb.com/public/3bd625bab3c7cae1c1257a0800495fac/ROB0149EN_D_LR.pdf. [Accessed: 10-Mar-2015].

[18] ROS, “About ROS.” [Online]. Available: http://www.ros.org/about-ros/. [Accessed: 08-Apr-2015].

[19] Generationrobots.com, “ROS - Robot Operating System.” [Online]. Available: http://www.generationrobots.com/en/content/55-ros-robot-operating-system. [Accessed: 08-Apr-2015].

[20] Osnabrück University, “3D Point Cloud Processing,” 2013. [Online]. Available: http://kos.informatik.uni-osnabrueck.de/icar2013/. [Accessed: 08-Apr-2015].