International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 13s,2025
https://theaspd.com/index.php`

Development Of Kalman Filter For Pedestrian Trajectory Prediction

Meng-Yun Chung[1], Chaur-Yang Chang[2], Hsin-Ru Wu[3]
Department of Electrical Engineering, National Taipei University of Technology, Taipei Taiwan

Abstract: *Computer vision is one of the main fields of artificial intelligence. Computer vision enables machines to extract useful information from images or videos. Applications include image recognition, autonomous driving, and pedestrian detection. In the context of traffic environments, the interaction between pedestrians and vehicles is a key focus of many studies. Since pedestrians often face more severe injuries in traffic accidents, therefore, accurately predicting pedestrian movement trajectories is crucial for improving traffic safety. To address this issue, combining object detection and tracking techniques to optimize pedestrian trajectory prediction has become a core direction of research. These technologies enable drivers or ADAS (Advanced Driver Assistance Systems) to have a more comprehensive understanding of the driving environment. This effectively reduces the risk of traffic accidents and enhances overall road safety. This study focuses on predicting the trajectories of pedestrians on the road. It calculates the pedestrian's relative speed and direction. It utilizes the Kalman filter algorithm to integrate object detection models, tracking models, and trajectory prediction models, by predicting the system state and correcting it based on new observed data. This continuously improves the accuracy of state estimation by using prediction and update steps, generating an optimal estimate through weighted averaging, in order to infer the pedestrian's potential future path. The research results show that the error between the x-axis data of the Kalman filter trajectory prediction and the real trajectory's x-axis data ranges under 0.1 %, while the error in the y-axis data ranges under 0.1 % in absolute value.*
Keywords-*Pedestrian trajectory prediction, object detection and tracking, Kalman filter, artificial intelligence.*

INTRODUCTION

Artificial intelligence is a field of technology and science that enables machines to simulate and perform human intelligence. It combines knowledge from various disciplines, including computer science, data analysis, image recognition, mathematics, and psychology, with the goal of developing systems capable of performing tasks similar to human intelligence [1]. In AI image recognition technology, AI object detection techniques can identify the location and size of specified objects within an image, assisting in monitoring whether an object appears, its location, size, movement direction, and speed. This can be applied to real-time image recognition needs across various fields. Among them, pedestrian detection has become one of the most important research directions in the field of computer vision and a key topic in deep learning. The core technology relies on intelligent video surveillance, traffic statistics in scenic areas, and other fields. Its accuracy is of great significance for the development of traffic safety. Although current detection algorithms have made significant progress in terms of accuracy and speed, pedestrian trajectory detection technology still faces many challenges. For example, when multiple pedestrians appear in the same scene, two or more pedestrians may obstruct each other, making accurate detection challenging. These uncertainties make the research on pedestrian detection technology exceptionally challenging [2]. To fully address the uncertainty in trajectory information discrimination within the discriminator and improve prediction accuracy, this paper proposes a trajectory prediction method based on the Kalman filter. The Kalman filter is an efficient recursive filter (self-regressive filter) that can estimate the state of a dynamic system from a series of incomplete and noisy measurements [3]. The Kalman filter generates estimates of unknown variables by considering the joint distribution of measurements at different times, based on their values at each time [4]. As a result, it provides more accurate estimates than methods that rely solely on a single measurement [5]. The Kalman filter is a commonly used filter in target state estimation algorithms. By establishing a target state model and estimating the target's velocity and acceleration, it can predict the future position of the target's centroid. This helps to narrow down the search area and overcome the tracking loss problem caused by partial occlusion of the target.

MATERIAL AND METHODS

The Kalman filter is a set of mathematical equations that provides an efficient computation of the least squares method [6]. The Kalman filter performs linear least-squares estimation on the state sequence of a dynamic system, using measurement values to correct the estimated states and provide reliable state estimates. It describes a dynamic system through state equations and observation equations. The state equation is shown in Equation 1.The observation equation is shown in Equation 2.

