

Article

# A Simple Interface for 3D Position Estimation of a Mobile Robot with Single Camera

Chun-Tang Chao, Ming-Hsuan Chung, Juing-Shian Chiou \* and Chi-Jo Wang

Department of Electrical Engineering, Southern Taiwan University of Science and Technology, 1, Nan-Tai St., Yongkang District, Tainan 71005, Taiwan; tang@stust.edu.tw (C.-T.C.); ma329201@stust.edu.tw (M.-H.C.); chijo@stust.edu.tw (C.-J.W.)

\* Correspondence: jschiou@stust.edu.tw; Tel.: +886-9-1621152; Fax: +886-6-3010069

Academic Editor: Seung-Bok Choi

Received: 20 January 2016; Accepted: 22 March 2016; Published: 25 March 2016

**Abstract:** In recent years, there has been an increase in the number of mobile robots controlled by a smart phone or tablet. This paper proposes a visual control interface for a mobile robot with a single camera to easily control the robot actions and estimate the 3D position of a target. In this proposal, the mobile robot employed an Arduino Yun as the core processor and was remote-controlled by a tablet with an Android operating system. In addition, the robot was fitted with a three-axis robotic arm for grasping. Both the real-time control signal and video transmission are transmitted via Wi-Fi. We show that with a properly calibrated camera and the proposed prototype procedures, the users can click on a desired position or object on the touchscreen and estimate its 3D coordinates in the real world by simple analytic geometry instead of a complicated algorithm. The results of the measurement verification demonstrates that this approach has great potential for mobile robots.

**Keywords:** arduino; robot interface; single camera; 3D position

---

## 1. Introduction

Robot technology has been successfully applied in many areas and is noted for its low cost, greater accuracy and reliability, suitability when there is a labor shortage, and its ability to provide entertainment. In particular robots can execute 3D (dirty, dangerous, and demeaning (or dull, difficult)) jobs for people. Compared with other modern autonomous robots, remote-controlled robots are much safer, which is a major concern in practical applications. Furthermore, increasingly more smartphone-controlled innovations are becoming available partly due to the adoption of smart phones as an everyday device by many people in highly-industrialized societies. However, controlling a mobile robot with a robotic arm by a smartphone or tablet involves a lot of motion actions which involve the visual control interface. This has caused many problems for engineers in the past. In order to improve the practicability of the mobile robot, real-time video transmission function should be added to the design, which allows the robot to operate out of sight of the user. Another crucial issue or function is the 3D position estimation of a target by a single camera. This paper is motivated by the issues mentioned above and aims to find applicable solutions for 3D position estimation with a mobile robot.

The Arduino Yun [1], a microprocessor board with a built-in Wi-Fi module, was utilized to implement the proposed design. Arduino is an open-source electronics platform based on easy-to-use hardware and software launched in 2005. The goal of Arduino is to create easy-to-use and inexpensive control devices for projects [2,3], which is consistent with the requirements of the proposed system.

Radio frequency (RF) wireless remote control is commonly used in commercially available products. Although lately there have been many debates over low-power Wi-Fi and RF (including ZigBee, 6LoWPAN, etc.) networks in the context of Internet of Things (IoT) applications and

sensor devices [4], since Wi-Fi is available on almost all smartphones, it is adopted as the wireless communication for the control commands and real-time video transmission in the proposed system.

Reconstructing a target's 3D location presents several problems. For example, an autonomous robot generally uses stereo cameras and range sensors [5,6], but these devices are not commonly used in ordinary remote-controlled mobile robots. Over the past decade, many tracking methods, such as Kalman filters [7], mean-shift [8], and Kanade-Lucas-Tomasi (KLT) algorithms [9] have been developed in computer vision. However, these tracking algorithms are mostly based on gradient changes or object features, which all require complicated computation or prior knowledge of the target objects. Karungaru *et al.* [10] applied a KLT tracker to obtain the distance to the target point for a mobile robot equipped with single camera using motion stereo vision, but the tracking of the feature points by optical flow analysis sometimes failed, further increasing the complexity of the approach. Another approach used a simple humanoid localization with a single camera [11]. With this approach, only simple analytic geometry was applied to determine the real-world coordinates of the robot using the goal posts in a soccer field as a landmark, making it practical and desirable for our system. However, in our application, no object can be used as a landmark.

