Pseudo-ground truth trajectory from contaminated data of object tracking using smoothing algorithms.

ELI G. PALE-RAMON[1], YURIY S. SHMALIY[1], LUIS J. MORALES-MENDOZA[2], MARIO GONZÁLEZ-LEE[2], JORGE A. ORTEGA-CONTRERAS[1], RENE FABIÁN VÁZQUEZ-BAUTISTA[2].

[1]Department of Electronic Engineering
Universidad Veracruzana
Salamanca, Guanajuato, 36885
MEXICO

[2] Electronics Engineerig Dept.
Universidad Veracruzana
Poza Rica, Veracruz, 93390
MEXICO

Abstract: - Object tracking is a study area of great interest to various researchers whose main objective is to improve the trajectory estimation for object tracking. In practical applications, the information available that allows the application of algorithms to improve the tracking process sometimes is missing. One of the main obstacles is obtaining ground truth, which takes a long processing time. There are manual methods and applications of reference algorithms. On the other hand, in most cases, the tracking information obtained using a camera is contaminated with noise during the acquisition process. In this paper, we applied smoothing algorithms to compute a pseudo-ground truth achieving lower estimation errors and higher precision than the measurement data. The test results showed that the proposed algorithms with the highest performance are q-lag UFIR and q-lag ML FIR. These smoothing algorithms can be useful in practical applications in object-tracking tasks.

Key-Words: - bounding box, pseudo-ground truth, ground truth, object tracking, smoothing algorithms, measurement noise, precision.

Received: May 25, 2022. Revised: July 21, 2023. Accepted: August 23, 2023. Published: October 2, 2023.

1 Introduction

Many times, the data about the trajectory of an object tracking is obtained through a tracking camera, these data are contaminated by noise in the tracking process, the factors causing this noise can be the movement of the camera, lighting, occlusion, rapid changes of direction, blur, among others. These factors cause the camera not to follow exactly the trajectory of the object, there being variations between the position measured by the camera and the true position of the object.

When tracking algorithms are implemented, it is necessary to know the ground truth trajectory to correctly evaluate the effectiveness of the tracking process. Estimator algorithms require the application of a method to eliminate this noise and compute a pseudo-ground truth that should be a more accurate estimation of the ground truth.

If a video of the object tracking process is available, it is possible to manually annotate each position of the object. This implies a slow process and human errors are possible as well when the complete information about the video is unavailable, such as frame rate and frame size.

Also, in the evaluation of object tracking algorithms, it is necessary to contrast the estimates obtained by the tracking algorithms against the ground truth to evaluate their performance. With an inadequate ground truth, we will have an evaluation of the algorithms that is further from the truth.

In this sense, smoothing algorithms are a suitable tool to remove noise from data. Therefore, smoothing algorithms are useful for reconstructing pseudo-ground truth, which can be used in the estimate process for object tracking [1].

This article shows the application of smoothing algorithms to reconstruct the ground truth and derive a pseudo-ground truth that is accurate enough to the ground truth. The results obtained using smoothing algorithms prove that they are useful for the estimation stage in object tracking.

Based on the test results, the smoothing algorithms provide pseudo-ground truth with lower estimation errors and higher precision than noise-contaminated measurement data. Being q-lag UFIR and q-lag ML FIR the algorithms that exhibit the best results.

Therefore, the pseudo-ground truth reconstruction through smoothing algorithms will have a practical application for those researchers who work in object tracking, where the ground truth is unavailable or incorrect, i.e., the measured tracking data is contaminated with noise.

2 Data of video object tracking

In the video object tracking process, image processing operations seek to identify the target at each position throughout the entire trajectory, which implies recognizing the appropriate features to differentiate the target from the background of the scene. The target information can be described through its properties. One of the most common methods of containing target information during object tracking is the bounding box [2].

The bounding box (BB) is a rectangular box that contains the target object information in a sequence of frames. The target position information is included in a BB array. The measurements of each BB represent the coordinates of the upper left and lower right corners of the box that encloses the target [3]. The BB matrix consists of the bounding box measurements, " x" coordinate, " y " coordinate, width (xw), and height (yh) for each of the object positions throughout the trajectory.

According to the measurements of each BB, the object centroid can be obtained in each position. This information represents the trajectory followed by the target object. As previously mentioned, this information is measured by the tracking camera, considered as contaminated by measurement noise. It is necessary to evaluate the performance through the information provided by the bounding boxes when using smoothing to reconstruct the pseudo-ground truth, that is, how accurate the pseudo-ground truth is compared to the ground truth. The most common method to evaluate the effectiveness of smoothing is by estimation error and precision.

3 Ground truth

In the computer vision field in object tracking tasks, the ground truth (GT) can be interpreted as the set of true data, that is known to be real or true positions of the object during the entire trajectory of the tracking process. These measurements can be represented through coordinates, bounding box measurements, camera pose measurements, etc. The ground truth information can be collected at the source or can be pre-programmed.

On the other hand, pseudo-ground truth (p-GT) can be interpreted as an estimation of the ground truth, which is used as reference data for the application of tracking algorithms and their performance evaluation. This data set can be obtained through hand annotation by a human operator or using a reference algorithm.

Generally, we can establish that there are two methods to obtain the ground truth. The first is through manual annotation of the ground truth data set and the second is a reference algorithm4].

Two of the most common annotation methods are:

- Bounding annotations. A box is drawn based on the characteristics of the object target.
- Point annotations. The position of the object target corresponds to the features extracted from a single point.

