# Improving Visual Object Detection using General UFIR Filtering

ELI G. PALE-RAMON[1], LUIS J. MORALES-MENDOZA[2], OSCAR G. IBARRA-MANZANO[1],
RENE FABIÁN VÁZQUEZ-BAUTISTA[2], JORGE A. ORTEGA-CONTRERAS[1],
YURIY S. SHMALIY[1]
[1]Department of Electronics Engineering
Universidad de Guanajuato
Salamanca, 36855
MEXICO
[2]Faculty of Electronics and Telecommunications Engineering
Universidad de Guanajuato
Poza Rica
MEXICO

*Abstract:* Object detection is a fundamental task in computer vision, which involves the identification and localization of objects within image frames or video sequences. The problem is complicated by large variations in the video camera bounding box, which can be thought of as colored measurement noise (CMN). In this paper, we use the general unbiased finite impulse response (GUFIR) approach to improve detection performance under CMN. The results are compared to the general Kalman filter (GKF) and two detection methods: "Faster-RCNN" and "Tensorflow PASCAL Visual Object Classes (VOC)". Experimental testing is carried out using the benchmark data "Car4". It is shown that GUFIR significantly improves the detection accuracy and demonstrates the properties of the effective tool for visual object tracking.

*Key-Words:* Object detection, colored measurement noise, precision, relative error, estimation.

Received: March 17, 2024. Revised: August 11, 2024. Accepted: September 8, 2024. Published: November 13, 2024.

## 1 Introduction

Object detection is a key task in computer vision, [1], [2], [3], [4], [5], that involves identifying objects and their locations in image frames or video sequences, [6], [7], [8], [9]. The problem arises in various research areas, including autonomous driving, surveillance, medical imaging, and robotics, among others, [10], [11], [12], [13]. The annoying thing is that discrepancies usually arise between the estimated positions of objects and the truth, primarily due to environmental factors, [14], [15], [16], [17]. Since discrepancies are not white, they can be treated as colored measurement noise (CMN), making accurate detection difficult, [18], [19], [20], [21], [22].

The goal is thus to discusses the potential for refining object detections through post-processing using filtering methods to improve detection or tracking accuracy under CMN. We do it by using the unbiased finite impulse response (UFIR) filtering approach, [23], [24]. We use the general Kalman filter (GKF) as a benchmark and perform a comparative analysis with two widely-used detection tools called "Faster R-CNN in CNKT" and "Tensorflow PASCAL Visual Object Classes (VOC)", [25], [26], [27], [28], [29], [30]. Testing is provided using video sequences, utilizing classic methods of object initialization and a combination of region labeling and contour search for object detection. The object position is represented by a bounding box (BB), whose coordinates serve as inputs for filtering algorithms. For the "Faster R-CNN in CNKT" and "Tensorflow PASCAL VOC" algorithms, the Visual Object Tagging Tool (VOTT) is employed. The VOTT facilitates the gathering of BB values from each frame. These assets are then converted into the "Faster R-CNN in CNKT" and "Tensorflow PASCAL VOC" formats for model training and obtaining object detection information.

## 2 Object Detection Process

The object detection process utilizes information extracted from images or video sequences to identify and locate objects. Pose information used with this aim is collected in the video camera BB, [31], [32], which is one of the most important tasks, [33], [34], [35]. The process starts with region labeling, divides the image into regions and identifies boundaries between them. Object is then described through the properties, requiring the extraction of parameters and properties for representation. Next, an object detection algorithm is applied to analyze the image and predict the location of objects in the scene. Post-

processing techniques, like detection refinement and filtering algorithms such as Kalman, can be used to improve detection precision, [36], [37], [38], [39], [40]. Finally, the results of object detection are evaluated for the ground truth data. Evaluation metrics include precision, Root Mean Square Error (RMSE), and Center of the Rectangle (CoR), among others, [41], [42].

### 2.1 Bounding Box Information

The BB is a rectangular frame that describes the location of a detected object within an image or a video sequence. The BB is represented by four coordinates: the x-coordinate and y-coordinate of the top-left corner of the box, and the width and height of the box, [43]. The BBs are commonly used to locate objects and provide spatial information about their positions.

### 2.2 Ground truth

The ground truth (GT) is the actual, real, or correct object position in a scene, [44]. The GT can be obtained by manual annotation, through a reference algorithm, or by automatic annotation using specialized software tools. Generating a reliable and valid reference annotation can be a time-consuming and complicated procedure. The ground truth includes information such as the object locations, and their coordinates in the image or video sequence. The GT serves as a benchmark for evaluating the accuracy, and effectiveness of detection. It is also used to evaluate tracking algorithms, [45].