In this paper, we desire to develop a feasible approach to estimate the target's 3D position for a mobile robot with a single camera. The camera needs to be properly calibrated and the odometer movement of the robot needs to be well designed to provide motion stereo vision. With the help of a touchscreen and a user-friendly visual control interface, the 3D position calculation will be implemented by simple analytic geometry and triangulation. This paper provides a practical reference for researchers attempting to design a visual remote-controlled system for a mobile robot. The rest of the paper is organized as follows: Section 2 presents the mobile robot design and the visual control interface. In Section 3, the steps for the target's 3D position estimation is provided and verified. Finally, Section 4 offers brief concluding comments.

## 2. The Mobile Robot Design and the Visual Control Interface

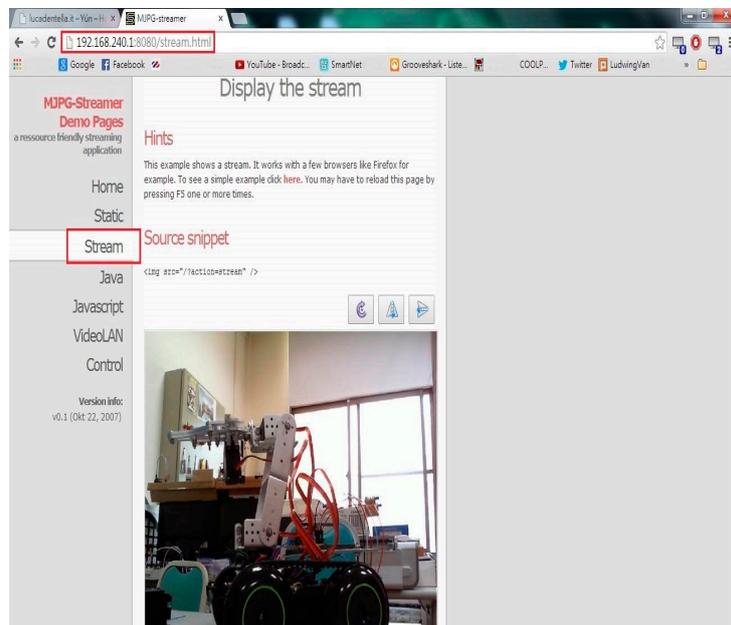
The proposed robotic remote control system is shown in Figure 1. Wi-Fi is used as the wireless communication because of its prevalence in many devices, such as personal computers, video-game consoles, and smartphones. The Arduino Yun board, based on the ATmega32u4 and the Atheros AR9331, is employed as the mobile robot controller. The board has built-in Ethernet and Wi-Fi support, a USB-A port, a micro-SD card slot, and a micro USB connection. The proposed mobile robot operates as a mobile Wi-Fi hotspot, which is specifically for the remote controller, and achieves smooth communication and control. The software installation or system configuration can be finished wirelessly as the Arduino Yun runs Linino (a MIPS GNU/Linux based on OpenWRT) with the Atheros AR9331 [1].



Figure 1. The robotic remote control system architecture.

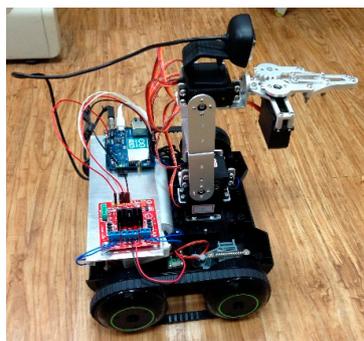
The Microsoft LifeCam HD-3000 Webcam with a USB interface is connected to the mobile robot for real-time video transmission. Software packages, such as the USB Video Class (UVC) driver, need to be installed on the Arduino Yun. Finally, the PuTTY terminal emulator is applied, which supports

SSH (Secure Shell), a terminal protocol for a secure connection between the computer and the Yun [12]. Figure 2 shows the network video taken from the webcam connected to the Arduino Yun, where 192.168.240 is the Yun IP and 8080 is the port number.



**Figure 2.** Network video from webcam connected to Arduino Yun.

Figure 3 shows the completed mobile car. A total of four DC servo motors are used for the robot arm. The bottom one (Motor A) is for base rotation and the top one (Motor D) is for the opening and closing of the claw. The middle two servo motors (Motors B and C) are for the stretch and retraction movement of the robotic arm.