4 Performance evaluation

We used standard metrics for evaluating the smoothing performance can be done using metrics, precision, and root mean square error (RMSE). Precision can be defined as the percentage of the number of correct predictions over the total number of predictions [5]-[9].

The RMSE is a measure of the variation between truth values and estimated values [10]. In the case of object tracking, it measures the difference between the truth trajectory and the estimated trajectory. The equation of RMSE is well known and is shown below.

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}} \tag{1}$$

Where N is the number of data points, i-th measurement, y is the truth value and $\hat{y}$ is the predicted value.

To calculate the precision, it is necessary to first calculate another metric, intersection over union (IoU), which indicates the percentage of overlap of the predicted bounding box over the True Bounding box (TBB). The variables used in the calculation of the precision are obtained from the comparison of the IoU result with an established threshold. The variables used for computing the precision are obtained from the comparison of the IoU result with an established threshold [5]-[7], [11]. The equations for calculating IoU and precision are (2) and (3), respectively.

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

$$\text{Precision } = \frac{\Sigma TP}{\Sigma TP + \Sigma FP} = \frac{\Sigma TP}{\text{All detections}} \qquad (3)$$

Where the IA is the area of intersection between the bounding box of the target object, the true bounding box (TBB), and the estimated bounding box (EBB). The TP is true positive, and FP is false positive.

The IoU metric allows establishing the degree or percentage of EBB overlap over TBB, for which it is necessary to establish an IoU threshold that works as the comparison parameter to establish whether it is a correct or incorrect detection. Usually, the IoU threshold is set to 0.5 and 0.75 [12].

Considering a single object tracking, many measures to evaluate the performance of the tracking algorithm are based on the overlap comparison of the EBB versus the TBB. The possible qualification of the bounding box overlap in object tracking compared to a given threshold is shown below [4][6]:

- True Positive (TP). It is a correct detection of a bounding box, that is, the IoU between the EBB and TBB is greater than or equal to the established threshold value.
- False positive (FP). It is an incorrect detection of an object or an off-site detection. The IoU is less than the given threshold value but greater than zero.
- False negative (FN). It is an undetected TBB.

5 State-Space Model

According to the motion of a physical system in space, the next position of the object can be calculated using Newton's equation of motion [13] as shown below:

$$cx = cx_0 + v_0\tau + \frac{1}{2}act^2, \qquad (4)$$

where cx is the object position, cx_0 is the object's initial position, v_0 is the object's initial velocity, ac is the object's acceleration, and τ is the time interval. The state equation is derived from Newton's equation of motion. So, we can construct the dynamic model, the model is represented in discrete-time state-space using the following state and observation equations:

$$x_n = A_n x_{n-1} + B_n w_n \qquad (5)$$

$$y_n = C_n x_n + v_n \qquad (6)$$

where $x_n \in \mathbb{R}^K$ is the state vector, $y_n \in \mathbb{R}^M$ is the observation vector, $v_n \in \mathbb{R}^M$ is the colored Gauss-Markov noise, and $A_n \in \mathbb{R}^{K\times K}$ is the state transition matrix, $B_n \in \mathbb{R}^{K\times P}$ is the gain matrix model, $C_n \in \mathbb{R}^{M\times K}$ is the measurement matrix.
The zero mean Gaussian noise vectors $w_n \sim \mathcal{N}(0, Q_n) \in \mathbb{R}^P$ and $\xi_n \sim \mathcal{N}(0, R_n) \in \mathbb{R}^M$ have the covariances Q_n and R_n and the property $E\{w_n \xi_k^T\} = 0$ for all n and k.

We estimated the state of the 4 coordinates of the bounding box. So, the state-space model is designed for the 4 measurements of the bounding box: left lower corner in x-axis (x_c) left upper corner in y-axis (y_c), BB width (x_w), and BB height (y_h).

For a constant velocity model [14], the state transition (A) is a block diagonal matrix with:

$$\begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \qquad (7)$$

where τ is the sample time. This block is repeated for the x_c, y_c, x_w and y_h to build the complete matrix A.

The gain matrix model (B) and observation matrix (C) are defined as shown below:

$$\begin{bmatrix} \frac{\tau^2}{2} & 0 & 0 & 0 \\ \tau & 0 & 0 & 0 \\ 0 & \frac{\tau^2}{2} & 0 & 0 \\ 0 & \tau & 0 & 0 \\ 0 & 0 & \frac{\tau^2}{2} & 0 \\ 0 & 0 & \tau & 0 \\ 0 & 0 & 0 & \frac{\tau^2}{2} \\ 0 & 0 & 0 & \tau \end{bmatrix}, \qquad (8)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (9)$$

6 Smoothing algorithms

6.1 Fixed-lag Kalman smoother

With equations (4) and (5) the Kalman filter (KF) estimates the state through observation of input and output. The KF can estimate the state dynamics of the system iteratively [15], [16], and consists of two steps: predict, where the optimal state $\hat{x}_n^-$ previous to observing y_n is calculated, and update, where after observing y_n the optimal posterior state $\hat{x}_n$ is calculated. Additionally, it computes the prior estimation error $\epsilon_n^- = x_n - \hat{x}_n^-$, the posterior estimation $\epsilon_n^- = x_n - \hat{x}_n^-$, the a priori estimate error covariance $P_n^- = E\{\epsilon_n^- \epsilon_n^{-T}\}$, and the posterior estimate error covariance $P_n = E\{\epsilon_n \epsilon_n^T\}$.

