

A Hybrid Method for Real-Time Vehicle Tracking from a Moving Camera with YOLO and KLT

Miroslav R. Tomov¹, Ivan G. Garvanov^{1, 2}

¹ University of Library Science and Information Technology, Sofia, Bulgaria

² Institute of Mathematics and Informatics, Sofia, Bulgaria

Emails: m.tomov@ips.bas.bg, i.garvanov@unibit.bg

Abstract: Deep neural detectors such as YOLO reliably detect vehicles, but running them on every frame is computationally expensive for real-time CPU performance. We propose a hybrid tracking method that requires no additional training. The detector operates only on every K keyframe, while intermediate bounding boxes are propagated using Kanade-Lucas-Tomasi (KLT) feature point tracking with forward-backward error validation. Since tracked points belong directly to the vehicles, the method inherently handles camera movement without requiring prior video stabilization. Implemented in MATLAB using YOLOv3, the approach was evaluated on a challenging night video featuring wet asphalt, headlight reflections, and camera displacement up to 190 px. At a detection interval of $K = 5$, the hybrid method achieves MOTA 0.73 and IDF1 0.76 (compared to 0.69 and 0.74 for full-frame YOLOv3), reduces identity switches from 55 to 13, and yields a nearly 4-fold speedup (9.3 fps vs. 2.4 fps). At $K = 15$, KLT maintains an IDF1 of 0.62, outperforming simple frame holding (0.38).

Keywords: *KLT tracking, YOLOv3, forward-backward error, key frame detection, moving camera, vehicle tracking.*

1. Introduction

Object detectors from the YOLO family [1-4] provide high accuracy and reliability in vehicle detection; however, their high computational complexity on standard central processing units (CPUs) leads to significant latency (hundreds of

milliseconds per frame), preventing real-time operation. An effective solution to this problem is temporal detection sparsification (executing the detector at specific frame intervals) and filling the intermediate gaps with lighter, faster tracking methods.

Popular algorithms such as SORT [5, 6] and DeepSORT [7], [8] associate detections using a Kalman filter and the Hungarian algorithm [9], [10], but require running the neural network on every single frame. On the other hand, approaches such as Deep Feature Flow [11], [12] and Detect or Track [13], [14] trigger the detector at larger intervals but rely on additional deep learning modules that require resource-intensive training.

In this work, we propose a novel hybrid tracking method that requires no additional training or video stabilization. The detector is executed only on designated keyframes, while intermediate bounding boxes are propagated by tracking feature points using the Lucas–Kanade (KLT) method [15]. To eliminate unreliable points, forward-backward error validation is applied [16]. Since the tracked points belong directly to the vehicle surfaces, the method inherently accounts for both target motion and ego-motion of the sensor, making it fully applicable to mobile camera scenarios.

2. Methodology and System Workflow

The proposed hybrid method processes the video sequence frame-by-frame at half resolution across four operational phases (Fig. 1).

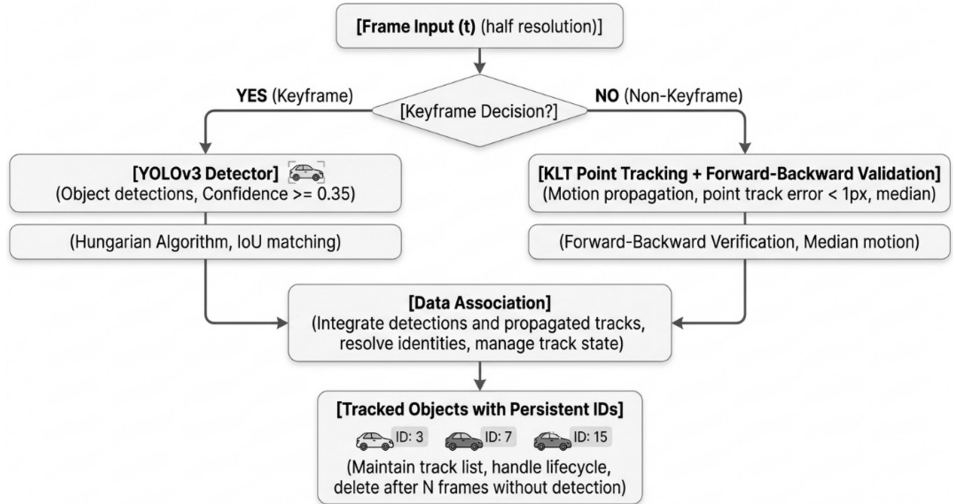


Fig. 1. Block diagram of the hybrid method: YOLOv3 operates only on keyframes, while KLT propagates the bounding boxes between them.

2.1. Keyframe Detection

At every K frame, the YOLOv3 detector identifies target objects (cars, motorcycles, buses, and trucks) with a confidence threshold above 0.35. Partial bounding boxes (e.g., detecting only headlights) are filtered out.

2.2. KLT Bounding Box Propagation

On intermediate frames, up to 40 feature points are selected within the inner region of each bounding box and tracked bidirectionally (forward-backward). Only points passing the consistency check are retained. The median displacement and scale factor of the valid points determine the updated position and size of the bounding box.

2.3. Data Association

On keyframes, the propagated bounding boxes are associated with new YOLOv3 detections using the Hungarian algorithm based on Intersection over Union (IoU). Upon successful matching, the bounding box is replaced by the new detection, resetting any accumulated KLT tracking drift.

2.4. Target Lifecycle Management

Unassociated detections instantiate new tracklets. Targets missed by the detector continue to be propagated passively via KLT while preserving their unique track IDs. If a target remains undetected for 45 consecutive frames (~ 1.5 s), it is terminated.