## 3 Performance Evaluation

The detection performance can be evaluated using the precision metric, RMSE, and CoR, [46]. The RMSE is a measure of the variation between truth values and estimated values, [47]. It is computed by taking the square root of the average of the squared differences between each truth value $y_i$ and its corresponding estimated value $\hat{y}_i$ for $N$ (observations), where $i$ represents the $i$-th measurement out of $N$ total observations. Precision is a measure of how well the estimated positions align with the ground truth, [42], [48]. Precision can be quantified using intersection over union (IoU), which indicates the percentage of overlap of the predicted BB over the true BB (TBB). To compute the precision, it needs comparing the IoU results with an established threshold, [42], [48], [49], [50], which can be done as follows:

$$\text{IoU} = \text{IA}(\text{TBB} - \text{EBB}) - \text{IA} \tag{1}$$

$$\text{Precision} = \text{TP}/\Sigma\text{TP} + \Sigma\text{FP} \tag{2}$$

where IA is the area of intersection between the BB of the target object, the TBB, and the estimated BB (EBB). Also TP is true positive and FP is false positive. The center of the rectangle is a metric to measure the distance between the estimated and ground truth BB (GT BB). This metric determines whether the detection is true or false. It is positive when the center of EBB is within the geometrical limit of the GT BB. The results can be presented by the percentage of CoR of estimated BB that is within the ground truth BB, [41]. The estimation error in object detection indicates the difference between the EBB and the ground truth position. The estimation error can be measured using diverse metrics. The relative error is a statistical metric, which is calculated as the result of the difference between the estimated value and GT value divided by the GT value, [51], and is often expressed in percentage.

## 4 State-Space Model with Measurement Disturbances

Input data are the object position information saved in the BB. Since these data are heavily disturbed by the environment, we consider the measurement disturbances as CMN. The object coordinates are stored at the BB coordinates $X_c$, $Y_c$, $X_w$, an $Y_h$. To detect the object in the discrete-time index $n$, we need measurements of the BB coordinates at every $n$. Therefore, we represent the object dynamics in discrete-time state-space by the following state and observation equations:

$$x_n = Fx_{n-1} + Bw_n\,, \tag{3}$$

$$v_n = \Psi v_{n-1} + \xi_n\,, \tag{4}$$

$$y_n = Hx_n + v_n\,, \tag{5}$$

where $x_n \in \mathbb{R}^K$, $K = 8$, is the partitioned state vector

$$x_n = \begin{bmatrix} X_c^T & Y_c^T & X_w^T & Y_h^T \end{bmatrix}^T, \tag{6}$$

in which the vector components are defined by

$$X_c = \begin{bmatrix} x_c \\ V_{x_c} \end{bmatrix}, Y_c = \begin{bmatrix} y_c \\ V_{y_c} \end{bmatrix}, X_w = \begin{bmatrix} x_w \\ V_{X_w} \end{bmatrix}, Y_h = \begin{bmatrix} y_h \\ V_{y_h} \end{bmatrix}.$$

The correspondent velocities $V_{X_c}, V_{Y_c}, V_{X_w}$, and $V_{Y_h}$ are considered constant, following, [52]. The system matrix $F$ is a block diagonal with the components

$$\bar{F} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \tag{7}$$

where the block is repeated for the "x","y", "width", and "height" spatial dimensions, and $\tau$ is the sampling time. The system noise matrix $\bar{B}$ is defined for each of the states by

$$\bar{B} = \begin{bmatrix} \dfrac{\tau^2}{2} \\ \tau \end{bmatrix}, \tag{8}$$

ad the observation matrix is

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (9)$$

We suppose that the object noise $w_n \sim \mathcal{N}(0, Q) \in \mathbb{R}^4$,

$$w_n = [w_{x_c n} \quad w_{y_c n} \quad w_{x_w n} \quad w_{y_h n}]^T,$$

has the known covariance $Q$. Since the measurement data are heavily affected by CMN, we treat the measurement noise $v_n \in \mathbb{R}^4$,

$$v_n = [v_{x_c n} \quad v_{y_c n} \quad v_{x_w n} \quad v_{y_h n}]^T,$$

as colored and represent by the Gauss-Markov model (4), where the components in the diagonal coloredness factor matrix

$$\Psi = \text{diag}\,[\psi_{x_c} \quad \psi_{y_c} \quad \psi_{x_w} \quad \psi_{y_h}]$$

are chosen such that $v_n$ remains stationary. The driving zero mean white Gaussian noise $\xi_n \sim \mathcal{N}(0, R) \in \mathbb{R}^4$ in (4),

$$\xi_n = [\xi_{x_c n} \quad \xi_{y_c n} \quad \xi_{x_w n} \quad \xi_{y_h n}]^T, \quad (10)$$

has the known covariance $R$. The noise vectors $w_n$ and $\xi_n$ are mutually uncorrelated, so the property $E\{w_n \xi_k^T\} = 0$ holds for all $n$ and $k$.