The prior state estimate is computed by (10), and the prior error covariance matrix is estimated by (11).

$$\hat{x}_n^- = A\hat{x}_{n-1} + B_n w_n \tag{10}$$

$$P_n^- = A_n P_n A + B_n Q_n B_n^T \tag{11}$$

Then, in the update phase, the current prior predictions are combined with the current state observation to redefine the state estimate and the error covariance matrix. The combination of the prediction with the current observation is used to calculate the optimal state estimate and is called the posterior state estimate. The measurement y_n is corrupted by colored measurement noise v_n. The measurement residual is (12).

$$y_n = C\hat{x}_{n-1} + \bar{v}_n \tag{12}$$

The residual covariance matrix is calculated as follow:

$$S_n = C_n P_n^- C_n^T + R_n \tag{13}$$

The optimal gain for Kalman is given by:

$$K_n = P_n^- C_n^T S_n^{-1} \tag{14}$$

A posteriori state estimate:

$$\hat{x}_n = \hat{x}_n^- + K_n(y_n - c\hat{x}_n^-) \tag{15}$$

A posteriori matrix of error covariance:

$$P_n = (I - K_n C)P_n^- \tag{16}$$

For the fixed-lag Kalman smoother, first it is run the standard Kalman filter and initialize the update of fixed-lag smoother for $i = l, \dots, q + l$[17], [18], as follows:

$$L_n = P_n^- C_n^T (S_n)^{-1} \tag{17}$$

$$P_{n+1}^{i,i} = P_n^{i-1,i-1} - P_n^{0,i-1} C_k^T L_n^T A_n^T \tag{18}$$

$$P_{n+1}^{(0,i)} = P_n^{(0,i-1)} \left[A_n - L_{k,0} C_n\right]^T \tag{19}$$

$$\hat{x}_{n+1-i} = \hat{x}_{n+2-i,n} + L_{n,i}(y_n - C_n \hat{x}_n) \tag{20}$$

6.2 q-lag ML FIR smoother

We used a batch q-lag maximum likelihood (ML) Finite Impulse Response (FIR) smoother, q-lag ML FIR, for full covariance matrices. The q-lag ML FIR smoother can be derived from the ML estimate at $k - q$ [20]. We use the (4) and (5) and extend them on $[m, k]$ in the conventional forms shown below.

$$x_{m,k} = A_{m,k} x_m + D_{m,k} W_{m,k} \tag{21}$$

$$y_{m,k} = C_{m,k} x_m + G_{m,k} W_{m,k} + v_{m,k} \tag{22}$$

The state x_{k-q} can be defined at $k - q$ for $u_k = 0$ as

$$x_{k-q} = A_{k-q}^{m+1} x_m + \bar{D}_{m,k}^{N-q} W_{m,k} \tag{23}$$

Where a matrix $\bar{D}_{m,k}^{N-q}$ can be represented with

$$\bar{D}_{m,k}^{N-q} = \left[A_{k-q}^{m+1} B_m \; A_{k-q}^{m+2} B_{m+1} \dots A_{k-q}^{k} B_{k-1} \; A_{k-q}^{k+1} B_k \right. \tag{24}$$

Rearranging the terms in equation (23), we represent the initial state as

$$x_m = \left(A_{k-q}^{m+1}\right)^{-1} x_{k-q} - \left(A_{k-q}^{m+1}\right)^{-1} \bar{D}_{m,k}^{N-q} W_{m,k} \tag{25}$$

The q-lag ML FIR estimate and estimation error in batch forms are calculated in the following. Substituting equation (25) in (23), which separates the regular terms and the random terms.

$$Y_{m,k} - H_{m,k}^{q} x_{k-q} = \mathcal{N}_{m,k} \tag{26}$$

Where $\left(H_{m,k}^{q}\right) = C_{m,k}^{q} W_{m,k} + V_{m,k}$ and the random term

$$\mathcal{N}_{m,k} = C_{m,k}\left(D_{m,k} - H_{m,k}^{q} \bar{D}_{m,k} x_{k-q}\right. \tag{27}$$

The likelihood of x_{k-q} can be written as

$$p\left(Y_{m,k} \mid x_{k-q}\right) \propto \exp\left\{-\frac{1}{2}\left(Y_{m,k} - H_{m,k}^{(q)} x_{k-q}\right)^T \Sigma_{m,k}^{-1}(\dots)\right\} \tag{28}$$

We determine the q-lag ML FIR estimate $\tilde{x}_{k-q}$ and $\Sigma_{m,k} = \mathcal{E}\{\mathcal{N}_{m,k}\mathcal{N}_{m,k}^T\}$ as

$$\tilde{x}_{k-q} = \underset{x_{k-q}}{\arg\min} \tag{29}$$

$$\left\{-\frac{1}{2}\left(Y_{m,k} - H^{(q)}_{m,k}x_{k-q}\right)^T \Sigma^{-1}_{m,k}(\dots)\right\}$$

Setting the derivative equal to zero

$$\begin{aligned}&\frac{\partial}{\partial x_{k-q}}\left(Y_{m,k} - H^{(q)}_{m,k}x_{k-q}\right)^T \Sigma^{-1}_{m,k}(\dots)\\ &= H^{(q)^T}_{m,k}\Sigma^{-1}_{m,k}\left(Y_{m,k} - H^{(q)^T}_{m,k}x_{k-q}\right) = 0,\end{aligned} \tag{30}$$

