Efficient Online Tracking-by-Detection with Kalman Filter

SIYUAN CHEN¹ AND CHENHUI SHAO¹

¹Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA (e-mail: siyuanc2@illinois.edu, chshao@illinois.edu)

Corresponding author: Chenhui Shao (e-mail: chshao@illinois.edu).

ABSTRACT Fast and reliable visual tracking of multiple objects in videos has a promisingly broad area of application in manufacturing, construction, traffic, logistics, etc., especially so in large-scale applications where it is not feasible to attach markers to many objects for traditional marker-enabled tracking methods. This paper presents a new approach, Kalman-intersection-over-union (KIOU) tracker, for multi-object tracking in videos that integrates a Kalman filter with IOU-based track association methods. The performance of the proposed KIOU tracker is quantitatively evaluated with UA-DETRAC, an open real-world multi-object detection and tracking benchmark. Experimental results show that the KIOU tracker outperforms the leading tracking methods. Additionally, the KIOU tracker has speed comparable to simple area overlap-based track association and quality comparable to methods with much higher computational costs, demonstrating its potential for online, real-time multi-object tracking.

INDEX TERMS Computer vision, Kalman filter, multi-object tracking, tracking-by-detection, online tracking, transportation

I. INTRODUCTION

Tracking of assets and related personnel are of practical value in a variety of usage cases, including traffic, healthcare, storage, logistics, construction and manufacturing sites, and efficient tracking methods are constantly deemed as a significant topic of research. Different tracking technologies, such as barcodes and radio-frequency identification (RFID) tags [1], Ultra WideBand (UWB) technology [2], global and indoor positioning system [3], ultrasonic sensors [4], RGB [5], RGBD [6], or depth-only cameras [7] as well as 3D structures calculated from Structure-from-Motion (SfM) algorithm [8], have been developed, deployed and commercialized at various sites. Among these, the RFID and UWB systems see the most success, with some solutions entering commercialization [9] thanks to their accuracy, reliability, and relatively low cost. These solutions are usually enabled by a marker, either passive (e.g., RFID) or active (e.g., UWB and wireless local area network) that are attached to valuable assets in the scene.

In contrast, vision-based tracking was usually deemed as impractical in earlier reviews [10]–[12]. Reviewers value marker-enabled indoor tracking method, including RFID, and UWB, for their reliability and localization range over camera tracking enabled by basic machine vision applications including barcode and QR-code.

However, vision-based tracking systems provide application opportunities that other alternatives fail to offer. A vision-based tracking system does not need a marker attached to each tracked object, which is a prerequisite for almost all other alternative tracking methods. Adapting a vision-based tracking system spares the user from the daunting task of instantiating, documenting, and maintaining the numerous passive or active markers for each tracked asset. Although advancements in mobile sensor technology allow easier housekeeping of the makers, users are limited to marking only the most valuable assets, in some other use cases it may become impossible to maintain a marker for each object, i.e., a parking lot or a logistics/distribution center, and it is generally unacceptable to track personnel/employees with the aforementioned markers. In such cases, a vision-based, marker-less tracking system provides potential for tracking significantly more objects at no additional cost, and may see application in high-flow rate applications like traffic analysis.

Following the rapid improvement of neural network-enabled image classifiers including Fast/Faster/Mask R-CNN [13]–[16], single shot multibox detector (SSD) [17], and fully convolutional networks (FCN) [18], [19], tracking of general objects in uncontrolled environments has also seen
increased interest, particularly in construction and manufacturing sites. Recent advances in computer vision, including point cloud analysis [8], tracking-by-detection [20], [21] and object re-identification across views [22] have all been found valuable in these usage scenarios.

Initial literature formulate the multi-object tracking-by-detection task as identifying optimal association of detections into tracks through global optimizations [23], [24]. For example, the Simple Online Realtime Tracker (SORT) [25] solves the global track association problem using the Hungarian algorithm. As an extension to SORT, Wojke et al. [26] incorporates visual information to aid in making associations over a longer time gap. Although the above tracking methods involving global optimization provide good performance, they do not permit on-line operation, which is preferred given the proliferation of fast and highly accurate image classifiers and detectors that run in nearly real-time. Many on-line tracking-by-detection approaches have been proposed [25], [27], [28] in pursuit of real-time, on-line tracking performance. These approaches perform frame association using simple spatio-temporal overlap measure to maximize frame rate, while relying on high-quality detections from a separate detector to achieve preferable tracking performance. An overlap is usually measured by the IOU of the detection bounding boxes between consecutive frames.

Among the various on-line tracking-by-detection methods, Use of probabilistic models [29], [30], particle filters [31], [32] and probability hypothesis density (PHD) filters [33]–[36] have been extensively discussed. Instead of relying on probabilistic models, the IOU tracker [27] achieves high speed by using only detection bounding boxes as the input while delivering performance improvements over previous optimization-based approaches. An extension to the IOU tracker, named V-IOU [28], implements a single-object visual tracker for each track in an effort to repair fragmented tracks due to intermittent missed detections across frames. This approach helps alleviate track fragmentation when minor, intermittent missed detections are present, but when tracks fail to associate with new detections for an extended period, i.e., due to occlusion, more costly techniques including visual re-identification must be performed in hope to re-acquire the missed object.

