Finite Impulse Response (FIR) Filters and Kalman Filter for Object Tracking Process

E. G. Pale-Ramon[1(✉)], Y. S. Shmaliy[1], L. J. Morales-Mendoza[2], and M. González Lee[2]

[1] Electronics Engineering Department, Universidad de Guanajuato, Salamanca, Guanajuato, Mexico
eg.paleramon@ugto.mx

[2] Electronics Engineering Department, Universidad Veracruzana, Poza Rica, Veracruz, Mexico

Abstract. Object tracking is a subject of great interest in different fields of research. In the process of estimating the trajectory of a target in a displacement sequence, the target is not always located exactly, the process tracking can be accompanied by variations in the position and size of the image. The differences observed can be considered as colored measurement noise (CMN). We treat the.se variations as Gauss-Markov color measurement noise. The standard Kalman, Optimal FIR, Optimal Unbiased FIR, and Unbiased FIR filters are tested in simulated displacement sequences to demonstrate the best performance. The object trajectory estimation process was carried out in two stages: "predict" and "update". The results showed good performance of the FIR and Kalman filters in the object tracking process under conditions of the process and data noise. While at higher data and process noise values, the FIR filters showed better performance.

Keywords: Object tracking · State estimation · Fir filters · Kalman filter

1 Introduction

Object tracking is a widely researched field within computer vision, mainly due to the many practical applications such as video surveillance and security, robotics, human-computer interaction, autonomous vehicle navigation, etc. [1–3]. Object tracking is a topic of study of great interest in the area of computer vision. In the same way, it is a study area of great interest in signal and image processing [3–5], where the data for estimating the trajectory of the object are generally obtained from the coordinates of a sequence of frames.

Object tracking can be understood as the problematic of estimating the trajectory of a target in a sequence of movement around a scene [5, 6]. Object tracking task generally presents problems for exact tracking of the object, since there are variations in the position of the frames in the scene, that is, the target is not followed exactly in the tracking process. These differences between the estimated position and the true position can be considered as color measurement noise (CMN) that is not white [7]. In addition to what has already been described, there are a variety of factors that affect the performance of tracking

Y. S. Shmaliy and A. Abdelnaby Zekry (Eds.): CCIE 2021, LNEE 920, pp. 665–684, 2022.
https://doi.org/10.1007/978-981-19-3927-3_66

algorithms such as occlusion, illumination, rapid movement, position changes, target scale variations, background clutter, etc. An example of variations in object tracking is shown in Fig. 1 for the "blurcar2" benchmark [8]. Where the object to be tracked is a white car, which is enclosed in a green bounding box, while bounding boxes in red represent erroneous estimates.

Various investigations have shown that the use of a motion model and state estimators is effective in avoiding large tracking errors [7–14]. With the correct specification of the model in the state-space, the dynamics of the object can be represented with great precision for different movements. However, the accuracy of the tracking algorithm will continue to depend heavily on residues representing noise measurement data and mismodeling errors.

Therefore, in this paper, the algorithms Kalman (KF), Optimal Finite Impulse Response (OFIR), Optimal Unbiased Finite Impulse Response (OUFIR), and Unbiased Finite Impulse Response (UFIR) filters are used to stabilize the bounding box trajectory during object tracking process, in which the differences in the scene can be considered as CMN. The state estimation strategy for both algorithms was developed in two different phases in the recursions and iterations: "predict" and "update".

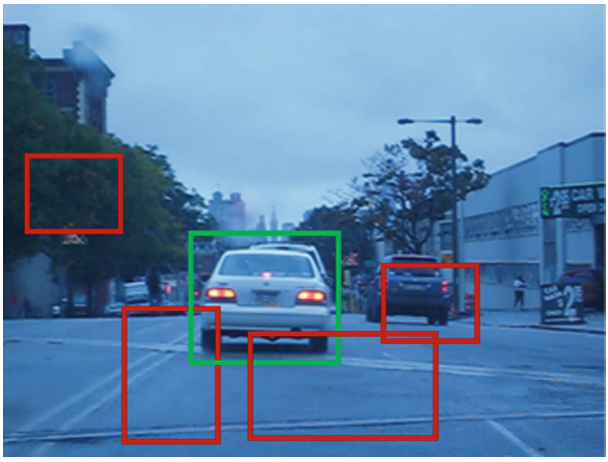

Fig. 1. Example car tracking in a video sequence when a desirable frame is green.

The FIR and KF algorithms were tested on simulated sequences. We evaluated the performance of the algorithms. Based on our experimental tests. We show that the Kalman and FIR algorithms showed favorable results with low data and process noise values, but increasing these values OUFIR and UFIR produced lower errors than KF and OFIR.

2 Object Representation: Bounding Box

An image can be described through its properties. To do this, it is necessary to calculate the mathematical properties of an image and use them as a basis subsequent classifications [15]. Therefore, the extraction of shape parameters is necessary for the image representations. One of the shape parameters most used in object tracking is the bounding box.

The bounding box can be defined as a rectangular box that encloses all the objects in an image or scene. It can be represented by the coordinates of the upper left and lower right corners of the box [16]. Using the bounding box as a shape parameter in target tracking, the information about the position of the objects in pixels is contained in an array of bounding boxes. The bounding box matrix contains the position of the minimum and maximum vertices of the box that encloses the detected object within the scene. The distribution of pixels within a frame begins in the upper left corner and ends in the lower right corner [17]. The bounding box matrix is distributed over n rows and 4 columns, the rows represent the number of recognized objects, and the columns contain the measurements for each bounding box located as follows:

$$Bounding\ Box = (X_c, Y_c, X_w, Y_h) \tag{1}$$

Where $X_{c,}$ Y_c, X_w, and Y_h are the coordinates of 4 corners of the bounding box: corners, weight, and height. In addition to these measurements, the centroid of the Bounding Box can be generated, in "x" and "y" coordinates, called C_x and C_y.

The algorithm to generate the bounding box in the tracking process must predict the four coordinates, X corner, Y corner, width, and height, for each bounding box. The center coordinates of the bounding box are based on the location of the filtering application in the tracking. Therefore, the coordinates of the centroid are used for the bounding box prediction. The coordinates of the centroid for the target are tracked through the prediction of movement that identifies each of the objects of interest with the same label in the sequence of images throughout the trajectory [18].

In the tracking process there may be misalignments and detection errors, an effective method to reduce them is to apply a filtering method. A filtering method is used to predict the centroid of each bounding box in the frame sequence and update the bounding box. The objective of using prediction and correction methods is to mitigate the noise present in the object tracking process. The prediction indicates the a posteriori position of the bounding box based on its previous position. The update is a correction step that includes the new measurement of the tracking model and helps improve filtering [19, 20].

3 Performance Evaluation of Tracking Algorithm

The tracking algorithms performance can be evaluated using metrics called precision and F-score. Precision is the percentage of correct objects detections, and F-score is one metric option to measure the accuracy.