$x(k+1)=A(k+1,k)x(k)+w(k)$ (Eq. 1)

$z(k)=H(k)x(k)+v(k)$ (Eq. 2)

In the state equation:x(k) is the state vector.z(k) is the observation vector. A(k+1,k) is the state transition matrix. H(k) is the observation matrix. w(k) is the system noise vector.v(k) is the observation noise vector. It is typically assumed to be a mutually uncorrelated zero-mean Gaussian white noise vector.

Using Kalman filter theory, the following prediction equations are shown in Equation 3, and the update equations are shown in Equations 4, 5, and 6. Equation 3 consists of two parts: the state prediction Equation 3a and the error covariance matrix prediction Equation 3b. State prediction equation: Equation 3a is the prediction at time step k based on all available observation data Z(1), Z(2), ... , Z(k) the optimal state estimate obtained. A(k+1,k)is the state transition matrix [7], which describes the system dynamics from time step k to time step k+1. $x'(k+1|k)$ is the predicted state estimate at time step k+1.This part represents using the optimal state estimate $x'(k+1|k)$ from the previous time step and the system's transition matrix A(k+1,k)to predict the state at the next time step.Error covariance matrix prediction equation:In Equation 3b, $P(k+1|k)$is the error covariance matrix at time step k, which describes the certainty or uncertainty of the state estimate. A(k+1,k)is the state transition matrix, which projects the error covariance matrix from time step k to time step k+1. $P(k+1|k)$ is the predicted error covariance matrix at time step k+1. Q(k) is the process noise covariance matrix, which represents the random disturbances or uncertainties that may arise in the model process. This part shows how to predict the error covariance matrix at the next time step based on the error covariance matrix from the previous time step $P(k+1|k)$ and the system's dynamic transition matrix A(k+1,k). At the same time, the process noise Q(k) is also included to account for the uncertainty in the model process.

The state prediction in Equation 3 uses the optimal state estimate from the previous time step $x'(k+1|k)$ and the state transition matrix A(k+1,k) to predict the state at the next time step $x'(k+1|k)$. The error covariance matrix prediction is based on the error covariance matrix from the previous time step $P(k+1|k)$ and the state transition matrix A(k+1,k) to predict the error covariance matrix at the next time step $P(k+1|k)$, while considering the process noise matrix Q(k).

Equation 4 calculates the Kalman gain matrix K(k+1) [8]. Where P(k+1|k) is the predicted error covariance matrix.H(k+1) is the observation matrix. It describes how to map from the system state to the observation values.R(K+1) is the observation noise covariance matrix. $HT(k+1)$is the transpose of the observation matrix. Equation 5 updates the state estimate $x'(k+1|k+1)$, This is done based on the predicted state $x'(k+1|k)$ and the actual observation value Z(k+1).The differenceZ(k+1)-H(k+1) $x'(k+1|k)$ is the prediction error, representing the gap between the predicted value and the actual observation.

The Kalman gain K(k+1) is used to incorporate this error into the new estimate. Equation 6 updates the error covariance matrix P(k+1|k+1), which represents the uncertainty of the new estimate. I is the identity matrix, K(k+1) is the Kalman gain, H(k+1) is observation matrix. The updated covariance matrix indicates that, after taking the observation data into account, the uncertainty in the system's estimation is reduced. The conclusion is that Equation 4 calculates the Kalman gain, which helps balance the influence of predictions and actual observations. Equation 5 is used to update the state estimate by combining the prediction error with the Kalman gain. Equation 6 is used to update the error covariance matrix, representing the accuracy of the state estimate after the update. By utilizing the prediction equations and update equations of the Kalman filter, the recursive method can continuously predict the position of a moving target in the next frame [9].

$$x'\{k+1|k\} = A(k+1,k)x'(k|k) \quad \text{(Eq. 3a)}$$

$$P\{k+1|k\} = A(k+1,k)P(k|k)A^T(k+1,k)Q(k) \quad \text{(Eq. 3b)}$$

$$K(k+1) = P(k+1|k)H^T(k+1)[H^T(k+1)P(k+1|k)H^T(k+1) + R(K+1)]^{-1} \quad \text{(Eq. 4)}$$

$$x'(k+1|k+1) = x'(k+1|k)x'(k+1|k)K(k+1)[Z(k+1) - H(k+1)x'(k+1|k) \quad \text{(Eq. 5)}$$

$$P(k+1|k+1) = [I - K(k+1)H(k+1)]P(K+1|k) \quad \text{(Eq. 6)}$$

The motion target tracking method based on Kalman filtering.