In this work, we extend the approach of [27] to further address the problem of track fragmentation by instantiating a Kalman filter [37] for each tracked object so that object dynamics are learned and tracked in a prediction-and-correction manner. A recursive Bayesian estimator for linear Gaussian dynamics are learned and tracked in a prediction-and-correction manner. A recursive Bayesian estimator for linear Gaussian

tracking-by-detection problems. We evaluate our approach—KIOU tracker—on the UA-DETRAC dataset in comparison with [27], [28] and other leading methods in the benchmark and demonstrate salient performance advantage over the state-of-the-art. We further show that this approach is suitable for real-time applications as it achieves comparable performance in multi-object tracking accuracy (MOTA), track fragmentation (FM) and identity switches (IDs) metrics with the current leading result while saving up to 80% of computational resources by using less input frame rate.

The remainder of this paper is organized as follows. Section II presents the proposed methodology. Main results are presented and discussed in Section III. Finally, Section IV concludes the paper. The source code for the proposed method has been made available online1.

II. METHODOLOGY

The KIOU tracker is an extension to the IOU tracker [27], which provides preferable performance on the UA-DETRAC dataset [45] at minimal computational cost. In this section, we provide an overview to the IOU tracker and demonstrate the improvements made by the proposed method.

A. IOU TRACKER

With continued performance improvements achieved by object detection and semantic segmentation methods, it is idealistically claimed that bounding boxes for generic objects in an image frame can be generated in a reliable and consistent manner, such that track association for multi-object tracking tasks become trivial [27].

Based on such belief, the IOU tracker operates on a list of bounding box locations in each frame, and performs track association in a greedy manner. Namely, for each existing track, the tracker calculates IOU metric between last known location of the object and all detections in the current frame. If the highest-overlapping detected bounding box also passes the pre-set IOU threshold, the said object is associated to the corresponding track. Given a detection \( d_f \) and a track whose last frame is \( t_{f-1} \), the IOU measure is defined as follows:

\[
\text{IOU}(d_f, t_{f-1}) = \frac{\text{Area}(d_f \cap t_{f-1})}{\text{Area}(d_f \cup t_{f-1})}
\]

For speed and simplicity, the IOU tracker does not tolerate missed detections. Tracks that fail to match with a current detection are terminated immediately, and all detections that are not matched with an existing track will be initiated as a new track at the end of the current iteration.

When paired with a modern object detector that provides high-quality detections, the IOU tracker provides preferable performance at surprisingly high speed. On dense traffic scenes featured in the UA-DETRAC dataset, the IOU tracker achieved leading performance while maintaining a frame rate of more than 100,000 frames per second (FPS).

1Source code available at https://github.com/siyuanc2/Kioutr.
The published results of the IOU tracker in a tracking-by-detection task are based on two assumptions—(1) a detector must generate as few missed detections as possible; and (2) the detector and tracker must collectively run at high frame rates such that detections of an object in consecutive frames have a high IOU rate. Nevertheless, these assumptions cannot be easily fulfilled when the algorithm is applied to on-line tracking.

It is at first unrealistic to assume that a perfect detector is used in tandem with the tracker. Missed detections and false positives exist in the output of nearly every detector and therefore a good tracker must address these flaws. Additionally, the high frame rates at which many trackers are evaluated may not be available in an on-line tracking setup. The EB detector [46] for vehicle detection reports 9-13 FPS on a high-end Nvidia TITAN X GPU, well below 25 FPS at which the UA-DETRAC dataset [45] is recorded. Therefore, the tracker may face at least 50% reduction in available frames, which invalidates the high FPS assumption set above.

### B. KALMAN FILTER

The Kalman filter [37] has long been treated as the go-to solution for various tracking and navigational tasks [47], and is also theoretically well poised for the visual tracking problem. We may assume that a tracked object’s movement has the form:

\[ x_{k+1} = \Phi x_k + w_k \]

where \( x_k \) is the state vector containing position and speed of the object at time \( k \); \( \Phi \) is the state transition matrix of the object movement from time \( k \) to \( k+1 \), assuming a constant speed; and \( w_k \) is the process noise that follows a normal distribution.

Observations on the position of the object can be modeled in the form:

\[ d_k = Hx_k + v_k \]

where \( d_k \) is the observed coordinate of the object at time \( k \); \( H \) is the observation matrix connecting the state vector and coordinate vector; and \( v_k \) is the measurement noise that follows a normal distribution. The covariances of the process noise and the measurement noise are given by:

\[ Q = E[w_kw_k^T] \]
\[ R = E[v_kv_k^T] \]

The goal of the Kalman filter is to estimate location and speed of the object, \( \hat{x} \). The error covariance matrix at time \( k \) is given by:

\[ P_k = E[(e_k e_k^T)] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \]

(5)

The Kalman filter is useful in visual tracking as one can obtain a prior estimate of the object coordinate \( \hat{d} \) from process knowledge, at least in the short term, without the current measurement:

\[ \hat{d}_k = H\hat{x}_k = H\Phi\hat{x}_{k-1} \]

(6)

Whenever a new measurement has been obtained, i.e., the coordinate of the object at time \( t \) has been found, the estimated state \( \hat{x} \) can be updated as follows:

\[ \hat{x}_k = \hat{x}_k' + K_k(d_k - H\hat{x}_k) \]

(7)

where \( K_k \) is the Kalman Gain for time \( k \), which can be obtained given the current measurement:

\[ K_k = P_k'HT(HP_k'HT + R)^{-1} \]

and the error covariance matrix can be updated by:

\[ P_{k+1} = \Phi P_k \Phi^T + Q \]

(8)

The Kalman filter recursively updates \( K_k \), \( P_k \) and \( \hat{x}_k \) and can be used to provide a smoothed trajectory for a track \( t = \{d_0,d_1,...,d_k\} \). When that track fails to match with a detection at time \( k \), process knowledge can be used to project the track \( \hat{d}_{k+1} \) in the short term, reducing track fragmentation caused by occasional missed detections as shown in later part of this paper.