**Figure 3.** The mobile robot.

A user-friendly control interface is important for manipulating the mobile robot. Figure 4 shows the visual control interface design with four buttons in the bottom left for the directional movement of the robot. The buttons in the bottom right control the robotic arm: the “T” and “R” buttons are for Motor D which controls the clamping and releasing ; the “right” and “left” buttons are for Motor A which controls base rotation; and the “forward” and “backward” buttons are for Motors B and C, which balance the claw and perform the claw’s stretch and retraction movement. Moreover, the top left button is for capturing video frame, which is stored in an SD card. Finally, with each click of all the bottom right buttons by the user, the Arduino Yun will send out PWM signals to control the corresponding servo motor to rotate at a fixed angle. The user can click the top right button

and a drop-down menu pops out with  $1^\circ$ ,  $5^\circ$ ,  $10^\circ$ ,  $20^\circ$ , or  $50^\circ$  of movement which is to set the fixed rotation angle. The video clips showing how the mobile robot grabs an object have been uploaded to YouTube [13] for the readers' reference.



Figure 4. The control interface design.

### 3. The Target's 3D Position Estimation

In this section, we show how to estimate the target's 3D position with motion stereo vision. Firstly, the proposed flow of the robot's movement and calculation is described. Following this, a practical estimation is conducted to verify the proposed approach.

#### 3.1. The Proposed Flow of Robot Movement and Calculation

First, the static mapping between the 3D real-world coordinates and the 2D image plane is shown in Figure 5, where  $f$  is the focal length and the pinhole camera is assumed. There are three coordinate systems in Figure 5, and each has its coordinate origin.

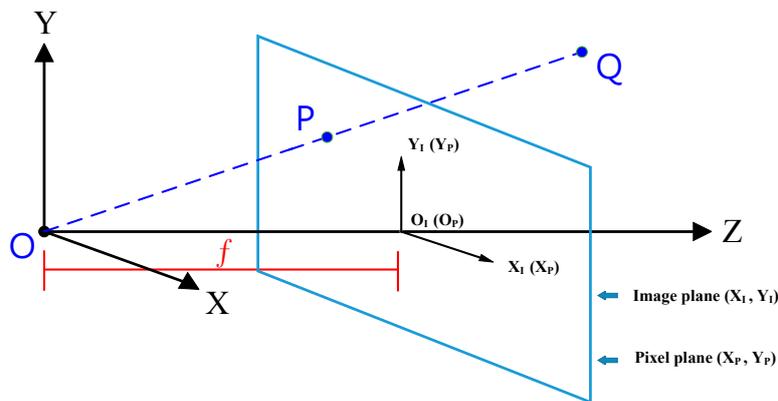


Figure 5. Image projection in relation to the coordinates.

The projection between the target object  $Q (Q_X, Q_Y, Q_Z)$  in the real world and the  $P$  point  $(P_X, P_Y)$  in the image can be derived by simple triangulation [14], as shown in Equation (1). It is worth mentioning that the  $P$  point has the same coordinate values in direction  $X$  and  $Y$  for both real world coordinates and image plane coordinates, that is,  $P_X = P_{X_i}$  and  $P_Y = P_{Y_i}$ .

$$Q_X = Q_Z \frac{P_X}{f} \text{ and } Q_Y = Q_Z \frac{P_Y}{f} \quad (1)$$

Equation (1) indicates the importance of  $Q_Z$ , the distance in  $Z$  direction. If we can obtain the coordinates of the  $P$  point in the image plane, we can get all of the coordinates of the  $Q$  point through

$Q_z$ . However, as it is well known, it is hard to obtain the distance to an object just by a single camera. One more transformation is also needed between the image plane coordinates and the pixel plane due to camera pixel specification, as shown in Equation (2):

$$X_P = \alpha X_I \text{ and } Y_P = \alpha Y_I \tag{2}$$

where  $\alpha$  is defined as the pixel number in 1 cm.