3.1 Precision

To calculate the precision values, it is necessary to calculate the Intersection over Union (IoU). The equations for calculating precision and IoU are (2) and (3), respectively. The variables used in the calculation of the precision are obtained from the comparison of the IoU result with an established threshold [20].

$$IoU = \frac{IA}{(TBB - PBB) - IA} \tag{2}$$

where IA is the intersection area of the real/true bounding box (TBB) and the estimated/predicted bounding box (PBB). The IoU value is calculated in each position of the bounding boxes.

$$Precision = \frac{\sum TP}{(\sum TP + \sum FP)} \tag{3}$$

where TP is true-positive, and FP is false-positive.

To determine the validity of object detection, that is, the correct detection of the target, a threshold value of IoU must be established. IoU is generally set to 0.5. Assuming the IoU threshold is 0.5, if the value is greater or equal to 0.5, the detection is classified as True Positive (TP). If IoU value is less than 0.5 it is considered as a wrong detection and classified as False Positive (FP). The IoU threshold can be set to a value of 0.5 or more, such as 0.75. 0.9, 0.95, or 1. Understanding that 1 would be an exact overlap of the bounding boxes.

3.2 F-Score

One metric for evaluating object tracking model using precision and recall is F-score. The F-score can provide important information about the model performing at various threshold values. Recall can be calculated as the number of correct detected objects divided by the total number of detections in the ground truth. In this sense, a very useful parameter is the F-score, which is the harmonic mean of the recall and precision. The F-score is a measure of accuracy test [21, 22]. This metric is based on the bounding box overlap obtained between the algorithm and the real trajectory to calculate the accuracy with which the algorithm operates on an object displacement sequence, the F-score is computed by Eq. (4).

$$F - score = 2\frac{Precision * Recall}{Precision + Recall} \tag{4}$$

4 Moving Object State Space Model

We consider a moving object represented in discrete time state-space with the following observation and state Eqs. (5) and (6):

$$x_n = A_n x_{n-1} + E_n u_n + B w_n \tag{5}$$

$$y_n = H_n x_n + v_n \tag{6}$$

where $x_n \in \mathbb{R}^k$ is the state vector, $y_n \in \mathbb{R}^M$ is the vector of observation. A is the model of the state transition, which is applied to project the previous state x_{n-1} to x_n. E is the input control model, u_n is the input control, B is the noise matrix. H is the model of observation. $w_n \in \mathbb{R}^P$ is the process noise $v_n \in \mathbb{R}^M$ is the colored Gauss-Markov noise with white Gaussian with zero mean $w_n \sim N(0, Q_n) \in \mathbb{R}^P$ and $v_n \sim N(0, R_n) \in \mathbb{R}^M$ have the covariances Q_n and R_n, and the property $E\{w_n v_k^T\} = 0$ for all n and k. Under the assumption that the two noise sequences and the initial state are uncorrelated and independent of each testing instant [23].

5 Kalman Filter

The Kalman filter uses the state equation of the linear system to estimate the state of the system through observation of input and output. Using KF requires knowledge of the system parameters, initial values, and measurement sequences. The KF can estimate the state sequences of the system iteratively [24].

The Kalman filter calculate the optimal estimate by recursively combining previous estimates with the new observations. This process consists of two stages: predict, in which the optimal state $\hat{x}_n^-$ before observing y_n is calculated and update, after observing y_n the optimal posterior state $\hat{x}_k$ is calculated. Also, it calculates the prior estimation error $\epsilon_n^- = x_n - \hat{x}_n^-$, the posterior estimation $\epsilon_n = x_n - \hat{x}_n$, a priori estimate error covariance $P_n^- = E\{\epsilon_n^- \epsilon_n^{-T}\}$, and posterior estimate error covariance $P_n = E\{\epsilon_n \epsilon_n^T\}$.

The a priori error covariance matrix is produced in the predict phase. Since the process noise w_n is assumed white Gaussian with zero mean, the a priori state estimate is given by (7), and the estimation of the a priori error covariance matrix is computed by (8).

$$\hat{x}_n^- = A\hat{x}_{n-1} + E_n u_n \tag{7}$$

$$P_n^- = A_n P_{n-1} A_n^T + B_n Q_n B_n^T \tag{8}$$

Then, in the update stage, the current a priori predictions are combined with the current state observation to redefine the state estimate and the matrix of error covariance. The current observation is used to improve the estimation, and it is called a posteriori estimation of the state.

The measurement y_n is corrupted by the noise v_n. Since v_n is assumed white with zero mean, this becomes (9), and the measurement residual (10).

$$y_n = H_n \hat{x}_{n-1} \tag{9}$$

$$z_n = y_n - H_n \hat{x}^- \tag{10}$$

The residual covariance matrix is given by (10):

$$S_n = H_n P_n^- H_n^T + R_n \tag{11}$$

The optimal Kalman gain:

$$K_n = P_n^- H_n^T S_n^{-1} \tag{12}$$

The a posteriori state estimate:

$$\hat{x}_n = \hat{x}_n^- + K_n(z_n - H\hat{x}_n^-) \tag{13}$$

The a posterior error covariance matrix:

$$P_n = (I - K_n H)P_n^- \tag{14}$$

Below is a KF pseudo-code listed as Algorithm 1.

Algorithm 1: Optimal Kalman Filter

Data: $y_n, u_n, \hat{x}_0, P_0, Q_n, R_n$
Result: $\hat{x}_n, P_n$
Begin
  for n= 1,2, ... ***do***
    $\hat{x}_n^- = A\,\hat{x}_{n-1} + E_n u_n$
    $P_n^- = A_n P_{n-1} A_n^T + B_n Q_n B_n^T$
    $S_n = H_n P_n^- H_n^T + R_n$
    $K_n = P_n^- H_n^T S_n^{-1}$
    $\hat{x}_n = \hat{x}_n^- + K_n(y_k - H_n\hat{x}_n^-)$
    $P_n = (I - K_n H)P_n^-$
  End for
End

6 Optimal Finite Impulse Response (OFIR)

Since the OFIR Batch form is generally computationally time consuming, due to the large dimensions of all extended vectors and matrices. In order to obtain faster processing, efficient computation of a posteriori state can be done such as an iterative algorithm. Its iterative computation on horizon [m, k] for given $\hat{\boldsymbol{x}}_{\boldsymbol{m}}$ and $\boldsymbol{P_m}$ provided by Kalman filter (Algorithm 1) if we change the auxiliary time-index i from $m+1$ to n and take output when $i = n$. The iterative form for the OFIR filter has been developed and tested in [25, 26].

The pseudo-code of the a posterior iterative OFIR filter is listed as Algorithm 2. Given $\hat{\boldsymbol{x}}_{\boldsymbol{m}}$ and $\boldsymbol{P_m}$, this algorithm is iteratively updates $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$ values from $i = m+1$ to $i = n$ using optimal recursions of Kalman filter (Algorithm 1) and obtains $\hat{\boldsymbol{x}}_{\boldsymbol{n}}$ and P_n, when $i = n$. The number of iterations can be limited by optimal horizon length, N_{opt} of the Unbiased FIR filter.

