

# Projectile Flight Trajectory and Position Estimation System Based on Stereo Vision

Hsiang-Yueh Lai\* and Hao-Yuan Ke

Department of Electrical Engineering, National Chin-Yi University of Technology  
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung City 41170, Taiwan (R.O.C)

(Received May 29, 2019; accepted October 16, 2019)

**Keywords:** trajectory estimation, stereo vision, Kalman filter, unscented Kalman filter

Projectile trajectory estimation systems, such as missiles, rockets, intelligent robots, and other parabolic trajectory prediction systems, are often used in our daily life. In this paper, we propose a projectile trajectory and position estimation system based on stereo vision. When a human captures a flying projectile, he quickly looks at it attentively to estimate its flight path and then catches it. To simulate the human eyes sensing dynamic images, two parallel network cameras capture the dynamic images of projectiles. The projectile target images are found from a complex background using the image process. The target's three-dimensional positions are calculated using the sum of absolute difference (SAD) algorithm and stereo vision, and a Kalman filter (KF) or unscented Kalman filter (UKF) algorithm is used to estimate the trajectory and landing position of a projectile. Experiments were carried out using different illumination backgrounds and different weights of projectiles. Experiment results showed that the proposed projectile trajectory and position estimation system based on stereo vision can estimate a projectile's three-dimensional trajectories, which correspond closely to actual trajectories, and UKF has higher performance than KF. Our proposed estimation system can be widely applied in daily life and industry.

## 1. Introduction

When a human captures a flying projectile, such as a baseball player, first, whose eyes look attentively at the ball, his brain analyzes the projected flight direction, speed, and height, and then judges the landing position on the ground, finally he runs to capture it. When a robot simulates a human receiving a projectile, the visual system and image processing are important technologies for it.

In this paper, we use two web cameras as human eyes, and a computer that simulates a human brain processes the stereo vision as computer vision. Computer vision is also popularly applied in, for example, gesture identification,<sup>(1)</sup> image monitoring, a fingerprint identification system,<sup>(2)</sup> a lane offset system,<sup>(3)</sup> and a stereoscopic vision system.

The images caught by cameras are dynamic continuous motion images. The optical flow is applied to the detection of moving objects from dynamic images, which utilizes the

---

\*Corresponding author: e-mail: anne@ncut.edu.tw  
<https://doi.org/10.18494/SAM.2019.2485>

instantaneous changes in pixel intensity to obtain moving object information.<sup>(4)</sup> Stereo vision and the sum of absolute difference (SAD) are applied to calculate target point coordinates in three-dimensional space using target points in different visual angle images.<sup>(5)</sup> In Ref. 6, the authors imitate the baseball outfielder that catches the ball with a single camera setup on a mobile robot on the basis of motion control laws. A monocular robotic ball catching system uses the view of an eye-in-hand camera with a visual servoing control to catch a thrown ball.<sup>(7)</sup> The projectile trajectory is assumed to be a 5th-order time polynomial, and the extended Kalman filter (KF) algorithm is used to estimate the landing position of the projectile.<sup>(8)</sup>

In this paper, we propose a projectile flight trajectory and position estimation system based on stereo vision. We use two web cameras that sense the projectile dynamic images as human eyes sensing the images, and the target is detected by image processing from complex background images. The target center point coordinates in the three-dimensional space are calculated using SAD and stereo vision. Finally, KF and an unscented Kalman filter (UKF) are used to estimate the projectile flight trajectory and landing position.

## 2. System Architecture

The flowchart of our projectile flight trajectory and landing position estimation system is shown in Fig. 1. Two web cameras capture dynamic images; the target is detected by image processing involving color recognition, binarization, morphology, and the center of gravity calculation. The target center position is a two-dimensional position with  $X$ - and  $Y$ -axis coordinates, which is determined by image processing. The  $Z$ -axis coordinate of the target is calculated using the SAD and stereo vision algorithm, then the target three-dimensional coordinates are calculated. Finally, KF and UKF are used to estimate the projectile flight trajectory and landing position.