Kalman filtering is a powerful recursive estimation technique widely used in motion target tracking. This method combines dynamic system models and measurement data to estimate the state of the system. For example, by utilizing the target's position information, accurate tracking of the moving target can be achieved. Through continuous iteration of the prediction and update cycle, the dynamic model's predictions are integrated with measurement data, enabling accurate tracking of the moving target amidst noise to obtain optimal estimates [10]. Figure 1 illustrates the algorithmic model flowchart of the motion target tracking using the Kalman filter, while Figure 2 shows the detection results of motion target tracking through the Kalman filter.Calculate the feature information of the moving target. To track the moving target, an enclosing rectangle is first used to mark the target, followed by calculating the centroid of the moving target as well as the width and height of the enclosing rectangle.Initialize the Kalman filter using the obtained feature information. Since the speed of the target and the rate of change of the enclosing rectangle are unknown during initialization, these quantities are initialized to 0.Use the Kalman filter to predict the corresponding target region in the next frame. When the next frame arrives, perform target matching within the predicted region.If a match is found, update the Kalman filter and record the target information in the current frame.

The algorithmic model flowchart for motion target tracking based on the Kalman filter is shown in Figure 1. The main steps will be discussed in detail below.

Motion target detection results.

Motion target feature extraction.

Establish inter-frame relationship matrix.

Detect the appearance of new targets, target matching, target occlusion, target separation, and target disappearance.

Handle various situations corresponding to step (4).

Confirm whether the target is matched

If not matched, record the Kalman prediction results from the previous frame and re-extract the motion target features.

If matched, update the target feature information.

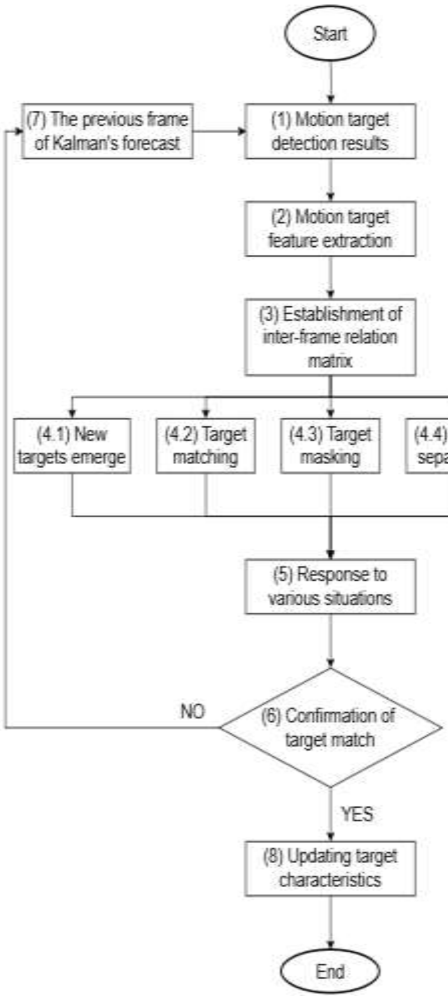


Fig. 1. The algorithmic model flowchart for motion target tracking using the Kalman filter.

Fig. 2. Detection results of motion target tracking using the Kalman filter.

Kalman filter performs prediction and estimation

State estimation is a crucial component of the Kalman filter. Generally, the estimation problem involves quantitatively inferring random variables based on observation data. Specifically, state estimation of dynamic behavior enables real-time operational status estimation and prediction functions [11]. Based on the above research on Kalman filter prediction and estimation, an experiment targeting pedestrian prediction is conducted. Assuming a pedestrian moves with an acceleration of 20 m/s^2 along the x-axis and 10 m/s^2 along the y-axis, observations are made for 10 seconds and 20 seconds respectively, and predictions and estimations are performed using the Kalman filter.

