sensors

MDPI

Article

Ball-Catching System Using Image Processing and an Omni-Directional Wheeled Mobile Robot

Sho-Tsung Kao and Ming-Tzu Ho *

Department of Engineering Science, National Cheng Kung University, Tainan 701401, Taiwan; n98971035@gs.ncku.edu.tw
* Correspondence: bruceho@mail.ncku.edu.tw

Abstract: The ball-catching system examined in this research, which was composed of an omni-directional wheeled mobile robot and an image processing system that included a dynamic stereo vision camera and a static camera, was used to capture a thrown ball. The thrown ball was tracked by the dynamic stereo vision camera, and the omni-directional wheeled mobile robot was navigated through the static camera. A Kalman filter with deep learning was used to decrease the visual measurement noises and to estimate the ball's position and velocity. The ball's future trajectory and landing point was predicted by estimating its position and velocity. Feedback linearization was used to linearize the omni-directional wheeled mobile robot model and was then combined with a proportional-integral-derivative (PID) controller. The visual tracking algorithm was initially simulated numerically, and then the performance of the designed system was verified experimentally. We verified that the designed system was able to precisely catch a thrown ball.

Keywords: visual servo; omni-directional mobile robot; Kalman filter; deep learning applications

check for updates

Citation: Kao, S.-T.; Ho, M.-T. Ball-Catching System Using Image Processing and an Omni-Directional Wheeled Mobile Robot. *Sensors* **2021**, *21*, 3208. https://doi.org/10.3390/s21093208

Academic Editor: Gregor Klancar

Received: 26 February 2021
Accepted: 30 April 2021
Published: 5 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The method in which visual information in a feedback control loop is used to precisely control the motion, position, and posture of a robot is called visual servoing. Visual servoing is well-known and commonly used in both dynamic and unstructured environments. Visual servoing tasks involving the use of a robot to catch a flying ball present a challenge.

In robotic ball-catching, there are several variations in the system configurations, methods of implementation, trajectory prediction algorithms, and control laws. In [1], the stereo camera system (two PAL cameras) was placed above the work area to catch a flying ball. In [2], two cameras were placed at the top and left of the work area for ping-pong ball catching and ball juggling. In [3], the cameras were located behind the robot for catching in-flight objects with uneven shapes. In [4], a high-speed vision system with two cameras was used for a ball juggling system.

In the above research, fixed cameras were used to predict the trajectory of the target object in the work area. The advantage is that the background image was fixed, and the image processing was relatively simple; however, it is difficult to achieve the desired results in an open space due to the field of view (FoV) constraints associated with the camera system and the limited workspace of a robotic arm.

In [5], an autonomous wheelchair mounted with a robotic arm used two vision sensors to accurately determine the location of the target object and to pick up the object. In [6], a robotic system endowed with only a single camera mounted in eye-in-hand configuration was used for the ball catching task. In [7], a robot with an eye-in-hand monocular camera was used for autonomous object grasping. The methods of [5–7] can resolve the stated problem. The eye-in-hand concept enables the camera to maneuver along with the robotic arm, which improves the trajectory prediction precision in an open space, especially when the object is near the robot. Another method by which to catch a ball in a wide-open space

is to combine a static stereo vision system with a mobile robot to complete the ball-catching task, as described below.

In [8], the ball was tracked using two ceiling-mounted cameras and a camera mounted on the base of the robot. Each camera performed ball detection independently. The 3D coordinates of the ball were then triangulated from the 2D image locations. In [9], a robotic system consisted of a high-speed hand-arm, a high-speed vision system, and a real-time control system. A method was proposed to estimate the 3D position and orientation by fusing the information observed by a vision system and the contact information observed by tactile sensors.

In [10], the robot maintained a camera fixation that was centered on the image of the ball and kept the tangent of the camera angle rising at a constant rate. The performance advantage was principally due to the higher gain and effectively wider viewing angle when the camera remained centered on the ball image. In [11], a movable humanoid robot with an active stereo vision camera was designed and implemented for a ball-catching task. Two flying balls were caught by mobile humanoid robots at the same time in a wide space.

The prediction of the trajectory is an important factor for the effective capture of a flying ball. Most methods assumed that a free-flying object's dynamic model was known. A parabola in 3D space was used to model the trajectory of a flying ball, and the least-squares method was used to estimate the model parameters.

In [10], the flying ball trajectory was predicted using a non-linear dynamic model that included air resistance with different parameter estimation algorithms. In [6], an estimate of the catching point was initially provided through a linear algorithm. Then, additional visual measurements were acquired to constantly refine the current estimate by exploiting a nonlinear optimization algorithm and a more accurate ballistic model with the influence of air resistance, visual measurement errors, and rotation caused by ground friction. One of the typical uses of the Kalman filter is in navigation and positioning technology.

In [12], an automated guided vehicle was combined with a Kalman filter with deep learning. The system was found to have good adaptability to the statistical properties of the noise system, which improved the positioning accuracy and prevented filter divergence. In our previous work [13], we provided a brief overview of an effective ball-catching system with an omni-directional wheeled mobile robot. In this paper, we extend the ball trajectory estimation method to improve the system's ball-catching performance. Furthermore, the experiments and application are described in detail.

In this research, we developed a combined omni-directional wheeled mobile robot and a multi-camera vision system to catch a flying ball in a large workspace. Figure 1 provides a schematic diagram of the proposed system.

To maneuver the robot around, an omni-directional wheeled setup was selected due to its superiority in terms of mobility. This robot can enable translational or rotational movements or any combination of these two movements. An active stereo vision camera assumed the visual tracking task for the flying ball with two pan-and-tilt cameras. To navigate the mobile robot to the ball's touchdown point, a static vision camera was used. Real-time processing of the image processing algorithms and control laws are necessary to accomplish these visual servoing tasks.

Therefore, digital signal processors (DSPs) were used to ensure the real-time ability to carry out these actions. Noise from the environment or other sources, such as the measurement of vision systems for caught balls, can deteriorate the performance of the proposed visual servoing system. In this work, estimating the position and velocity of the ball was achieved using the Kalman filter and a linear dynamic model of a flying ball. The use of deep learning in Kalman filtering improved both the accuracy and robustness of the results. In this paper, the experimental setup is presented, and the results of the simulation and experiments are provided to demonstrate the performance of the proposed system.

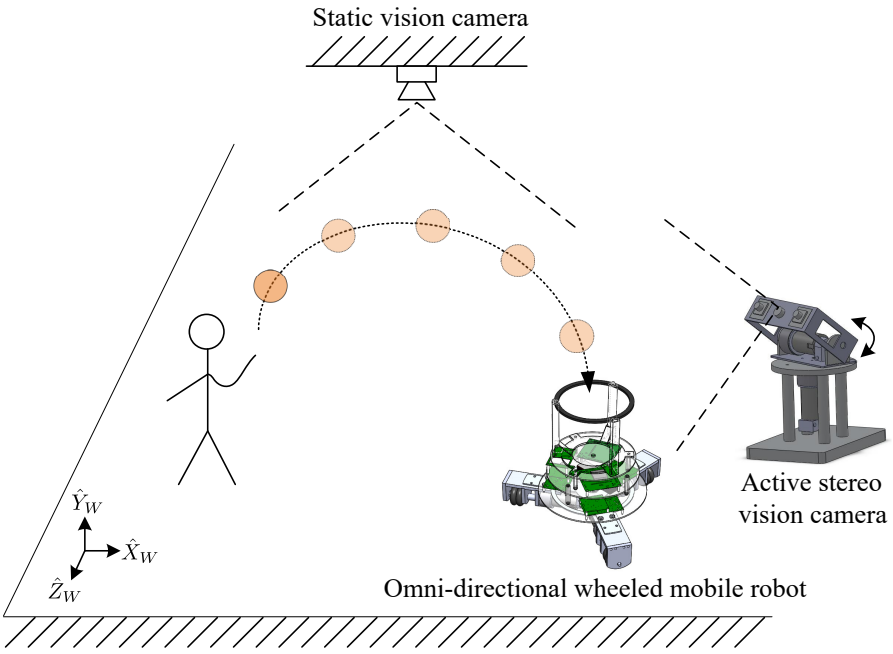


Figure 1. Schematic overview of the proposed system.

The remainder of this paper is composed as follows: In Section 2, the relationship between the coordinates used in this work is described (specifically, the world and image coordinates). In addition, the image processing algorithms are also introduced. Section 3 describes the active stereo vision camera. Section 4 describes the prediction and trajectory estimation method for a flying ball. Section 5 offers a discussion of the control law of the omni-directional wheeled mobile robot. Section 6 presents the implementation of the designed system. Section 7 presents the results for the simulation and the experiments. Finally, Section 8 provides our concluding remarks.

2. Image Processing and Visual Measurement

Vision systems are used to obtain the location of the ball and robot in a three-dimensional (3D) Cartesian coordinate system. The position determination of an object from an image is based on a pinhole camera model of the vision system [14,15]. The position of an object can be given in the vector form as

$$\lambda p = K_I[R_E|T]P_o \tag{1}$$

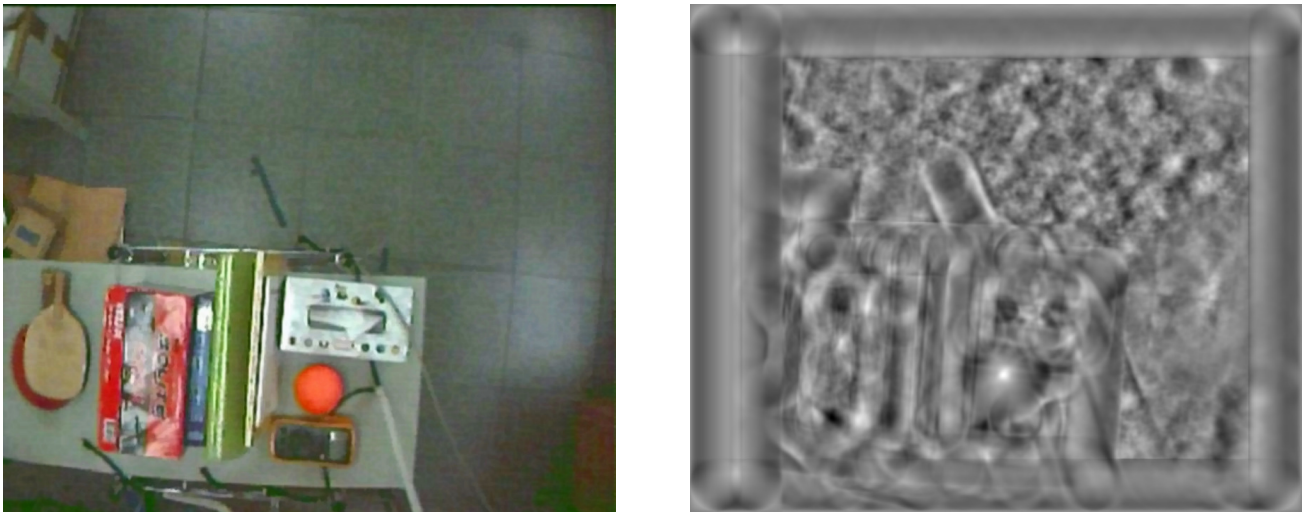
where vector $p = [x_{im}\ y_{im}\ 1]^{\mathrm{T}}$ represents the 2D homogeneous coordinates of an image point in the image coordinate system. The vector $P_o = [X_W\ Y_W\ Z_W\ 1]^{\mathrm{T}}$ represents the 3D homogeneous coordinate of a target object point in the world coordinate system. The 3 by 3 matrix R_E and 3D vector T are external camera parameters that define the rotation and translation between the world frame and camera frame, respectively. λ is a scaling factor, and K_I is the internal parameter matrix of the camera. It is given by

$$K_I = \begin{bmatrix} f_{sx} & f_{s\theta} & o_x \\ 0 & f_{sy} & o_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

In the matrix K_I, (o_x, o_y) is the principal point in the image coordinate system in pixels; f_{sx} and f_{sy} are the size of the unit lengths in the horizontal and vertical pixels, respectively, and $f_{s\theta}$ is the skewness factor of the pixels. The camera calibration procedure in [14] is used to calibrate the internal camera parameters beforehand.