We derive the q-lag ML FIR smoothing estimate in the canonical maximum likelihood form.

$$\begin{aligned}\hat{x}_{k-q} &= \left(H^{(q)^T}_{m,k}\Sigma^{-1}_{m,k}H^{(q)}_{m,k}\right)^{-1}H^{(q)^T}_{m,k}\Sigma^{-1}_{m,k}Y_{m,k}\\ &= \mathcal{H}^{\mathrm{ML}(q)}_{m,k}Y_{m,k}.\end{aligned} \tag{31}$$

This leads to the q-lag ML FIR smoother gain.

$$\mathcal{H}^{\mathrm{ML}(q)}_{m,k} = \left(H^{(q)^T}_{m,k}\Sigma^{-1}_{m,k}H^{(q)}_{m,k}\right)^{-1}H^{(q)^T}_{m,k}\Sigma^{-1}_{m,k} \tag{32}$$

6.3 q-lag UFIR smoother

The q-lag Unbiased Finite Impulse Response smoother, q-lag UFIR smoother, can be designed to satisfy the unbiasedness condition.

$$\mathcal{E}\{\tilde{x}_{k-q|k}\} = \mathcal{E}\{x_{k-q}\}, \tag{33}$$

where the q-lag estimate can be defined as

$$\tilde{x}_{k-q} \triangleq \tilde{x}_{k-q|k} = \hat{\mathcal{H}}^{(q)}_{m,k}Y_{m,k} \tag{34}$$

The state model is represented by the $(N-q)$ the row vector of the extended state equation (35) on $[m,k]$ as

$$x_{k-q} = \mathcal{A}^{m+1}_{k-q}x_m + \bar{D}^{(N-q)}_{m,k}W_{m,k} \tag{35}$$

where $\bar{D}^{(N-q)}_{m,k}$ is the $(N-q)$ th row vector in $\bar{D}_{m,k}$.

The batch forms of q-lag UFIR smoother are given by the following equations. Applying the condition (33) to (34) and (35) gives two unbiasedness constraints, the UFIR smoother gain $\hat{\mathcal{H}}^{(q)}_{m,k}$ is given by.

$$\hat{\mathcal{H}}^{(q)}_{m,k} = \mathcal{A}^{m+1}_{k-q}\left(C^T_{m,k}C_{m,k}\right)^{-1}C^T_{m,k} \tag{36}$$

Referring to $\mathcal{A}^{m+1}_{k-q} = \left(\mathcal{A}^{k-q+1}_{k}\right)^{-1}\mathcal{A}^{m+1}_{k}$, then we transform (36) to

$$\hat{\mathcal{H}}^{(q)}_{m,k} = \left(\mathcal{A}^{k-q+1}_{k}\right)^{-1}\hat{\mathcal{H}}_{m,k} \tag{37}$$

Where $\mathcal{H}_{m,k} = \mathcal{A}^{m+1}_{k-q}\left(C^T_{m,k}C_{m,k}\right)^{-1}C^T_{m,k}$ is the UFIR filter gain and it gives the homogeneous smoothed estimate.

$$\begin{aligned}\tilde{x}^{\mathrm{h}}_{k-q} &= \left(\mathcal{A}^{k-q+1}_{k}\right)^{-1}\hat{\mathcal{H}}_{m,k}Y_{m,k}\\ &= \left(\mathcal{A}^{k-q+1}_{k}\right)^{-1}\hat{x}_k\end{aligned} \tag{38}$$

Where $\hat{x}_k$ is the UFIR filtering estimate. In this work we use a system without input, $u_k = 0$, so the smoothing estimate is obtained by simple projection of (38) as described in [19], [20].

It is significant to mention that the q-lag UFIR and q-lag ML FIR are of FIR type, i.e. FIR filtering structures are bounded input bounded output (BIBO) stable by design [19]. On the other hand, the Fixed-lag Kalman is a Kalman filter structure, and it is known to be asymptotically stable even when the initial state is unknown [21].

7 Ground truth approximation tests

7.1 Numerical simulation tests

We conducted a computer simulation using the moving object tracking model. In this case, the simulation only represents one possible trajectory followed by an object. The dynamic model corresponds to a constant velocity. The moving object model can be described by (4) and (5). A discrete constant velocity model is simulated where acceleration is a zero-mean Gaussian white noise process. The dynamics of simulated movement correspond to a trajectory in the x and y plane with the following matrices.

$$A = \begin{bmatrix}1 & \tau & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & \tau\\ 0 & 0 & 0 & 1\end{bmatrix}, B = \begin{bmatrix}\frac{\tau^2}{2} & 0\\ \tau & 0\\ 0 & \frac{\tau^2}{2}\\ 0 & \tau\end{bmatrix},$$

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

In addition to computing the actual trajectory of the object tracking, the ground truth, we created a trajectory that simulates the tracking data by a camera in the presence of noise that affects the tracking measurement. We called this trajectory the measurement data and we used it as input data for the smoothing algorithms. The purpose of this was to prove that the pseudo-ground truth obtained by the smoothing algorithms is suitable in cases where the ground truth is unknown.

The numerical stability of the simulation was verified on the basis that a numerical method is stable if small changes in the initial data cause small effects in the final numerical solution. On the other hand, it is numerically unstable if it produces large errors in the final solution [22]-[24]. We used the measurement data, where we perturbed the initial state x_0, and it was compared to the RMSE obtained between the estimate calculated by q-lag UFIR and the actual trajectory with the original initial state x_0. The results are shown in Table 1, which corroborates that small disturbances to the initial state generate small changes in the solution obtained by the UFIR q-lag algorithm. So, we can establish that the simulation method used is stable.