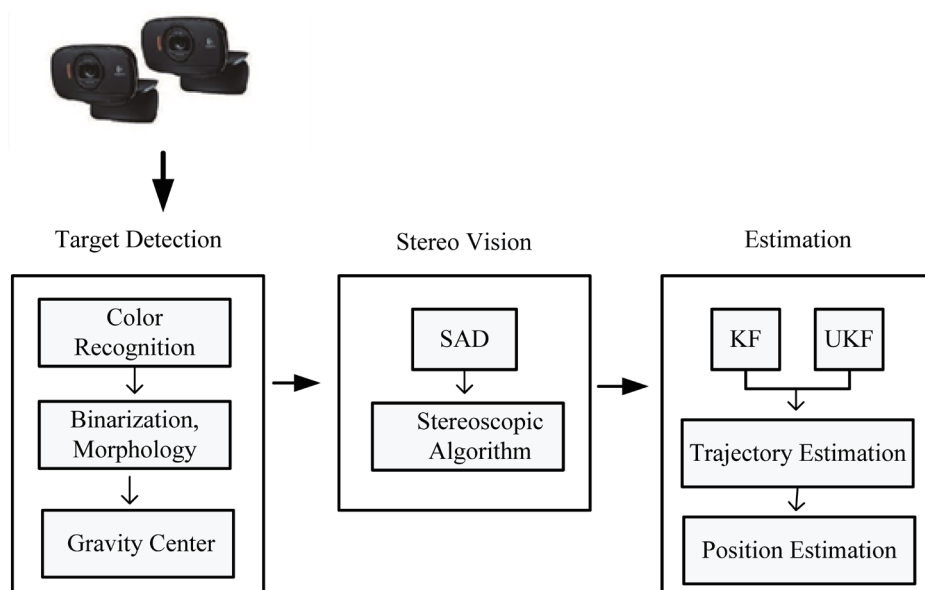


Fig. 1. Projectile flight trajectory and landing position estimation system flowchart.

### 3. Target Detection

Two web cameras capture dynamic images; the projectile object is detected from a complex environment by image processing involving: color recognition, binarization, morphology, and the center of gravity calculation.

The color image model of the camera capturing images is the red, green, and blue (RGB) color model. We convert the RGB to YUV color space using Eq. (1), where  $Y$  means the luminance, and  $U$  and  $V$  components represent chrominance.

$$\begin{aligned} Y &= (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \\ U &= (-0.1687 \times R) - (0.3313 \times G) + (0.5 \times B) \\ V &= (0.5 \times R) - (0.4187 \times G) - (0.0813 \times B) \end{aligned} \quad (1)$$

After color space conversion, the source image color space is converted to the grayscale components  $src(x, y)$ , followed by binarization using Eq. (2).  $dst(x, y)$  is the binarization result of  $src(x, y)$ . If  $dst(x, y)$  is equal to 255, we define it as logic 1, otherwise, as logic 0.

$$dst(x, y) = \begin{cases} 255, & \text{if } src(x, y) > threshold \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

After binarization, the target binarization image still has some noise and voids. The erosion and dilation of the morphology are carried out to delete the noise and voids. The erosion step eliminates noise and small protruding points of the target image, and the expansion step fills the voids in the target image. To obtain a clear image, we first adopt the erosion step and then the expansion step to remove the image noise.

After morphology, the gravity center coordinates  $(\bar{x}, \bar{y})$  of the target object are defined using Eq. (3).  $(x, y)$  are the logic 1 point coordinates of the target binarization image, and  $m$  and  $n$  are the total numbers of logic 1 points of the  $X$ - and  $Y$ -coordinates, respectively.

$$(\bar{x}, \bar{y}) = \left( \frac{1}{m} \sum x, \frac{1}{n} \sum y \right) \quad (3)$$

## 4. Stereo Vision

### 4.1 Stereoscopic algorithm

The stereoscopic vision uses different cameras to obtain different visual angle images of a three-dimensional object.<sup>(5)</sup> In Fig. 2, we use two cameras with different visual angles to catch a target  $P$ , as the characters of human eyes.  $O_l$  and  $O_r$  are the center points of the left and right camera lenses, respectively, and  $P(x_i, y_i, z_i)$  are the target coordinates. Moreover,  $P_l(x_l, y_l, z_l)$  and  $P_r(x_r, y_r, z_r)$  are the target image coordinates of the left and right cameras on the dotted line