In the image plane, to obtain the precise position of a target object, a series of image processing algorithms are used to process the images captured from the camera. The source image with a complex background acquired from the camera is shown in Figure 2a. In this study, the template matching approach [16] is used to find the image of the target object that matches a template object image in the whole image. The template matching process compares the sub-image of the source image and template object image, from left to right and from top to bottom, to obtain the correlation between these two images.

Additionally, to detect the target object in the three-dimensional space, the template object image is resized during the comparison process. Figure 2b shows the normalized cross-correlation image, in which the brightest one represents the most similarity between two images.



(a) (b)

Figure 2. (**a**) Image with a complicated background captured by the camera and (**b**) the normalized cross-correlation image.

The simulation result of the template matching was compared to the one obtained by the color matching method. The result when using the template matching method to perform ball detection is shown in Figure 3a. The result based on the color matching method is shown in Figure 3b. In a complex background, when comparing the object extraction step in Figure 3a with the thresholding process in Figure 3b, the use of template matching for ball detection is not interfered with by other objects, which is suitable for the environment used in this study.

The location of the processed image was taken using the centroid of the object as follows:

$$(x_c, y_c) = \left(\frac{1}{n_1} \sum x_{im}, \frac{1}{n_1} \sum y_{im} \right) \tag{3}$$

where (x_c, y_c) are the center coordinates of the object, (x_{im}, y_{im}) are the coordinates of the white pixel, and n_1 is the number of white pixels. The actual centroid of the ball in Figure 3 obtained by the manual image segmentation is $(x_c, y_c) = (336, 368)$. The centroid of the ball obtained from the template matching method is $(x_c, y_c) = (334, 365)$, and the centroid of the ball obtained from the color matching method is $(x_c, y_c) = (322, 355)$. The template matching method provided a more accurate result.

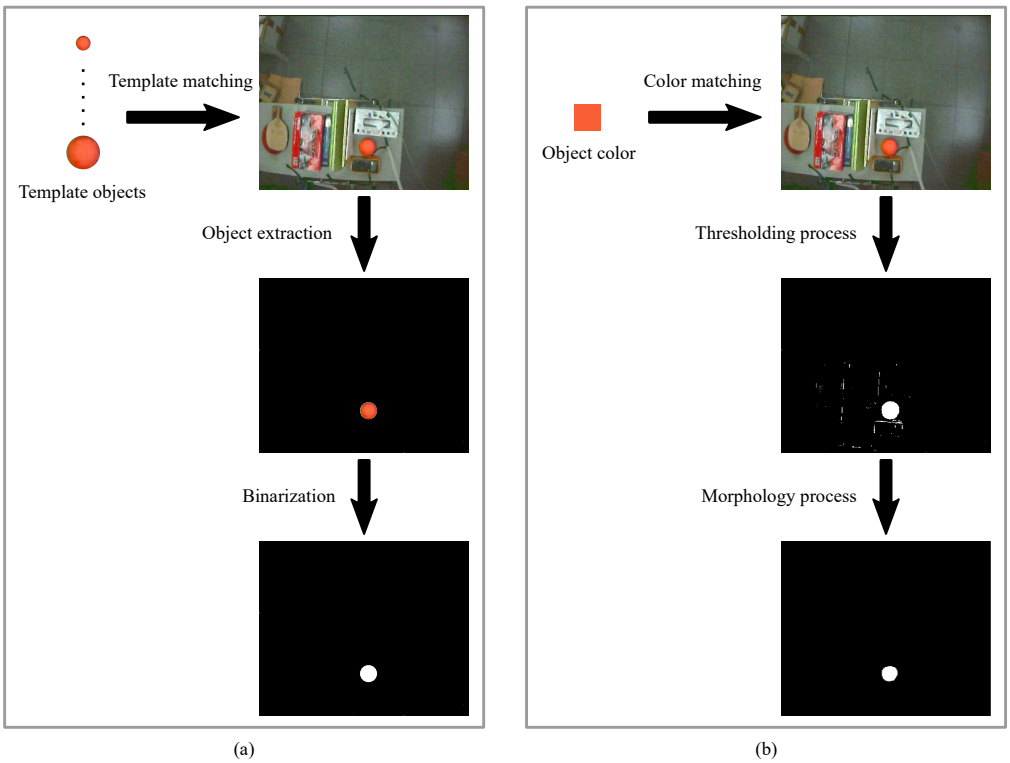


Figure 3. Simulation results of the (**a**) template matching method and (**b**) color matching method.

3. Active Stereo Vision

In this study, an active stereo vision camera was used to locate and trace the flying ball. The active stereo vision camera included two cameras mounted on a pan-and-tilt platform, and the cameras were parallel to each other. The coordinate systems of this active vision camera are described in Figure 4, for which the parameters are listed below.

P	target object in the word coordinate system.
r	the distance between the stereo coordinate and the geodetic coordinate.
d	the distance between the center of camera A and camera B.
ϕ_s	the angle of the pan axis.
θ_s	the angle of the tilt axis.
$(O_A, \hat{X}_A, \hat{Y}_A, \hat{Z}_A)$	coordinate system of camera A.
$(O_B, \hat{X}_B, \hat{Y}_B, \hat{Z}_B)$	coordinate system of camera B.
$(O_S, \hat{X}_S, \hat{Y}_S, \hat{Z}_S)$	stereo coordinate system located in the center of the camera frame A and camera frame B to point to the target.
$(O_G, \hat{X}_G, \hat{Y}_G, \hat{Z}_G)$	geodetic coordinate system fixed relative to the ground with the pan axis aligned on the $\hat{Y}_G$-axis and the tilt axis aligned on the $\hat{X}_G$-axis, where both axes are connected and considered to be zero. Three-axes are determined using a right-handed coordinate system.

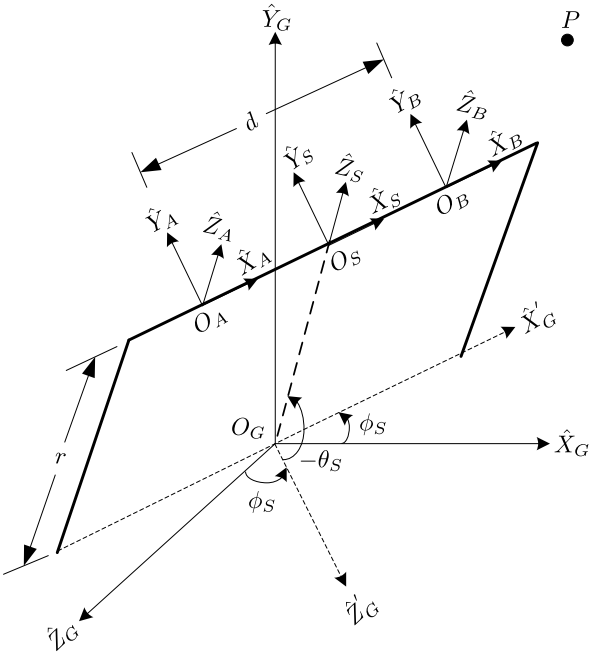


Figure 4. Coordinate frames.

The pan angle ϕ_S rotates along the $\hat{Y}_G$ axis, and the tilt angle θ_S rotates along the $\hat{X}'_G$ axis. The position vector of the target object P relative to camera A is denoted as $[x_A \; y_A \; z_A]^{\mathrm{T}}$. In this study, we assumed that cameras A and B were identical. Based on (1), after obtaining the image coordinates of the objects of camera A and camera B, respectively, the position of the target object in the coordinate frame of camera A can be written as:

$$\begin{aligned} x_A &= \frac{d}{x_{A_im} - x_{B_im}}[x_{A_im} - o_x - \frac{f_{s\theta}}{f_{sy}}(y_{A_im} - o_y) \\ y_A &= d\frac{f_{sx}}{f_{sy}} \cdot \frac{y_{A_im} - o_y}{x_{A_im} - x_{B_im}} \\ z_A &= f_{sx}\frac{d}{x_{A_im} - x_{B_im}} \end{aligned} \tag{4}$$

where (x_{A_im}, y_{A_im}) and (x_{B_im}, y_{B_im}) are the coordinates of the target object obtained from cameras A and B, respectively. According to Figure 4, the relationship of the coordinate system between the stereo coordinate and the camera A coordinate can be given by:

$$\begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} - \begin{bmatrix} d/2 \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

where $[x_S \; y_S \; z_S]^{\mathrm{T}}$ is the position vector of target object P relative to the stereo coordinate frame. The position vector of target object P converts to the geodetic frame using a homogeneous transformation matrix H_S^G, which is given by:

$$\begin{bmatrix} x_G \\ y_G \\ z_G \\ 1 \end{bmatrix} = H_S^G \begin{bmatrix} x_S \\ y_S \\ z_S \\ 1 \end{bmatrix} \tag{6}$$

where $[x_G \; y_G \; z_G]^{\mathrm{T}}$ is the position of target object P relative to the geodetic coordinate system and

$$H_S^G = \begin{bmatrix} \cos\phi_s & -\sin\phi_s \sin\theta_s & \sin\phi_s \cos\theta_s & r\sin\phi_s \cos\theta_s \\ 0 & \cos\theta_s & \sin\theta_s & r\sin\theta_s \\ -\sin\phi_s & -\cos\phi_s \sin\theta_s & \cos\phi_s \cos\theta_s & r\cos\phi_s \cos\theta_s \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Using Equations (4)–(6), the pixel position of the target object in the image frames of the two cameras can be used to determine the position vector of target object P in the world frame.

Figure 5 illustrates the position of target object P in the geodetic frame. From Figure 5 using simple geometry, the angular displacement of the pan-axis can be determined by

$$\phi_S = \sin^{-1} \frac{x_G}{\sqrt{x_G^2 + z_G^2}} \tag{7}$$

where the angular displacement of the tilt-axis is:

$$\theta_S = \sin^{-1} \frac{y_G}{\sqrt{x_G^2 + y_G^2 + z_G^2}} \tag{8}$$

The direct current (DC) servo motors drive the pan-and-tilt platform to maintain the continuous tracking of θ_S and ϕ_S. The θ_S and ϕ_S angular commands are sent in real-time to keep the target object in the FoV of the cameras.

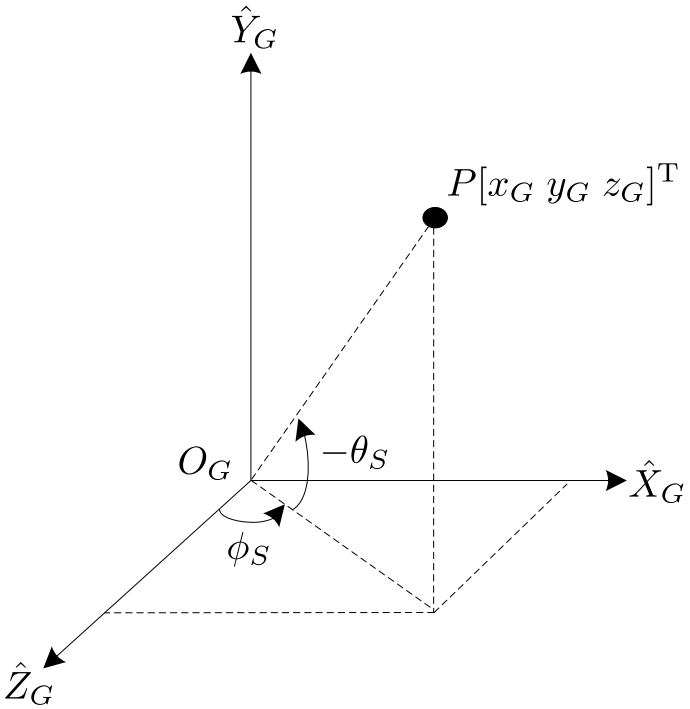


Figure 5. Angular motor displacements.