The standard UFIR and KF algorithms cannot be applied to models with CMN. Therefore, we first convert the model (3)–(5) to the standard form with white Gaussian noise components. To this end, we will use the measurement differencing approach proposed in [53], in the form developed for the Euler forward method-based state model (3) in [54].

### 4.1 State-Space Model Transformation

To avoid CMN in the observation equation (5), we use the measurement differencing approach, [53], [55], and introduce a new observation vector $z_n = y_n - \Psi y_{n-1}$ as

$$z_n = Hx_n + v_n - \Psi H x_{n-1} - \Psi v_{n-1}, \quad (11)$$
$$z_n = \bar{H} x_n + \bar{v}_n, \quad (12)$$

where $\bar{H} = H - \Gamma, \Gamma = \Psi H F^{-1}$, and $\bar{v}_n = \Gamma B w_n + \xi_n$. In the new observation equation (12), the noise $\bar{v}_n$ is white Gaussian with the properties:

$$E\{\bar{v}_n \bar{v}_n^T\} = \Gamma\Phi + R, \quad (13)$$
$$E\{\bar{v}_n w_n^T\} = \Gamma B Q, \quad (14)$$
$$E\{w_n \bar{v}_n^T\} = Q B^T \Gamma^T, \quad (15)$$

where $\Phi = BQB^T\Gamma^T$. So, $\bar{v}_n$ and $w_n$ are time-correlated.

To implement the robust UFIR filter and optimal KF for the new state space model (3) and (12), we need new bias correction gains for time-correlated $\bar{v}_n$ and $w_n$. Our transformations will be based on the following measures: $\hat{x}_n^- \triangleq \hat{x}_{n|n-1}$ is the *a priori* estimate, $\hat{x}_n \triangleq \hat{x}_{n|n}$ is the *a posteriori* estimate, $\epsilon_n^- = x_n - \hat{x}_n^-$ is the *a priori* estimation error, $\epsilon_n = x_n - \hat{x}_n$ is the *a posteriori* estimation error, $P_n^- \triangleq P_{n|n-1} = E\{\epsilon_n^- \epsilon_n^{-T}\}$ is the *a priori* error covariance, and $P_n \triangleq P_{n|n} = E\{\epsilon_n {\epsilon_n}^T\}$ is the *a posteriori* error covariance.

## 5 General Filters for CMN

In this section, we will follow, [24], and develop the GKF and GUFIR algorithms for CMN using measured information contained in BBs.

### 5.1 General KF for CMN

There are two options to develop GKF under time-correlated $w_n$ and $\hat{v}_n$, [24]: 1) derive a new bias correction gain or 2) de-correlate the noise vectors. Since the ultimate algorithms are equivalent and do not have significant advantages over each other, we will base our developments on the first option, which implies a new bias correction gain. A pseudo code of the GKF developed for object detection under Gauss-Markov CMN with time-correlated $w_n$ and $\hat{v}_n$ is listed as Algorithm 1, [21], [24]. In the predict phase, this algorithm computes the *a priori* state estimate $\hat{x}_n^-$ and the *a priori* state estimation error covariance $P_n^-$. In the update phase, it uses the new observation equation $z_n$,

**Algorithm 1:** GKF for Object Detection under CMN with Time-Correlated $w_n$ and $\bar{v}_n$

**Data:** $\mathbf{y}_n$, $\hat{x}_0$, $P_0$, $Q$, $R$
**Result:** $x_n$, $P_n$
**begin**
  $\Gamma = \Psi H F^{-1}$ ; $\bar{H} = H - \Gamma$ ;
  $\Phi = BQB^T\Gamma^T$ ;
  **for** $n = 1, 2, \cdots$ **do**
    $z_n = y_n - \Psi y_{n-1}$ ;
    $P_n^- = F P_{n-1} F^T + BQB^T$ ;
    $S_n = \bar{H} P_n^- \bar{H}^T + R + H\Phi + \Phi^T \bar{H}^T$ ;
    $K_n = (P_n^- \bar{H}^T + \Phi) S_n^{-1}$ ;
    $\hat{x}_n^- = F \hat{x}_{n-1}$ ;
    $\hat{x}_n = \hat{x}_n^- + K_n(z_n - \bar{H}\hat{x}_n^-)$ ;
    $P_n = (I - K_n \bar{H}) P_n^- - K_n \Phi^T$ ;
  **end for**
**end**

computes the innovation covariance $S_n$, and updates the bias correction gain $K_n$, the *a posteriori* estimate $\hat{x}_n$, and the *a posteriori* error covariance $P_n$. The subsequent minimization of the trace of $P_n$ by $K_n$ gives the optimal gain $K_n$ for the GKF. Note that the zero coloredness factor $\Psi = 0$ makes $\Gamma = 0$ and $\Phi = 0$ and converts GKF to the standard KF.