To estimate the distance of a static object from a single camera, we need to conversely move the robot to show the distance information on the image plane, or the visual interface. Figure 6 shows the proposed flow of robot movement in order to obtain the motion stereo vision for the target’s 3D position. The user’s touchscreen is designed to have a vertical line in the middle of the screen. The proposed steps are outlined as follows:

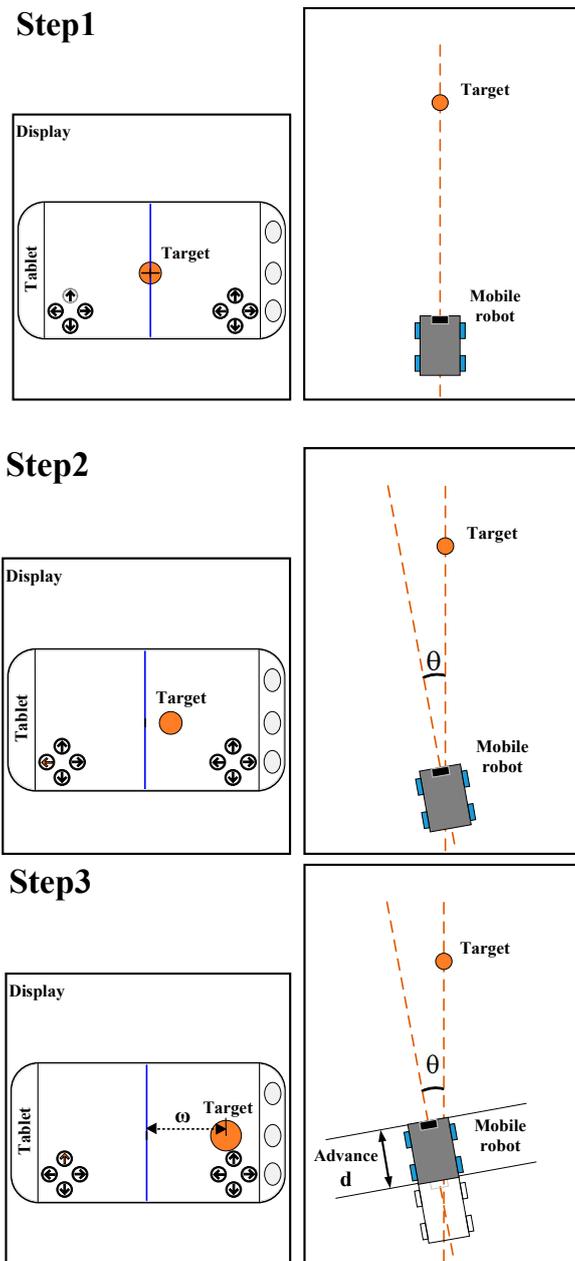


Figure 6. Flow of robot movement.



Moreover, angle  $\varphi$  can be calculated by:

$$\phi = \pi - \beta - \left(\frac{\pi}{2} - \theta\right) = \frac{\pi}{2} - \beta + \theta \quad (5)$$

Finally, we can finish the calculation by obtaining angle  $\beta$  through the following equation:

$$\beta = \tan^{-1} \frac{w}{f} \quad (6)$$

Based on the above derivation, it will be easier to obtain the 3D position of target  $Q_1$  with respect to origins  $O$  or  $O_{NEW}$ . As shown in Figure 7,  $P_O$  and  $P_N$  are the projected points of  $Q_1$  in the original pixel plane and the new pixel plane, respectively. The respective coordinates  $(P_{O(X_P)}, P_{O(Y_P)})$ , where  $P_{O(X_P)} \cong 0$ , and  $(P_{N(X_P)}, P_{N(Y_P)})$  are recorded during the user's two clicks in Step 1 and Step 3. Then, the real-world coordinate of  $Q_1$  with respect to origin  $O$  will be:

$$Q_1 \text{ (w.r.t } O) = \left(0, \overline{OQ_1} \frac{P_{O(Y_P)}}{f\alpha}, \overline{OQ_1}\right) \quad (7)$$

where the  $Y$  coordinate value is obtained by Equations (1) and (2), as shown below:

$$Q_1(Y) \text{ (w.r.t } O) = Q_1(Z) \frac{P_{O(Y)}}{f} = \overline{OQ_1} \frac{P_{O(Y_i)}}{f} = \overline{OQ_1} \frac{P_{O(Y_P)}}{f\alpha} \quad (8)$$

We can also get the real-world coordinate of  $Q_1$  with respect to origin  $O_{NEW}$ . First, the  $Z$  direction will be changed as  $Z_{NEW}$  direction, as shown in Figure 7. The distance of  $Q_1$  in  $Z_{NEW}$  direction can be calculated by:

$$\overline{O_{NEW}G} = \overline{OG} - d = \overline{OQ_1} \cos \theta - d \quad (9)$$