4. Trajectory Estimation and Prediction

Visual tracking is challenging due to image variations caused by camera noise, background changes, illumination changes, and object occlusion. The above-mentioned problems will deteriorate the tracking performance and may even cause the loss of target objects. In this work, a Kalman filter [17,18] was applied to enhance the robustness of the designed visual tracking system for the purpose of estimating the target object's position and velocity. The projectile motion trajectory for the flying ball was used to predict the touchdown point. The mobile robot was commanded to catch the ball at the appropriate location. A brief introduction to the Kalman filter is given below.

A state-space system model is described by

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + \varepsilon_k \\ y_k &= Cx_k + \omega_k \end{aligned} \tag{9}$$

where x_k is the state of the system; u_k is the input, and y_k is the measurement. Matrices A, B, and C are the state transition matrix, input matrix, and output matrix, respectively. The state and measurement noises are denoted as ε_k and ω_k. The zero-mean normal-distribution Gaussian white noise is assumed to be these two noises, and the covariances are Q_k and R_k, respectively.

The use of the Kalman filter involves two major procedures: the time updating step and the measurement updating step [17,18]. Time updating is used to estimate the probability outcome of the next state. The measurement updating step is used to update the estimated state with the measured information. This updated state is used in the time updating step of the next cycle. The Kalman filter is done recursively, and the recursive formulas are

- Time updating step:

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^{\mathrm{T}} + Q_k. \end{aligned} \tag{10}$$

- Measurement updating step:

$$K_k = P_k^- C^{\mathrm{T}}(CP_k^- C^{\mathrm{T}} + R_k)^{-1} \tag{11}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-) \tag{12}$$

$$P_k = (I - K_k C)P_k^-. \tag{13}$$

In (10)–(13), $\hat{x}_k^-$ denotes the a priori predicted state, and $\hat{x}_k$ is the optimal estimated state after the measurement is updated. P_k^- and P_k are the a priori and a posteriori estimate error covariances, respectively. K_k is known as the Kalman gain, and $y_k - \hat{y}_k$ is called the measurement residual. Based on (11), the a posteriori estimate error covariance P_k is minimized by K_k, and $\hat{x}_k$ is, hence, optimized.

Now, we consider the dynamics of a flying ball. The position and velocity vector of the ball are denoted as $[x_W \; y_W \; z_W]^{\mathrm{T}}$ and $[v_x \; v_y \; v_z]^{\mathrm{T}}$ in the world coordinate system, respectively. We assumed that the flying trajectory of the ball is not interfered with by other objects. The forces considered in this work are the gravitational force, the buoyancy force, and the drag force. Other non-stated forces, such as the Magnus force [19], are ignored.

The buoyancy force vector is denoted as $[0 \;\; 0 \;\; \frac{4}{3}\pi R^3 \rho g]^{\mathrm{T}}$, where the radius of the ball is R (m); the air density is ρ (kg/m^3), and the gravitational acceleration at sea level is g (m/s^2). The drag force is assumed to be proportional to the velocity. According to Stokes's law [19], the drag force is $6\pi\mu R v_w$, where μ (kg/m·s) is the dynamic viscosity of the air, and $v_w = [v_x \; v_y \; v_z]^{\mathrm{T}}$. Let the mass of the ball be denoted as m (kg). The equation that governs the motion of a flying ball can be written as

$$m\frac{dv_w}{dt} = \begin{bmatrix} 0 \\ -m'g \\ 0 \end{bmatrix} - bv_w \tag{14}$$

where

$$b \triangleq 6\pi\mu R \tag{15}$$

$$m' \triangleq m - \frac{4}{3}\pi R^3 \rho. \tag{16}$$

By discretizing (14) with the sampling period Δt, we obtain the dynamic model of the system as shown below:

$$\begin{aligned} x_W(t) &= v_x(t-\Delta t)\Delta t + x_W(t-\Delta t) && (17)\\ y_W(t) &= v_y(t-\Delta t)\Delta t + y_W(t-\Delta t) && (18)\\ z_W(t) &= v_z(t-\Delta t)\Delta t + z_W(t-\Delta t) && (19)\\ v_x(t) &= (1-\frac{b}{m})v_x(t-\Delta t)\Delta t && (20)\\ v_y(t) &= -\frac{m'}{m}g\Delta t + (1-\frac{b}{m})v_y(t-\Delta t)\Delta t && (21)\\ v_z(t) &= (1-\frac{b}{m})v_z(t-\Delta t)\Delta t. && (22) \end{aligned}$$

From Equations (17)–(22), a discrete-time linear state-space form (9) can be further written for the system model as

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} x_W(t) & y_W(t) & z_W(t) & v_x(t) & v_y(t) & v_z(t) \end{bmatrix}^{\mathrm{T}}\\ x_k &= \begin{bmatrix} x_W(t-\Delta t) & y_W(t-\Delta t) & z_W(t-\Delta t) \\ v_x(t-\Delta t) & v_y(t-\Delta t) & v_z(t-\Delta t) \end{bmatrix}^{\mathrm{T}}\\ u_k &= -g \end{aligned} \quad (23)$$

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & (1-\frac{b}{m})\Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-\frac{b}{m})\Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-\frac{b}{m})\Delta t \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{m'}{m}\Delta t & 0 \end{bmatrix}^{\mathrm{T}}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Based on the above model, the Kalman filter is applicable for visual tracking to estimate the ball's trajectory. Therefore, for the visual tracking of the flying ball, the Kalman filter and a constant acceleration model were used in this work.

Using the estimated position, the estimated velocity, and the projectile motion Formulas (17)–(22), the future trajectory and touchdown point were predicted for the flying ball. Since the uncertainty of the initial value P_0 affects the estimation accuracy and convergence of the Kalman filter, this can cause the active stereo vision camera to fail to track the ball, and therefore the ball leaves the FoV. The inaccuracy of the initial value is the result of an unreasonable flight trajectory, such as the ball not being thrown into the air.

In this work, deep learning was used to obtain the initial value of the state estimation error covariance P_0. Figure 6 describes the application using the Kalman filter combined with deep learning. After the camera obtains the position and velocity of the flying ball, the Kalman filter estimates the next position and commands the active stereo vision camera to track the ball.

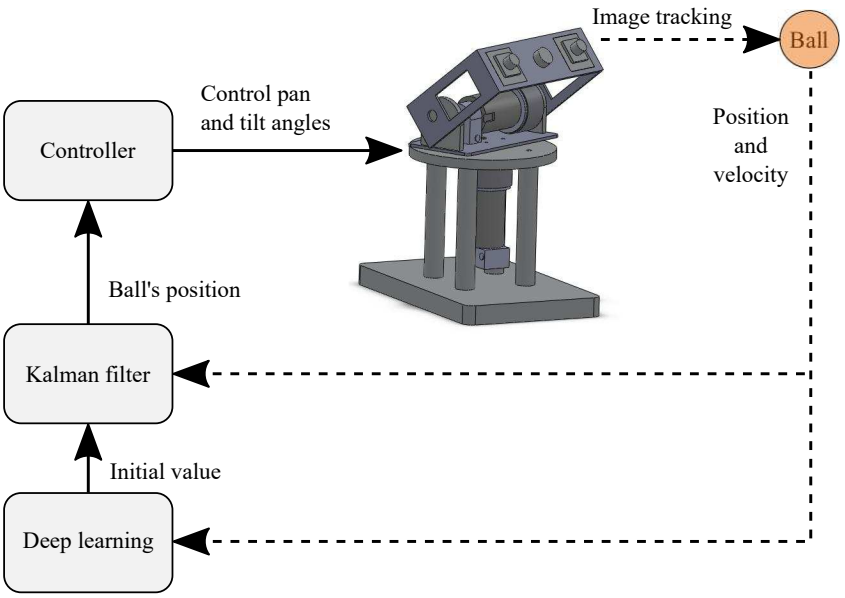


Figure 6. Kalman filter combined with deep learning.

Figure 7 describes the deep learning architecture. The input layer receives the input x_k and x_{k-1}, which is the data learned by the neural network. The network is based on AlexNet [20], which is a deep convolutional neural network. These input matrices represent the position vector and velocity vector of the flying ball. The last layer is called the output layer, which outputs an initial estimated error covariance P_0, representing the neural network's result. The hidden layers are performed in the layers between the input and output layers. Deep learning is helpful to attenuate the initial value deviation in the filtering process. Once the ball's predicted touchdown point is obtained, the point-to-point path planning is used to command the mobile robot to move toward the touchdown point in advance to catch the ball. This ball-catching strategy is discussed in a later section.

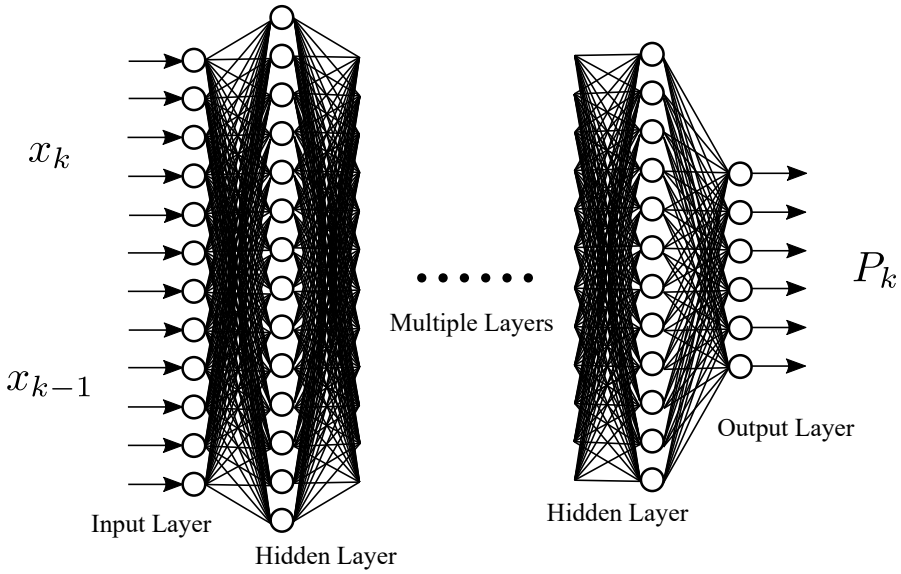


Figure 7. The architecture of the deep learning.

5. Controller Design of the Omni-Directional Wheeled Mobile Robot

Figure 8 shows a schematic diagram of an omni-directional wheeled mobile robot from the top view. This mobile robot consists of a rigid body and three omni-directional wheels labeled 1, 2, and 3. The wheels were arranged at an equal distance from the center of the robot platform with a 120° interval. This setup allowed the robot to move freely in

any direction. In this study, $(O_W, \hat{Z}_W, \hat{X}_W)$ is the world coordinate system. $(O_M, \hat{Z}_M, \hat{X}_M)$ is the body coordinate system with the origin attached to the center of mass of the robot. The direction of the $\hat{X}_M$-axis is aligned to wheel 1, as shown in Figure 8. We assumed that the wheels can roll without slipping.