Algorithm 2: Iterative OFIR filter

Data: $y_n, u_n, \hat{x}_m, P_m, Q_n, R_n, N$
Result: $\hat{x}_n$
Begin
  For $n=1,2, \ldots$ ***do***
    $m = n - N + 1$ if $k > N - 1$ and $m = 0$ otherwise
    For $i=m+1,m+2,\ldots, n$ ***do***
      Algorithm 1: $\hat{x}_i, P_i$
    end for
  end for
  $\hat{\boldsymbol{x}}_{\boldsymbol{n}}, P_n$
End

7 Unbiased Finite Impulse Response (UFIR) filter

Unlike the KF and iterative OFIR, the Unbiased FIR does not require any information about initial conditions and noise, except for the zero mean assumption [9, 14, 27, 28, 29]. Therefore, the Unbiased FIR filter is more suited for object tracking, where measurement and process noises are not exactly known. However, the Unbiased FIR requires an optimal horizon length, called N_{opt}, from $m = n - Nopt + 1$ to n, to minimize the Mean Square Error (MSE), and it cannot ignore the CMN V_n, which violates the zero mean assumption at short horizons.

As already mentioned, the UFIR algorithm does not require noise statistics, so the prediction step calculates a single value, a priori state (15).

$$\hat{x}_l^- = A\hat{x}_{l-1} + E_l u_l \tag{15}$$

In the update step, the state estimated is combined with the current observation state to upgrade the state. The estimate is iteratively updated to the a posteriori state estimate using equations (16)–(19).

Generalized noise power gain:

$$G_l = \left[H_l^T H_l + (A_L G_{l-1} A_l^T)^{-1}\right]^{-1} \tag{16}$$

The measurement residual:

$$z_l = y_l - H_l \hat{x}^- \tag{17}$$

The bias correction gain:

$$Gain_l = y_l - H_l \hat{x}^- \tag{18}$$

The a posteriori state estimate is given by (19):

$$\hat{x}_l = \hat{x}_l^- + Gain_l (z_l - H\hat{x}_l^-) \tag{19}$$

Below a pseudo-code of UFIR algorithm is listed as algorithm 3. To initialize iterations, the algorithm requires a short measurement vector $y_{m.k} = \left[y_m \ldots y_k\right]^T$ and matrix (20).

$$C_{m,k} = \begin{bmatrix} H_m\left(A_k^{m+1}\right)^{-1} \\ H_{m+1}\left(A_k^{m+2}\right)^{-1} \\ \vdots \\ H_{k-1}A_k^{-1} \\ H_k \end{bmatrix} \tag{20}$$

Algorithm 3: Unbiased Finite Impulse Response filter

Data: y_n, u_n, N
Result: $\hat{x}_n$
Begin
  For $n = N-1, N, \ldots$ ***do***
    $m = n - N + 1$, $s = n - N + k$
    $G_S = (C_{m,s}^T C_{m,s})$
    $\tilde{x}_s = G_s C_{m,s}^T (Y_{m,s} - L_{m,s}u_{m,s}) + S_{m,s}^k u_{m,s}$
    For l=s+ k: **do**
    $\tilde{x}_l^- = A\hat{x}_{l-1} + Eu$
    $G_l = \left[H_l^T H_l + \left(A_l G_{l-1} A_l^T\right)^{-1}\right]^{-1}$
    $Gain_l = G_l H_l^T$
    $\tilde{x}_l = \hat{x}_l^- + Gain_l(y_l - H_i\tilde{x}_l^-)$
    end for
  end for
  $\hat{x}_n = \tilde{x}_n$
End

Where $S_{m,s}$ and $L_{m,s}$ are given by (21) and (22) respectively, $S_{m,s}^k$ is de *Kth* row vector in (21).

$$S_{m,s} = \begin{bmatrix} E_0 & 0 & \cdots & 0 & 0 \\ A_{m+1}E_m & E_{m+1} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ A_{s-1}^{m+1}E_m & A_{s-1}^{m+2}E_{m+1} & \cdots & E_{s-1} & 0 \\ A_s^{m+1}E_m & A_s^{m+2}E_{m+1} & \cdots & A_s E_{s-1} & E_s \end{bmatrix} \tag{21}$$

$$L_{m,s} = diag(C_{m,s})S_{m.s} \tag{22}$$

8 Optimal Unbiased Finite Impulse Response (OUFIR)

In some cases, requiring an initial state $\hat{\boldsymbol{x}}_m$ and error covariance P_m. Generally, in real applications, all the information on the initial conditions of the model is not available, will be needed to tackle the problem. The OUFIR filter is very indifferent to the initial conditions. In this case, we apply an iterative Optimal Unbiased (OUFIR) filter [30].

8.1 Iterative OUFIR

The iterative OUFIR algorithm is stated below. The prediction phase calculates a single value, a priori state, considering input (u) equal to zero the a priori state is computed by (23.

$$\tilde{x}_{\boldsymbol{l}} = A_l \hat{x}_l^- + Gain_l(y_l - H_i \tilde{x}_l^-) \tag{23}$$

In the update stage, the state estimated is combined with the current observation state to refine the state. In the same way as UFIR, the estimate is iteratively updated to the a posterior state estimate using equations (24)–(28).

The residual covariance matrix is given by (10):

$$S_n = H_n P_n^- H_n^T + R_n \tag{24}$$

The OUFIR filter gain:

$$K_l = P_n^- H_n^T S_n \tag{25}$$

$$K_l^- = (I - K_l H_l)\, G_l N_l G_l^T H_l^T S_n \tag{26}$$

$$Gain = K_l + K_l^- \tag{27}$$

The a posteriori state estimate:

$$\tilde{x}_{\boldsymbol{l}} = \hat{x}_l^- + Gain_l(y_l - H_i \tilde{x}_l^-) \tag{28}$$

A pseudo-code of the Optimal Unbiased FIR filter is listed as Algorithm 4.