The following conditions can be assumed:

The pedestrian's acceleration along the x-axis is $a_x = 20\mathrm{m}/s^2$.

The pedestrian's acceleration along the y-axis is $a_y = 10\mathrm{m}/s^2$.

Simulation time: 20 seconds.
Assume the initial position and velocity are 0, and the pedestrian maintains constant acceleration during this period.
Experimental Steps:
Motion simulation: Based on the initialization and acceleration, the position and velocity at each second can be calculated using motion equations. It is assumed that at t=0, the pedestrian's initial position is $(x_0, y_0) = (0,0)$,the velocity is $(v_{xo}, v_{y0}) = (0,0)$, and the acceleration is constant. The motion equations are as shown in Equation 7 and Equation 8. In these motion equations, t represents time, and a_x and a_y are the accelerations along the x-axis and y-axis, respectively.

$$x(t) = x_0 + v_{x0} \cdot t + \frac{1}{2} a_x \cdot t^2 \quad \text{(Eq. 7)}$$

$$y(t) = y_0 + v_{y0} \cdot t + \frac{1}{2} a_y \cdot t^2 \quad \text{(Eq. 8)}$$

Adding Noise: Actual observations are affected by measurement noise, so random noise needs to be added to the simulated data at each time point to model this type of error. Typically, the noise is assumed to be Gaussian noise.
Using Kalman filtering for prediction and correction, a Kalman Filter is established to filter out the noise and estimate the pedestrian's true position and velocity.
The Kalman filter prediction and estimation are based on the following two steps:
Prediction Step: Predict the state at the next time point based on the dynamic model.
Update Step: Adjust the prediction based on the observed values.
The Trajectory Prediction Process of the Kalman Filter
The Kalman filter is a recursive method for estimating the next state of a system, commonly used in dynamic systems with time-dependent parameters, such as motion equations. The trajectory prediction within this filter is analogous to the motion equations used for estimation and prediction in frame sequences [12]. The experimental process outlined below covers the core steps of the Kalman filter, including state initialization, prediction steps, measurement update steps, and dynamic data fusion during the iterative process. The flowchart of the pedestrian prediction algorithm model based on the Kalman filter is shown in Figure 3. The main steps will be discussed in detail below.
Initialization:
Set the initial state vector x_0(position and velocity).
Set the initial covariance matrix P_0,representing the uncertainty of the initial estimate.
Set the state transition matrix *F*, measurement matrix *H*, and measurement noise matrix R.
Prediction step:
Predict the current state based on the state transition matrix *F* as shown in Equation 9. Update the covariance matrix (assuming the presence of a process noise matrix) as shown in Equation 10.

$$x = F \cdot x_{prev} \quad \text{(Eq. 9)}$$

$$P = F \cdot P_{Prev} \cdot F^T + Q \quad \text{(Eq. 10)}$$

Update Step:
The calculation of innovation (measurement error) is as shown in Equation 11, the calculation of the innovation covariance is as shown in Equation 12, the calculation of the Kalman gain is as shown in Equation 13, the update of the estimated value is as shown in Equation 14, and the update of the covariance is as shown in Equation 15.

$$y = z - H \cdot x \quad \text{(Eq. 11)}$$

$S = H \cdot P \cdot H^T + R$ (Eq. 12)

$K = P \cdot H^T \cdot S^{-1}$ (Eq. 13)

$x = x + K \cdot y$ (Eq. 14)

$P = (I - k \cdot H) \cdot P$ (Eq. 15)

Repeat Steps: Repeat the prediction and update steps until all measurement data is processed.

Update the predicted trajectory data after processing all measurement data.

