Automation of Hydroponic Installations using a Robot with Position Based Visual Feedback

Niels F. Tanke[1*], Guoming A. Long[1], Dhruv Agrawal[1], Abhinav Valada[1], George A. Kantor[1]
[1] *The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[*] *Corresponding author. E-mail: ntanke@cmu.edu*

Abstract

This paper presents a robot for the automation of hydroponic farms using Position Based Visual Feedback. In recent years advances in agricultural engineering have resulted in higher crop yields and more environmentally friendly practices. To meet the increasing world food demand even more advances in agriculture have to be made. A way to do this is by intensifying agriculture systems. Hydroponics and robotics are proven areas that contribute to intensifying agricultural practices, but automating these hydroponics systems using current automation techniques requires a large capital investments. The designed system is a low cost and simple addition to existing Nutrient Film Technique infrastructures. This robot manipulates seedlings and plants without human intervention and can be used as a monitoring system to give parameters of the crop and environment to the grower. A low cost alternative to stereo cameras, a Microsoft Kinect and a Position Based Visual Feedback algorithm were used to detect the plants and position the robot. It is demonstrated that by using a Microsoft Kinect the positioning is accurate enough to manipulate plants on an hydroponic system.

Keywords: Hydroponics, Nutrient Film Technique, Kinect, Position Based Visual Feedback

1 Introduction

Agriculture is the cornerstone of human civilization and is responsible for the production of quality food for the human population. To ensure the availability of inexpensive and quality products to meet the increasing food demand, intensification of the growing practices is necessary (Matson et al., 1997). One of these intensification methods is hydroponics and in particular Nutrient Film Technique (NFT) (Graves, 1983). The advantage of NFT systems is the closed loop water circuit, which means easier quality control of nutrients, resulting in decreased environmental impact, less waste products and improved crop quality (Jensen, 1999).

Another technique to intensify food production is robotics. In recent years several advances have been made in the field of robotics, the operating speed of robots have surpassed humans and robots are catching up with other skills. Robots can work continuously and consistently with minimal maintenance (Koren, 1985). Hydroponics is a good platform for robotic automation, because it requires periodic labor, a systematic approach, repetitive motion and a structured environment. By combining hydroponics and robotics big improvements can be made on the efficiency of growing plants. Besides being used for labor tasks, robotic systems can also be used for monitoring tasks. By combining the robot with a monitoring system a grower can remotely monitor parameters of the crop and the environment (Kohanbash et al., 2011).

For the robot to be able to work on a hydroponic farm, it needs to observe its environment to react to it, this can be done using 3D vision. A Microsoft Kinect was used as a low cost alternative to stereo cameras.

Large-scale systems that automate hydroponic farms already exist, but require a large capital investment (Michael, 2001). Current automation systems fail to reach the stage of commercialization, because their low operating speeds, low success rates and high costs (Grift et al., 2008).

The goal of this project is to provide a low cost and easy to implement robot for existing hydroponic farms that can act on and observe the crop. The focus is on the implementation of a Position Based Visual Feedback (PBVF) algorithm in combination with a Microsoft Kinect.

2 Method

The research platform consists of: NFT system, robotic hardware, operating software, vision system and a Position Based Visual Feedback algorithm.

2.1 NFT system

The NFT system is an AmHydro 612 NFT Production Unit. It is a 1.8 m by 3.65 m by 0.9 m production unit (Figure 1) that can store 144 plants and 144 seedlings and uses a closed loop water system. The plants are put into small holes in long gullies so that the roots of the plants are submerged in a small stream of flowing water with nutrients. The NFT system consists of the following parts: table, gullies, water reservoir, tubing and a water pump. Above the NFT system are artificial lights to improve indoor growing conditions. The 612 NFT Production Unit is a kit aimed at hobby growers and is a simple and affordable system. In this case the NFT system was used to grow lettuce.

Figure 1 shows the layout of the NFT system. The gullies lay on an inclined (angle ϕ) table so that water flows passively to the end of the gullies. At the end of the gullies water is collected and directed to the water reservoir. A pump is positioned in the water reservoir to pump water to the top of the gullies.

2.2 Robotic hardware

To manipulate the plants on the NFT system without human intervention a robot was developed (Figure 1). The robot was designed as a gantry with four v-groved wheels running on two inverted angle iron tracks (x-axis). On top of the gantry is a carriage that can move back and forth over the gantry (y-axis), this is perpendicular to the x-axis. On the carriage is a mechanism to move an arm up and down (z-axis), down being the negative direction. At the end of the arm is a two degree of freedom (DOF) gripper that can open, close and rotate around the y-axis.