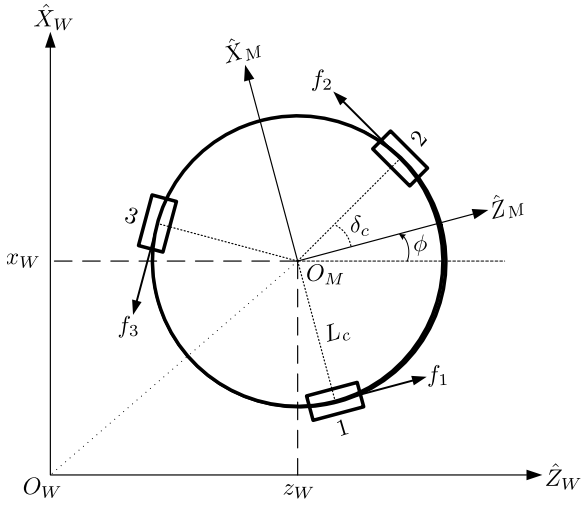


Figure 8. Top view of the omni-directional mobile robot.

The parameters for an omni-directional mobile robot are listed as follows:

M_t	mass of the mobile robot.
L_c	radius of the mobile robot.
f_1, f_2, f_3	the reaction force applied by the ground to the omni-directional wheel, where the direction is vertical to wheel axes 1, 2, and 3, respectively.
R_w	radius of the omni-directional wheels.
δ_c	the angle between wheel 2 and the $\hat{Z}_M$-axis of the mobile coordinate system; the value is fixed to 30°.
ϕ	the rotation angle of the omni-directional mobile robot.
I_y	moment of inertia of the omni-directional mobile robot about $\hat{Y}_M$-axis.
$\theta_1, \theta_2, \theta_3$	the rotation angle of omni-directional wheels 1, 2, and 3, respectively.
$\omega_1, \omega_2, \omega_3$	the angular velocity of omni-directional wheels 1, 2, and 3, respectively.
$[z_W \; x_W]^{\mathrm{T}}$	the position for the center of mass of the robot relative to the world coordinate system.

According to [21], the dynamics and kinematics of the mobile robot can be described by:

$$\begin{bmatrix} \ddot{z}_W \\ \ddot{x}_W \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} M_t & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & I_y \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ L_c & L_c & L_c \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{R_w} \begin{bmatrix} 1 & 0 & L_c \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & L_c \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & L_c \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{z}_W \\ \dot{x}_W \\ \dot{\phi} \end{bmatrix}. \tag{25}$$

In this study, the brushed DC servo motors drove the omni-directional wheels. We also assumed that the motor's electrical time constant was smaller than the mechanical time constants and that the motor friction was negligible. The model of a DC motor is, thus, reduced to

$$\tau_m = \frac{K_t}{R_a} u - \frac{{K_t}^2}{R_a} \omega_m. \tag{26}$$

In (26), τ_m, u, K_t, ω_m, and R_a represent the motor torque, control voltage, motor torque constant, angular velocity of the motor, and armature resistance, respectively. The traction force f of the wheel is given by

$$f = \frac{n}{R_w} \tau_m \tag{27}$$

where n is the gear ratio. The three motors used in this mobile robot were assumed to be identical. Therefore, combining (26) and (27), the relationship between f, u, and ω is given by

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \frac{nK_t}{R_w R_a} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - \frac{n^2 {K_t}^2}{R_w R_a} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \tag{28}$$

From (24), (25), and (28), the dynamics of the robot can be presented as

$$\ddot{P}_w = A_w \dot{P}_w + B_w(\phi) U_C \tag{29}$$

where

$$P_w = \begin{bmatrix} z_W & x_W & \phi \end{bmatrix}^{\mathrm{T}}, \; U_C = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^{\mathrm{T}} \tag{30}$$

$$A_w = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_1 & 0 \\ 0 & 0 & a_2 \end{bmatrix} \tag{31}$$

$$B_w(\phi) = \begin{bmatrix} 2b_1 \cos(\phi) & -b_1 \cos(\phi) - \sqrt{3} b_1 \sin(\phi) & -b_1 \cos(\phi) + \sqrt{3} b_1 \sin(\phi) \\ 2b_1 \sin(\phi) & -b_1 \sin(\phi) + \sqrt{3} b_1 \cos(\phi) & -b_1 \sin(\phi) - \sqrt{3} b_1 \cos(\phi) \\ b_2 & b_2 & b_2 \end{bmatrix} \tag{32}$$

with

$$a_1 = \frac{-3n^2 K_t^2}{2R_w^2 M_t R_a}, \; a_2 = \frac{-3n^2 K_t^2 L_c^2}{R_w^2 I_y R_a}, b_1 = \frac{nK_t}{2R_w M_t R_a}, \; b_2 = \frac{nK_t L_c}{R_w I_y R_a}. \tag{33}$$

We assume that the predicted touchdown point of the ball in the world coordinate system is $[z_{bw} \; x_{bw}]^{\mathrm{T}}$. The position reference to the mobile robot is set to be:

$$P_{bw}(t) = \begin{bmatrix} z_{bw}(t) & x_{bw}(t) & \phi \end{bmatrix}^{\mathrm{T}}$$

where the rotation angle ϕ is assumed to be 0. The tracking error is defined as follows:

$$e(t) = P_{bw}(t) - P_w(t).$$

This gives

$$\ddot{e} = \ddot{P}_{bw} - [A_w \dot{P}_w + B_w(\phi) U_C]. \tag{34}$$

A new control input U [22] is defined as follows:

$$U \triangleq \ddot{P}_{bw} - [A_w \dot{P}_w + B_w(\phi) U_C]. \tag{35}$$

From (35) and (34), we obtain

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} U. \tag{36}$$

From (34), the feedback control U_c can be written as

$$U_C = B_w(\phi)^{-1}[\ddot{P}_{bw} - A_w\dot{P}_w - U].$$

In the form used in (36), the system is decoupled into a linear system, and the PID control algorithm is used for tracking control. In this case, the following PID control:

$$\begin{aligned} \dot{\varepsilon} &= e \\ U &= -K_p e - K_d\dot{e} - K_i\varepsilon \end{aligned} \tag{37}$$

where K_d, K_p, and K_i are 3 by 3 diagonal PID gain matrixes that equal diag$\{k_{d_i}\}$, diag$\{k_{p_i}\}$, and diag$\{k_{i_i}\}$, respectively, and i = 1, 2, and 3. From (36) and (37), it follows that the closed-loop tracking error system is given by:

$$\frac{d}{dt}\begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -K_i & -K_p & -K_d \end{bmatrix}\begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix}.$$

According to the Routh–Hurwitz criterion [23], PID gain values k_p, k_i, and k_d must satisfy:

$$k_{i_i} < k_{d_i}k_{p_i},\ i = 1,2,3.$$

To obtain closed-loop stability, the PID control gain values are based on the control design method proposed in [24]. The phase margin and the gain margin were set to 45° and 6.0206 dB in this work.

6. Implementation of the Designed System

Figure 9 shows a block diagram of the proposed ball-catching system. This system consisted of an omni-directional wheeled mobile robot and an image processing system that included an active stereo vision camera and a static vision camera.

Figure 10a shows the active stereo vision camera. The MT9P001 image sensor was used, which is a complementary metal-oxide-semiconductor (CMOS). It can capture 640 × 480 pixels in the quantized RGB format at 60 frames per second (FPS). The cameras were attached onto a field-programmable gate array (FPGA) board. This FPGA board was used to configure the cameras and to acquire images. In the real-time image process for the acquired images, a DSP (TMS320DM6437) board was used.

An optical encoder with the resolution of 500 pulses/rev was used to measure the motors' angular displacement in the pan-and-tilt platform. Another DSP board (TMS320F2812), with two quadrature encoder pulse (QEP) units and one pulse width modulation (PWM) signal generator unit was used to acquire the angular displacement and rotational direction of the motor from the quadrature encoder and to control the pan-and-tilt platform motors. In addition, the Kalman filter was implemented to mitigate the measurement noises and to predict the motion of the ball.

The static vision camera was mounted above the work area, where the FoV of the camera covered the entire work area (length: 2.5 m, width: 2.5 m, and height: 3 m) as shown in Figure 1. This camera was used to locate and navigate the omni-directional wheeled mobile robot. As with the active stereo vision camera, the Kalman-filter-based vision tracking and image processing algorithms were implemented in the DSP board (TMS320DM6437).

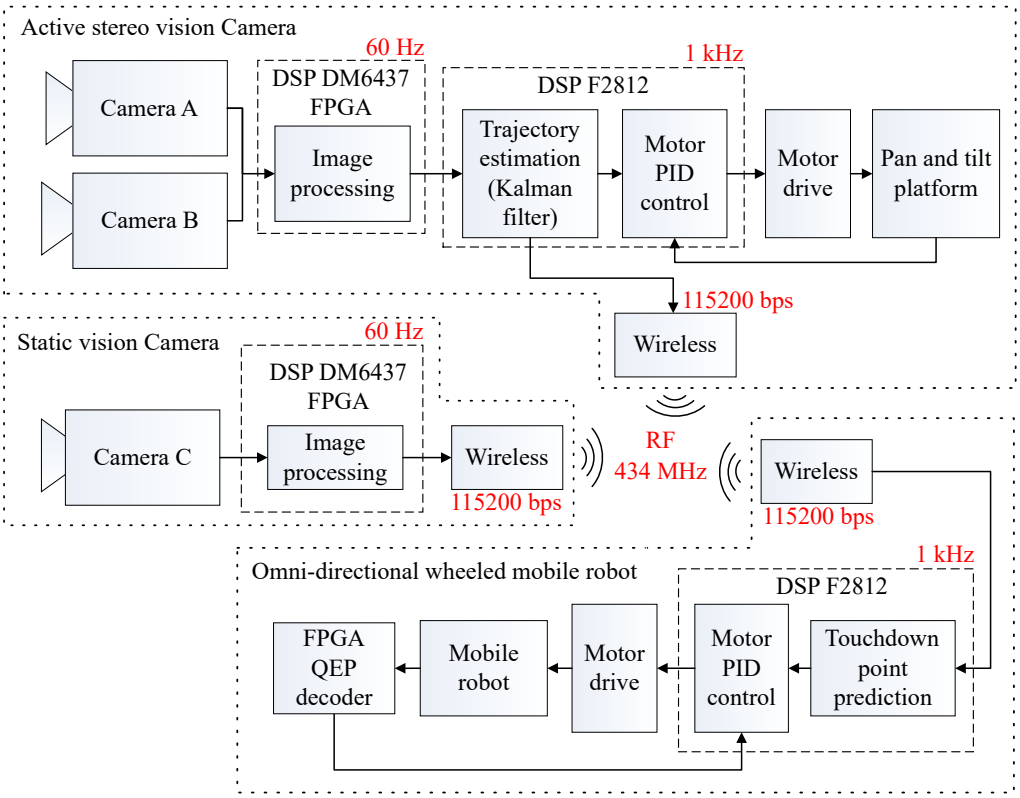
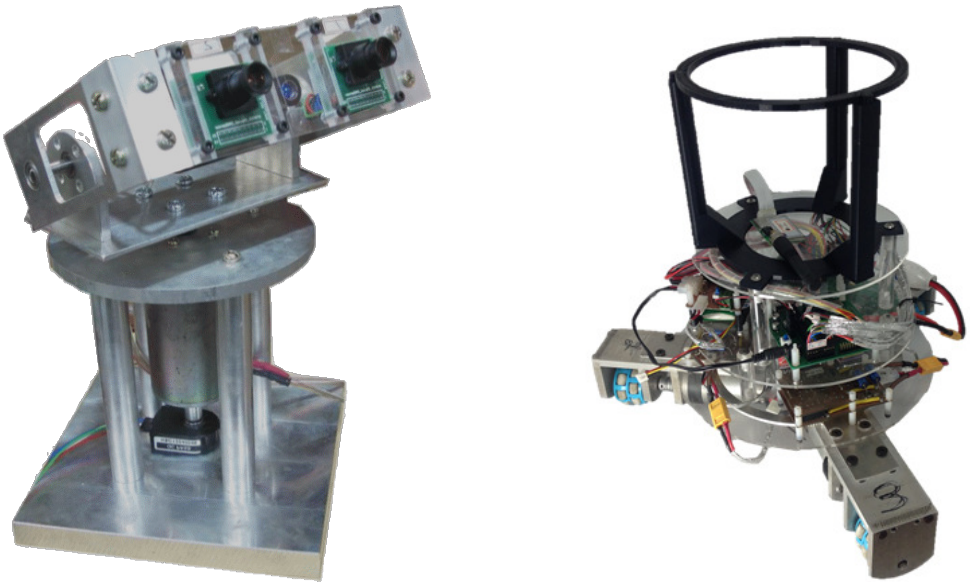


Figure 9. Block diagram of the proposed system.