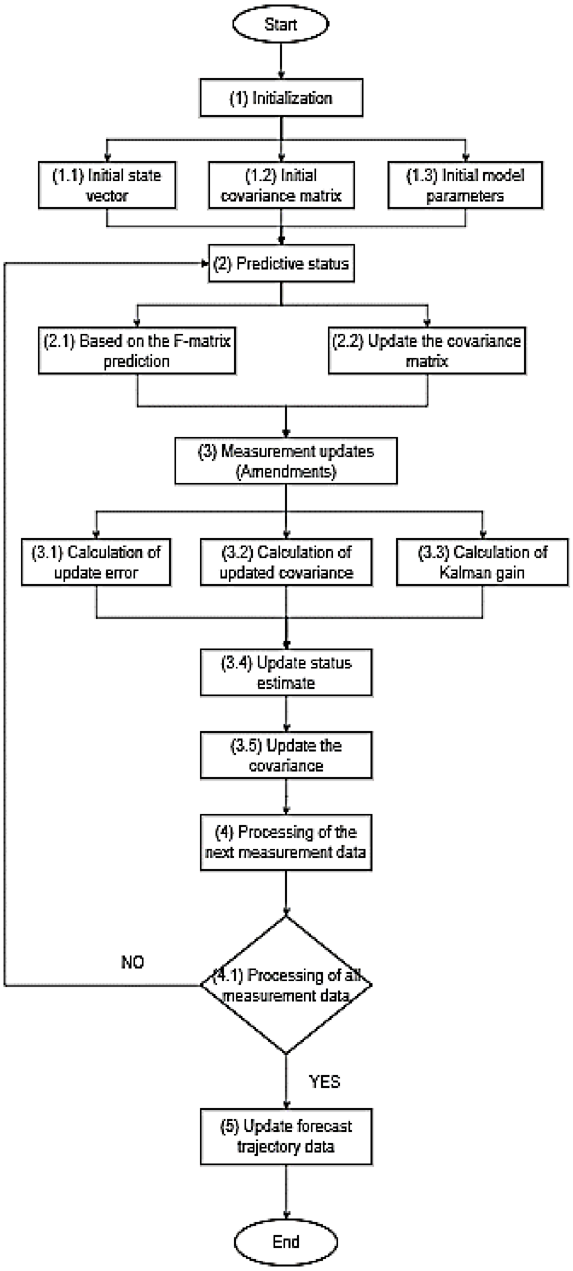


Fig. 3. Flowchart of the pedestrian prediction algorithm model using the Kalman filter.

Kalman filter prediction implementation

This study combines the linear dynamic system model with the Gaussian noise assumption to predict the target state using a Kalman filter [13-14]. The main objectives are:

To track the target's position and velocity in a dynamic system.

To integrate model predictions with measurement data for a smoothed estimation of the target trajectory.

To provide accurate predictions of the target's future state, applicable to trajectory tracking or control decision-making.

This experiment will cover the core steps of the Kalman filter, including state initialization, prediction steps, measurement update steps, and dynamic data fusion during the iterative process. Additionally, it will simulate a typical motion target tracking problem to demonstrate the predictive capability of the Kalman filter in a noisy environment. This experiment initializes the observed values, assuming they are the coordinates (x, y) and the horizontal and vertical velocities (v_x, v_y) of the object $State(x)$, along with the time interval (dt)。

Simulating the real motion trajectory: The pedestrian's position and velocity are calculated using the quadratic motion equation [15], and noise is added at each time step to simulate actual measurement data.

The following is the Kalman process noise covariance matrix Q, as shown in Equation 16.

$Q = np.array([\ [1, 0, 0, 0],$
$[0, 1, 0, 0],$
$[0, 0, 1, 0],$
$[0, 0, 0, 1]\ ])$ (Eq. 16)

The following is the Kalman measurement noise covariance matrix R, as shown in Equation 17.

$R = np.array([\ [1, 0],$
$[0, 1]\ ])$ (Eq. 17)

The execution steps of the Kalman filter include state prediction and measurement update (correction). The Kalman filter is established to filter out noise and estimate the pedestrian's true position. This process involves making predictions based on the motion model and updating the estimation using measurement data.

Kalman prediction steps

Kalman predicted state.

Kalman predicted covariance.

Assume the observed position includes measurement noise.

Create Kalman Gain.

Kalman Update Steps.

6.1) Kalman Updated State.

6.2) Kalman Updated Covariance.