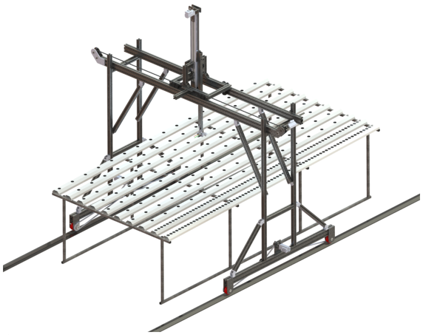
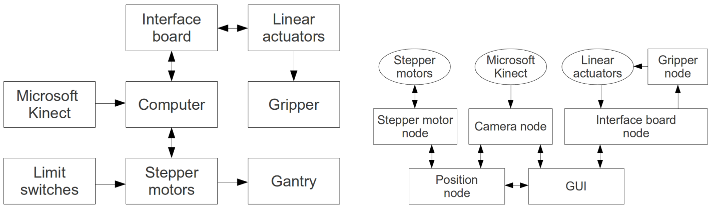


Figure 1: *Robotic manipulator and the NFT system.*

The structure is made primarily from Bosch Rexroth aluminum framing that allows the robot to be adjusted to accommodate different sized NFT systems. To drive the three independent axes, various driving mechanisms were designed. The x-axis is driven by a stepper motor (Applied Motion STM24SF-3RN) and a chain. The carriage on the gantry is connected to the stepper motor (Applied Motion STM23S-3RN) by a timing belt. The arm on the carriage is balanced by a counterweight and is driven by a stepper motor (Applied Motion STM23S-3RN) and a chain. The gripper at the end of the arm is actuated by three linear actuators, two linear actuators (Firgelli PQ12-100-12-P) are used to open and close the gripper and the other linear actuator (Firgelli L12-50-210-12-P) is used to rotate the gripper around the y-axis. All three linear actuators are driven by a 12V relay board that communicates with a Phidgets PhidgetInterfaceKit 1019. The Phidgets interfaceboard is connect to the main computer (SolidLogic Atom JT01), which is running Ubuntu Server 11.04 x64. The Kinect vision system is mounted on the carriage so that the optical axis is along the negative z-axis. The overview of the hardware layout is shown in Figure 2(a).



(a) Hardware layout of the robot. (b) Software layout of the robot.

Figure 2: *Diagram of the hardware and software layout of the robot. Arrows indicate communication paths.*

2.3 Operating software

The framework used to implement the control software is Robotic Operating System (ROS). ROS is a package of tools aimed at the development of robotics applications (Quigley et al., 2009). All software is programmed in C++.

The overview of the software layout is shown in Figure 2(b). Every hardware component (Figure 2(a)) communicates with its own ROS node. A ROS node is a C++ program running in the ROS environment that communicates using ROS topics. The main hardware nodes are: stepper motor node, gripper node, interface board node, position node and Kinect node. The position node keeps track of the x, y and z-position of the robot and a GUI was designed to provide low level control of the system.

2.4 Vision system

To observe the NFT system a camera system was needed. A Microsoft Kinect was added to the system to provide visual information. This camera combines a standard RGB with a time-to-flight camera, making it a RGB-Depth (RGB-D) camera. It is designed for home-entertainment purposes by PrimeSense. The Kinect is a low cost and readily available RGB-D camera that is an affordable alternative to stereo cameras (Ramey et al., 2011). It captures RGB images and per-pixel depth information. It is widely adopted by the engineering and science community, so there are easily accessible and robust tools available (Villaroman et al., 2011). The Microsoft Kinect produces two kinds of images, a 640 x 480 pixel RGB color image and a 640 x 480 pixel 11-bit (0 - 2047) gray scale depth image (Garstka and Peters, 2011) that is provided by an Infra-Red (IR) sensor (Ramey et al., 2011).

By combining classical 2D image analysis techniques and IR based depth measurement the 3D position of the plants are extracted. The vision algorithm was implemented in C++ using the Open Computer Vision (OpenCV) library (Intel Corporation, 2001).

The Kinect is located on the carriage and is facing downwards (negative z-axis). The Kinect is positioned in such a way as to ensure the plants in its field of view are at a maximum distance of 1.5 m, because the accuracy of the Kinect decreases quadratically with distance. Up to a distance of 1.5 m the accuracy is 10 mm and the precision of the Kinect is 1 mm (Willow Garage, 2011a). With the Kinect mounted on the carriage and facing downwards there is a field of view of 0.8 m by 1.15 m in x and y-direction.