Table 1 Simulation stability.

Initial state x_0	RMSE difference
0.1	0.0103
0.3	0.0600
0.5	0.0020
0.7	0.0111
0.9	0.0469

For the first simulation we consider that an object target is disturbed by white Gaussian acceleration noise with a standard deviation of $\sigma_w = 5\ \mathrm{m/s^2}$. The data noise originates from white Gaussian with $\sigma_\xi = 3$ m. The simulation of the trajectory was 500 points with sample time $T = 0.05$s seconds, $P_0 = 0,\ Q = \sigma_w^2,\ R = \sigma_v^2$.

The RMSE results assessed from smoothing algorithms, fixed lag Kalman, q-lag UFIR, and q-lag ML FIR, and measurement data are presented in Table 2. For the computed RMSE with the q-lag UFIR and q-lag ML FIR, the N_{opt} was 4. The results show that q-lag UFIR smoother presented a higher performance since the value is lower than the other algorithms, followed by ML FIR, which was only surpassed by a small value of 4^{-7}. Fixed-lag Kalman presents a higher RMSE value compared to the other smoothing algorithms. However, it reduces the estimation error of the measurement data.

Fig 1. presents the trajectories reconstructed through the smoothing algorithms, as well as the measurement data. It describes the resulting smoothing of measurement data as observed, reducing the noise and calculating a pseudo-ground truth that closely follows the ground truth. Results of the q-lag UFIR and the q-lag ML FIR smoothers are similar, as mentioned above UFIR is slightly better than ML FIR, which is consistent with the RMSE results.

Table 2 RMSE results of simulated data 1.

Data	RMSE results of algorithms	
	RMSE Value	Performance ranking
Measurement data	3.0353	4
Fixed-lag Kalman	2.6938	3
q-lag UFIR	2.1054	1
q-lag ML FIR	2.1054	2

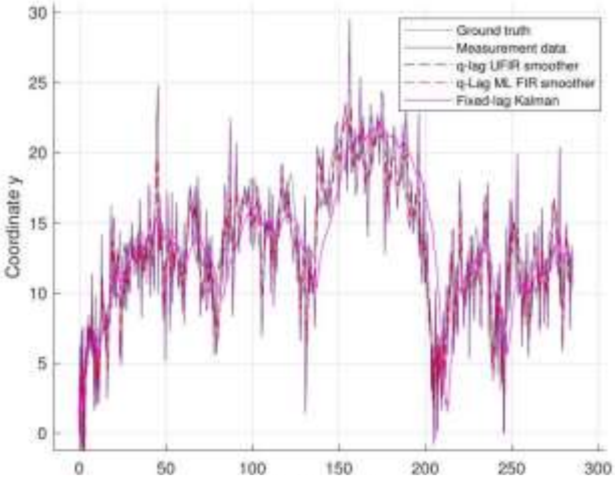

Fig. 1 Smoothing and measurement trajectories of simulated data 1.

Fig.2 presents separately ground truth, measurement data, and the smoothed estimates to clarify the smoothing algorithms' performance. The tracking measurement data, which represents the measurements obtained by a tracking camera under noise conditions, presents large estimation errors. Fixed-lag Kalman smooths the estimates, which is similar to the ground truth; However, the behaviour pattern differs. On the other hand, q-lag UFIR and q-lag ML FIR perform higher at smoothing the estimates, computing a pseudo-ground truth close to the ground truth.

To corroborate the effect of smooth we performed another simulation test. For this test, the moving object model is the same as the example above with the same matrices. The model was developed with a standard deviation of $\sigma_w = 10\ \mathrm{m/s^2}$, and the data noise with $\sigma_v = 30$ m. The simulated trajectory with 500 points with a sample time $T = 0.05$ seconds.

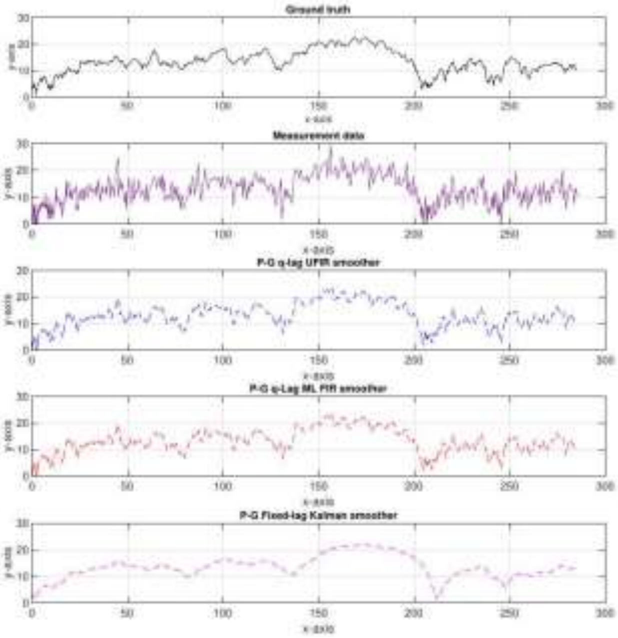

Fig. 2 Smoothing and measurement trajectories (separated) of simulated data 1.