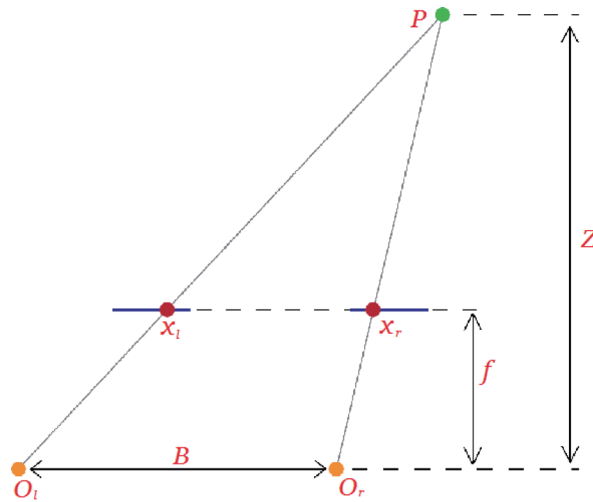


Fig. 2. (Color online) Stereoscopic geometric relationship diagram of stereo vision.

$x_l, x_r$ , respectively,  $x_l$  is the distance between  $P_l$  and  $O_l$ ,  $x_r$  is the distance between  $P_r$  and  $O_r$  in the  $X$ -axis direction,  $f$  is the camera's focal length, and  $B$  is the distance of the base line between  $O_l$  and  $O_r$ . The two cameras center points  $O_l$  and  $O_r$  have the same  $Y$ -axis coordinate.

The stereoscopic geometric relationship diagram of stereo vision is shown in Fig. 2. Equation (4) can be deduced from the relationship of the two similar triangles of  $\Delta x_l P x_r$  and  $\Delta O_l P O_r$  using the  $X$ - and  $Z$ -axis coordinates in Fig. 2, where  $Z$  is the distance between the target  $P$  and the base line  $O_l O_r$ . The distance between  $x_l$  and  $x_r$  is defined as the parallax  $d$ . Equation (4) can be rewritten as Eq. (5).

$$\frac{B - (x_l - x_r)}{Z - f} = \frac{B}{Z} \quad (4)$$

$$Z = \frac{fB}{d} \quad (5)$$

## 4.2 Regional alignment algorithm

As the characteristics of stereo vision shown in Fig. 2, a target  $P$  has a visual angle image  $P_l$  with the visual angle of  $O_l$  and a visual angle image  $P_r$  with the visual angle of  $O_r$ . Equation (6) shows the SAD in the region alignment, where  $m$  and  $n$  are the row and column pixel values of an image, respectively. We select a reference point  $(i, j)$  in the left visual angle image and use a  $5 \times 5$  block to calculate the gray value  $W_L(i, j)$ . We find the target point  $(i, j)$  in the right visual angle image and calculate the gray value  $W_R(i, j)$  using a  $5 \times 5$  block. We move the target point to find the minimum  $V_{SAD}$  point using Eq. (6). The distance between the target point and the minimum  $V_{SAD}$  point is  $d$ , and  $d$  is the parallax between the left and right visual angle images.

$$V_{SAD} = \sum_{i=0}^{i=m} \sum_{j=0}^{j=n} |W_L(i, j) - W_R(i + d, j)| \quad (6)$$

## 5. Projection Trajectory Estimation Algorithms

### 5.1 KF algorithm

KF is a highly efficient recursive filter that estimates the state equation and measurement equation of a dynamic system from incomplete and noise-containing states.<sup>(9)</sup> The major KF applications are filtering and estimation, which have been widely applied in radar, aviation, computer vision, satellite navigation, and other aspects. The KF state equation and measurement equation are Eqs. (7) and (8), respectively.

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \boldsymbol{\varepsilon}_k, \quad \boldsymbol{\varepsilon}_k \sim N(0, \mathbf{Q}_k) \quad (7)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \boldsymbol{\omega}_k, \quad \boldsymbol{\omega}_k \sim N(0, \mathbf{R}_k) \quad (8)$$

Here,  $\mathbf{x}_k$  and  $\mathbf{x}_{k-1}$  are the state matrices at times  $k$  and  $k - 1$ , respectively.  $\mathbf{A}$  is the state transition matrix,  $\mathbf{u}_{k-1}$  is the input at time  $k - 1$ ,  $\mathbf{B}$  is the input matrix,  $\boldsymbol{\varepsilon}_k$  is the state noise at time  $k$ ,  $\mathbf{y}_k$  is the observation matrix at time  $k$ ,  $\mathbf{C}$  is the output matrix,  $\boldsymbol{\omega}_k$  is the observation noise,  $\boldsymbol{\varepsilon}_k$  and  $\boldsymbol{\omega}_k$  are Gaussian white noises with the mean value 0, and  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  are covariance matrices. There are two major steps of KF estimation, namely, state estimation and measurement update.