9 Experimental Tracking Test

Now we study the performance of the standard Kalman and FIR filters under study (Algorithm 1–Algorithm 4). In this section, we test the algorithms numerically by different simulated data. Our main goal is to evaluate the performance in object tracking using precision and F-score metrics. We consider the two-state model and suppose an object is disturbed by white Gaussian acceleration noise with a given value of standard deviation. The model of a moving target in a two-dimensional space was specified by (5) and (6) with matrices:

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} T \\ 1 \end{bmatrix}, H = [10] \tag{29}$$

In simulation data four scenarios will be considered:

1) Simulated data 1. An object target is disturbed by white Gaussian acceleration noise with a standard deviation of $\sigma_w = 5\,\mathrm{m/s^2}$. The for the data noise (CMN) originates from white Gaussian $\sigma_v = 10\,\mathrm{m}$. The simulation of the trajectory was 1000 points with sample time $T = 1$ s, $P_0 = 0$, $Q = \sigma_w^2$, $R = \sigma_V^2$, on a short horizon $\mathrm{N_{opt}} = 5$.
2) Simulated data 2. A trajectory is simulated at 2000 points $\sigma_w = 50\,\mathrm{m/s^2}$, $\sigma_v = 20\,\mathrm{m}$. With sample time $T = 0.05$ s, $P_0 = 0$, $Q = \sigma_w^2$, $R = \sigma_V^2$, on a short horizon $\mathrm{N_{opt}} = 9$.
3) Simulated data 3. A trajectory is simulated at 2000 points with $\sigma_w = 50\,\mathrm{m/s^2}$, $\sigma_v = 4\,\mathrm{m}$. With sample time $T = 0.05$ s, $P_0 = 0$, $Q = \sigma_w^2$, $R = \sigma_V^2$, on a short horizon $\mathrm{N_{opt}} = 5$.
4) Simulated data 4. A trajectory is simulated at 2000 points $\sigma_w = 60\,\mathrm{m/s^2}$, $\sigma_v = 30\,\mathrm{m}$. With sample time $T = 0.03$ s, $P_0 = 0$, $Q = \sigma_w^2$, $R = \sigma_V^2$, on a short horizon $\mathrm{N_{opt}} = 15$.

Algorithm 4: Iterative OUFIR filter.

Data: y_n, Q_n, R_n, N
Result: $\hat{x}_n$
Begin
  For $n = N-1, N, \ldots$ ***do***
    $m = n - N + 1$, if $k > N - 1$ and $m = 0$ otherwise
    Compute $\hat{x}_{m+1}$ and P_{m+1}
    For l=m+2, m+3, … n **do**
      $P_l^- = A_l P_{l-1} A_l^T + B_l Q_l B_l^T - A_l P_l H_l^T S_l H_l P_n A_l^T$
    $G_l = A_l(I - P_n^- H_l^T S_l H_l^T) G_{l-1}$
    $S_l = H_l P_l^- H_l^T + R_l$
    $N_l = \left(N_{l-1}^{-1} + G_l^T H_l^T S_l H_l G_l\right)^{-1}$
    $K_l = P_n^- H_n^T S_n$
    $K_l^- = (I - K_l H_l)\, G_l N_l G_l^T H_l^T S_n$
    $Gain = K_l + K_l^-$
    $\tilde{x}_l = \hat{x}_l^- + Gain_l(y_l - H_i \tilde{x}_l^-)$
    end for
  end for
  $\hat{x}_n = \tilde{x}_n$
End

9.1 Test on Simulated Data 1

Then we examine the results of algorithms in the object tracking simulation using the bounding box measurements or coordinates as a tool for metric evaluation. In Fig. 2 the true trajectory of the object and the estimations made by the algorithms are shown, where the black line is the real data, the blue line is KF, the red line is UFIR, the yellow line is OUFIR, and the green line is OFIR. Given that the N_{opt} for the FIR filters was 5, the estimates started from this.

Under the conditions given in the first simulation, with low values of white Gaussian acceleration noise and data noise. The Kalman and FIR filters showed similar behavior. To provide a more complete view, we calculate the root mean square error (RMSE). The RMSE values were 347.5495 for KF, 347.2659 for OFIR, 353.3701 for OUFIR, and 353.3701 for UFIR. According to these results, we consider that the algorithms presented a good performance, where OFIR showed the lowest value.

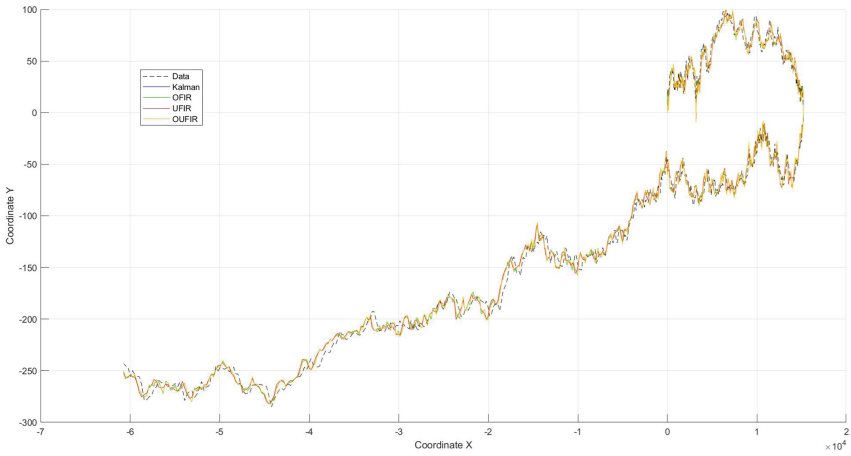

Fig. 2. Estimation of data 1 trajectory in the x-y plane using Kalman and FIR filters.

Performance Evaluation of Algorithms: Precision and Accuracy. The precision results of the test of simulated data 1 are shown in Fig. 3. The OFIR and KF algorithms produced a high precision with similar behavior on the thresholds from 0 to 0.8, from which to decayed. The precision was greater than 0.8 until the threshold of 0.8. It can be inferred that in each detection the overlap of the Predicted Bounding Box (PBB) on the True Bounding Box (TBB) was good. It can be inferred that each detection of the Kalman and FIR filters covers 80% of the TBB area. Since the most used threshold values are 0.5% and 0.75% [20]. Therefore, we consider that Kalman and FIR filters algorithms gave favorable results in the most widely used threshold IoU range.

To measure the accuracy, the F-score metric was used as already mentioned, this parameter allows combining the precision and recall results as a harmonic mean. This metric is based on the bounding box overlap obtained between the algorithm and the real trajectory to calculate the accuracy with which the algorithm operates on an object displacement sequence. The results of the F-score for simulated data 1 are shown in Fig. 4. The Kalman and FIR algorithms produced a high accuracy from 0 to 0.75 threshold, from which to decay. The OFIR filter showed the highest accuracy with 0.7979, closely followed by KF with 0.7979, OUFIR with 0.7912, and UFIR with 0.7886.

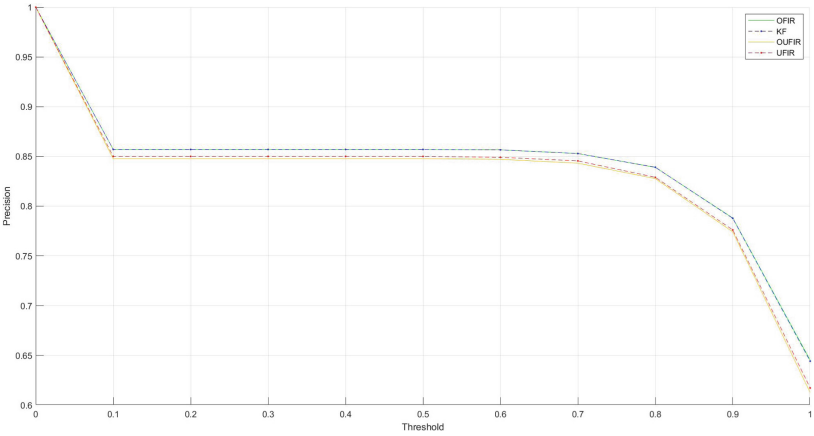

Fig. 3. Precision of Kalman and FIR filters in simulated data 1.