The RMSE results obtained by smoothing algorithms, fixed-lag Kalman, q-lag UFIR, and q-lag ML FIR, and measurement data are shown in Table 3. For the computed RMSE with the q-lag UFIR and q-lag ML FIR, the N_{opt} was 6. In the same way, as in the previous simulation test, q-lag UFIR smoother presented a higher performance with a lower value than the other algorithms, followed by ML FIR, which was only surpassed by a small value of 1.1^{-7}. The UFIR and ML FIR obtain an estimation error equivalent to half that generated by measurement data. In this case fixed-lag Kalman with a higher RMSE value compared to the other smoothing algorithms. However, it reduces the estimation error of the measurement data.

Table 3 RMSE results of simulated data 2.

Data	RMSE results of algorithms	
	RMSE Value	Performance ranking
Measurement data	9.7477	4
Fixed-lag Kalman	7.1000	3
q-lag UFIR	4.8168	1
q-lag ML FIR	4.8168	2

Fig 3 shows the trajectories reconstructed using the smoothing algorithms and the measurement data. As in the previous test, the measurement data represents data measured by a tracking camera, which explain the observed high estimation errors. The smoothing algorithms reduce estimation errors providing a pseudo-ground truth more similar to the ground truth. Results of the q-lag UFIR smoother and the q-lag ML FIR smoother are similar, as mentioned above UFIR slightly performs higher than ML FIR. In this case, fixed-lag Kalman shows poor performance, reducing the estimation errors to a lesser extent.

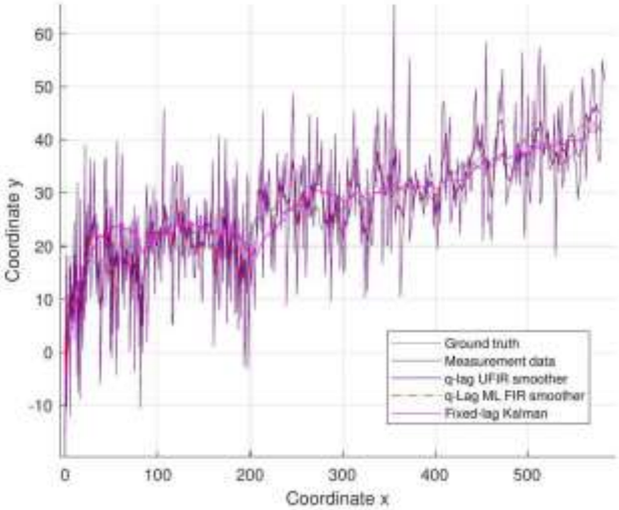

Fig. 3 Smoothing and measurement trajectories of simulated data 2.

For a broader visualization of the smoothing algorithms' performance, in Fig. 4 ground truth, measurement data, and the smoothed estimates are shown separately. Fixed-lag Kalman smoothing with poor performance, although pseudo-ground truth has similarities with ground truth, the pattern of behaviour is different. While q-lag UFIR and q-lag ML FIR perform better in obtaining a pseudo-ground truth whose behaviour is more similar to the GT.

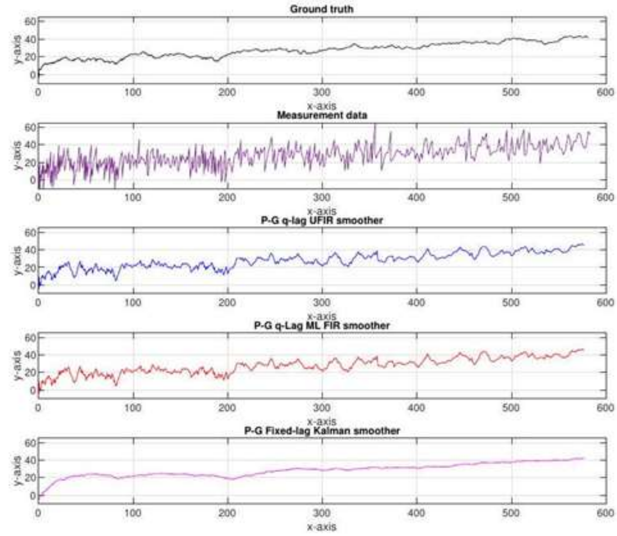

Fig. 4 Smoothing and measurement trajectories (separated) of simulated data 2.

According to the results of the simulation data, the performance of the q-lag UFIR and q-lag ML FIR show great capacity to reduce disturbances, providing a smoothed pseudo-ground truth with fewer estimation errors. This gives another proof that these smoothing algorithms are suitable for obtaining a pseudo-ground truth when the ground truth is unavailable.

7.2 Results of experimental test

The smoothing algorithms have shown a high performance capable of reducing estimation errors by close to 50%. Therefore, we decided to test these algorithms with true tracking data. For this purpose, we use the data called "Remotecar" available in [25]. In this case, we used the bounding box data to estimate and assess the performance of the smoothing algorithms.

The experimental test was performed using the moving object tracking model and matrices proposed in section V. In the object tracking model, we considered that the car target is disturbed by white Gaussian acceleration noise with a standard deviation of $\sigma_w = 30\ \mathrm{m/s^2}$. The data noise originates from white Gaussian $\sigma_v = 20\ \mathrm{m}$. The sample time $\mathrm{T} = 0.05$ seconds, $P_0 = 0, Q = \sigma_w^2, R = \sigma_V^2$, on a short horizon $N_opt = 10$. The model of a moving target is completed according to what is established in section V.