#### 5.1.1 State estimation

The KF state estimation equation and estimation covariance matrix are shown in Eqs. (9) and (10), respectively.

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \quad (9)$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}_k \quad (10)$$

Here,  $\hat{\mathbf{x}}_{k-1}$  is the state estimation vector at time  $k - 1$ ,  $\hat{\mathbf{x}}_{k|k-1}$  is the state estimation vector at time  $k$  using the previous time  $k - 1$  state  $\hat{\mathbf{x}}_{k-1}$ ,  $\mathbf{P}_{k|k-1}$  is the estimated covariance matrix, and  $\mathbf{Q}_k$  is the process noise covariance matrix.

#### 5.1.2 Measurement update

The Kalman gain  $\mathbf{K}_k$  and the system optimal estimation state vector  $\hat{\mathbf{x}}_k$  are shown in Eqs. (11) and (12), respectively.

$$K_k = P_{k|k-1} C^T (C P_{k|k-1} C^T + R_k)^{-1} \tag{11}$$

$$\hat{x}_k = x_{k|k-1} + K_k (y_k - C x_{k|k-1}) \tag{12}$$

$$P_k = (I - K_k C) P_{k|k-1} \tag{13}$$

Here,  $R_k$  is the measured noise covariance matrix and  $y_k - C\hat{x}_{k|k-1}$  is the residual of measurement. Finally, the current time estimated covariance matrix is updated using Eq. 13. The flow of KF recursive calculation is shown in Fig. 3. The system initial state vector is  $\hat{x}_0$ , the initial covariance matrix is  $P_0$ , the state estimation uses Eqs. (9) and (10), and the measurement update uses Eqs. (11)–(13).

### 5.2 UKF algorithm

The basic KF algorithm is developed on the basis of the linear system state equation and linear measurement equation. However, the state equations or observation equations of most practical systems are nonlinear in daily life. To solve the nonlinear conditions of the estimation problem, Julier *et al.* proposed the UKF algorithm<sup>(10)</sup> to solve the nonlinear estimation problem. The discrete system of the UKF algorithm is as follows:

$$X_{k+1} = f(X_k) + w_k, \tag{14}$$

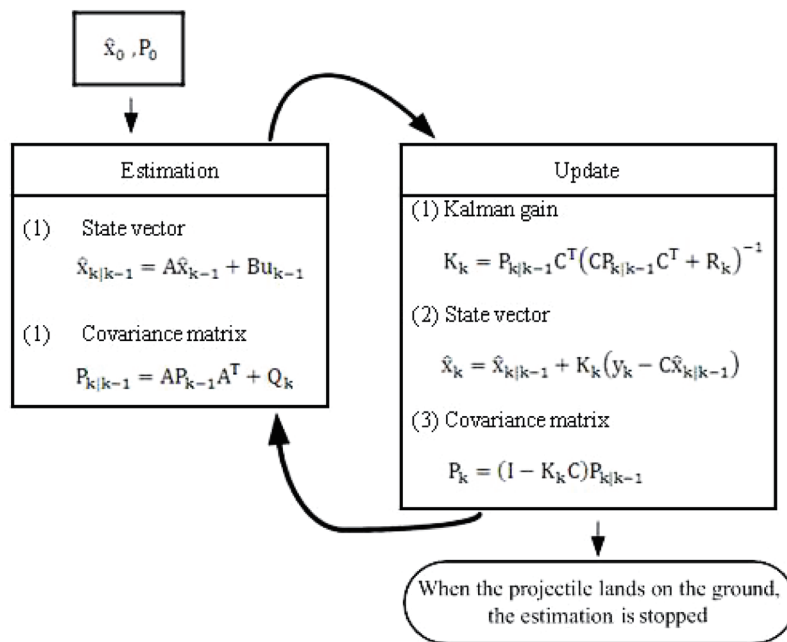


Fig. 3. Flow of KF recursive calculation.