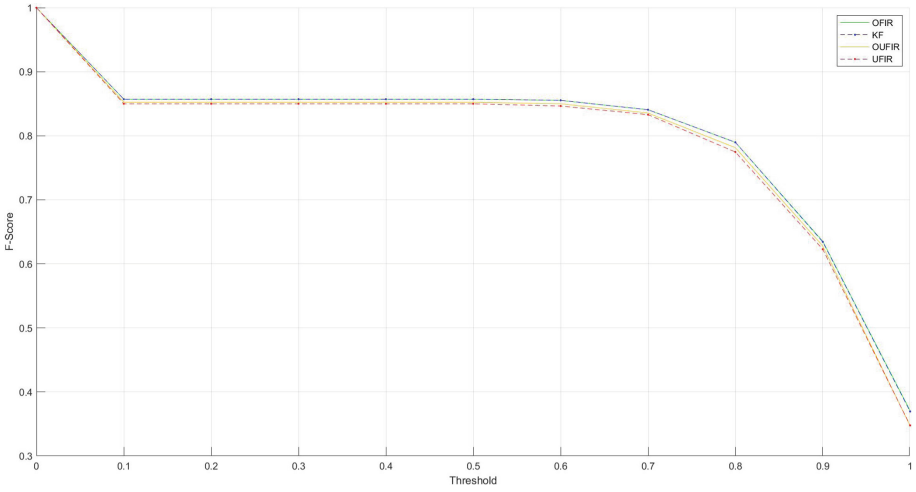

Fig. 4. Accuracy of Kalman and FIR filters in simulated data 1.

9.2 Test on Simulated Data 2

We examine the results of algorithms in the object tracking simulation data 2. In Fig. 5 the true trajectory of the object and the estimations made by the algorithms are shown. The algorithms are identified with the same line colors as the previous section. The horizon length, N_{opt}, for the FIR filters was 9. In this case, a significant difference is observed in the estimates made by the algorithms, where the OUFIR and UFIR are the ones that are closest to the real trajectory with an RMSE value of 4575.38, and 4737.23, respectively. OFIR and KF differ in a greater way, the RMSE values were 8463.93 and 8463.55, respectively. This may be due to the fact that the data simulation was carried out with larger values in data and process noise.

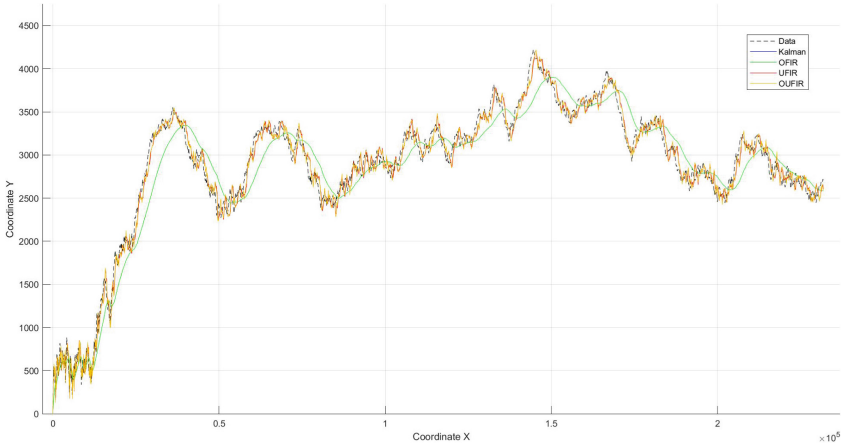

Fig. 5. Estimation of data 2 trajectory in the x-y plane using Kalman and FIR filters.

Performance Evaluation of Algorithms: Precision and Accuracy.
The precision results of a test of simulated data 2 are shown in Fig. 6. OUFIR and UFIR presented a better performance, these produced a precision over 70% from 0 to 0.9 threshold, from which to decay. It can be inferred that each detection covers at least 70% of the TBB area. While OFIR and KF presented low precision values below 50% in the entire threshold range. The average precision for OUFIR was 0.7841, for UFIR was 0.7821, for OFIR was 0.4045, and for KF was 0.4040. Since usually, the threshold used is 0.5, we considered that the algorithms gave favorable results for OUFIR and UFIR.

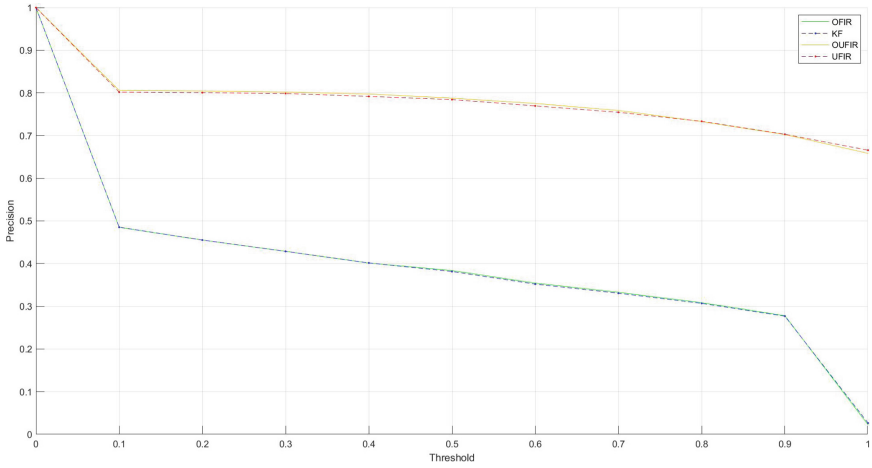

Fig. 6. Precision of Kalman and FIR filters in simulated data 2.

The accuracy results of simulated data 2, the F-score metric, are shown in Fig. 7. The OUFIR and UFIR algorithms produced accuracy values over 0.7 from 0 to 0.6 threshold,

from which to decayed. The value of accuracy towards the 1 threshold is close to 0.5. The OFIR and KF presented a low performance, the F-score values are less than 0.5 in the threshold of 0.1. The accuracy value decreases as the threshold value increases. The OUFIR filter showed the highest average accuracy with 0.7273, closely followed by UFIR with 0.7231, the average accuracy for OFIR was 0.3791, and for KF was 0.3788. According to the results, we consider that OUFIR and UFIR present a good performance in the object tracking task under the given conditions.