The estimated smoothing trajectories and measurement data are shown in Fig. 5. In this case, a more complex trajectory with greater variation between states is observed. As in the tests with simulated data, the q-lag UFIR and q-lag ML FIR present a higher performance with similar results, an overview of these results indicates that these algorithms manage to reduce the noise present in the measurement to a greater degree. Likewise, fixed-lag Kalman performed with lower performance, but managed to reduce the noise of the measurement data.

In Fig. 6 ground truth, measurement data, and the smoothed estimates are shown separately. It can be observed that the measurement data contains a high variation concerning to the ground truth, representing the measurements obtained by a tracking camera in noise conditions. Fixed-lag Kalman had a lower performance. While q-lag UFIR and q-lag ML FIR smooth the estimates in a better way. The performance of the smoothing algorithms will be better analyzed by the precision metric.

We have the complete information on the object tracking available for assessing the precision, therefore we have the measurements of the bounding box at each point of the trajectory so we can evaluate the performance of the smoothing algorithms.

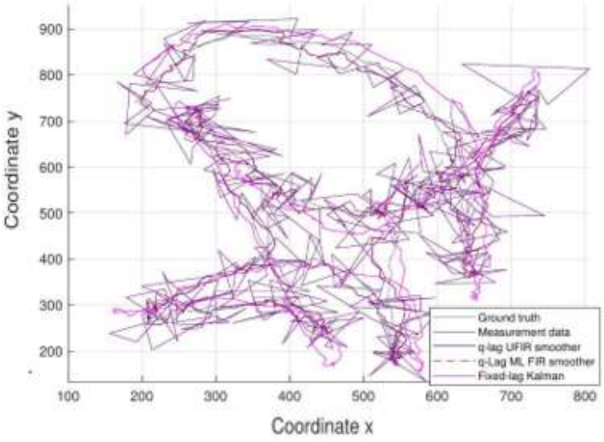

Fig. 5 Smoothing and measurement trajectories of Remotecar

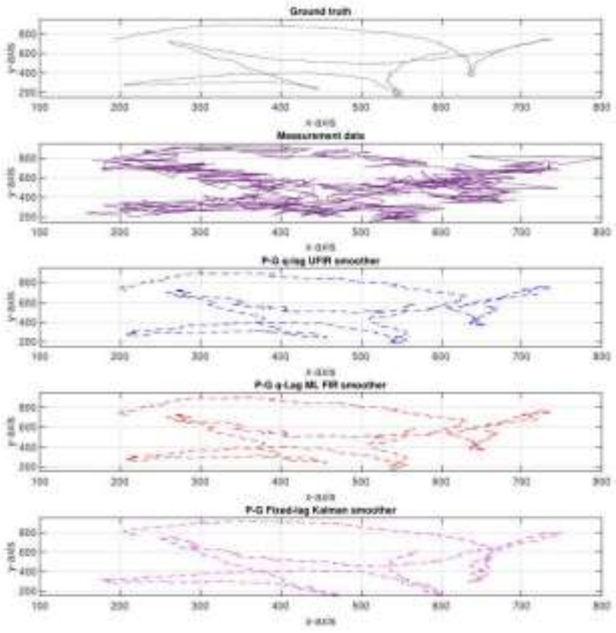

Fig. 6 Smoothing and measurement trajectories (separated) of Remotecar

The precision values of each of the smoothing filters in the entire intersection over union (IoU) threshold range are shown in Fig.7. The q-lag UFIR and q-lag ML FIR smoothers presented the best performance.

Setting the IoU threshold equal to 0.5, the precision of all smoothing algorithms is close to 20%, obtaining a higher precision than the measurement data which is below 10%. The average precision over the full range of the IoU threshold of the smoothing algorithms and the measurement data are shown in Table 4. With these results it is confirmed that q-lag UFIR and q-lag ML FIR generate a pseudo-ground truth reducing the noise of the measurement data, providing adequate information to use as a reference in the development of object tracking tasks when the ground truth does not is available.

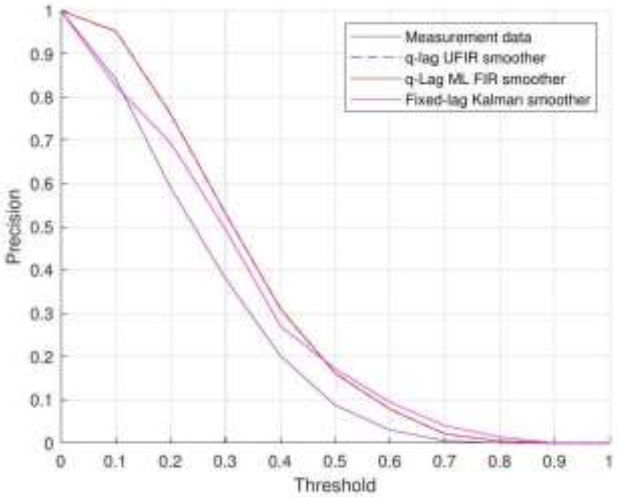

Fig. 7 Smoothing algorithms precision

Table 4 Precision results of Remotecar

Data	Precision results of algorithms	
	Precision	Performance ranking
Measurement data	27%	4
Fixed-lag Kalman	31%	3
q-lag UFIR	37%	1
q-lag ML FIR	37%	2

4 Conclusion

Smoothing algorithms for deriving the ground truth from the measurement data proved to be able to reduce noise, producing pseudo-ground truth with less estimation error than the measured data.

With both simulated data and truth object tracking data, the q-lag UFIR and q-lag ML FIR algorithms exhibited the highest performance, being able to provide a pseudo-ground truth with higher precision and lower estimation error.