### C. KIOU TRACKER

In observation of the aforementioned challenges faced by the IOU tracker, we propose the KIOU tracker by extending the IOU tracker in seek of addressing the challenges of missed detections and low FPS, while preserving the merits of the original approach. Namely, the proposed approach leverages the Kalman filter’s ability to consider a series of measurements with presence of noise. By instantiating a separate Kalman filter instance for each track, the KIOU tracker estimates object dynamics along with location information, and use the estimated speed to perform track association in a prediction-and-correction manner.

As is with the original implementation in [27], [28], track association is done by evaluating IOU measure of a track with candidates from each frame. Searching for detections with highest overlap with existing tracks is expected to lead to correct track association, provided that the frame rate is high enough such that the difference between two consecutive frames is as small as possible. At lower frame rates, the IOU threshold \( \sigma_{IOU} \) must be significantly relieved, from which the results may be prone to identity switches and false track associations. This complication can be alleviated though, as a Kalman filter may estimate the trajectory and project the track to next time steps. It takes two associated frames for the Kalman filter to start making predictions, after which the IOU measure can be evaluated between current detections \( d_k \) and projected tracks \( \hat{d}_k' \).

Incorporating a Kalman filter to the KIOU tracker brings pronounced benefits. First, a track remains current even with the presence of intermittent missed detections: the Kalman filter may continue the track without an observation should the detection is missed in the current frame. Thanks to this feature, the KIOU tracker can keep current tracks pending for extended periods before termination, resulting in less track...
fragments and identity switches. Gaps in a track resulting from missed detections can be easily filled by interpolation if needed.

The ability to predict next frames further allowed for opportunities to save computational cost and meet the performance requirements for on-line tracking. This is however not to claim that our method has a lower computational complexity than that of [27]; in fact, the KIOU tracker runs at 160.85 FPS on the DETRAC-Train dataset, which is no comparable with IOU tracker that achieves 10329 FPS on the same hardware setup. Rather, we claim that when the Kalman filter is combined with a suitable state transformation matrix that correctly characterizes the objects’ motion pattern, the tracker can achieve little performance loss while using a fraction of the frames, effectively mitigating the computational burden from the detector. We believe this merit is of practical value considering that the computational complexity of an object detector is much larger than that of the KIOU tracker. The EB detector which is used for detecting vehicles in the DETRAC-Train dataset runs at 9-13 FPS as reported in [46], while more versatile detectors may run below 10 FPS on a high-end GPU [15]. When the KIOU tracker is supplied with detections from the EB detector, skipping 50% of the frames brings the duo to real-time tracking range, and tracking performance does not suffer from significant degradation until more than 70% of frames are dropped. We present detailed results in the Section III.

The KIOU algorithm is presented in detail in Algorithm 1. Detections from each frame \(D_k\) are first filtered by detection confidence threshold \(\sigma_l\). Track association is facilitated by evaluating the IOU measure between detections \(d\) and active tracks \(T\), which is further guarded by IOU threshold \(\sigma_{IOU}\). Tracks that fail to associate with a current detection are kept pending for up to \(\sigma_f\) frames, after which obsolete tracks longer than \(\sigma_l\) are terminated.

We use a constant speed linear movement model for the Kalman filter to adapt to object movements in the UA-DETRAC dataset. In this case, the Kalman filter projects object location by assuming that the object moves under a constant speed and same direction as estimated from previous observations. Other motion models can be used where appropriate. Although no significant impact to tracking results can be seen when there is few or no missed detections, unsuitable motion models that does not capture object dynamics will eventually lead to deviations in position estimation and false track association when a track remain unobserved for extended periods.

### III. RESULTS

In this section, we demonstrate the performance of the KIOU tracker by quantitatively evaluating the algorithm with publicly available object tracking datasets, and compare the results with major open-source implementations.