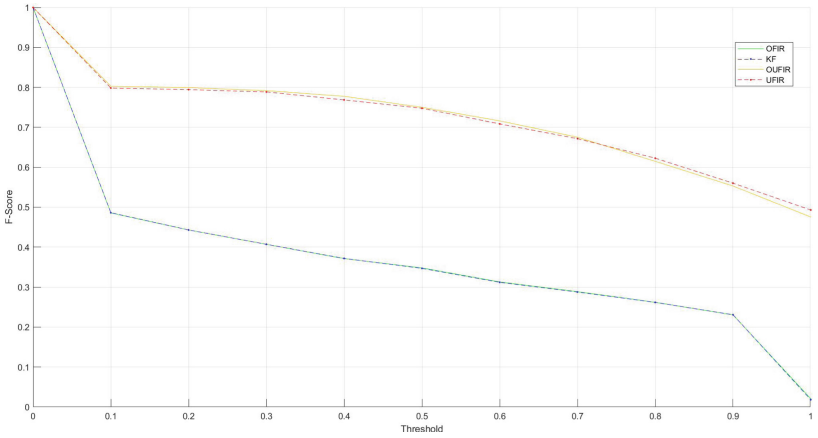

Fig. 7. Accuracy of Kalman and FIR filters in simulated data 2.

9.3 Test on Simulated Data 3

Figure 8 are shown the results of estimation of algorithms in the object tracking of simulation data 3. The algorithms are identified with the same line colors as the previous sections. The horizon length, N_{opt}, for the FIR filters was 5. In this case, there is also a significant difference in the estimates made by the algorithms, where OUFIR and UFIR provided estimates close to the real trajectory, while KF and OFIR showed erroneous estimates, the estimated trajectories are smoothed. The RMSE value for OUFIR, UFIR, OFIR, and KF were 3334.01, 5848.31, 6909.94, and 6910.42, respectively. OUFIR presented the best result. OFIR and KF differ in a greater way than the real data.

Performance Evaluation of Algorithms: Precision and Accuracy.
Figure 9 are shown the precision results of the test of simulated data 3. OUFIR and UFIR presented a better performance, these produced a precision around 80% from 0 to 0.7 threshold. In the full range of the threshold, the precision was over 70%. It can be inferred that each detection covers at least 70% of the TBB area. OFIR and KF presented a poor performance in the correct estimation of the trajectory, the precision was below 50% in most of the full range of the threshold. The average precision for UFIR was 0.8173, for OUFIR was 0.8153, for OFIR was 0.4025, and for KF was 0.4020. Since

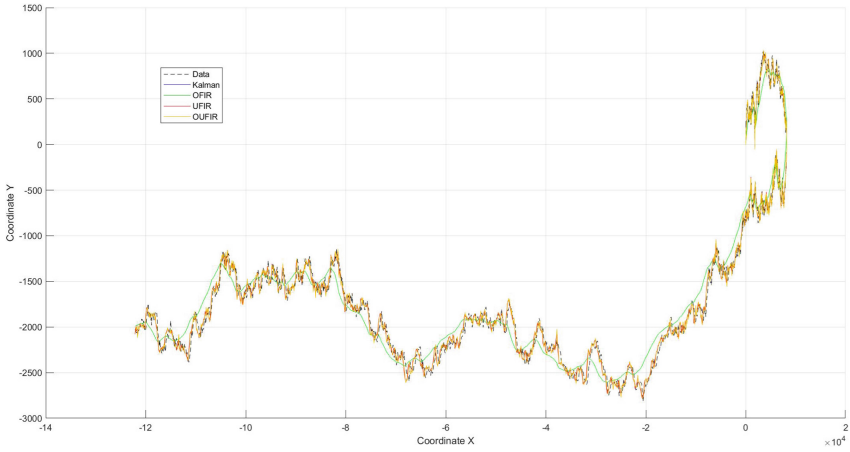

Fig. 8. Estimation of data 3 trajectory in the x-y plane using Kalman and FIR filters.

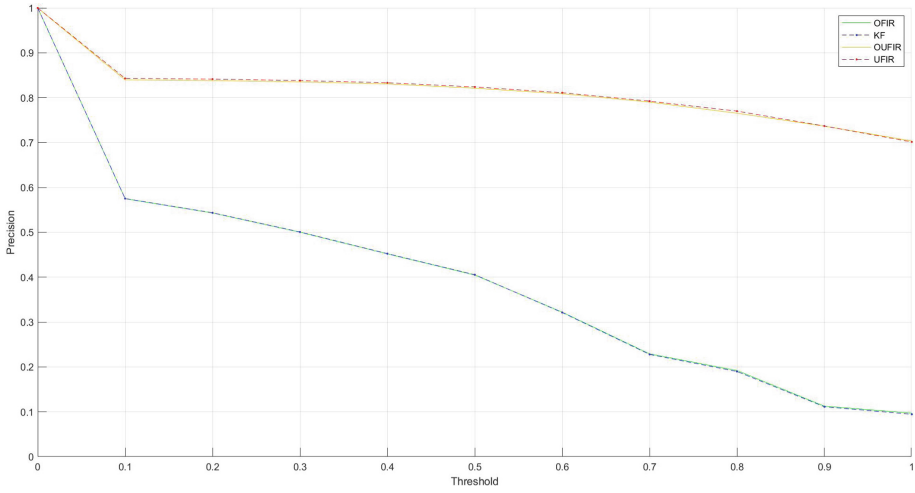

Fig. 9. Precision of Kalman and FIR filters in simulated data 3.

usually, the threshold used is 0.5, according to the results, it can be determined that the OUFIR and UFIR algorithms gave favorable results in the object tracking task.

In Fig. 10 the accuracy results of simulated data 3. The OUFIR and UFIR algorithms produced accuracy values over 0.7 from 0 to 0.6 threshold, from which to decayed. The value of accuracy towards the 1 threshold is close to 0.4. The OFIR and KF presented a poor performance from 0.3 to 1 threshold, with an accuracy value below 0.5. The accuracy value decreases as the threshold value increases. The OUFIR filter showed the highest average accuracy with 0.7402, closely followed by UFIR with 0.7379, the average accuracy for OFIR was 0.3150, and for KF was 0.3152. According to the results,

we can be determined that the OUFIR and UFIR present a good performance in the object tracking process under the given conditions.

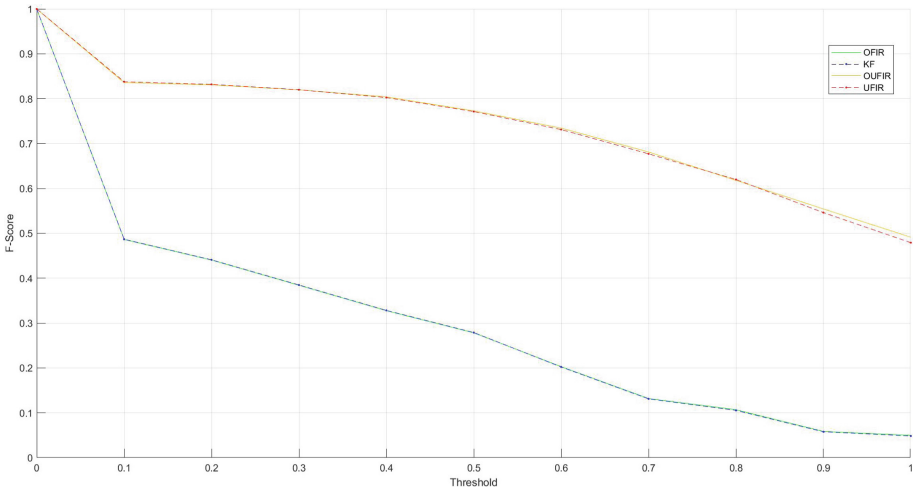

Fig. 10. Accuracy of Kalman and FIR filters in simulated data 3.