Then by applying Equations (1) and (2), we can get the real-world coordinate of  $Q_1$  with respect to origin  $O_{NEW}$  as in the following equation:

$$Q_1 \text{ (w.r.t } O_{NEW}) = \left(\overline{O_{NEW}G} \frac{P_{N(X_P)}}{f\alpha}, \overline{O_{NEW}G} \frac{P_{N(Y_P)}}{f\alpha}, \overline{O_{NEW}G}\right) \quad (10)$$

Finally, let us note that for  $Q_1 \text{ (w.r.t } O)$  and  $Q_1 \text{ (w.r.t } O_{NEW})$ , their respective  $Y$  coordinate values which represent the height of the target, should be the same. So we have the following equation by Equations (7) and (10):

$$\overline{OQ_1} \frac{P_{O(Y_P)}}{f\alpha} = \overline{O_{NEW}G} \frac{P_{N(Y_P)}}{f\alpha} \quad (11)$$

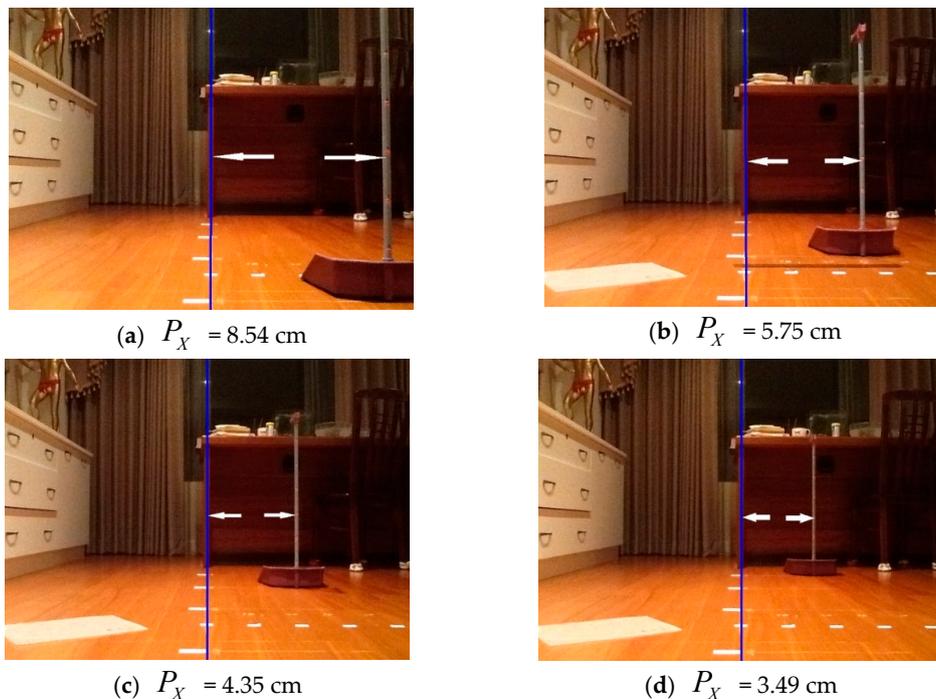
### 3.2. Measurement Verification

To verify the proposed prototype method and the derived calculation, a tablet with a 9.7 inch display and a  $960 \times 720$  pixels camera was applied as the visual control interface. Figure 8 shows the tablet on top of a metal box which was sandwiched between two bookshelves. Using a long ruler and protractor, we moved the box to simulate the movement and rotation of the mobile robot.



**Figure 8.** Measurement verification by tablet.

First, the parameter  $\alpha$  is about 48.7 pixel/cm according to the tablet screen size and the pixel specification ( $960 \times 720$  pixels) of the camera. Furthermore, the measurement of focal length  $f$  is necessary for a calibrated camera system. In our measurement, a room was plastered with stickers at different distances on the floor, with a movable vertical pole labelled with different heights. Without loss of generality, we will focus on  $Q_X = Q_z \frac{P_X}{f}$  in Equation (1) to estimate focal length  $f$ , so  $f$  can be calculated by  $f = Q_z \frac{P_X}{Q_X}$ . In Figure 9, the vertical pole was put at distances ranging from near to far away, with  $Q_z$  ranging from 100 cm to 250 cm, but  $Q_X$  was kept at 36 cm. The distance between the white double arrow in each picture indicates  $P_{Xp}$  (pixel) or  $P_X$  (cm). All of the data for the four different cases is shown in Table 1. The average  $f$  was calculated as 24.03, which is adopted in the subsequent calculations.