### Algorithm 1: Kalman IOU tracker

| Input : \(D = \{D_1, D_2, ... D_F\}\) |
| Output: \(T_{finished}\) |
| \(T_{active} \leftarrow \{\}\) |
| \(T_{pending} \leftarrow \{\}\) |
| \(T_{finished} \leftarrow \{\}\) |
| \(D \leftarrow \{\{d_i \mid d_i \in D_j, \sigma_l \geq \sigma_f \} \mid D_j \in D\}\) |
| for \(k = 1 \text{ to } F\) do |
| \(\text{for } t_i \in T_{active} \text{ do} \) |
| \(d_{best} = \arg \max_{d \in D_k} IOU(d, \hat{d}_{t_i,k})\) |
| if \(IOU(d, \hat{d}_{t_i,k}) \geq \sigma_{IOU}\) then |
| \(K_{t_i,k} = P_{t_i,k}^{\prime}H^{T}(HP_{t_i,k}^{\prime}H^{T} + R)^{-1}\) |
| \(\hat{x}_{t_i,k} = \hat{x}_{t_i,k}^{\prime} + K_{t_i,k}(d_{best} - H\hat{x}_{t_i,k}^{\prime})\) |
| \(P_{t_i,k} = (I - K_{t_i,k}H)P_{t_i,k}^{\prime}\) |
| \(\hat{x}_{t_i,k+1} = \Phi \hat{x}_{t_i,k}\) |
| \(P_{t_i,k+1} = \Phi P_{t_i,k} \Phi^{T} + Q\) |
| \(D_f.remove(d_{best}); t_i.insert(d_{best})\) |
| end if |
| end for |
| \(\text{for } t_i \in T_{pending} \text{ do} \) |
| if \(t_i \text{ inactive for more than } \sigma_f\) then |
| \(\text{if length}(t_i) \geq \sigma_l \text{ then} \) |
| \(T_{finished}.insert(t_i)\) |
| end if |
| \(T_{pending}.remove(t_i)\) |
| end if |
| \(d_{best} = \arg \max_{d \in D_k} IOU(d, \hat{d}_{t_i,k})\) |
| if \(IOU(d, \hat{d}_{t_i,k}) \geq \sigma_{IOU}\) then |
| \(K_{t_i,k} = \hat{P}_{t_i,k}^{\prime}H^{T}(H\hat{P}_{t_i,k}^{\prime}H^{T} + R)^{-1}\) |
| \(\hat{x}_{t_i,k} = \hat{x}_{t_i,k}^{\prime} + K_{t_i,k}(d_{best} - H\hat{x}_{t_i,k}^{\prime})\) |
| \(P_{t_i,k} = (I - K_{t_i,k}H)\hat{P}_{t_i,k}\) |
| \(\hat{x}_{t_i,k+1} = \Phi \hat{x}_{t_i,k}\) |
| \(P_{t_i,k+1} = \Phi \hat{P}_{t_i,k} \Phi^{T} + Q\) |
| \(D_f.remove(d_{best}); T_{active}.insert(t_i)\) |
| end if |
| end if |
| end for |
| \(\text{end for} \) |
| \(\text{for } d_i \in D_f \text{ do} \) |
| \(t_{new} \leftarrow \{d_i\}\) |
| \(T_{active}.insert(t_{new})\) |
| end for |
| end for |

\(T_{finished} \leftarrow T_{finished} + T_{active}\)

return \(T_{finished}\)
A. DATASET

We evaluated the performance of the proposed tracker on the DETRAC benchmark [45]. Consisting of over 10 hours of dense traffic scene recordings under various weather and lighting conditions, the DETRAC benchmark targets at multi-object detection and tracking tasks. The videos were recorded at 25 FPS with a resolution of 960 by 540, and baseline detections from DPM, ACF, R-CNN and CompACT are available.

B. EXPERIMENTAL ENVIRONMENT

The performance of a tracking-by-detection algorithm is highly dependent on the upstream detector. In general, the higher the performance of the detector to generate bounding boxes in a high-quality and consistent manner, the more likely an track association algorithm will generate high quality tracks. Although more advanced detectors, including Faster R-CNN, SSD, and Mask R-CNN may lead to higher performance on the DETRAC dataset, we opt not to report them based on the belief that our goal is to compare the performance of the downstream tracking algorithm instead of the upstream detector. As the performance of a tracker is dependent on that of a tracker, using a more advanced tracker gives us an advantage over the other methods being compared. Therefore, we report results generated with detections from CompACT, EB and Mask R-CNN detector; the former is a tracker with reasonably good performance and was used by all methods across the leader board, while the current best results were generated with the latter two as of time of writing. We do not report results based on DPM, ACF and R-CNN as their detections are generally of low quality that effective track association cannot be performed.

The behavior of the proposed tracker can be controlled by a series of tunable parameters. These parameters are determined by performing a comprehensive search in the parameter space using the training data provided by the DETRAC benchmark dataset. It is necessary to tune these parameters if the algorithm is applied to a different detector, as the output from different detectors can exhibit vastly different behaviors. For example, it may be necessary to use a low \( \sigma_{\text{IOU}} \) if a detector fail to generate bounding boxes of consistent sizes, and the confidence threshold \( \sigma_{f} \) may be increased should a detector that generates many low-confidence false positive detections. Parameters used to generate the reported results are listed in Table 2.

To consider the performance of object detection and tracking jointly in evaluation, multi-object tracking literature [48]–[50] have introduced a set of performance metrics including identity switches (IDs), mostly tracked (MT), mostly lost (ML), false positives (FP), false negatives (FN), multi-object tracking accuracy (MOTA) and multi-object tracking precision (MOTP). \( \text{IDS} \) reflects the accuracy of tracks by describing the number of times that the identity of object being tracked changes in an track, and a track generated by the evaluated method is labeled as mostly tracked if it contains at least 80% of frames in the ground-truth track. Similarly, a track is labeled as mostly lost if it contains less than 20% of frames of the ground-truth. FP and FN are obtained by comparing tracker output with ground-truth on by-frame basis.