9.4 Test on Simulated Data 4

Finally, we analyze the results obtained for the simulated data test 4. The real trajectory and the estimates are shown in Fig. 11; the identification colors remain as already mentioned previously in this paper. The N_{opt}, for the FIR filters, was 15. In this test, the estimates of OUFIR and UFIR were the best compared with those obtained by KF and OFIR. To provide a more complete view, we calculate the root mean square error (RMSE). The RMSE values were 6596.12 for OUFIR, 6973.98 for UFIR, 12932.46 for OFIR, and 12932.46 for KF. According to these results, we consider that the OUFIR and UFIR algorithms presented a good performance, where OUFIR showed the lowest value of RMSE.

Performance Evaluation of Algorithms: Precision and Accuracy. Analyzing the precision metric, it is observed that the performance of the algorithms was lower than in the previous tests. The results are shown in Fig. 12. Again, the OUFIR and UFIR showed higher precision compared to OFIR and KF. However, the precision is between 0.5 and 0.7 in the threshold range of 0.1 to 0.9. Taking as a reference that the Threshold most used to evaluate the precision is 0.5, we can determine that the performance of the algorithms was poor, since in this Threshold the precision just exceeds 0.6 for OUFIR and UFIR, while for OFIR and KF is at 0.2.

The F-score values, accuracy, are shown in Fig. 13. The FIR and Kalman algorithms produced accuracy values less than 0.7 in the entire Threshold range. OUFIR and UFIR

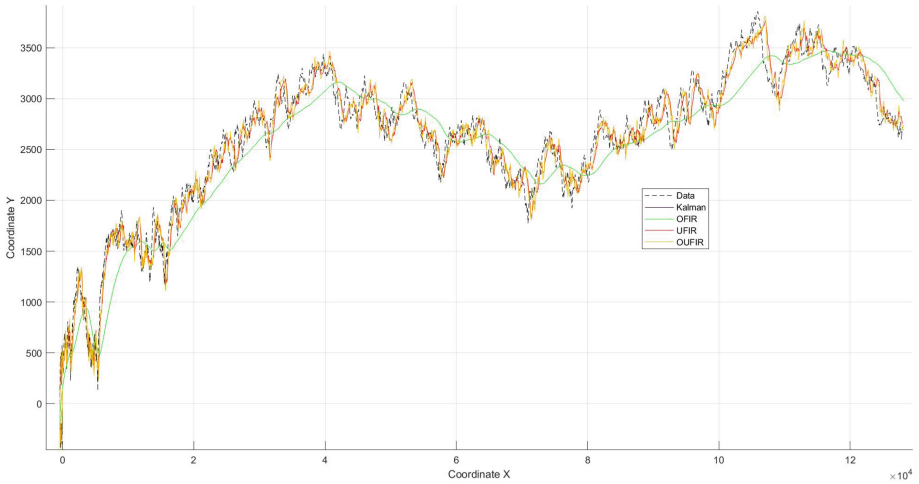

Fig. 11. Estimation of data 4 trajectory in the x-y plane using Kalman and FIR filters.

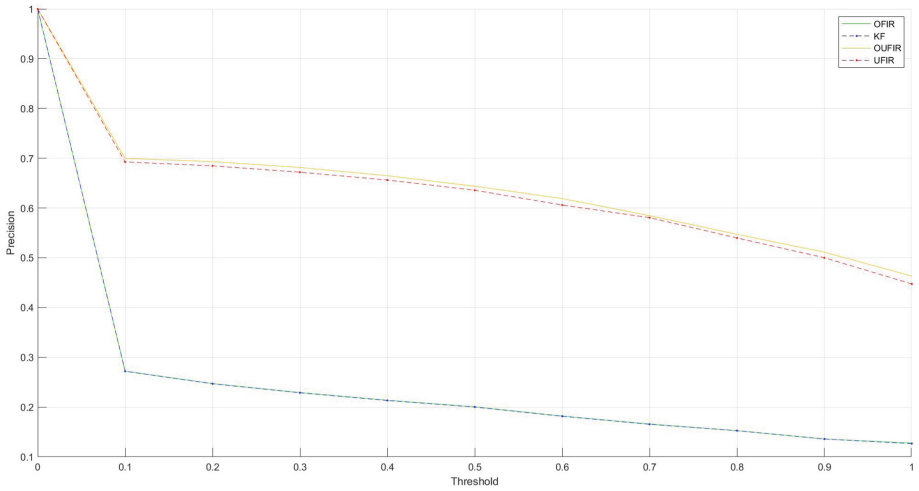

Fig. 12. Precision of Kalman and FIR filters in simulated data 4.

algorithms produced accuracy values over 0.5 from 0.1 to 0.6 threshold, from which to decay. The value of accuracy towards the 1 threshold is close to 0.3. The OFIR and KF presented a low performance. The accuracy obtained was poor, the F-score values are less than 0.3 in the entire Threshold range. In this test, the OUFIR filter showed the highest average accuracy with 0.5715, closely followed by UFIR with 0.5631, the average accuracy for OFIR was 0.2600, and for KF was 0.2596. According to the results, we consider that OUFIR and UFIR present a poor performance in the object tracking task under the given conditions, with higher data and process noise.

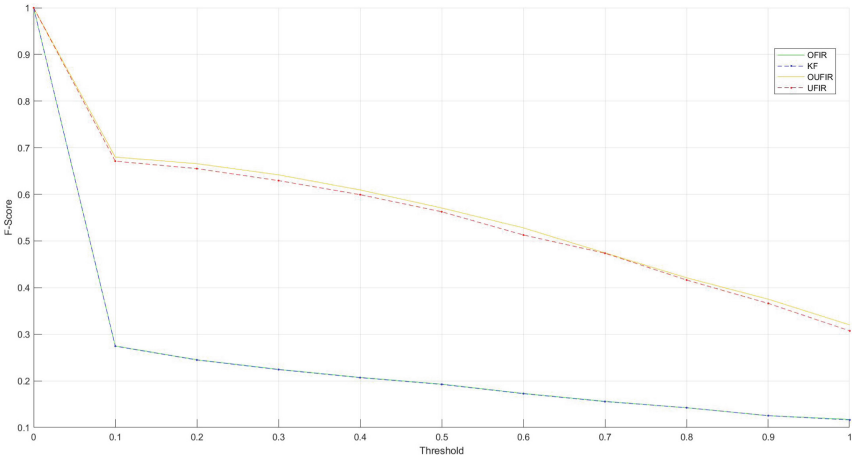

Fig. 13. Accuracy of Kalman and FIR filters in simulated data 4.