### 5.2 General UFIR Filter for CMN

Unlike GKF, the general UFIR (GUFIR) filter does not require any prior knowledge about noise, except for the zero mean assumption, and initial values. Therefore, $w_n$ and $\bar{v}_n$ can be ignored in the model (3) and (12). This means that the GUFIR is invariant to time-correlation between $w_n$ and $\bar{v}_n$, [24], [56]. The GUFIR filter, being of the FIR type, operates without the feedback. The unbiasedness condition assumes orthogonality between the linear estimator and the observation. In this sense, the structure of GUFIR resembles the Gaussian least squares. Moreover, a GUFIR filter does not require initial conditions, [57]. However, the GUFIR filter cannot ignore CMN, which violates the zero-mean assumption at short horizons, [24]. Also, the GUFIR filter processes data over the averaging horizon $[m, n]$ of $N$ points, from $m = n - N + 1$ to $n$, and minimizes the MSE when the horizon is set optimally as $N_{\text{opt}}$, [58]. Note that the FIR filter theory is given in [24].

A pseudo code of the GUFIR filter developed for CMN in [21], [24], and modified for object detection is listed as Algorithm 2. The GUFIR filter operation is divided into two parts: 1) batch initial values and 2) iterative update. The short batch forms are used to initialize the iterations. Accordingly, the algorithm requires a short measurement vector $y_{m,s} = [y_m \dots y_s]^T$, where $s = n - N + K$, and an auxiliary partitioned matrix $C_N$ computed by [24], [57]

$$C_N = \begin{bmatrix} \bar{H}F^{-(N-1)} \\ \vdots \\ \bar{H}F^{-1} \\ \bar{H} \end{bmatrix}, \tag{16}$$

The initial state $x_s$ is also computed in batch form (line 7).

Similarly to GKF, iterations in the GUFIR are performed in two phases: predict and update. In the predict phase, only the *a priori* state estimate $x_s$ is computed. Recall that GUFIR does not require the noise statistics. In the update phase, the state estimate is combined with the actual observation state to refine the state. The *a posteriori* state estimate is updated iteratively using the generalized noise power gain (GNPG) $G_l$, the new observation $z_l$, and the bias correction gain $K_l$. Finally, the *a posteriori* state estimate $\bar{x}_l$ goes to the GUFIR filter output.

**Algorithm 2:** GUFIR Filtering Algorithm for Object Detection under CMN

**Data:** $y_n$
**Result:** $\hat{x}_n$
1 **begin**
2 $\quad \Gamma = \Psi H F^{-1}$ ; $\bar{H} = H - \Gamma$ ; $\Phi = BQB^T\Gamma^T$ ;
3 $\quad$ **for** $n = N-1, N, \cdots$ **do**
4 $\quad\quad m = n - N + 1$ , $s = n - N + K$ ;
5 $\quad\quad G_s = (C_N^T C_N)^{-1}$ ;
6 $\quad\quad Y_{m,s} = [y_m \quad y_{m+1} \quad \cdots \quad y_s]^T$ ;
7 $\quad\quad \bar{x}_s = G_s C_N^T Y_{m,s}$ ;
8 $\quad\quad$ **for** $l = s+1 : n$ **do**
9 $\quad\quad\quad z_l = y_l - \Psi y_{l-1}$ ;
10 $\quad\quad\quad G_l = [\bar{H}^T\bar{H} + (FG_{l-1}F^T)^{-1}]^{-1}$ ;
11 $\quad\quad\quad K_l = G_l\bar{H}^T$ ;
12 $\quad\quad\quad \bar{x}_l^- = F\bar{x}_{l-1}$ ;
13 $\quad\quad\quad \bar{x}_l = \bar{x}_l^- + K_l(z_l - \bar{H}\bar{x}_l^-)$ ;
14 $\quad\quad$ **end for**
15 $\quad\quad \hat{x}_n = \bar{x}_n$ ;
16 $\quad$ **end for**
17 **end**

## 6 Experimental Results

To perform testing of the proposed algorithms, we choose the benchmark data "Car4" available from [59]. Before starting detecting objects using GUFIR filter and GKF, we tune them under the following assumptions. By analyzing the car trajectory, we compute the standard deviation of the acceleration noise to be $\sigma_w = 3\,\text{m/s}^2$ and that suppose that the CMN originates from white Gaussian noise with the standard deviation of $\sigma_v = 2\,\text{m}$. To obtain the ground truth, we manually annotate the positions of the object using VOTT, [60], for the sample time of $\tau = 1\,\text{s}$ and the coloredness factor of $\Psi = 0.3$. This procedure gives $Q = \sigma_w^2$, $R = \sigma_V^2$, and $N_{\text{opt}} = 20$. The object dynamics is described by (3) as shown in Section 4.