The mobile robot, as stated in Section 6, consisted of three brushed DC motors used to drive the omni-directional wheels. A DSP board (TMS320F2812) was used for PID control of the motors and the touchdown point prediction for the ball. To obtain the wheel's angular displacement, an optical encoder with a 500 pulse/rev resolution was mounted to the shaft of each wheel. These optical encoders generate quadrature encoder signals to the QEP circuit on an FPGA board for decoding. The wheels' angular velocities were obtained by differentiating the angular displacement using the sampling time, and a low-pass filter was then applied to attenuate the high-frequency noises.

The robot's position and orientation were determined with the static vision camera and a dead reckoning algorithm based on the motor encoder measurements. The PWM signals were generated according to the designed PID control laws to drive each of the motors. Figure 10b shows a basket 0.16 m in diameter mounted on the top of the robot for the purpose of catching the ball.

All of the systems stated above were communicated with using wireless communication modules, as shown in Figure 9. The active stereo vision camera obtained the position and velocity of the ball and then sent it to the mobile robot to predict the touchdown point. The static vision camera obtained the position and direction of the mobile robot, and then sent it to the mobile robot for navigation and positioning through wireless communication.



(a) (b)

Figure 10. Omni-directional wheeled mobile robot.

7. Simulation and Experimental Results

7.1. Touchdown Point Prediction

The prediction of the touchdown point of the target ball was first verified through a numerical simulation using MATLAB/Simulink. The mass and the radius of the ball were set at 0.07 kg and 0.004 m, respectively. The initial location of the ball in meters was $[3.9 \ 1.1 \ 5.15]^{\mathrm{T}}$, and the initial velocity in meters per second was $[-2.1 \ 5.2 \ -1.9]^{\mathrm{T}}$. The sampling period was 0.0167 s. Although the dynamics of a flying ball were considered, it was impossible to precisely model the ball's rotation and the airflow field conditions during flight.

Q_k mainly represents these noises. Since these noises have little effect on the flight of the ball, Q_k can be reasonably obtained according to experimental measurements. R_k models the light variation noise, which cannot be processed by the camera calibration method; however, the slight variation was not significant in this study. A reasonable value of R_k can be obtained through experimental measurements. Thus, the covariance matrices Q_k and R_k were assumed to be constant during the motion, and those used in this simulation are given below.

$$Q_k = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.005 \end{bmatrix}$$

$$R_k = \begin{bmatrix} 0.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.16 \end{bmatrix}.$$

The initial error covariance matrix prediction was

$$P_0 = \begin{bmatrix} 1.6 \times 10^{-9} & 0 & 0 & 3.2 \times 10^{-7} & 0 & 0 \\ 0 & 1.6 \times 10^{-9} & 0 & 0 & 3.2 \times 10^{-7} & 0 \\ 0 & 0 & 1.6 \times 10^{-9} & 0 & 0 & 3.2 \times 10^{-7} \\ 3.2 \times 10^{-7} & 0 & 0 & 6.4 \times 10^{-5} & 0 & 0 \\ 0 & 3.2 \times 10^{-7} & 0 & 0 & 6.4 \times 10^{-5} & 0 \\ 0 & 0 & 3.2 \times 10^{-7} & 0 & 0 & 6.4 \times 10^{-5} \end{bmatrix}.$$

The simulated results of the estimated, measured, and actual trajectories of the flying ball are shown in Figure 11. We observed that the initial estimated and actual trajectories were significantly different. After several iterations, the estimated trajectory overlapped the actual trajectory with reasonable accuracy. The predicted touchdown point obtained using Equations (17)–(22) in each step of the estimation is plotted in Figure 12. The initially predicted location was quite far from the actual one; however, it converged to the real touchdown point with additional iterations.

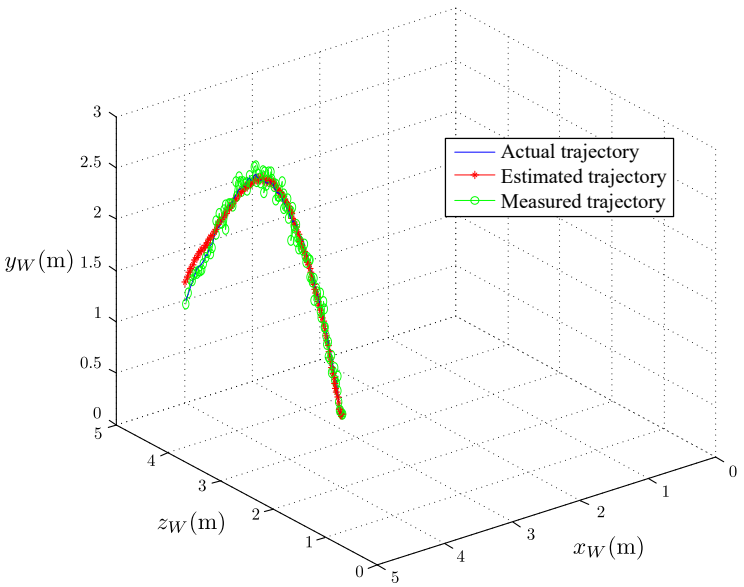


Figure 11. Comparison of the actual trajectory, estimated trajectory, and measured trajectory.

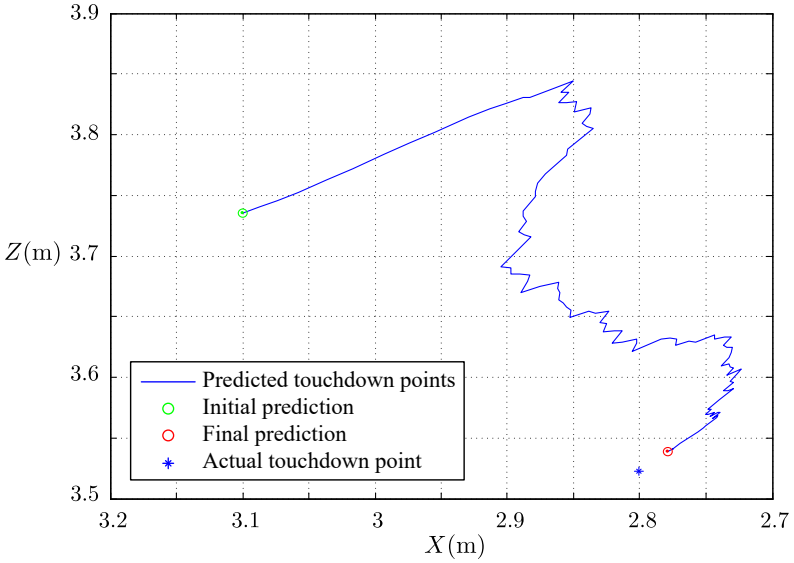


Figure 12. Prediction of the touchdown points.

7.2. Improvement of Projectile Prediction

Deep learning was used to improve the projectile prediction. The training data sets are shown in Figure 13, with a total of 80 projectile data. The deep learning ANN was trained using many projectile trajectory data in Figure 13 to accurately obtain the ball's projectile in the initial state to set a reasonable P_0 value. The covariance matrices Q_k and R_k were the same as shown in the above simulation, and the initial estimate error covariance matrix P_0 was obtained through the deep learning ANN after two iterations of measurements, and the Kalman filter was set to update P_k during the ball motion. Figure 14 shows the result of predicting the touchdown point using P_0 obtained by the deep learning ANN. From the figure, the use of deep learning increased the accuracy of predicting the touchdown point.

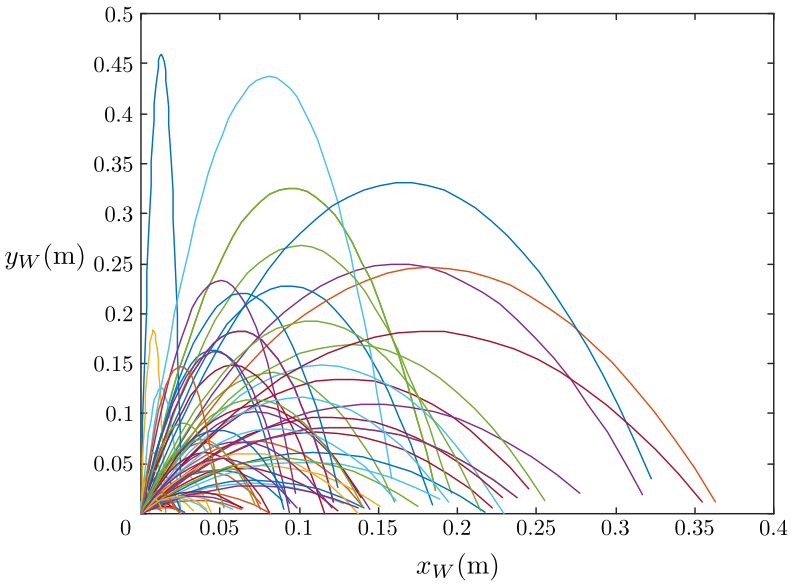


Figure 13. The training data sets.

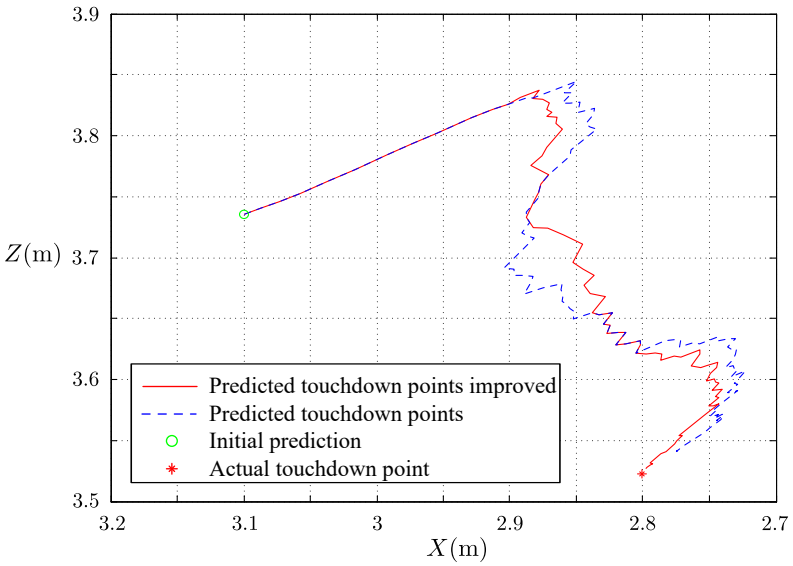


Figure 14. Prediction of the touchdown points improved by deep learning.

7.3. Free Falling Ball Experimental

For further validation, the proposed visual servoing system for ball-catching was developed and performed in the experimental setup shown in Figure 15. An active stereo vision camera achieved the visual tracking task for the flying ball with two pan-and-tilt cameras. The omni-directional wheeled mobile robot was navigated to catch the ball.