$$\mathbf{z}_k = h(\mathbf{X}_k) + \mathbf{v}_k. \quad (15)$$

Here,  $\mathbf{X}_k$  is the state vector of the system,  $f(\cdot)$  is the nonlinear system state function, and  $\mathbf{w}_k$  is the system process noise, which is assumed as zero mean Gaussian white noise.  $\mathbf{z}_k$  is the measurement vector,  $h(\cdot)$  is a nonlinear observation function, and  $\mathbf{v}_k$  is the system measurement noise. Similarly, UKF has KF with the state estimation and measurement update steps. The nonlinear system using the UKF process is as follows:

### 1. State estimation

$$\mathbf{X}_{i(k|k-1)} = f\left[\mathbf{X}_{i(k-1)}\right], (i = 0, 1, 2, \dots, 2n), \quad (16)$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} \mathbf{W}_i^m \mathbf{X}_{i(k|k-1)}, \quad (17)$$

$$\mathbf{P}_{x(k|k-1)} = \sum_{i=0}^{2n} \mathbf{W}_i^c \left[ \mathbf{X}_{i(k|k-1)} - \hat{\mathbf{x}}_{k|k-1} \right] \left[ \mathbf{X}_{i(k|k-1)} - \hat{\mathbf{x}}_{k|k-1} \right]^T + \mathbf{Q}_k. \quad (18)$$

### 2. Measurement update

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left( \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \right), \quad (19)$$

$$\mathbf{P}_{x(k)} = \mathbf{P}_{x(k|k-1)} - \mathbf{K}_k \mathbf{P}_{z(k)} \mathbf{K}_k^T, \quad (20)$$

$$\mathbf{K}_k = \mathbf{P}_{xz(k)} \mathbf{P}_{z(k)}^{-1}, \quad (21)$$

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n} \mathbf{W}_i^m h\left[\mathbf{X}_{i(k|k-1)}\right], \quad (22)$$

$$\mathbf{P}_{z(k)} = \sum_{i=0}^{2n} \mathbf{W}_i^c \left\{ h\left[\mathbf{X}_{i(k|k-1)}\right] - \hat{\mathbf{z}}_{k|k-1} \right\} \left\{ h\left[\mathbf{X}_{i(k|k-1)}\right] - \hat{\mathbf{z}}_{k|k-1} \right\}^T + \mathbf{R}_k, \quad (23)$$

$$\mathbf{P}_{xz(k)} = \sum_{i=0}^{2n} \mathbf{W}_i^c \left\{ \mathbf{X}_{i(k|k-1)} - \hat{\mathbf{x}}_{k|k-1} \right\} \left\{ h\left[\mathbf{X}_{i(k|k-1)}\right] - \hat{\mathbf{z}}_{k|k-1} \right\}^T. \quad (24)$$

The UKF gain matrix is  $\mathbf{K}_k$ . The flow of UKF recursive calculation is similar to that of KF recursive calculation as shown in Fig. 3.

## 6. Experimental Results

In our experiment, two web cameras catch the projectile dynamic images. The image process step detects the target from a complex background and calculates the target center point of the  $X$ - and  $Y$ -axis coordinates. The  $Z$ -axis coordinate of the target center point is calculated using the SAD and stereo vision algorithm. Finally, the KF and UKF steps estimate the projectile trajectory and landing position, respectively. The state equation vector  $\mathbf{x}$  of the KF and UKF state equations indicates the target center point position coordinates  $(p_x, p_y, p_z)$  and speed  $(v_x, v_y, v_z)$ . The gravity acceleration  $g$  is the input  $u$ . The current time  $k$  state is estimated using the previous time  $k - 1$  state shown in Sects. 5.2 and 5.3.

In our experiments, the projectile dynamic images are captured using two web cameras, and we select 21 image frames of the projectile dynamic images to observe the actual projectile trajectory. The flight duration of each experiment trajectory is about 0.8 to 1 s. The weight of the green ball is 32.5 g and that of the blue ball is 157 g. Our experiments are carried out under different daylight illuminations. The actual trajectory of each ball is observed using the dynamic images from the two cameras. Under different daylight illuminations, the trajectories are estimated using KF and UKF. Figure 4(a) shows the three-dimensional coordinates and Fig. 4(b) shows the time function of the blue ball trajectory under 515 lux illumination. The height of the projectile trajectory is about 1 m. The “KF trajectory estimation” is the projectile trajectory estimation using KF. The “UKF trajectory estimation” is the projectile trajectory estimation using UKF. The “actual trajectory” is the actual projectile trajectory, which is observed using the dynamic images from the two cameras.