2.4.1 Plant detection algorithm

To make the robot react to the visual information, a Position Based Visual Feedback algorithm was implemented. This algorithm was used to detect plants on the NFT system and position the robot to manipulate plants. The algorithm first detects the gullies, because the plants are only located on the gullies. All gullies are oriented along the x-axis and are straight. The straight lines were detected using an Probabilistic Hough Transform for line detection (Guo et al., 2008). By filtering the detected lines using prior knowledge about the NFT system, the edges of the gullies were identified. After the identification of the edges, the lines were grouped, resulting in a segmentation of the gullies.

The plants are grown in round cups and were detected using an Hough Transform for circles detection (Yuen et al., 1990). All plants are located on the gullies so detected circles not on the segmented gullies are deleted. After filtering, the coordinates $p_i = (u_i, v_i)$ of the plants in the image frame are known. Point $(u, v) = (0, 0)$ is defined as the top left corner of the image. The depth information d_i was extracted from the depth image by getting the value at point (u_i, v_i). The OpenNI (Villaroman et al., 2011) driver transforms the IR sensor values into distances in meters by using a fitting function (Willow Garage, 2011b; Garstka and Peters, 2011). To reduce the noise, multiple consecutive frames are averaged to calculate the plant coordinates. This results in the plant coordinates $p_i = (u_i, v_i, d_i)$ in the image frame.

2.5 Position Based Visual Feedback

The plant coordinates form the control input for the robot. This robot uses stepper motors because of the lower costs and complexity. A drawback of stepper motors as compared to servo motors is the absence of continuous control, the output only depends on the current state and the control input. This means closed loop Position Based Visual Servoing (PBVS) techniques (Chaumette and Hutchinson, 2006) can't be implemented, so an open-loop control algorithm was used.

2.5.1 Coordinate transformation

To be able to pick up a plant, the image frame coordinates $p_i = (u_i, v_i, d_i)$ have to be transformed to gantry coordinates $P_i = (x_i, y_i, z_i)$ (Figure 3).

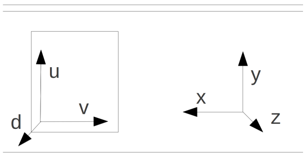


Figure 3: *Top view of the NFT system. u, v and d are the coordinates in the image frame. x, y and z are the coordinates in the gantry frame.*

To transform the image frame coordinates to gantry coordinates a modified transformation proposed by Garstka and Peters (2011) was used. The transformation equation are (1), (2) and (3).

$$\bar{x}_i = (v_i - c_v) \cdot d_i / f_v + x_0 \tag{1}$$

$$\bar{y}_i = (u_i - c_u) \cdot d_i / f_u + y_0 \tag{2}$$

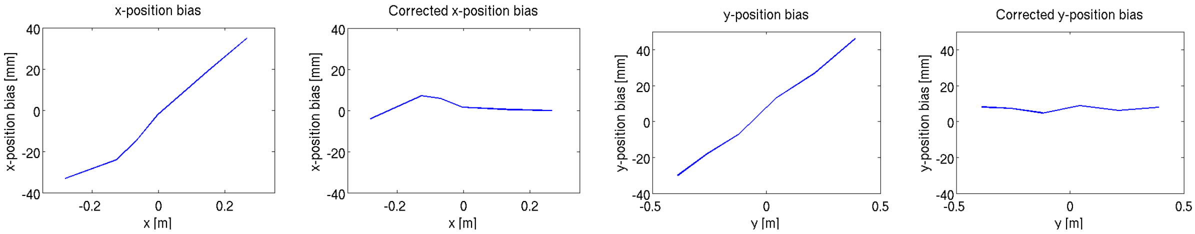
$$\bar{z}_i = d_i + z_0 + (\tan \phi \cdot \bar{x}_i) \tag{3}$$

Because the Kinect is not located on the gripper, all coordinates have to be offset (x_0, y_0 and z_0). These offsets are dependent on the position of the Kinect relative to the gripper. The z-coordinate has to be offset by an extra value, because the NFT table is under an angle $\phi = 2.2°$. In this transformation $\tilde{p} = (c_u, c_v)$ is the principal point in pixels of the depth sensor. The focal lengths in pixels are f_u and f_v. The values in Table 1 are experimentally quantified by calibrating the Kinect (Burrus, 2011).