Additionally, two high-level metrics are provided. The MOTA metric provides a summary of the performance of the evaluated method by considering a collection of metrics over the evaluated video sequences:

\[
\text{MOTA} = 1 - \frac{\sum_v \sum_t (FN_{v,t} + FP_{v,t} + \text{IDS}_{v,t})}{\sum_v \sum_t GT_{v,t}}
\]

where \( FN_{v,t} \) is the number of false negative track frames, \( FP_{v,t} \) is the number of false positive track frames, \( \text{IDS}_{v,t} \) is the number of identity switches, and \( GT_{v,t} \) represents the total number of ground-truth objects at time \( t \) of sequence \( v \). The MOTP metric on the other hand is calculated as the overlap between correctly assembled tracks and the ground-truth.

The DETRAC dataset used custom versions of the metrics mentioned above, by evaluating the tracker under increments of detection confidence thresholds and integrating along the detector precision-recall curve to obtain the final metrics score. Pronouncedly, the number of detections available to the tracker will be decreased by the increasing detection confidence threshold, thus the leading performance numbers
are lower than that of other datasets of similar difficulty levels.

C. EVALUATION

We present the results generated by the KIOU tracker when evaluated on the DETRAC-Test dataset in Table 3. The algorithm is implemented in Python as a single-thread program. Per guidelines of the UA-DETRAC dataset, only the location, size, and confidence score of detection boxes in each frame is available as input to the tracker. Note that although a tracker may be evaluated with multiple detectors, only the best tracker-detector combinations are reported to the benchmark for evaluation on the DETRAC the detector will run at only 9-13 FPS. Test set.

In summary, the KIOU tracker outperforms the other available approaches in tracking accuracy (PR-MOTA), percentage mostly tracked (PR-MT), percentage mostly lost (PR-ML), and false negative detections (PR-FN). These results indicate that the proposed method maintains the highest number of valid tracks among the methods compared. The results reaffirm our observation that performance of a tracker is highly dependent on the underlying detector, as the results from the IOU tracker vary greatly depending on the detector used. Namely, the results from EB detector has a higher tracking accuracy but relatively low precision when compared with that from the R-CNN detector, which indicates that EB detector generates more detections but are generally less precise than the R-CNN detector. In this case, it is more reasonable to compare tracker performance on the same detector. On the EB detector, the KIOU tracker also achieves lower identity switches (IDs) and fragmentation (FM) than the IOU tracker, a benefit brought by the Kalman filter’s ability to continue tracks correctly under occasionally missing detections.

To further inspect the effect of the Kalman filter in track continuing, a comparison between the original IOU tracker, the V-IOU tracker, which is an extension of the IOU tracker and at the time of writing (January 2021) is the best publicly available tracker on this dataset, and our method on the DETRAC-Train dataset is presented in Table 4. Two detectors are used in the comparison: EB is evaluated in DETRAC-Test official results; and Mask R-CNN that allows overall higher performance than the former. Our method is evaluated in multiple runs with the detectors while dropping a variable portion of the frames, with the most difficult run using only 20% of the available frames. Considering that the UA-DETRAC evaluation metrics pose penalties for each missed frame, the bounding boxes in the skipped frames are generated off-line by simple linear interpolation for a fair comparison. One may claim that if a tracker is capable of tracking the location of an object at every other frame, then the true location at the skipped frames are more or less unimportant, especially when the frames are only separated by 0.04 seconds (at 25 FPS). Therefore, the tracker can be used on-line, without post-interpolation, while delivering the same level of performance as reported as the state-of-the-art.

The comparison in Table 4 shows that the KIOU tracker provides a significant performance gain in MOTA, IDs and FM metrics over the baseline method. When using detections from the EB detector, the IDs metric was reduced by 89.8%, and FM was reduced by 85.7% when compared to the IOU tracker. In comparison, the reduction in IDs and FM metric are 46.5% and 29.1%, respectively, which indicates that Mask R-CNN generally has a lower number of missing or incorrect detections compared to EB. This also suggests that with a sufficiently sophisticated detector that generates high-quality detections one may resort to simpler frame association methods in favor of higher frame rates.

Visualized results in Table 5 also show that the Kalman filter aids in reducing track fragmentation and identity switches resulted from missing detections or occlusion. We used best settings reported in [28] for the IOU tracker, and the settings reported in Table 2 for the KIOU tracker. The detector used in this visualization is the Mask R-CNN, which has the lowest number of missing or incorrect detections among all detectors evaluated in this paper. For each missing detection caused by occlusion, the IOU tracker must initiate a new track when the object is detected again in the following frames away from its last known location, while the KIOU tracker can estimate and predict the location of the object, thus increasing the likelihood of continuing the track.

Another notable observation from the test results is that the performance of our method is not significantly impacted by the reduced number of frames available. Skipping 1/2 or 2/3 of the frames offers a significant performance gain in effective frame rates at a cost of minor performance down in MT, ML and MOTA metrics. Some other metrics, most significantly the IDs and FM actually benefits from skipped frames. The most probable cause for this performance improvement is that some erroneous/misleading detections are skipped among the majority of frames.