Since the q-lag UFIR and q-lag ML FIR algorithms are more robust against measurement data under noise, they provide a reliable pseudo-ground truth for use as a reference in object-tracking research in the field of computer vision. Being practical and robust methods against the lack of ground truth information, data noise, and when complete information on video object tracking is unavailable, they can be useful for state estimation applied with different artificial intelligence methodologies, neural networks, and machine learning, among others, to improve the tracking object process.

Due to the higher robustness of the q-lag UFIR and q-lag ML FIR smoothing algorithms, we are currently working on efficient algorithms that use smoothing and state estimator algorithms for object tracking and plan to report the results in the near future.

References:

[1] E. G. Pale-Ramon, L. J Morales-Mendoza, M. González-Lee, O. G. Ibarra-Manzano, J. A. Ortega-Contreras, and Y. S. Shmaliy. Improving Visual Object Tracking using General UFIR and Kalman Filters under Disturbances in Bounding Boxes. *IEEE Access*, Vol. 11, 2023, pp. 57905-57915

[2] W. Burger, M. J. Burge, M. J. Burge, and M. J. Burge, *Principles of digital image processing.* Springer, Vol. 111, 2009

[3] K. Choeychuen, P. Kumhom, and K. Chamnongthai, An efficient implementation of the nearest neighbor based visual objects tracking, in *2006 International Symposium on Intelligent Signal Processing and Communications*. IEEE, 2006, pp. 574-577.

[4] J. Ferryman, Tenth IEEE international workshop on performance evaluation of tracking and surveillance (pets 2007), *IEEE Computer Society*, 2007.

[5]] B. Karasulu and S. Korukoglu, A software for performance evaluation and comparison of people detection and tracking methods in video processing, *Multimedia Tools and Applications*, Vol. 55, No. 3, 2011. pp. 677723.

[6] D. L. Olson and D. Delen, *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[7] R. Padilla, W. L. Passos, T. L. Dias, S. L. Netto, and E. A. da Silva, A comparative analysis of object detection metrics with a companion open-source toolkit, *Electronics*, Vol. 10, No. 3, 2021.

[8] E. Ranguelova, B. Weel, D. Roy, M. Kuffer, K. Pfeffer, and M. Lees, Image based classification of slums, built-up and non-built-up areas in kalyan and bangalore, india, *European journal of remote sensing*, Vol. 52, No. sup1, 2019, pp. 40-61.

[9] F. Sun, H. Li, Z. Liu, X. Li, and Z. Wu, Arbitrary-angle bounding box based location for object detection in remote sensing image,

European Journal of Remote Sensing, Vol. 54, No. 1, 2021, pp. 102-116.

[10] A. G Barnston, Correspondence among the Correlation, RMSE, and Heidke Forecast Verification Measures; Refinement of the Heidke Score. *Wea. Forecasting*, Vol. 7, 1992, pp. 699–709.

[11] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, Visual tracking: An experimental survey, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 36, No. 7, 2013, pp. 1442-1468.

[12] R. Padilla, W. L. Passos, T. L. Dias, S. L. Netto, and E. A. Da Silva. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, Vol. 10, No. 3, 2021, pp. 279.

[13] C. Xiu, X. Su, and X. Pan, Improved target tracking algorithm based on camshift, in *2018 Chinese control and decision conference (CCDC)*. IEEE, 2018, pp. 4449-4454.

[14] X. R. Li and V. P. Jilkov, Survey of maneuvering target tracking. part I. dynamic models, *IEEE Transactions on aerospace and electronic systems*, Vol. 39, No. 4, 2003, pp. 1333-1364.

[15] R. G. Brown and P. Y. Hwang, *Introduction to random signals and applied kalman filtering: with matlab exercises and solutions*, John Wiley & Sons, Inc., 2012.

[16] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[17] S. Poddar and J. L. Crassidis, Adaptive lag smoother for state estimation, *Sensors*, Vol. 22, No. 14, 2022, pp. 5310.

[18] D. Simon, Optimal state estimation: *Kalman, H_∞, and nonlinear approaches*. Hoboken,NJ: John Wiley & Sons, 2006.

[19] Y. S. Shmaliy and S. Zhao, *Optimal and robust state estimation: Finite Impulse Response (FIR) and Kalman approaches*. John Wiley & Sons, 2022.

[20] S. Zhao and Y. S. Shmaliy, Unified maximum likelihood form for bias constrained fir filters, *IEEE Signal Processing Letters*, Vol. 23, No. 12, 2016, pp. 1848-1852.

[21] C. Lee and Y. Chang, Solution bounds for the discrete riccati equation and its applications, *J. Optim. Theory Appl.*, Vol. 99, No. 2, 1998, pp. 443–463.

[22] T. Belytschko, W. K. Liu, B. Moran and K. Elkhodary. *Nonlinear finite elements for continua and structures*. John wiley & sons, 2014.

[23] Eberly, D. Stability analysis for systems of differential equations. *Geometric Tools, LLC*, 2008, pp. 1-15.

[24] W. K. Macura, Numerical Stability. From MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein. https://mathworld.wolfram.com/NumericalStability.html. accessed: 20 september 2023.

[25] E. G. Pale-Ramon, *Database*, https://drive.google.com/drive/folders/1emoP9O7R0vYNGd343TSvkkUd2eitBS1h?usp=sharing., 2021, accessed: 28 July 2023.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The funding was received from Universidad Veracruzana for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)