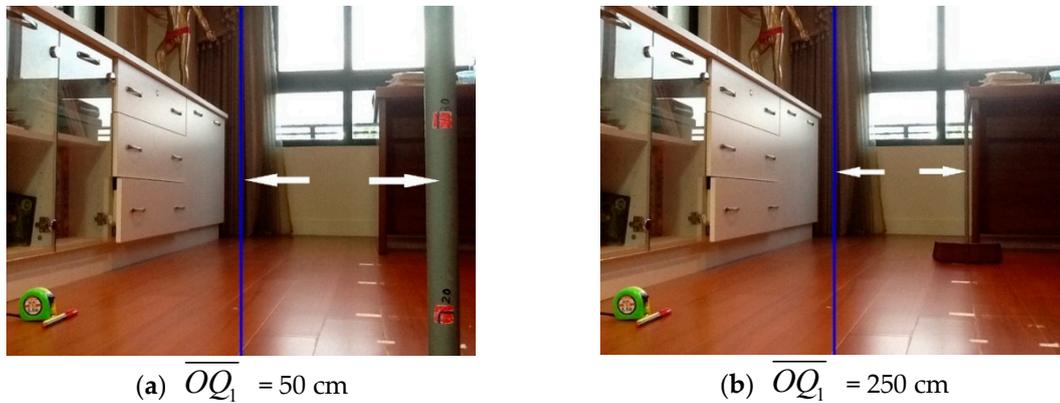


**Figure 9.** Focal length  $f$  measurement ( $Q_X = 36$  cm). (a)  $P_X = 8.54$  cm; (b)  $P_X = 5.75$  cm; (c)  $P_X = 4.35$  cm; (d)  $P_X = 3.49$  cm.

**Table 1.** Results of focal length  $f$  measurement ( $Q_X = 36$  cm).

$P_{X_p}$ (pixel)	$P_X$ (cm)	$Q_Z$ (cm)	$f$ (cm)
416	8.54	100	23.72
280	5.75	150	23.96
212	4.35	200	24.17
170	3.49	250	24.27

Next, we will verify the proposed flow for estimating the 3D position of a target object. The  $\theta$  is set to  $12^\circ$  and  $d$  is 20 cm. For simplicity, we will calculate  $\overline{OQ_1}$  in Equation (3) for the following verification. For the first step in the proposed process, we put the vertical pole in the middle of the screen. Then in the Step 2, the robot was rotated  $12^\circ$  to the left and the vertical pole moved to the right on the screen. Finally, as the robot advanced 20 cm in Step 3, the vertical pole moved to the right again. Figure 10 shows two extreme cases for the target distance measurement after Step 3 was finished, where  $\overline{OQ_1}$  equals 50 cm and 250 cm, which corresponds to the nearest and farthest cases, respectively. The distance between the white double arrows in each picture indicates  $P_{N(X_p)}$  (pixel) or  $w$  (cm).

**Figure 10.** Measurement of distance to the target. (a)  $\overline{OQ_1} = 50$  cm; (b)  $\overline{OQ_1} = 250$  cm.

Using the derived equations from the last subsection, the step by step procedure of  $\overline{OQ_1}$  calculation for the target's 3D estimation algorithm can be summarized as:

- Given constant values:  $\theta, d, \alpha, f$ .
- Get the coordinates  $(P_{O(X_p)}, P_{O(Y_p)})$  and  $(P_{N(X_p)}, P_{N(Y_p)})$  at the start (Step 1) and end (Step 3) of the proposed approach. The variable  $w$  equals to  $\frac{P_{N(X_p)}}{\alpha}$ .
- Calculate  $\beta$  by Equation (6).
- Calculate  $\varphi$  by Equation (5).
- Calculate  $\overline{DQ_1}$  by Equation (4).
- Calculate  $\overline{OQ_1}$  by Equation (3).
- Estimation for the 3D coordinates of target  $Q_1$  with respect to origins  $O$  or  $O_{NEW}$  can be accomplished by Equations (7) and (10).