Figure 1 illustrates the trajectories based on the centroids of the BBs detected by GKG, GUFIR, faster-RCNN, and TPVOC algorithms.

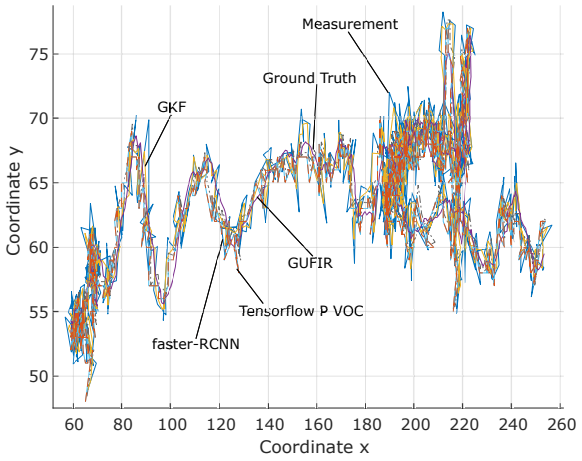


Fig. 1: Measured and estimated positions of the "Car4" benchmark trajectory

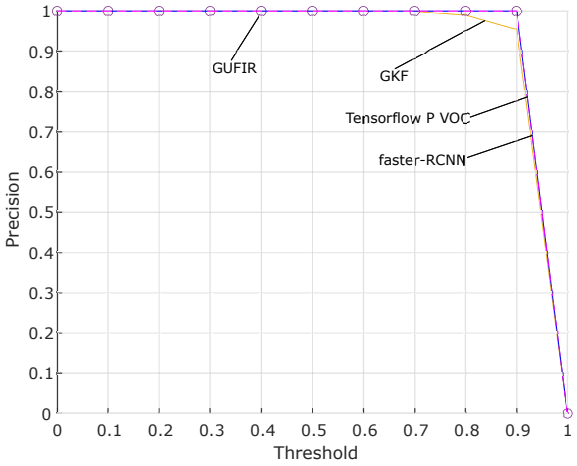


Fig. 2: Precision of the filtering algorithms

The ground truth trajectory and the measured trajectory are also shown here. Inherently, the measurements produce the highest level of noise, while all filtering algorithms reduce variations with respect to the GT trajectory. It also follows that GUFIR effectively smoothes data and reduces noise much better than GKF, which exhibits greater variation. Meanwhile, both "Faster-RCNN" and "Tensorflow PASCAL VOC" algorithms perform in-between. So, we see that GUFIR algorithm outperforms others. To evaluate the performance, the precision metrics were calculated as shown in Figure 2.

As can be seen, GKF demonstrates the high precision up to an IoU threshold of 0.8, with precision decreasing to around 96% between the IoU thresholds of 0.8 and 0.9. In the meantime, the remaining algorithms exhibit the highest precision between the IoU thresholds of 0 and 0.9. Even so, all algorithms demonstrate low precision at the IoU threshold equal to 1. In other words, no algorithm has the ability to estimate BB that overlaps 100% with the GT BB.

Although the GUFIR, "Faster-RCNN" and "Tensorflow PASCAL VOC" algorithms exhibit similar performances, as can be seen in Figure 1, GUFIR is much more successful in noise reduction. The consistency observed in the precision metric might be attributed to the confinement of values within a certain range. If we set a commonly employed IoU threshold of 0.5, [42], then the estimated BB overlaps at least 50% with the GT BB. In this case, the precision becomes the same as for the levels of 53% or 59%.

The relative error, illustrating the estimation error of each algorithm compared to the ground truth, are sketched in Figure 3. Here, each line represents the difference between the centroids of the EBB and the centroid of the GT BB. While all algorithms exhibit