RESULTS AND DISCUSSION

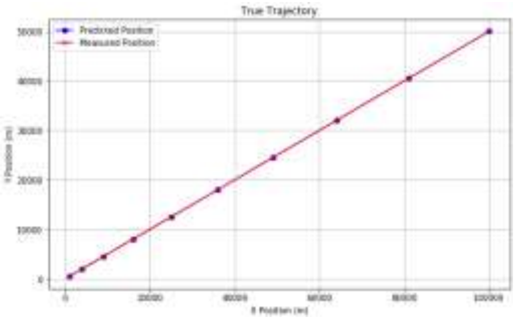
The results of this experiment involve a trajectory prediction experiment for pedestrian movement based on the Kalman filter. When the pedestrian's acceleration is 20 m/s^2,along the x-axis and 10 m/s^2along the y-axis, observations were conducted for 10 seconds and 20 seconds, respectively. Trajectory predictions were performed using the Kalman filter, and the experimental results were compared with the true trajectory of a pedestrian moving with constant acceleration.Figure 4(a) shows the trajectory prediction results using the Kalman filter. The red line represents the observed data with noise over 10 seconds. The blue dotted line represents the trajectory estimated by the Kalman filter within 10 seconds, which closely approximates the true trajectory.

Figure 4(b) shows the trajectory prediction results using the Kalman filter. The red line represents the observed data with noise over 20 seconds. The blue dotted line represents the trajectory estimated by the Kalman filter within 20 seconds.

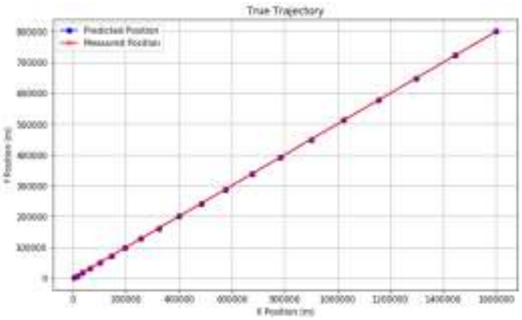
Figure 4(c) shows the light blue line representing the simulated true trajectory of a pedestrian moving with constant acceleration.

Table 1 presents a comparison between the x-axis and y-axis data of the trajectory prediction results within 10 seconds using the Kalman filter and the x-axis and y-axis data of the true trajectory. The unit is in meter.

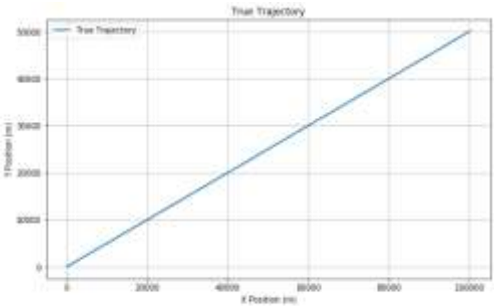
Table 2 shows the error values of the x-axis and y-axis data between the trajectory prediction results using the Kalman filter and the true trajectory.



(a)



(b)



(c)

Fig. 4. Trajectory prediction results using the Kalman filter: (a) within 10 seconds, (b) within 20 seconds, (c) true trajectory.

Table 1. Comparison table of the x- and y-axis.

Prediction position (x, y)	True Trajectory (x, y)
(1000.31, 499.62)	(1000.00, 500.00)
(4000.98, 1999.25)	(4000.00, 2000.00)
(9003.53, 4497.91)	(9000.00, 4500.00)
(16004.40, 7997.67)	(16000.00, 8000.00)
(25005.20, 12497.60)	(25000.00, 12500.00)
(36005.40, 17997.10)	(36000.00, 18000.00)
(49005.32, 24494.70)	(49000.00, 24500.00)
(64005.40, 31993.61)	(64000.00, 32000.00)
(81005.60, 40493.10)	(81000.00, 40500.00)
(100004.00, 49991.89)	(100000.00, 50000.00)

Table 2. Error values of the x- and y-axis.