The projectile trajectories of UKF and KF, and the actual trajectory are very similar in our experiment results. The estimation errors of the blue and green balls under different illuminations are shown in Tables 1 and 2, respectively. The maximum error is less than 0.05 m. The average error is controlled at 0.03 m. The blue ball UKF errors are less than the KF errors under different illuminations. The green ball has the same result. The projectile trajectory

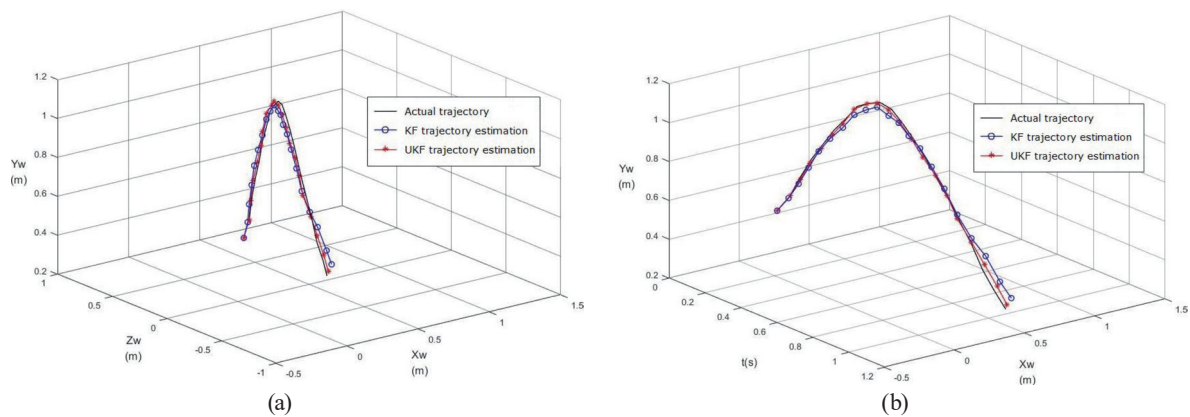


Fig. 4. (Color online) (a) Three-dimensional coordinates and (b) time function of blue ball trajectory under 515 lux illumination.



Table 1  
Blue ball trajectory error.

	Error (m)					
	515 lux			215 lux		
	Average	Maximum	Minimum	Average	Maximum	Minimum
KF (X-axis)	0.033572	0.043973	0.017853	0.035574	0.047793	0.016091
UKF (X-axis)	0.031893	0.038589	0.011714	0.034896	0.047189	0.015854
KF (Y-axis)	0.030896	0.037731	0.013458	0.032576	0.051904	0.013726
UKF (Y-axis)	0.029383	0.036542	0.012973	0.031232	0.050083	0.011293
KF (Z-axis)	0.031232	0.041289	0.011586	0.033389	0.046896	0.014282
UKF (Z-axis)	0.029892	0.039583	0.009731	0.031974	0.045138	0.012531

Table 2  
Green ball trajectory error.

	Error (m)					
	515 lux			215 lux		
	Average	Maximum	Minimum	Average	Maximum	Minimum
KF (X-axis)	0.031381	0.051393	0.008793	0.033581	0.042134	0.008796
UKF (X-axis)	0.030796	0.047986	0.004531	0.032976	0.041558	0.008326
KF (Y-axis)	0.027538	0.045045	0.010913	0.030598	0.038575	0.009732
UKF (Y-axis)	0.026986	0.041913	0.009316	0.030112	0.037853	0.008873
KF (Z-axis)	0.033198	0.047574	0.009385	0.029381	0.040972	0.010589
UKF (Z-axis)	0.031574	0.044913	0.005164	0.029002	0.039851	0.009585

Table 3  
Blue ball position error.

	Error (m)					
	515 lux			215 lux		
	Average	Maximum	Minimum	Average	Maximum	Minimum
KF (X-axis)	0.027255	0.040937	0.011573	0.027243	0.042387	0.012093
UKF (X-axis)	0.023513	0.037859	0.009167	0.024928	0.037986	0.009871
KF (Z-axis)	0.030518	0.045245	0.015579	0.030759	0.045443	0.016076
UKF (Z-axis)	0.024142	0.039513	0.008372	0.023984	0.036588	0.009788

Table 4  
Green ball position error.