The above observations do not necessarily suggest that an unlimited portion of frames can be skipped. Skipping increasing number of frames offers diminishing performance improvement but in turn magnifies performance deterioration in terms of tracking accuracy and other major metrics. Most notably, the overlap threshold, which is the primary means for associating frames, must be lowered to accommodate increasing number of missing frames. From Table 2 one may observe that for Mask R-CNN detections, the $\delta_{IOU}$ threshold is decreased to as low as 0.1 from the normal threshold of 0.6 when no frames are skipped. Continuing reducing number of frames input may prevent the tracker from making any correct track associations. Judging from the results from Table 4, the KIOU tracker outperforms the baseline in most metrics when up to 2/3 or 3/4 frames are dropped from results of EB and Mask R-CNN detectors, respectively, and dropping more frames beyond that may result in a severe performance loss, most significantly in mostly tracked (MT) and tracking accuracy (MOTA) metrics.

Although highly dependent on the nature of the data, it appears that dropping 1/2 or 2/3 frames is a reasonable choice
computational cost in real-time operation. This high-speed processing on the videos within the DETRAC dataset. EB detector is selected as it is relatively light-weight marker-less tracking approach may prove suitable for many use cases in manufacturing, traffic, logistics, among others, where tracking are generally unable to reliably process missing detections. At this configuration the proposed tracker can deliver the reported performance at 25-36 FPS, even though the frames. At this configuration the proposed tracker can deliver the reported performance at 25-36 FPS, even though the detector will run at only 9-13 FPS. The detector will run at only 9-13 FPS.

### IV. CONCLUSION

Simple track association approaches by spatial-temporal overlap measure provide decent tracking performance at high speed but are generally unable to reliably process missing detections. We present an innovative method to perform track associations under missing detections through Kalman filter. We further exploit this characteristic by manually introducing missing frames in favor for significantly higher processing speed. Our approach outperforms the state-of-the-art on the evaluated tracking dataset while using up to 80% less computational cost in real-time operation. This high-speed marker-less tracking approach may prove suitable for many use cases in manufacturing, traffic, logistics, among others, where tracking are predominantly performed with the use of markers.

### REFERENCES

[1] J. Song, C. Haas, C. Caldas, E. Ergen, and B. Akinci, “Automating pipe spool tracking in the supply chain,” Automation in Construction, vol. 15, no. 2, pp. 166–177, 2006.
[2] T. Cheng, M. Venugopal, J. Teizer, and P. Vela, “Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments,” Automation in Construction, vol. 20, no. 8, pp. 173–189, 2011.
[3] K. Kim, J. Jun, S. Kim, and B. Y. Sung, “Medical asset tracking application with wireless sensor networks,” in 2008 Second International Conference on Sensor Technologies and Applications (sensortcomm 2008), pp. 531–536, IEEE, 2008.
[4] S. E. Li, G. Li, J. Yu, C. Liu, B. Cheng, J. Wang, and K. Li, “Kalman filter-based tracking of moving objects using linear ultrasonic sensor array for road vehicles,” Mechanical Systems and Signal Processing, vol. 98, pp. 173–189, 2018.
[5] I. Brilakis, M.-W. Park, and G. Jog, “Automated vision tracking of project related entities,” Advanced Engineering Informatics, vol. 25, no. 4, pp. 713–724, 2011.
[6] J. Gong and C. H. Caldas, “Data processing for real-time construction site spatial modeling,” Automation in Construction, vol. 17, no. 5, pp. 526–536, 2008.
[7] L. Jia and R. J. Radke, “Using time-of-flight measurements for privacy-preserving tracking in a smart room,” IEEE Transactions on Industrial Informatics, vol. 10, no. 1, pp. 689–696, 2013.
[8] Y. Fang, J. Chen, Y. Cho, and P. Zhang, “A point cloud-vision hybrid approach for 3d location tracking of mobile construction assets,” in 33rd International Symposium on Automation and Robotics in Construction

---

**TABLE 3.** Comparison of proposed method and other tracking-by-detection methods on DETRAC-Test set.

| Tracker Detector | PR-MOTA | PR-MOTP | PR-MT | PR-ML | PR-IDs | PR-FM | PR-FP | PR-FN | Speed (fps) |
|------------------|---------|---------|--------|--------|--------|-------|-------|-------|-------------|
| CEM CompACT      | 5.1%    | 35.2%   | 3.0%   | 35.3%  | 267.9  | 352.3 | 12341.2 | 260390.4 | 4.62        |
| CMOT CompACT     | 12.6%   | 36.1%   | 16.1%  | 18.6%  | 285.3  | 1516.8 | 57885.9 | 167110.8 | 3.79        |
| GOG CompACT      | 14.2%   | 37.0%   | 13.9%  | 19.9%  | 3334.6 | 3172.4 | 32092.9 | 180183.8 | 390         |
| DCT R-CNN        | 11.7%   | 38.0%   | 10.1%  | 22.8%  | 758.7  | 742.9  | 336561.2 | 210855.6 | 0.71        |
| H^2T CompACT     | 12.4%   | 35.7%   | 14.8%  | 19.4%  | 852.2  | 1117.2 | 51765.7 | 173899.8 | 3.02        |
| IHTLS CompACT    | 11.1%   | 36.8%   | 13.8%  | 19.9%  | 953.6  | 3556.9 | 14796.5 | 171806.8 | 19.79       |
| IOU R-CNN        | 16.0%   | 38.3%   | 13.8%  | 20.7%  | 5029.4 | 5795.7 | 22535.1 | 193041.9 | 100840      |
| IOU EB           | 19.4%   | 28.9%   | 17.7%  | 18.4%  | 2311.3 | 2445.9 | 14796.5 | 171806.8 | 6902        |
| KIOU EB          | 21.1%   | 28.6%   | 21.9%  | 17.6%  | 462.2  | 712.1  | 19046.8 | 159178.3 | 160.85      |