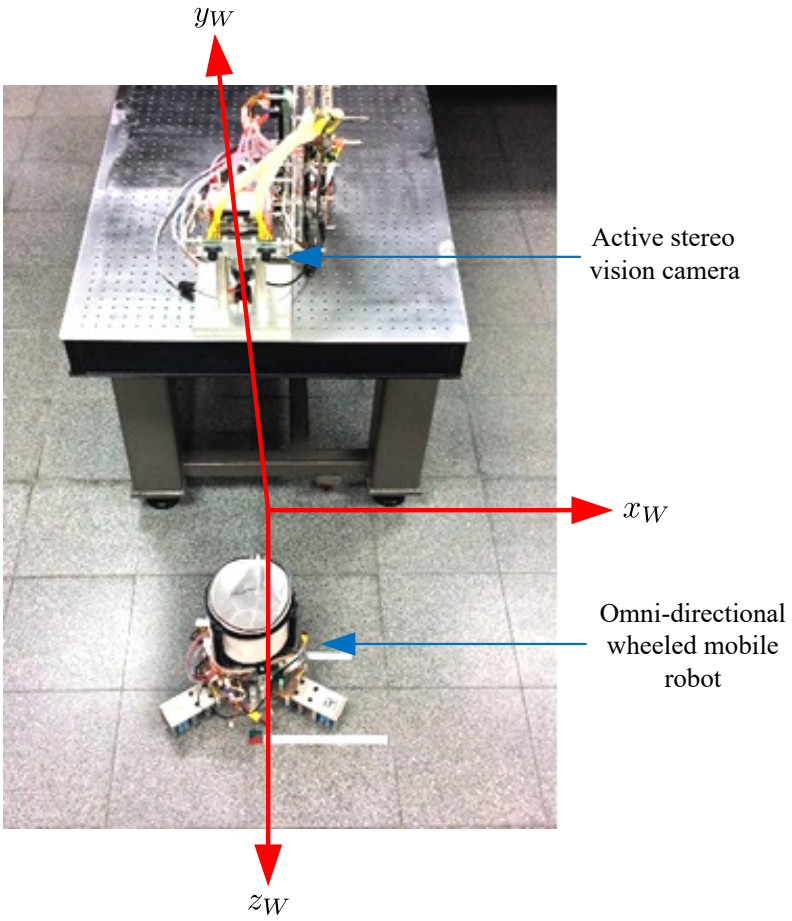


Figure 15. The experimental setup.

To verify the performance of the active stereo vision camera, an orange ball was dropped a suitable distance in front of the system. The visual tracking system was expected to be able to track the ball and keep it in the FoV of both cameras. Figure 16 shows a series of images taken by the active stereo vision camera. In both image series (image A and image B), the ball was kept in the images. The results of a free falling ball visual tracking of this study and the results of our previous work are given in the following Figures 17 and 18.

Figure 17 shows the estimated trajectory of the ball, and Figure 18 shows the command and time response of the pan and tilt angles, respectively. In the case of a free-drop object, the object only moved in the vertical direction with no horizontal displacement. As shown in Figure 18a, the command and time response of the tilt angle changed over time, while the pan angle remained at 0 the entire time. Both the pan and tilt motor were able to precisely follow the commands. Figure 18b, compared with our previous results, shows that the tracking response was improved in this study. With these results, we concluded that the active stereo vision camera was able to successfully perform visual tracking.

7.4. Catching a Flying Ball Experiment

Next, the active stereo vision camera was tested to track a flying ball. In this test, we threw the target ball at a high initial velocity. Figure 19 shows a series of images captured by the active stereo vision camera. Similar to the free-drop ball case, the visual tracking system was able to keep the ball in the FoV of the cameras. The results of a flying ball visual tracking of this study and the results of our previous work are given in Figures 20 and 21.

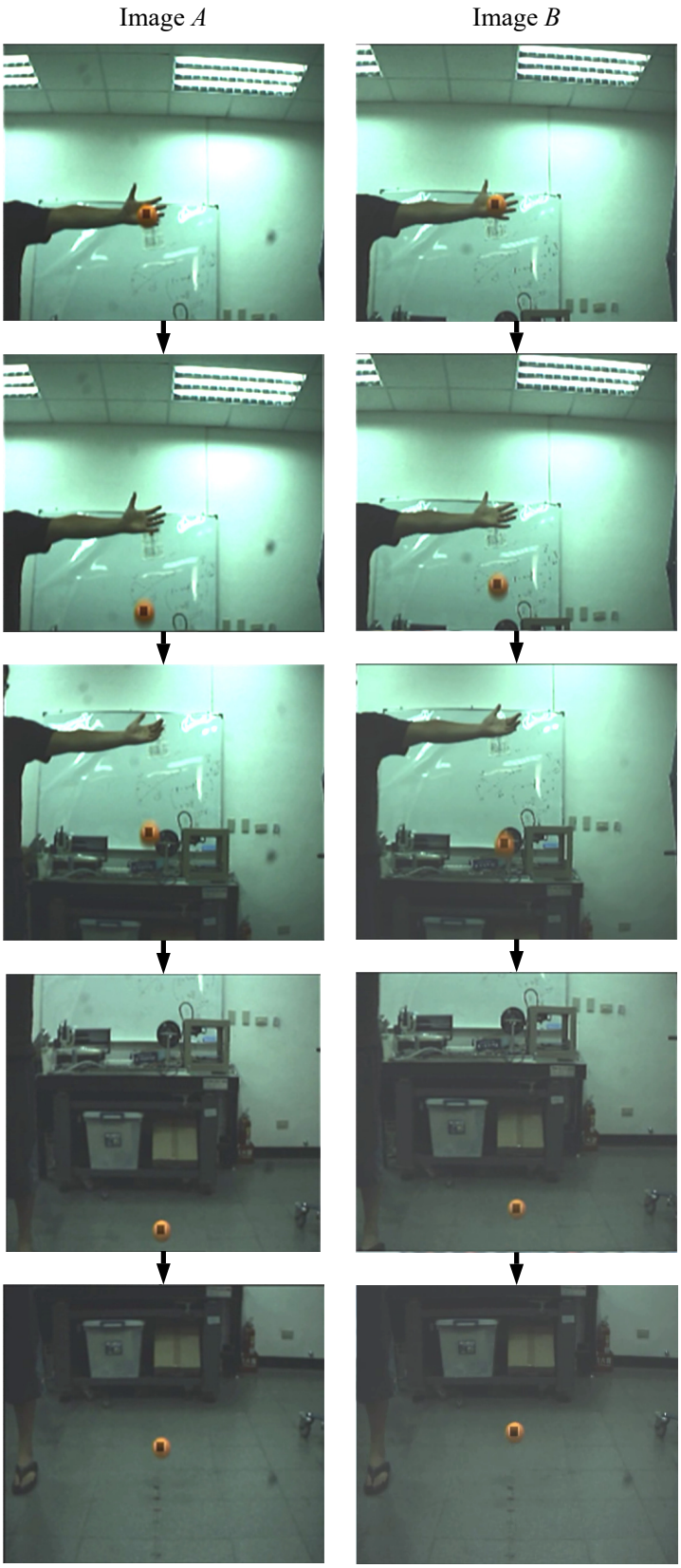


Figure 16. A free falling ball: the sequence of images captured.

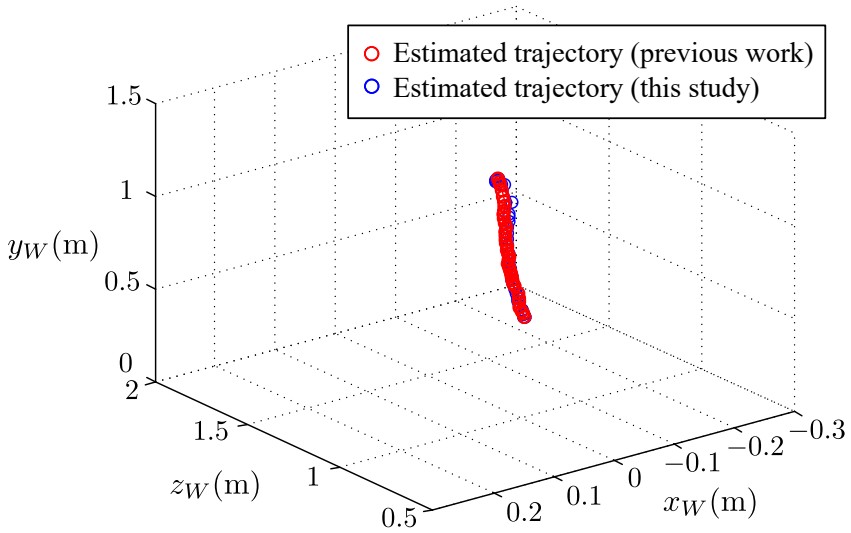


Figure 17. A free falling ball: estimated trajectory of the ball.

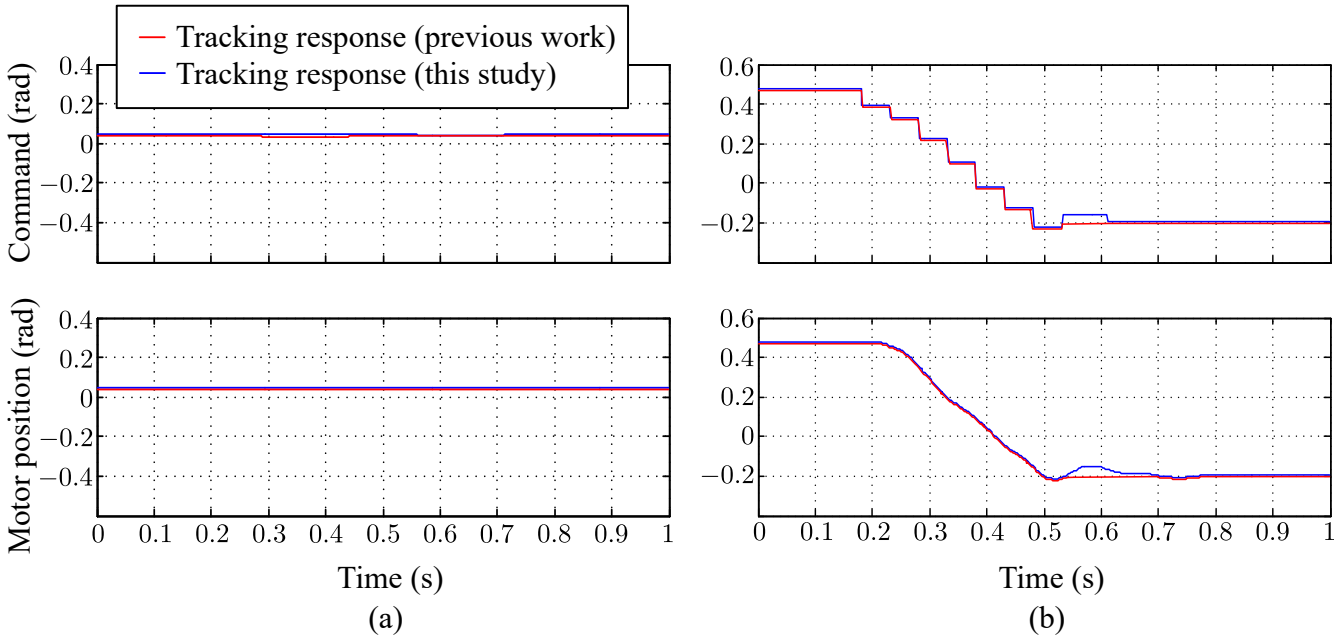


Figure 18. A free falling ball: tracking response of (**a**) the pan-axis motor and (**b**) the tilt-axis motor.