Table 2 shows the input  $P_{N(X_p)}$  and the resulting parameters and estimated  $\overline{OQ_1}$  for the two cases in Figure 10. It seems that the estimation result meets the expectation that the farther away the target is, the larger the error will be, which may be true for any 2D image method used in 3D distance estimation.

**Table 2.** Estimation results of distance to the target.

	$P_{N(X_p)}$ (Pixel)	$w$ (cm)	$\beta$ (In Degree)	$\varphi$ (In Degree)	True $OQ_1$ (cm)	Estimated $OQ_1$ (cm)
Case 1	414	8.50	19.48	82.52	50	51.23
Case 2	270	5.54	12.98	89.02	250	262.05

Let us review Figures 6 and 7 again, which show the whole process and mathematical derivation of the target's 3D position estimation. To make the process work, "the robot automatically rotates  $\theta$  degree to the left" in Step 2 is an important step. However, the key point in Step 2 is that the projection center of the camera, the  $O$  point in Figure 7, must be kept at the same point after the rotation of the mobile robot. As mentioned earlier, we can check this by observing target points at different distances, like  $Q_1$  and  $Q_2$  in Figure 7, and these should project to the points on the screen with the same  $X$  coordinate value. But in practice, it may be hard to drive a mobile robot to achieve this. So, in practice, Step 2 and Step 3 can be combined to first obtain the final correct position of the mobile robot by manual operation. After the final ideal position of the robot is obtained, we can then begin designing a method to make the mobile robot reach the correct position automatically.

To implement the automatic process for target's 3D position estimation, in addition to the above points for Step 2 and Step 3, we should also reconsider the hardware design of the mobile robot. For example, to improve the accuracy, the DC motor which controls the wheels may need to be replaced by a stepping motor. Furthermore, the odometer data often includes some errors due to tire slips, so high friction tires may be necessary.

A remote-controlled mobile robot with video transmission is practical and effective when the operator is unable to see the environment or visibility is poor, but when the mobile robot is fitted with a robotic arm, it could be difficult and time-consuming for an inexperienced user to manipulate the robotic arm to grasp an object. With the proposed target's 3D position estimation, the user can activate the automatic process when approaching the target, then the mobile robot and robotic arm may reach the appropriate position by inverse dynamics in a short time automatically. Therefore, it has functional applications for robotic grasping tasks, such as obstacle removal or bomb disposal.

For the proposed Wi-Fi mobile robot design in Section 2, it appears that communication is smooth when using the video transmission and control commands. However, when the target's 3D position estimation in this section is added, the computational consumption may cause a time delay between the click on the touchscreen and the motor reaction. Fortunately, the proposed three steps for the target's 3D position estimation is activated by the user, and then executed by the system in an automatic way. Thus, the interactive design can be incorporated to reduce the effect of the time delay. Finally, the pitch angle of the camera will affect the positioning performance; thus, the camera and the robotic arm should be reset to the standard position once the target's 3D position estimation is activated to avoid any error caused by the pitch angle of the camera.

#### 4. Conclusions

In this paper, a visual control interface for a mobile robot was proposed, and a simple prototype method for estimating the 3D position of a target was derived and verified. Wi-Fi technology was applied to provide real-time video transmission, which enables the robot to operate in different environments or areas where visibility is poor, limited, or the object is out of sight. It was found that, in addition to the basic movements of the robot, more control actions are needed for the manipulation of the robotic arm and a user-friendly interface is necessary for human operators. In the proposed visual control interface, there are 12 buttons in all for controlling two DC motors and four DC servo motors. Among these buttons, one special button controls a drop-down menu which allows the user to select the degree of movement which allows for a more flexible way to control the servo motor.

The authors have uploaded the operation videos to YouTube for the readers' reference. The videos show how a skilled user operates the mobile robot to grasp an object with a three-axis robotic arm. However, this will not always be an easy task for an amateur or in an emergency situation. Thus, it will be helpful if we can estimate the 3D position of the desired target, then activate the corresponding automatic process to finish the job by robotic arm inverse dynamics. Although stereo cameras and range sensors can be applied in 3D position reconstruction, we have shown that a simple approach for the target's 3D estimation with a single, ordinary camera is still a worthwhile research goal. Instead of a complicated theory, this paper proposes a prototype method for estimating the target's 3D position by simple geometry and triangulation. The proposed steps have been clearly verified and, in future, we hope to make the robot motion and 3D position estimation more accurate.