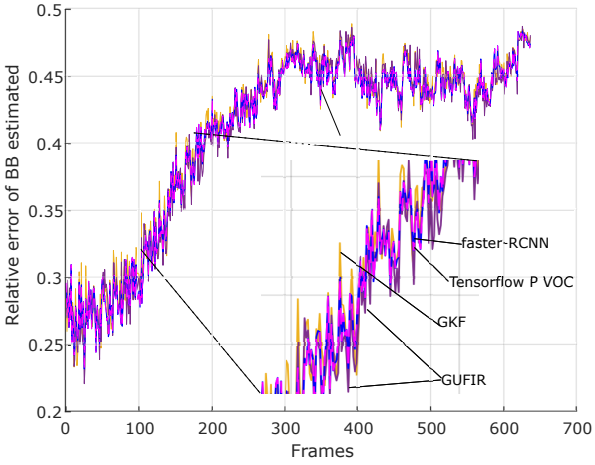


Fig. 3: The relative error of the estimated BB

similar overall behaviors, GUFIR produces smaller errors in each object detection. On the contrary, GKF exhibits gives larger errors, and the "Faster-RCNN" and "Tensorflow PASCAL VOC" algorithms give intermediate error values.

In Table 1, we list the performance metrics, Root Mean Square Errors (RMSEs), and relative Errors, for the GKF, GUFIR, "Faster-RCNN", and "Tensorflow PASCAL VOC" algorithms.

Table 1. RMSE and Relative Errors Produced by the Algorithms

| Data | Metric results | |
|---|---|---|
| | **RMSE** | **Average** |
| GKF | 4.1381 | 0.4053 |
| GUFIR | 2.8707 | 0.3936 |
| faster-RCNN | 3.2883 | 0.4024 |
| Tensorflow Pascal VOC | 2.1054 | 0.4024 |

Table 2. Percentage of Centroids within the Ground Truth

| Data | Percentage |
|---|---|
| GKF | 19% |
| GUFIR | 23% |
| faster-RCNN | 22% |
| Tensorflow Pascal VOC | 22% |

As can be seen, the GKF has the highest RMSE of 4.1381, indicating a significant deviation from the GT values. Moreover, its relative error of 0.4053 suggests that, in average, the GKF predictions deviate from the ground truth by approximately 40.53%. The GUFIR has the best performance, having the RMSE of 2.8707 and reducing the relative prediction error to 0.3936. This means a slightly lower average deviation from GT values, around 39.36%. In the meantime, the standard "Faster-RCNN", and "Tensorflow PASCAL VOC" algorithms exhibit intermediate integral performances.

Additionally, we calculated another metric, Center of Rectangle (CoR), Table 2. This metric evaluates the accuracy of detections by determining if the center of EBB is within the geometrical limits of the GT BB. The results are presented as the percentage of CoR of the estimated BB that is within the GT BB for each of the proposed algorithms. GUFIR exhibits slightly better performance than the "Faster-RCNN", "Tensorflow PASCAL VOC" and GKF algorithms, while GKF shows the lowest performance. These results confirm that GUFIR demonstrates superior performance, effectively improving the detection process with lower estimation error and high precision.

# 7 Conclusion

The GUFIR filtering algorithm developed in this paper for visual object tracking using information about the bounding box coordinates has demonstrated superior precision over the GKF and comparable with the standard "Faster-RCNN" and "Tensorflow PASCAL VOC" algorithms, especially in the 0 to 0.9 threshold range. This has become possible by treating the environmental disturbances as Gauss-Markov colored noise. The GUFIR algorithm effectively mitigates estimation errors and improve detection accuracy. Therefore, it can be recommended as a useful tool for visual object detection.

We are now exploring the integration of GUFIR filter into real-time detection systems and its adaptation to different environmental conditions for robust performance. The results will be reported it the near future.

*References:*

[1] B. J. Scholl, Z. W. Pylyshyn, and J. Feldman, "What is a visual object? Evidence from target merging in multiple object tracking," *Cognition*, vol. 80, no. 1-2, pp. 159–177, 2001.

[2] P. Zhang, D. Wang, and H. Lu, "Multi-modal visual tracking: Review and experimental comparison," *Computational Visual Media*, vol. 10, pp. 193–214, 2024.

[3] M. Dunnhofer, A. Furnari, G. M. Farinella, and C. Micheloni, "Visual object tracking in first person vision," *Int. J. Comput. Vision*, vol. 131, pp. 259–283, 2023.

[4] T. I. Amosa, P. Sebastian, L. I. Izhar, O. Ibrahim, L. S. Ayinla, A. A. Bahashwan, A. Bala, and Y. A. Samaila, "Multi-camera multi-object tracking: A review of current trends and future advances," *Neurocomputing*, vol. 552, p. 126558, 2023.

[5] Z. Tang, T. Xu, H. Li, X.-J. Wu, X.-F. Zhu, and J. Kittler, "Exploring fusion strategies for accurate rgbt visual object tracking," *Information Fusion*, vol. 99, p. 101881, 2023.

[6] A. S. Jalal, "The state-of-the-art in visual object tracking," *Informatica*, vol. 36, pp. 227–248, 2012.