Table 1: *Depth camera properties. $\tilde{p} = (c_u$,$c_v)$ is the principal point. f_u and f_v are the x and y-direction focal lengths.*

	Value	**Unit**
c_u	314.54	pixels
c_v	247.81	pixels
f_u	587.49	pixels
f_v	588.47	pixels

With the proposed transformation by Garstka and Peters (2011), the positioning system has a bias (Figure 4(a) and Figure 4(c)). The calculated x, and y-position of the plants away from the center of the image are biased.



(a) x-position bias (b) x-position bias after correction. (c) y-position bias (d) y-position bias after correction.

Figure 4: *Measured x and y-position bias before and after correction.*

Figure 4(a) and Figure 4(c) show a linear relationship between the position of the robot and the position bias. This bias is removed by a linear scaling of the x and y-coordinates, as seen in (4).

$$x_i = \alpha \cdot \bar{x}_i \qquad y_i = \beta \cdot \bar{y}_i \tag{4}$$

The values for α and β were found by a linear regression analysis on the data of Figure 4(a) and Figure 4(c). This results in $\alpha = 1.132$ ($R^2 = 0.9973$) and $\beta = 1.098$ ($R^2 = 0.9747$). Figure 4(b) and Figure 4(d) show the x and y-position bias after correction.

2.6 Performance metrics

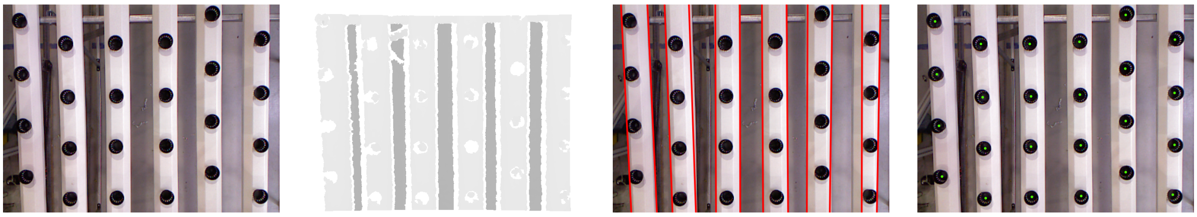
To evaluate the performance of the positioning and control algorithm, the x, y and z-position error between the final position of the gripper $P_i = (\hat{x}, \hat{y}, \hat{z})$ and the plant coordinate $P_0 = (x, y, z)$ was measured. P_0 was defined as 20 mm above the center of the cup. On the top of the cup a cross-hair is drawn to mark the center. P_i is defined as the middle between the points of the gripper so the x, y and z-position error can be measured. For every sample the camera view was analyzed to detect the plants. With these coordinates the robot moves to a plant. At the final stopping position the positioning error was measured with a ruler. The robot then returned to the same starting position. To be able to pick up a plant the gripper must be ±15 mm in x-direction, ±20 mm in y-direction and ±10 mm in z-direction from the center of the cup.

3 Results

The results of the detection and positioning algorithm were evaluated with the criteria of section 2.6.

3.1 Detection algorithm

The detection algorithm was tested on the designed robot. Figure 5 shows the view from the Kinect on the carriage and the output of the detection algorithm for both the RGB and depth image. In the depth image darker pixels indicate larger distance and white pixels indicate no depth information (Garstka and Peters, 2011).



(a) Kinect RGB image. (b) Kinect depth image. (c) Segmented gullies. (d) Detected plants.

Figure 5: *Figure 5(a) and Figure 5(b) show the unprocessed RGB and depth image. Figure 5(c) shows the gully segmentation. Figure 5(d) shows the detected plants.*

From these images with detected plants the gantry coordinates of the plants were calculated. These gantry coordinates were the input for the positioning algorithm so that the robot can be positioned to pick up the plants.

3.2 Positioning performance

Because there is no encoder feedback while the robot is moving, the accuracy of the detection has to be high enough to ensure that the plant can be picked up. The performance of the positioning is shown in Figure 6 and Table 2. The results were obtained by evaluating 25 samples ($n = 25$).

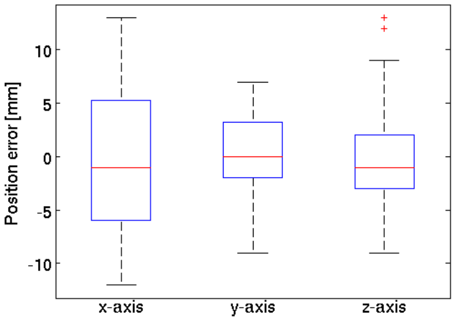


Figure 6: *Position error in the x, y and z-direction in millimeters* ($n = 25$).