**TABLE 4.** Comparison of the results of the KIOU tracker with that of the IOU and V-IOU trackers on DETRAC-Train dataset.

| Tracker Detectors | PR-MOTA | PR-MOTP | PR-MT | PR-ML | PR-IDs | PR-FM | PR-FP | PR-FN | PR-MOTA | PR-MOTP |
|-------------------|---------|---------|--------|--------|--------|-------|-------|-------|---------|---------|
| IOU All | 32.34% | 12.88% | 20.93% | 7958.8 | 163739.9 | 4129.4 | 4221.9 | 35.8 | 40.8 |
| KIOU 1/2 All | 34.5% | 9.5% | 22.1% | 6803.5 | 155556.0 | 472.8 | 596.9 | 38.0 | 40.9 |
| KIOU 1/3 All | 31.9% | 10.9% | 23.4% | 7611.4 | 160566.9 | 483.2 | 630.9 | 37.0 | 40.9 |
| KIOU 1/4 All | 20.8% | 12.6% | 32.8% | 11163.4 | 192857.4 | 628.1 | 711.1 | 30.8 | 40.9 |
| KIOU 1/2 V-IOU All | 37.22% | 11.46% | 24.24% | 50096.9 | 171714.1 | 1022.0 | 929.5 | 34.4 | 45.4 |
| KIOU 1/4 V-IOU All | 38.32% | 8.36% | 26.24% | 50294.8 | 174052.0 | 448.2 | 293.7 | 34.0 | 40.7 |
| KIOU 1/2 IOU All | 39.10% | 12.88% | 20.93% | 7958.8 | 163739.9 | 4129.4 | 4221.9 | 35.8 | 40.8 |
| KIOU 1/4 IOU All | 42.87% | 9.5% | 22.1% | 6803.5 | 155556.0 | 472.8 | 596.9 | 38.0 | 40.9 |
| KIOU 1/2 V-IOU All | 39.10% | 11.46% | 24.24% | 50096.9 | 171714.1 | 1022.0 | 929.5 | 34.4 | 45.4 |
| KIOU 1/4 V-IOU All | 38.32% | 8.36% | 26.24% | 50294.8 | 174052.0 | 448.2 | 293.7 | 34.0 | 40.7 |
| KIOU 1/2 IOU All | 39.10% | 11.46% | 24.24% | 50096.9 | 171714.1 | 1022.0 | 929.5 | 34.4 | 45.4 |
| KIOU 1/4 IOU All | 38.32% | 8.36% | 26.24% | 50294.8 | 174052.0 | 448.2 | 293.7 | 34.0 | 40.7 |
TABLE 5. Visual Comparison of the output of the KIOU tracker with that of the IOU tracker on DETRAC-Train dataset. Continuous estimation of the target’s dynamic model aids in continuing the track in cases of missing detections or occlusion.

| Tracker | Frame # |
|---------|---------|
| Sequence #40152 | 31       |
|          | 37       |
|          | 40       |
| IOU     |         |
| KIOU    |         |

| Sequence #40161 | 252       |
|                | 254       |
|                | 256       |
| IOU            |         |
| KIOU           |         |

(ISARC 2016), Proceedings of the International Symposium on Automation and Robotics in Construction, vol. 33, pp. 1–7, 2016.

[9] R. J. Fontana, E. Richley, and J. Barney, “Commercialization of an ultra wideband precision asset location system,” in IEEE Conference on Ultra Wideband Systems and Technologies, 2003, pp. 369–373, IEEE, 2003.

[10] N. M. Scala and J. A. Pazour, “A value model for asset tracking technology to support naval sea-based resupply,” Engineering Management Journal, vol. 28, no. 2, pp. 120–130, 2016.

[11] A. Haddad, J. C. Dugger, and H. Lee, “Manufacturing control, asset tracking, and asset maintenance: Assessing the impact of RFID technology adoption,” Journal of International Technology and Information Management, vol. 24, no. 2, pp. 35–III, 2015.

[12] A. Alrasheed, A. S. Atkins, and R. Campion, “Developing a guideline for hospital tracking and monitoring systems evaluation,” in Proceedings of the 2018 International Conference on Computers in Management and Business, pp. 74–78, ACM, 2018.

[13] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, pp. 1440–1448, 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, pp. 91–99, 2015.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in Computer Vision (ICCV), 2017 IEEE International Conference on, pp. 2980–2988, IEEE, 2017.

[16] K. Chen, X. Song, X. Zhai, B. Zhang, B. Hou, and Y. Wang, “An integrated deep learning framework for occluded pedestrian tracking,” IEEE Access, vol. 7, pp. 26060–26072, 2019.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision, pp. 21–37, Springer, 2016.

[18] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440, 2015.

[19] M. A. Zulkifley and N. Trigoni, “Multiple-model fully convolutional neural networks for single-object tracking on thermal infrared video,” IEEE Access, vol. 6, pp. 42790–42799, 2018.

[20] M. Neuhausen, J. Teizer, and M. König, “Construction worker detection and tracking in bird’s-eye view camera images,” in ISARC, Proceedings of the International Symposium on Automation and Robotics in Construction, vol. 35, pp. 1–8, IAARC Publications, 2018.