[7] F. Chen, X. Wang, Y. Zhao, S. Lv, and X. Niu, "Visual object tracking: A survey," *Comput. Vision Image Understand.*, vol. 222, p. 103508, 2022.

[8] E. Araujo, C. R. Silva, and D. J. B. S. Sampaio, "Video target tracking by using competitive neural networks," *WSEAS Trans. Signal Process.*, vol. 8, no. 4, pp. 420–431, 2008.

[9] K. Sundaraj, "Real-time face detection using dynamic background subtraction," *WSEAS Informat. Sci. Appl.*, vol. 11, no. 5, pp. 420–431, 2008.

[10] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm Computing Surveys (CSUR)*, vol. 38, no. 4, pp. 1–45, 2006.

[11] J. Viitanen, M. Happonen, P. Patama, and J. Rajamäki, "Near border procedures for tracking information," *WSEAS Trans. Systems*, vol. 3, no. 9, pp. 223–232, 2010.

[12] Z. Li, M. Dong, S. Wen, X. Hu, P. Zhou, and Z. Zeng, "Clu-cnns: Object detection for medical images," *Neurocomputing*, vol. 350, pp. 53–59, 2019.

[13] G. Xu, A. S. Khan, A. J. Moshayedi, X. Zhang, and Y. Shuxin, "The object detection, perspective and obstacles in robotic: a review," *EAI Endorsed Trans. AI Robot.*, vol. 1, no. 1, 2022.

[14] B.-F. Wu, Y.-H. Chen, and P.-C. Huang, "A demand-driven architecture for web-based tracking systems," *WSEAS Trans. Informat. Sci. Appl.*, vol. 12, no. 8, pp. 477–486, 2011.

[15] Y. Xu, Y. S. Shmaliy, X. Chen, and Y. Li, "UWB-based indoor human localization with time-delayed data using EFIR filtering," *IEEE Access*, vol. 5, pp. 16 676–16 683, 2017.

[16] A. J. Frhan, "Detection and tracking of real-world events from online social media user data using hierarchical agglomerative clustering based system," *WSEAS Trans. Comput.*, vol. 16, pp. 355–365, 2017.

[17] D. Lokesh and N. V. Uma Reddy, "Energy efficient routing design for target tracking in wireless sensor network," *WSEAS Trans. Informat. Sci. Appl.*, vol. 19, pp. 132–137, 2022.

[18] Y. Yoon, A. Kosaka, and A. C. Kak, "A new Kalman-filter-based framework for fast and accurate visual tracking of rigid objects," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1238–1251, 2008.

[19] M. K. Tyagi, M. Srinivasan, and L. S. S. Reddy, "Design of traditional/hybrid software project tracking technique: State space approach," *WSEAS Trans. Informat. Sci. Appl.*, vol. 11, no. 10, pp. 345–355, 2013.

[20] R. Haider, F. Mandreoli, and R. Martoglia, "Effective aggregation and querying of probabilistic RFID data in a location tracking context," *WSEAS Trans. Informat. Sci. Appl.*, vol. 12, pp. 148–160, 2015.

[21] 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*, 2023.

[22] A. İftar, "Robust tracking and disturbance rejection for decentralized neutral distributed-time-delay systems," *WSEAS Trans. Syst. Contr.*, vol. 18, pp. 307–315, 2023.

[23] Y. S. Shmaliy, "An iterative Kalman-like algorithm ignoring noise and initial conditions," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2465–2473, 2011.

[24] Y. S. Shmaliy and S. Zhao, *Optimal and Robust State Estimation: Finite Impulse Response (FIR) and Kalman Approaches*. John Wiley & Sons, 2022.

[25] S. Vasuhi and V. Vaidehi, "Target detection and tracking for video surveillance," *WSEAS Trans. Signal Process.*, vol. 10, pp. 168–117, 2014.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances Neural Informat. Process. Syst.*, vol. 28, 2015.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[28] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[29] L. Konwar, A. K. Talukdar, and K. K. Sarma, "Robust real time multiple human detection and tracking for automatic visual surveillance system," *WSEAS Trans. Signal Process.*, vol. 17, pp. 93–98, 2021.

[30] M. Benvenuti, M. G. Colantonio, S. Di Bono, G. Pieri, and O. Salvetti, "Tracking of moving targets in video sequences," in *Proc. 6th WSEAS Int. Conf. on Neural Networks, Lisbon, June 16-18*, 2005, pp. 20–25.

[31] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object detection," in *Computer Vision: A Reference Guide*. Springer, 2021, pp. 875–883.

[32] F. Jalled and I. Voronkov, "Object detection using image processing," *arXiv preprint arXiv:1611.07791*, 2016.