10 Conclusions

The KF and FIR estimation algorithms appear to be efficient in the object tracking process under conditions of low the process and data noise. The algorithms showed favorable results in the tracking tests and provided an estimate of the state with greater precision and accuracy. In other hand, when the process and data noise is higher, the OUFIR and UFIR estimation algorithms are more efficient, showing favorable results with high precision and accuracy values, which can be useful in many visual tracking applications since it gives favorable results. Remarking that, UFIR does not require information about the noise and knowing the initial position. In the same way, the OUFIR is highly insensitive to the initial conditions.

Although the Unbiased FIR filter does not require knowing the noise statistics, it can produce larger errors than in KF, because it does not guarantee the optimization. According to the accuracy and precision results, the OUFIR filters showed better performance for tracking objects. Consequently, the OUFIR filter in general shows greater robustness against initial conditions and noise statistics than UFIR, OFIR, and KF.

Therefore, we conclude that the incorporation of state estimators can provide further development of object tracking algorithms for wide variety of application areas.

Regarding the good performance of OUFIR and UFIR filters algorithms, we are now working on develop practical modified algorithms for object tracking to mitigate CMN.

References

1. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with Applications to Tracking and Navigation. Wiley, New York (2001)
2. Bishop, A.N., Savkin, A.V., Pathirana, P.N.: Vision-based target tracking and surveillance with robust set-valued state estimation. IEEE Signal Process. Lett. **17**(3), 289–292 (2009)
3. Brown, R.G., Hwang, P.Y.C.: Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises, 4th edn. Wiley, Hooboken (2012)

4. Burger, W., Burger, M.: Principles of Digital Image Processing, vol. 111. Springer, London (2009)
5. Choeychuent, K., Kumhomtand, P., Chamnongthait, K.: An efficient implementation of the nearest neighbor based visual objects tracking. In: International Symposium on Intelligent Signal Processing and Communication Systems, pp. 574–577. IEEE, Japan (2006)
6. Deepak, P., Suresh, S.: Design and utilization of bounding box in human detection and activity identification. emerging ict for bridging the future. In: Satapathy, S., Govardhan, A., Raju, K., Mandal, J. (eds.) Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2. Advances in Intelligent Systems and Computing, vol. 338, pp. 59–70. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-13731-5_8
7. Farhadi, A., Redmon, J.: YOLOv3: an incremental improvement computer vision and pattern recognition. arXiv preprint arXiv:1804.02767 (2018)
8. Grewal, M.S., Andrews, A.: Kalman Filtering: Theory and Practice with MATLAB. Wiley, Hoboken (2014)
9. Kang, T.K., Mo, Y.H., Pae, D.S., Ahn, C.K., Lim, M.T.: Robust visual tracking framework in the presence of blurring by arbitrating appearance-and feature-based detection measurement. J. Int. Meas. Confed. **95**, 50–69 (2017)
10. Karasulu, B., Korukoglu, S.: A software for performance evaluation and comparison of people detection and tracking methods in video processing. Multimed. Tools Appl. **55**(3), 677–723 (2011)
11. Liang, P., Blasch, E., Ling, H.: Encoding color information for visual tracking: algorithms and benchmark. IEEE Trans. Image Process. **24**(12), 5630–5644 (2015)
12. Murray, S.: Real-time multiple object tracking-a study on the importance of speed. arXiv preprint arXiv:1709.03572 (2017)
13. Padilla, R., Passos, W., Dias, T., Netto, S., Da Silva, E.: A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics **10**(3), 279 (2021)
14. Parekh, H.S., Thakore, D.G., Jaliya, U.K.: A survey on object detection and tracking methods. Int. J. Innov. Res. Comput. Commun. Eng. **2**(2), 2970–2978 (2014)
15. Parmar, M.: A survey of video object tracking methods. Int. J. Eng. Dev. Res. **4**, 519–524 (2016)
16. Shmaliy, Y.S.: Linear optimal fir estimation of discrete time-invariant state-space models. IEEE Trans. Signal Process. **58**(6), 3086–3096 (2010)
17. Shmaliy, Y.S.: An iterative kalman-like algorithm ignoring noise and initial conditions. IEEE Trans. Signal Process. **59**(6), 2465–2473 (2011)
18. Shmaliy, Y.S., Andrade-Lucio, J., Pale-Ramon, E.G., Ortega-Contreras, J., Morales-Mendoza, L.J., González-Lee, M.: Visual object tracking with colored measurement noise using Kalman and UFIR filters. In: 2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 1–6. EEE, Mexico City (2020)
19. Shmaliy, Y.S., Zhao, S., Ahn, C.: Unbiased FIR filtering: an iterative alternative to Kalman filtering ignoring noise and initial conditions. IEEE Control Syst. Mag. **37**(5), 70–89 (2017)
20. Shmaliy, Y.S., Zhao, S., Ahn, C.K.: Kalman and UFIR state estimation with colored measurement noise using backward Euler method. IET Signal Process. **14**(2), 64–71 (2020)
21. Simon, D.: Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. Wiley, Hoboken (2006)
22. Smeulders, A.W.: Visual tracking: An experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1442–1468 (2013)
23. Computer Vision Lab 2013 Visual Tracker Benchmark (2013). http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html
24. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A Survey. ACM Comput. Surv. **38**(4), 1–45 (2006)

25. Yoon, Y., Kosaka, A., Kak, A.: A new Kalman-filter-based framework for fast and accurate visual tracking of rigid objects. IEEE Trans. Robot. **24**(5), 1238–1251 (2008)
26. Zhao, S., Shmaliy, Y.S., Liu, F.: Fast Kalman-like optimal unbiased FIR filtering with applications. IEEE Trans. Signal Process. **64**(9), 2284–2297 (2016)
27. Zhao, S., Shmaliy, Y.S., Khan, S., Ji, G.: Iterative form for optimal FIR filtering of time-variant systems. Recent Adv. Electrosci. Comput. **114** (2015)
28. Zhao, S., Shmaliy, Y.S., Liu, F.: Fast computation of discrete optimal FIR estimates in white Gaussian noise. IEEE Signal Process. Lett. **22**(6), 718–722 (2014)
29. Zhao, S., Shmaliy, Y.S., Ahn, C.: Bias-constrained optimal fusion filtering for decentralized WSN with correlated noise sources. IEEE Trans. Signal Inf. Process. Netw. **4**(4), 727–735 (2018)
30. Zhou, X., Li, Y., He, B., Bai, T.: GM-PHD-based multi-target visual tracking using entropy distribution and game theory. IEEE Trans. Ind. Inform. **10**(2), 1064–1076 (2014)