Table 2: *Root mean square (RMS) error and standard deviation (STD) in millimeters* ($n = 25$).

	RMS error [mm]	**STD [mm]**
x-axis	0.32	6.89
y-axis	0.48	4.09
z-axis	0.28	5.44

4 Discussion

The performance of the system is within the requirements as stated in section 2.6 implying that plants can be manipulated on an NFT system using a Kinect and an open-loop control algorithm.

As explained by Garstka and Peters (2011) depth information can not be obtained for every pixel in the depth image. By averaging the depth information of the surrounding pixels, the depth information can be approximated, but this introduces inaccuracies in the positioning. By adding extra sensors the approximation can be improved and the positioning accuracy will increase.

Because this implementation uses an open-loop control scheme the final position of the gripper is determined by the initial position measurement. By adding an extra camera on the gripper a confirmation can be given when the gripper is close to a plant. This can improve the accuracy of the final position of the gripper.

5 Conclusion

It is demonstrated that accurate positioning can be achieved by using a Microsoft Kinect and an open-loop control algorithm. This accuracy is sufficient to manipulate plants on an NFT system. A vision based approach is useful because hard coding the layout of the NFT is a time consuming activity and can be prone to errors. By using a vision system, this robot system can be easily deployed at different farms with different layouts and because of the hardware choices the robot is a low cost alternative to existing hydroponic automation systems.

Acknowledgment

The authors wish to thank D. Kohanbash, A. Valada, C. Whittaker and W. Liu for their advice during hardware and software development.

References

Burrus, N. (2011). Kinect rgbdemo v0.6.1. `http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV6`.

Chaumette, F. and Hutchinson, S. (2006). Visual Servo Control, Part I: Basic Approaches. *IEEE Robotics Automation Magazine*, 13(4):82–90.

Garstka, J. and Peters, G. (2011). View-dependent 3D Projection using Depth-Image-based Head Tracking. In *8th IEEE International Workshop on Projector Camera Systems PROCAMS*, pages 52–57.

Graves, C. (1983). *The Nutrient Film Technique*, pages 1–44. John Wiley and Sons Inc.

Grift, T., Zhang, Q., and Kondo, N. (2008). A Review of Automation and Robotics for the Bioindustry. *Journal of Biomechatronics Engineering*, 1(1):37–54.

Guo, S., Kong, Y., Tang, Q., and Zhang, F. (2008). Probabilistic Hough Transform for Line Detection Utilizing Surround Suppression. In *Machine Learning and Cybernetics, 2008 International Conference on*, volume 5, pages 2993–2998.

Intel Corporation (2001). *Open Source Computer Vision Library*. Available: www.developer.intel.com.

Jensen, M. H. (1999). Hydroponics Worldwide. *Acta Horticulturae (ISHS)*, 481:719–730.

Kohanbash, D., Valada, A., and Kantor, G. (2011). Wireless Sensor Networks and Actionable Modeling for Intelligent Irrigation. In *2011 ASABE Annual International Meeting*.

Koren, Y. (1985). *Robotics for Engineers*. McGraw-Hill, New York.

Matson, P. A., Parton, W. J., Power, A. G., and Swift, M. J. (1997). Agricultural Intensification and Ecosystem Properties. *Science*, 277(5325):504–509.

Michael, K. (2001). Agricultural Automation in the New Millennium. *Computers and Electronics in Agriculture*, 30(1–3):237–240.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. (2009). ROS: an Open-Source Robot Operating System. In *ICRA Workshop on Open Source Software*.

Ramey, A., González-Pacheco, V., and Salichs, M. (2011). Integration of a Low-Cost RGB-D Sensor in a Social Robot for Gesture Recognition. In *Proceedings of the 6th international conference on Human-robot interaction*, HRI '11, pages 229–230, New York, NY, USA.

Villaroman, N., Rowe, D., and Swan, B. (2011). Teaching Natural User Interaction using OpenNI and the Microsoft Kinect Sensor. In *Proceedings of the 2011 conference on Information technology education*, SIGITE '11, pages 227–232, New York, NY, USA.

Willow Garage (2011a). Precision of the Kinect Sensor. `http://www.ros.org/wiki/openni_kinect/kinect_accuracy`.

Willow Garage (2011b). Technical Description of Kinect Calibration. `http://www.ros.org/wiki/kinect_calibration/technical`.

Yuen, H., Princen, J., Illingworth, J., and Kittler, J. (1990). Comparative Study of Hough Transform Methods for Circle Finding. *Image and Vision Computing*, 8(1):71–77.