[33] W. Burger, M. J. Burge, M. J. Burge, and M. J. Burge, *Principles of Digital Image Processing*. Springer, 2009, vol. 54.

[34] B. Jahne, *Practical Handbook on Image Processing for Scientific and Technical Applications*. CRC press, 2004.

[35] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[36] S.-Y. Hou, H.-S. Hung, Y.-C. Chang, and S.-H. Chang, "Multitarget tracking algorithms using angle innovations and extended Kalman filter," *WSEAS Trans. Syst.*, vol. 3, no. 8, pp. 420–429, 2009.

[37] X. Sun, H. Qin, and J. Niu, "Comparison and analysis of GNSS signal tracking performance based on Kalman filter and traditional loop," *WSEAS Trans. Signal Process.*, vol. 3, no. 9, pp. 99–108, 2013.

[38] I. Vasilev, D. Slater, G. Spacagna, P. Roelants, and V. Zocca, *Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*. Packt, 2019.

[39] M. H. Assaf, V. Groza, and E. M. Petriu, "The use of Kalman filter techniques for ship track estimation," *WSEAS Trans. Systems*, vol. 19, pp. 7–13, 2020.

[40] S. Chen and C. Shao, "Efficient online tracking-by-detection with kalman filter," *IEEE Access*, vol. 9, pp. 147 570–147 578, 2021.

[41] S. Brenton, "Overview of two performance metrics for obejct detection algorithms evaluation."

[42] 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, p. 279, 2021.

[43] K. Choeychuen, P. Kumhom, and K. Chamnongthai, "An efficient implementation of the nearest neighbor based visual objects tracking," in *2006 Int. Symp. Intell. Signal Process. Commun.*, 2006, pp. 574–577.

[44] Y. Xu, Y. S. Shmaliy, W. Ma, X. Jiang, T. Shen, S. Bi, and H. Guo, "Improving tightly LiDAR/Compass/Encoder-integrated mobile robot localization with uncertain sampling period utilizing EFIR filter," *Mobile Networks Appl.*, vol. 26, pp. 440–448, 2021.

[45] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, pp. 303–338, 2010.

[46] L. Čehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1261–1274, 2016.

[47] A. Barnston, "Correspondence among the correlation [root mean square error] and heidke verification measures; refinement of the heidke score notes and correspondence, climate analysis center 1992," 2020.

[48] B. Karasulu and S. Korukoglu, "A software for performance evaluation and comparison of people detection and tracking methods in video processing," *Multimed. Tools Appl.*, vol. 55, no. 3, pp. 677–723, 2011.

[49] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, 2013.

[50] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. Springer Science & Business Media, 2008.

[51] L. Fan, C. Kang, X. Zhang, and S. Wan, "Real-time tracking method for a magnetic target using total geomagnetic field intensity," *Pure Appl. Geophys.*, vol. 173, pp. 2065–2071, 2016.

[52] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Trans. Aero. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, 2003.

[53] A. Bryson Jr and L. Henrikson, "Estimation using sampled data containing sequentially correlated noise," *J. Spacecraft Rockets*, vol. 5, no. 6, pp. 662–665, 1968.

[54] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, "Kalman and UFIR state estimation with coloured measurement noise using backward Euler method," *IET Signal Process.*, vol. 14, no. 2, pp. 64–71, 2020.

[55] A. Bryson and D. Johansen, "Linear filtering for time-varying systems using measurements containing colored noise," *IEEE Trans. Automat. Contr.*, vol. 10, no. 1, pp. 4–10, 1965.

[56] S. Zhao, Y. S. Shmaliy, and C. K. Ahn, "Bias-constrained optimal fusion filtering for decentralized WSN with correlated noise sources," *IEEE Trans. Signal Inform. Process. Netw.*, vol. 4, no. 4, pp. 727–735, 2018.

[57] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, "Unbiased finite impluse response filtering: An iterative alternative to Kalman filtering ignoring noise and initial conditions," *IEEE Contr. Syst. Mag.*, vol. 37, no. 5, pp. 70–89, 2017.

[58] F. Ramirez-Echeverria, A. Sarr, and Y. S. Shmaliy, "Optimal memory for discrete-time FIR filters in state-space," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 557–561, 2014.

[59] (2015) Datasets-visual tracker benchmark. [Online]. Available: http://www.visual-tracking.net

[60] Microsoft, "Visual object tagging tool: An electron app for building end to end object detection models from images and videos."

## Declaration of Generative AI and AI-assisted Technologies in the Writing Process

The authors wrote, reviewed and edited the content as needed they have not utilised artificial intelligence (AI) tools. The authors take full responsibility for the content of the publication.

## 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

No funding was received for conducting this study.

## Conflicts 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)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US