**Acknowledgments:** The authors would like to express the sincere thanks to the reviewers for their invaluable comments and suggestions. This work is supported by the Ministry of Science and Technology, Taiwan, under Grant nos. MOST 104-2221-E-218-039.

**Author Contributions:** Juing-Shian Chiou and Chun-Tang Chao developed the methodology and drafted the manuscript. Moreover, Chi-Jo Wang derived the 3D formulas and Ming-Hsuan Chung implemented the hardware experiments. The authors approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Arduino Yun. Available online: <https://www.arduino.cc/en/Main/ArduinoBoardYun> (accessed on 17 January 2016).
2. Palma, D.; Agudo, J.E.; Sánchez, H.; Macías, M.M. An Internet of Things example: Classrooms access control over near field communication. *Sensors* **2014**, *14*, 6998–7012. [[CrossRef](#)] [[PubMed](#)]
3. Chao, C.T.; Wang, C.W.; Chiou, J.S.; Wang, C.J. An arduino-based resonant cradle design with infant cries recognition. *Sensors* **2015**, *15*, 18934–18949. [[CrossRef](#)] [[PubMed](#)]
4. Chen, Y.; Han, F.; Yang, Y.H.; Ma, H.; Han, Y.; Jiang, C.; Lai, H.Q.; Claffey, D.; Safar, Z.; Liu, K.J.R. Time-reversal wireless paradigm for green Internet of Things: An overview. *IEEE Internet Things J.* **2014**, *1*, 81–98. [[CrossRef](#)]
5. Sabattini, L.; Levratti, A.; Venturi, F.; Amplo, E.; Fantuzzi, C.; Secchi, C. Experimental comparison of 3D vision sensors for mobile robot localization for industrial application: Stereo-camera and RGB-D sensor. In Proceedings of the 2012 IEEE International Conference on Control Automation Robotics & Vision (ICARCV), Guangzhou, China, 5–7 December 2012; pp. 823–828.
6. Cai, X.; Dean-Leon, E.; Somani, N.; Knoll, A. 6D image-based visual serving for robot manipulators with uncalibrated stereo cameras. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, TX, USA, 14–18 September 2014; pp. 736–742.
7. Hu, X.; Hu, Y.; Xu, B. Generalized Kalman filter tracking with multiplicative measurement noise in a wireless sensor network. *IET Signal Process.* **2014**, *8*, 467–474. [[CrossRef](#)]
8. Wen, Z.; Cai, Z. Mean shift algorithm and its application in tracking of objects. In Proceedings of the 2006 International Conference on Machine Learning and Cybernetics, Dalian, China, 13–16 August 2006; pp. 4024–4028.
9. Jang, W.; Oh, S.; Kim, G.A. hardware implementation of pyramidal KLT feature tracker for driving assistance systems. In Proceedings of the 2009 IEEE International Conference on Intelligent Transportation Systems (ITSC '09), St. Louis, MO, USA, 4–7 October 2009; pp. 1–6.
10. Karungaru, S.; Ishitani, A.; Shiraiishi, T.; Fukumi, M.A. simple interface for mobile robot equipped with single camera using motion stereo vision. *Int. J. Mach. Learn. Comput.* **2012**, *2*, 633–639. [[CrossRef](#)]
11. Sudin, M.N.; Nasrudin, M.F.; Abdullah, S.N.H.S. Humanoid localization in a robot soccer competition using a single camera. In Proceedings of the 2014 IEEE Tenth International Colloquium on Signal Processing & Applications (CSPA), Kuala Lumpur, Malaysia, 7–9 March 2014; pp. 77–81.
12. Using the Command Line for Communication with SSH and cURL. Available online: <https://www.arduino.cc/en/Tutorial/LinuxCLI> (accessed on 17 January 2016).

13. A Wi-Fi Mobile Robot with Real-Time Video Transmission. Available online: [https://youtu.be/8pajGYlh\\_UQ](https://youtu.be/8pajGYlh_UQ) (accessed on 17 January 2016).
14. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer: New York, NY, USA, 2011.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).