	Error (m)					
	515 lux			215 lux		
	Average	Maximum	Minimum	Average	Maximum	Minimum
KF (X-axis)	0.025792	0.039897	0.009647	0.025691	0.041134	0.010089
UKF (X-axis)	0.021293	0.035921	0.004661	0.021495	0.037589	0.007392
KF (Z-axis)	0.028397	0.043192	0.013588	0.029958	0.044198	0.014481
UKF (Z-axis)	0.022386	0.038976	0.005796	0.022088	0.037983	0.006193

estimations using UKF correspond more closely to the actual trajectory than those using KF. The color of the test ball does not affect the projectile trajectory estimations.

When a projectile lands on the floor, the Y-axis coordinate of the projectile trajectory is zero. We estimate the landing position using UKF and KF, and the position estimation errors of the blue and green balls under different illuminations are shown in Tables 3 and 4, respectively. The maximum error is less 0.05 m and the average error is about 0.03 m for all samples. The results of experiments on projectile landing position estimation using KF and UKF correspond closely to the actual landing position. The projectile landing position estimations using UKF correspond more closely to the actual position than those using KF.

## 7. Conclusion

A projectile three-dimensional trajectory and position estimation system based on stereo vision is implemented in our research. After cameras sense the projectile, target images are identified from complex backgrounds, and then the three-dimensional coordinates of the projectile target are calculated using the stereoscopic visual algorithm. After obtaining the target image information, we use KF or UKF to estimate the trajectory and landing position of the projectile. It is found that UKF can estimate the projectile and landing position more accurately than KF. Our system can intelligently track the projectile trajectory and landing position, and can be widely applied in daily life and industrial intelligent robots. The robot senses a target, actively tracks the target trajectory, estimates the landing position, and judges where to capture the target or beat the target. Our system can be widely applied in intelligent robot industries, such as intelligent trash cans, badminton robots, and rehabilitation robots.

## References

- 1 H. Y. Lai, H. Y. Ke, and Y. C. Hsu: *Sens. Mater.* **30** (2018) 4. <https://myukk.org/SM2017/article.php?ss=1790>
- 2 B. Saropourian: *Proc. 2009 2nd IEEE Int. Conf. Computer Science and Information Technology (IEEE, 2009)* 158. <https://ieeexplore.ieee.org/document/5234593>
- 3 N. Hautière, R. Labayrade, and D. Aubert: *IEEE Trans. Intell. Transp. Syst.* **7** (2006) 2. <https://ieeexplore.ieee.org/abstract/document/1637675>
- 4 J. L. Barron, D. J. Fleet, and S. S. Beachemin: *Int. J. Comput. Vis.* **12** (1994) 1. <https://link.springer.com/article/10.1007/BF01420984>
- 5 H. Y. Lai and M. H. Sung: *ICIC Express Letters Part B: Applications* **7** (2016) 12. [http://www.ijicic.org/elb-7\(12\).htm](http://www.ijicic.org/elb-7(12).htm)
- 6 J. A. Borgstadt and N. J. Ferrier: *Proc. IEEE Robotics and Automation (IEEE, 2000)* 3189. <https://ieeexplore.ieee.org/document/845154>
- 7 V. Lippiello and F. Ruggiero: *Proc. 10th IFAC Symp. Robot Control Federation of Automatic Control (IFAC, 2012)* 229. <https://www.sciencedirect.com/science/article/pii/S1474667016336151>
- 8 M. Bratt, C. Smith, and H. I. Christensen: *2007 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IEEE, 2007)* 2710. <https://ieeexplore.ieee.org/abstract/document/4398989>
- 9 R. E. Kalman: *J. Basic Eng.* **82** (1960) 1628. <http://fluidsengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1430402>
- 10 S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte: *Proc. 1995 American Control Conf. (IEEE, 1995)*. <https://ieeexplore.ieee.org/document/529783>

## About the Authors



**Hsiang-Yueh Lai** received her Ph.D. degree from the Department of Engineering and System Science, National Tsing Hua University, Hsinchu, Taiwan, in 2010. She is currently an associate professor in the Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan. Her current research interests include intelligent systems, embedded systems, and solid-state electronic devices.



**Hao-Yuan Ke** received his M.E. degree from the Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, in 2017. He is currently an engineer in the Department of Automation, HIWIN Company, Taichung, Taiwan. His current research interests include factory automation, industrial robots, and vision applications.