[21] B. Zhang, Z. Zhu, A. Hammad, and W. Aly, “Automatic matching of construction onsite resources under camera views,” Automation in Construction, vol. 91, pp. 206–215, 2018.

[22] E. Konstantinou and I. Brilakis, “Matching construction workers across views for automated 3D vision tracking on-site,” Journal of Construction Engineering and Management, vol. 144, no. 7, p. 04018061, 2018.

[23] A. Andriyenko and K. Schindler, “Multi-target tracking by continuous energy minimization,” in CVPR 2011, pp. 1265–1272, IEEE, 2011.

[24] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li, “Multiple target tracking based on undirected hierarchical relation hypergraph,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1283–1289, 2014.

[25] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468, IEEE, 2016.

[26] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking
with a deep association metric,” in 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649, IEEE, 2017.

[27] E. Bochinski, V. Eislein, and T. Sikora, “High-speed tracking-by-detection without using image information,” in International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017, (Lecce, Italy), Aug. 2017.

[28] E. Bochinski, T. Senst, and T. Sikora. “Extending iou based multi-object tracking by visual information,” in IEEE International Conference on Advanced Video and Signals-based Surveillance, (Auckland, New Zealand), pp. 441–446, Nov. 2018.

[29] Y. Ban, S. Ba, X. Alameda-Pineda, and R. Horaud. “Tracking multiple persons based on a variational bayesian model,” in European Conference on Computer Vision, pp. 52–67, Springer, 2016.

[30] S.-H. Bae and K.-J. Yoon. “Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking,” IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 5, pp. 595–610, 2018.

[31] S. Li, S. Zhao, B. Cheng, E. Zhao, and J. Chen. “Lightweight particle filter for robust visual tracking,” IEEE Access, vol. 6, pp. 32310–32320, 2018.

[32] Y. Wang, X. Ban, H. Wang, X. Li, Z. Wang, D. Wu, Y. Yang, and S. Liu. “Particle filter vehicles tracking by fusing multiple features,” IEEE Access, vol. 7, pp. 133694–133706, 2019.

[33] Z. Liu, S. Chen, H. Wu, and K. Chen. “Robust student’s mixture probability hypothesis density filter for multi-target tracking with heavy-tailed noises,” IEEE Access, vol. 6, pp. 39208–39219, 2018.

[34] Z. Liu, L. Ji, F. Yang, X. Qu, Z. Yang, and D. Qin. “Cubature information gaussian mixture probability hypothesis density approach for multi extended target tracking,” IEEE Access, vol. 7, pp. 103678–103692, 2019.

[35] N. L. Baisa. “Online multi-object visual tracking using a gm-phd filter with deep appearance learning,” in 2019 22nd international conference on information fusion (FUSION), pp. 1–8, IEEE, 2019.

[36] Q. Zhu, T. Li, J. Pan, and Q. Bao. “The modified probability hypothesis density filter with adaptive birth intensity estimation for multi-target tracking in low detection probability,” IEEE Access, vol. 8, pp. 43690–43710, 2020.

[37] R. E. Kalman. “A new approach to linear filtering and prediction problems,” Journal of basic Engineering, vol. 82, no. 1, pp. 35–45, 1960.

[38] J. B. Pearson and E. B. Stear. “Kalman filter applications in airborne radar tracking,” IEEE Transactions on Aerospace and Electronic Systems, no. 3, pp. 319–329, 1974.

[39] V. J. Aidala. “Kalman filter behavior in bearings-only tracking applications,” IEEE Transactions on Aerospace and Electronic Systems, vol. 1, pp. 29–39, 1979.

[40] Y. Chan, A. Hu, and J. Plant. “A kalman filter based tracking scheme with input estimation,” IEEE transactions on Aerospace and Electronic Systems, no. 2, pp. 237–244, 1979.

[41] G. M. Siouris, G. Chen, and J. Wang. “Tracking an incoming ballistic missile using an extended interval kalman filter,” IEEE Transactions on Aerospace and Electronic Systems, vol. 33, no. 1, pp. 232–240, 1997.

[42] S.-K. Weng, C.-M. Kuo, and S.-K. Tu. “Video object tracking using adaptivekalman filter,” Journal of Visual Communication and Image Representation, vol. 17, no. 6, pp. 1190–1208, 2006.

[43] X. Li, K. Wang, W. Wang, and Y. Li. “A multiple object tracking method using kalman filter,” in The 2010 IEEE international conference on information and automation, pp. 1862–1866, IEEE, 2010.
CHENHUI SHAO is an assistant professor in the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign. He received his B.E. degree in Automation from the University of Science and Technology of China; M.S.E. degree in Industrial and Operations Engineering, M.A. degree in Statistics, and Ph.D. degree in Mechanical Engineering, all from the University of Michigan, Ann Arbor. His research interests focus on smart manufacturing, machine learning, statistics, big data analytics in manufacturing, and materials joining. He is a recipient of the National Science Foundation CAREER Award, ASME Chao and Trigger Young Manufacturing Engineer Award, IIEE Manufacturing and Design (M&D) Division Outstanding Young Investigator Award, SME Barbara M. Fossum Outstanding Young Manufacturing Engineer Award, SME 30 Under 30 Award, and multiple best paper awards.

***