The estimated trajectory of the ball is shown in Figure 20, and the command and time response of the pan and tilt angles are shown in Figure 21, respectively. The results are similar to those obtained for the free-drop ball case, where both of the motors were able to precisely respond to the commands. This experimental results indicate that the active stereo vision camera can track a target and keep it in the FoV even at a high initial velocity.

The effectiveness of the touchdown point prediction was investigated through experiments. In this test, the ball was thrown toward the stereo vision camera, and then the system estimated the ball's position and velocity. With the estimated position and velocity, the future trajectory and touchdown point of the ball will be predicted by projectile motion Formulas (17)–(22). Figure 22 shows the experimental results. In this figure, the predicted trajectories converge to the measured and estimated trajectories, and the predicted and real touchdown points converge with a reasonable degree of accuracy. For comparison, the results of the touchdown point prediction of the previous work are given in Figure 22a. The results are improved.

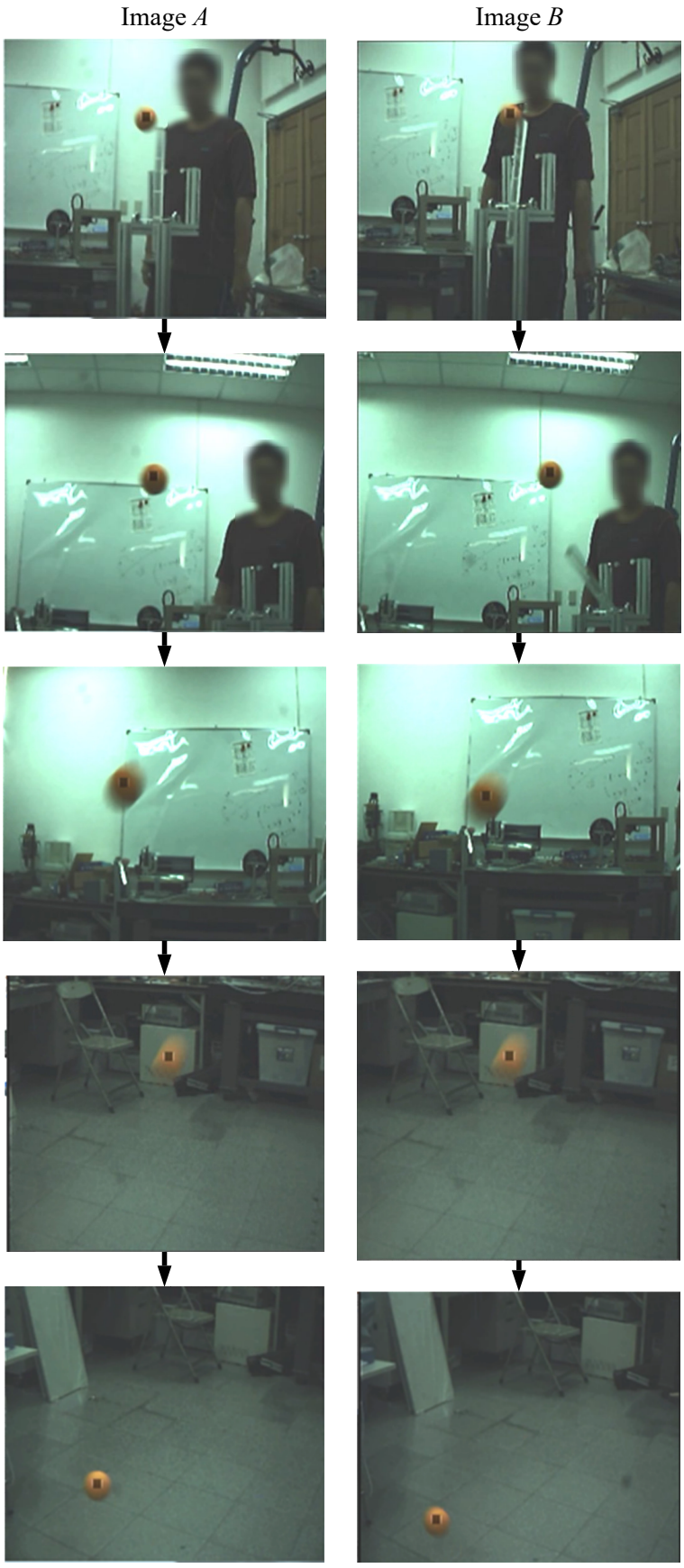


Figure 19. A flying ball: the sequence of images captured.

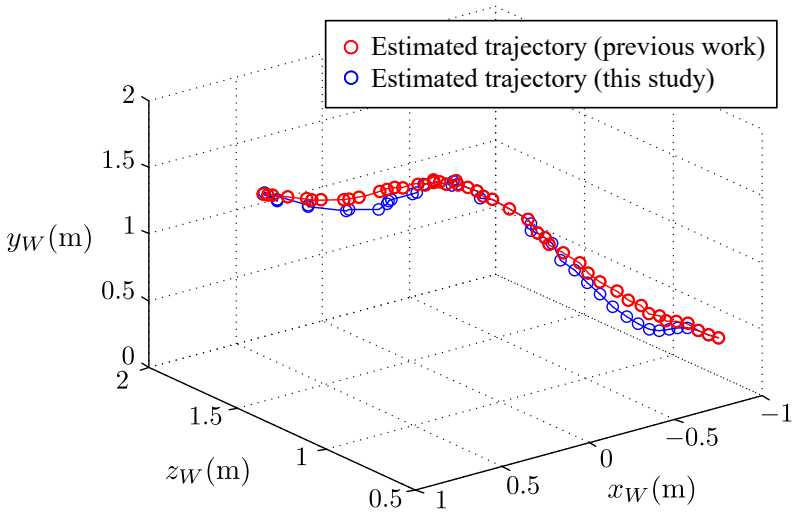


Figure 20. A flying ball: estimated trajectory of the ball.

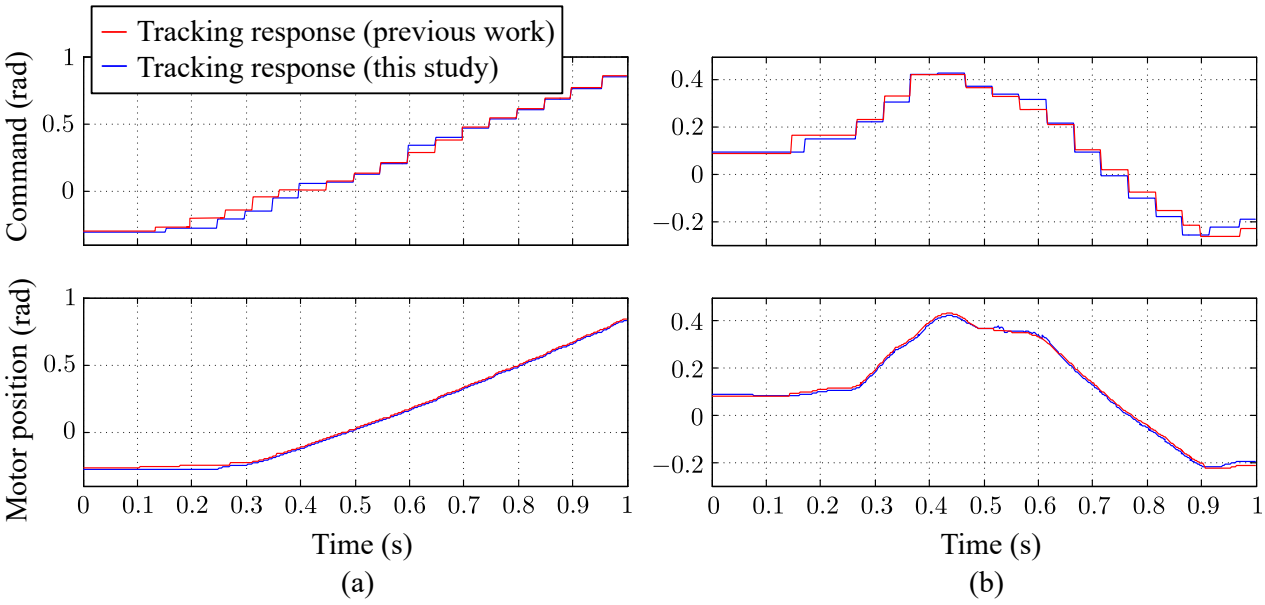


Figure 21. A flying ball: tracking response of (**a**) the pan-axis motor and (**b**) the tilt-axis motor.

Finally, we tested the complete ball-catching task. In this test, the ball was thrown a 2-m distance from the mobile robot. The robot moved to the predicted point as soon as possible after receiving the command, in order to catch the ball before it touches the ground. The results of the ball-catching task of this study and the results of our previous work are given in Figures 23 and 24.

In Figure 23 (3-directional view) and Figure 24 (view from the XZ plane) shown the ball's trajectories of estimation and measurement and the robot's path of movement. The result shows that the final predicted touchdown point and the location of the robot match very well. The residual difference between the measured trajectory and the estimated trajectory is the measurement residual. The mean value and standard deviation of the measurement residual of our previous work are 0.0529 m and 0.034 m, while those of this study are 0.0372 m and 0.019 m, respectively.

The methods proposed in this paper provided a smaller residual and variance and, therefore, more accurate measurements and more precise prediction of the touchdown point. This indicates that the robot could accomplish the ball-catching task effectively. Readers can access a link to a video clip to watch the developed system in action (https://youtu.be/En-6XcmkeBs, accessed on 30 April 2021). To determine the success rate, we

performed 50 throws with random initial positions. The robot successfully caught the ball 44 times out of 50 total trials, giving an overall success rate of 88%. The success rate of our previous work was about 74%.

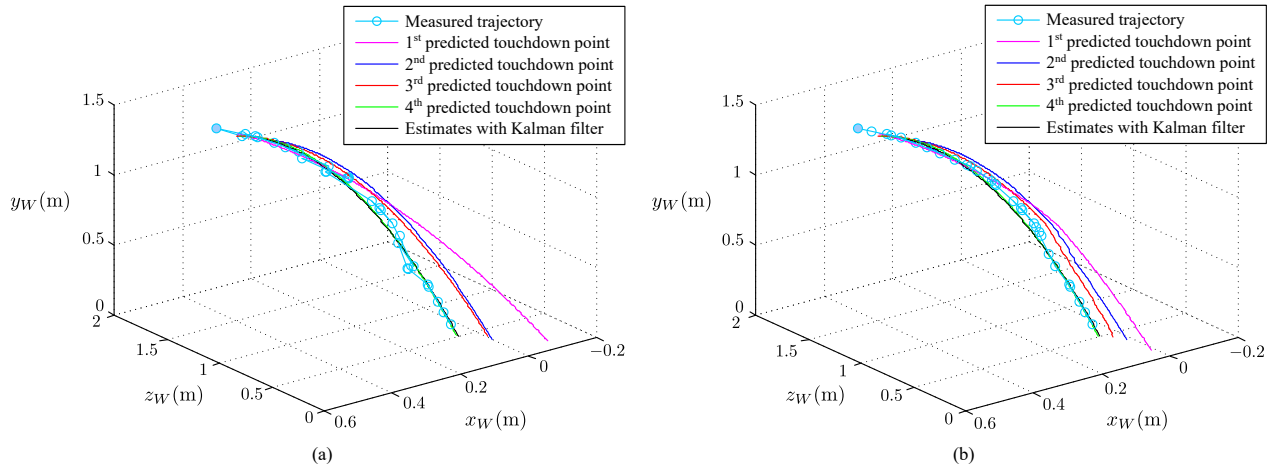


Figure 22. Prediction of touchdown points: (**a**) the previous work and (**b**) this study.