x-axis	y-axis
0.03%	-0.08%
0.02%	-0.04%
0.04%	-0.05%
0.03%	-0.03%
0.02%	-0.02%
0.01%	-0.02%
0.01%	-0.02%
0.01%	-0.02%
0.01%	-0.02%
0.00%	-0.02%

CONCLUSIONS

This study utilizes the core concept of the Kalman filter to continuously improve the accuracy of state estimation through system state prediction and corrections based on new observational data. The prediction step relies on the dynamic model of the system, while the update step combines new observational data with the predictions to produce an optimal estimate through a weighted average. The Kalman filter assumes that the system dynamics and observation models are linear, and all noise is Gaussian white noise, which forms its theoretical foundation. The strength of the Kalman filter lies in its ability to provide relatively accurate state estimations in real-time within dynamic environments by continuously updating and correcting, as well as effectively handling noise from various sources. In this study, a practical experiment was conducted using the Kalman filter to predict pedestrian trajectories. The accuracy of the predicted trajectories was observed over the period from 10 to 20 seconds, comparing noisy observations with the Kalman filter's predictions. The conclusions are as follows:

The true trajectory data of a pedestrian was calculated by simulating a scenario where the pedestrian moves with constant acceleration using the equations of motion.Based on the noisy observation data over 10 seconds and 20 seconds, the Kalman filter continuously corrected its predictions. Observations revealed that the accuracy of the pedestrian trajectory state estimation was effectively improved.The study results involved calculating the error values between the Kalman filter's predicted trajectory data (x-axis and y-axis) and the true trajectory data (x-axis and y-axis). Statistical analysis revealed that the error values for the x-axis data ranged under 0.1 %, while the error values for the y-axis data ranged under 0.1 % in absolute value.

Competing interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding

The Ministry of Science and Technology (NSTC 113-2622-E-027-024 -).

REFERENCE

1. Schank, R. C. (1991). Where's the AI?. *AI magazine*, *12*(4), 38-38.
2. Liu, Lisang, et al. "Research on Pedestrian Detection Algorithm Based on MobileNet-YoLo."*Computational intelligence and neuroscience* 2022.1 (2022): 8924027.
3. Kim, Y., & Bang, H. (2018). Introduction to Kalman filter and its applications. In *Introduction and implementations of the Kalman filter*. IntechOpen.

4. Maybeck, P. S. (1990). The Kalman filter: An introduction to concepts. In *Autonomous robot vehicles* (pp. 194-204). New York, NY: Springer New York.
5. Galleani, L., & Tavella, P. (2010). Time and the Kalman filter. *IEEE control systems magazine*, *30*(2), 44-65.
6. Welch, Greg, and Gary Bishop. "An introduction to the Kalman filter." (1995): 2.
7. Rui, X., Wang, G., Lu, Y., & Yun, L. (2008). Transfer matrix method for linear multibody system. *Multibody System Dynamics*, *19*, 179-207.
8. Juang, J. N., Chen, C. W., & Phan, M. (1993). Estimation of Kalman filter gain from output residuals. *Journal of Guidance, Control, and Dynamics*, *16*(5), 903-908.
9. Shanks, J. L. (1967). Recursion filters for digital processing. *Geophysics*, *32*(1), 33-51.
10. Schechter, S. (1962). Iteration methods for nonlinear problems. *Transactions of the American Mathematical Society*, *104*(1), 179-189.
11. Zhang, Jinghe, et al. "A two-stage Kalman filter approach for robust and real-time power system state estimation." *IEEE Transactions on Sustainable Energy* 5.2 (2013): 629-636.
12. Farahi, Fahime, and Hadi Sadoghi Yazdi. "Probabilistic Kalman filter for moving object tracking." *Signal Processing: Image Communication* 82 (2020): 115751.
13. Davison, E. (1966). A method for simplifying linear dynamic systems. *IEEE Transactions on automatic control*, *11*(1), 93-101.
14. Russo, F. (2003). A method for estimation and filtering of Gaussian noise in images. *IEEE Transactions on Instrumentation and Measurement*, *52*(4), 1148-1154.
15. Majhi, A., & Radhakrishnan, R. (2022). Problem of identity and quadratic equation.