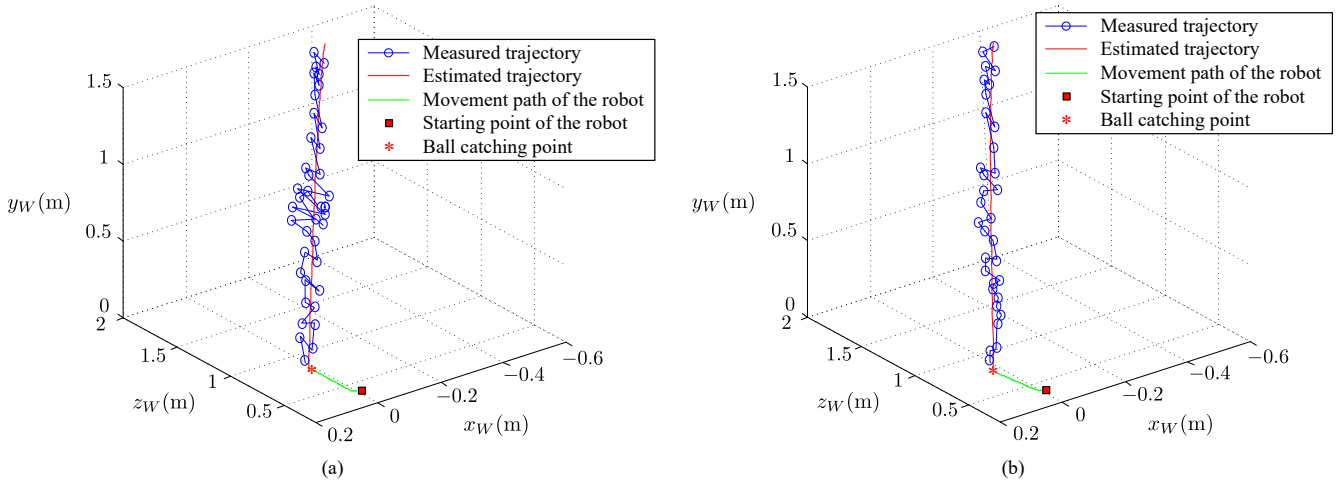


Figure 23. The ball's trajectories of estimation and measurement, and the robot's path of movement: (**a**) the previous work and (**b**) this study.

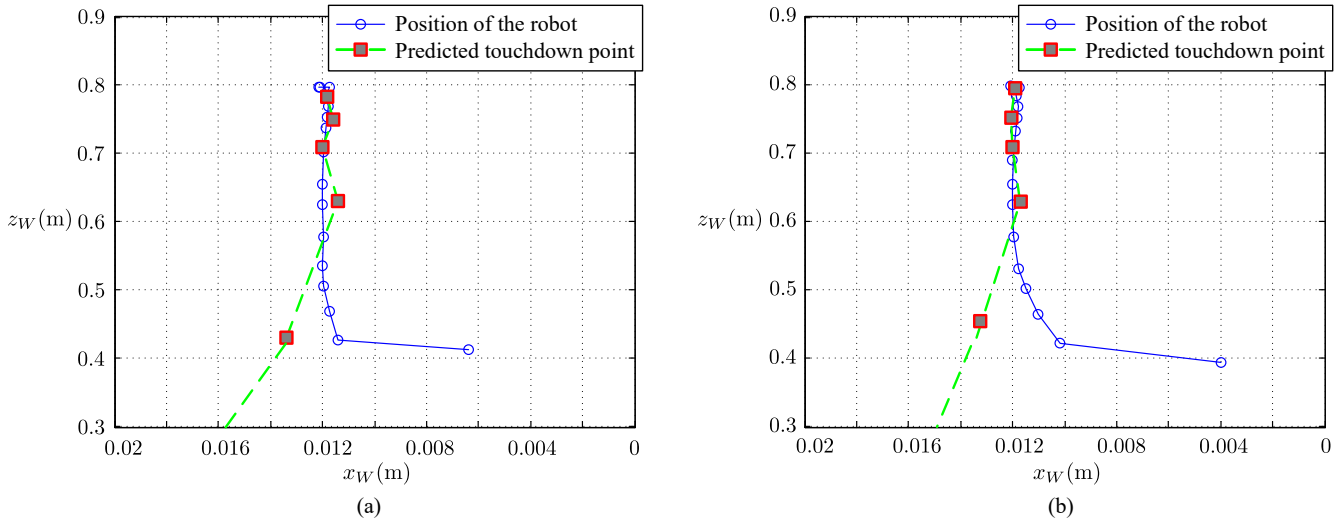


Figure 24. Moving path of the robot to catch the ball: (**a**) the previous work and (**b**) this study.

8. Concluding Remarks

In this research, a robotic ball-catching system was presented. This system consisted of multi-camera vision systems, an omni-directional wheeled mobile robot, and wireless communication. In the multi-camera vision system, the ball's motion was tracked with an active stereo vision camera while a static vision camera navigated a mobile robot. Using a Kalman filter and the motion governing equations of a flying ball, the ball's touchdown point was predicted with reasonable accuracy. For Kalman filtering, we found that the use of deep learning improved the accuracy and robustness.

The robot was controlled to move toward the predicted point to catch the ball before it hit the ground. The performance of the sub-systems and the proposed algorithms was verified through simulations and experiments. The results of the simulation matched the experimental results well. The experimental results confirmed that the developed robotic system combined with multi-camera vision systems could catch a flying ball.

The main contribution of this paper is to present the main issues and technical challenges of the design and system integration of a vision-based ball-catching system. Compared with the existing vision-based ball-catching systems, by combining an omni-directional wheeled mobile robot and an active stereo vision system, the ball-catching system proposed in this paper can be in a large workspace.

In future research, the image capture system used in this paper can be improved through the use of better Kalman filtering methods to restrain noise problems. Deep learning will be applied to image pre-processing, and it will be used on complex backgrounds with balls of different colors to verify the recognition accuracy. Different sensors (such as a laser range finder or RGB-D cameras) will be used for experimental comparisons. Different control laws will be applied to the omni-directional wheeled mobile robot in an attempt to increase the speed and accuracy of movement. In addition, worst-case conditions (long-distance movement or disturbance during the movement) will be applied to verify the robot's abilities. In the simulations and experiments, different ball conditions (speed, height, etc.) and disturbances during the flight will be used to verify the system robustness.

Author Contributions: Conceptualization, M.-T.H.; Methodology, S.-T.K. and M.-T.H.; Software, S.-T.K.; Validation, S.-T.K.; Formal analysis, M.-T.H.; Data curation, S.-T.K.; Writing-original draft, S.-T.K.; Writing-review and editing, S.-T.K. and M.-T.H.; Supervision, M.-T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Technology of Taiwan under Grant No. MOST 103-2221-E-006-184.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data collected through research presented in the paper are available on request from the corresponding author.

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

References

1. Bauml, B.; Wimbock, T.; Hirzinger, G. Kinematically Optimal Catching a Flying Ball with a Hand-Arm-System. In Proceedings of the International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 2592–2599.
2. Rapp, H.H. A Ping-Pong Ball Catching and Juggling Robot: A Real-Time Framework for Vision Guided Acting of an Industrial Robot Arm. In Proceedings of the 2011 International Conference on Automation, Robotics and Applications, Wellington, New Zealand, 6–8 December 2011; pp. 430–435.
3. Kim, S.; Shukla, A.; Billard, A. Catching Objects in Flight. *IEEE Trans. Robot.* **2014**, *30*, 1049–1065. [CrossRef]
4. Oka, T.; Komura, N.; Namiki, A. Ball juggling robot system controlled by high-speed vision. In Proceedings of the IEEE International Conference on Cyborg and Bionic Systems, Beijing, China, 17–19 October 2017.
5. Karuppiah, P.; Metalia, H.; George, K. Automation of a wheelchair mounted robotic arm using computer vision interface. In Proceedings of the IEEE International Instrumentation and Measurement Technology Conference, Houston, TX, USA, 14–17 May 2018.

6. Cigliano, P.; Lippiello, V.; Ruggiero, F.; Siciliano, B. Robotic Ball Catching with an Eye-in-Hand Single-Camera System. *IEEE Trans. Control. Syst. Technol.* **2015**, *23*, 1657–1671. [CrossRef]
7. Tongloy, T.; Boonsang, S. An image-based visual servo control system based on an eye-in-hand monocular camera for autonomous robotic grasping. In Proceedings of the International Conference on Instrumentation, Control and Automation, Bandung, Indonesia, 29–31 August 2016.
8. Carter, E.J.; Mistry, M.N.; Carr, G.P.K.; Kelly, B.A.; Hodgins, J.K. Playing catch with robots: Incorporating social gestures into physical interactions. In Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, UK, 25–29 August 2014.
9. Namiki, A.; Itoi, N. Ball catching in kendama game by estimating grasp conditions based on a high-speed vision system and tactile sensors. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 18–20 Novemer 2014.
10. Sugar, T.G.; McBeath, M.K.; Suluh, A.; Mundhra, K. Mobile robot interception using human navigational principles: Comparison of active versus passive tracking algorithms. *Auton. Robot.* **2006**, *21*, 43–54. [CrossRef]
11. Bauml, B.; Birbach, O.; Wimbock, T.; Frese, U. Catching Flying Balls with a Mobile Humanoid: System Overview and Design Considerations. In Proceedings of the IEEE International Conference on Humanoid Robots, Bled, Slovenia, 26–28 October 2011; pp. 513–520.
12. Wang, R.; Liu, M.S.; Zhou, Y.; Xun, Y.Q.; Zhang, W.B. A deep belief networks adaptive Kalman filtering algorithm. In Proceedings of the IEEE International Conference on Software Engineering and Service Science, Beijing, China, 26–28 August 2016.
13. Kao, S.T.; Wang, Y.; Ho, M.T. Ball catching with omni-directional wheeled mobile robot and active stereo vision. In Proceedings of the IEEE 26th International Symposium on Industrial Electronics, Edinburgh, UK, 19–21 June 2017.
14. Faugeras, O. *Three Dimensional Computer Vision*; MIT Press: Cambridge, MA, USA, 1993.
15. Ma, Y.; Soatto, S.; Kosecka, J.; Sastry, S. *An Invitation to 3-D Vision: From Images to Geometric Models*; Springer: New York, NY, USA, 2003.
16. Brunelli, R. *Template Matching Techniques in Computer Vision: Theory and Practice*; John Wiley & Sons: New York, NY, USA, 2009.
17. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Department of Computer Science, University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 2006.
18. Grewal, M.S.; Andrews, A.P. *Kalman Filtering: Theory and Practice with MATLAB*; Wiley: New York, NY, USA, 2014.
19. Elger, D.F.; Williams, B.C.; Crowe, C.T.; Roberson, J.A. *Engineering Fluid Mechanics*; Wiley: New York, NY, USA, 2012.
20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
21. Watanabe, K.; Shiraishi, Y.; Tzafestas, S.G.; Tang, J.; Fukuda, T. Feedback Control of an Omnidirectional Autonomous Platform for Mobile Service Robots. *J. Intell. Robot. Syst.* **1998**, *22*, 315–330. [CrossRef]
22. d'Andrea Novel, B.; Bastin, G.; Campion, G. Dynamic feedback linearization of nonholonomic wheeled mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, 12–14 May 1992; pp. 2527–2532.
23. Golnaraghi, F.; Kuo, B.C. *Automatic Control Systems*; Wiley: New York, NY, USA, 2009.
24. Ho, M.T.; Wang, H.S. PID Controller Design with Guaranteed Gain and Phase Margins. *Asian J. Control.* **2003**, *5*, 374–381. [CrossRef]