3. Mathematical Formulation

Let $t \in N$ denote the discrete frame index, and let $K \in N^+$ represent the detector interval. The set of keyframes $\mathcal{T}_K$ is defined as:

$$\mathcal{T}_K = \{t \in N | (t - 1) \bmod K = 0\} \quad (1)$$

For each tracked target j , up to $N_{max} = 40$ feature points p_i are extracted within the central 80% ROI of its bounding box using the Shi–Tomasi corner detector [17]. Points are tracked into the subsequent frame using a pyramidal Lucas–Kanade tracker (3 scale levels, 15×15 pixel window) [15].

To prune unstable vectors, a forward-backward error validation is applied [16]. A point p_i^{t-1} is deemed valid if its cyclic discrepancy e_i satisfies:

$$e_i = \left\| p_i^{t-1} - \text{LK}_{t \rightarrow t-1} \left(\text{LK}_{t-1 \rightarrow t} (p_i^{t-1}) \right) \right\| < 1 \text{ px} \quad (2)$$

where $LK_{a \rightarrow b}$ represents the optical flow mapping operator from frame a to frame b . The spatial displacement d_j and scale adjustment factor s_j for target j are derived from the valid point set via median aggregation:

$$d_j = \text{median}_i(p_i^t - p_i^{t-1}) \quad (3)$$

$$s_j = \text{clip}_{[0.95, 1.05]} \left(\text{median}_{i < l} \frac{\|p_i^t - p_i^{t-1}\|}{\|p_i^{t-1} - p_i^{t-2}\|} \right) \quad (4)$$

The center coordinates c_j^t and dimensions (w_j^t, h_j^t) of the bounding box are updated as follows:

$$c_j^t = c_j^{t-1} + d_j, \quad (w_j^t, h_j^t) = s_j (w_j^{t-1}, h_j^{t-1}) \quad (5)$$

If the number of inlier points n_j falls below $n_{min} = 6$, the bounding box is translated using its last known velocity vector.

On keyframes ($t \in \mathcal{T}_K$), data association between propagated bounding boxes B_j and new detections D_k is performed using the Hungarian algorithm [10] based on Intersection over Union (IoU):

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

$$\pi^* = \arg \max_{\pi} \sum_{(j,k) \in \pi} \text{IoU}(B_j, D_k), \text{ subject to } \text{IoU}(B_j, D_k) \geq 0.3 \quad (7)$$

An unassociated detection D_k instantiates a new target if and only if it does not significantly overlap with existing matched detections M :

$$\text{New Tracklet from } D_k \Leftrightarrow \max_{l \in M} \text{IoU}(D_k, D_l) \leq 0.6 \quad (8)$$

Target visibility $v_j^t \in \{0, 1\}$ and termination are governed by the elapsed frames since the last detection a_j :

$$v_j^t = 1 \Leftrightarrow n_j^t \geq 6 \wedge a_j < K \quad (9)$$

A tracklet is terminated when $a_j = 45$ consecutive frames elapse without detection matching.

4. MATLAB Implementation

The method is implemented using the Computer Vision Toolbox. Equation (2) is embedded into `vision.PointTracker` via the `MaxBidirectionalError` property. Listing 1 implements (2)–(5), while data association (7) is reduced to `matchpairs` (Listing 2).

Listing 1. MATLAB function for bounding box propagation, equations (2)–(5):

```
function [Bn, ok] = propagateKLT(g0, g1, B, P, tracker)
N = size(B, 1); Bn = B; ok = false(N, 1); [H, W] = size(g0);
pts = zeros(0, 2, 'single'); owner = zeros(0, 1);
for k = 1:N
    lo = max(floor(B(k,1:2) + 0.1*B(k,3:4)), 1);
    hi = min(floor(B(k,1:2) + 0.9*B(k,3:4)), [W H]);
    if any(hi - lo < 6), continue; end
    c = detectMinEigenFeatures(g0, 'MinQuality', 0.01, 'ROI', [lo hi-lo]);
    c = selectStrongest(c, P.maxPts);
    pts = [pts; c.Location]; owner = [owner; repmat(k, c.Count, 1)];
end
if isempty(pts), return; end
release(tracker); initialize(tracker, pts, g0);
[p1, valid] = step(tracker, g1);
for k = 1:N
    s = (owner == k) & valid;
    if nnz(s) < P.minPts, continue; end
    a0 = double(pts(s, :)); a1 = double(p1(s, :));
    d = median(a1 - a0, 1); % (3)
    [i, l] = find(triu(true(size(a0, 1)), 1));
    d0 = hypot(a0(i,1) - a0(l,1), a0(i,2) - a0(l,2));
    d1 = hypot(a1(i,1) - a1(l,1), a1(i,2) - a1(l,2));
    sc = 1; q = d0 > 5;
    if nnz(q) > 3, sc = median(d1(q) ./ d0(q)); end
    sc = min(max(sc, 0.95), 1.05); % (4)
    c = B(k,1:2) + B(k,3:4)/2 + d; wh = B(k,3:4) * sc; % (5)
    Bn(k, :) = [c - wh/2, wh]; ok(k) = true;
end
end
```

Listing 2. MATLAB function for keyframe data association, equation (7):

```
function pairs = associate(Pb, D, iouMin)
M = bboxOverlapRatio(Pb, D);
cost = 1 - M; cost(M < iouMin) = 1e6;
pairs = matchpairs(cost, 1e5);
pairs = pairs(M(sub2ind(size(M), pairs(:,1), pairs(:,2))) >= iouMin, :);
end
```

5. Experimental Results and Analysis

The proposed hybrid tracking framework was evaluated on the high-definition test sequence IMG_2942 (1920×1080 resolution, 30 frames per second, 455 frames total). The scene presents challenging nighttime urban tracking conditions, including wet pavement reflections, acute headlight glare, and substantial handheld camera ego-motion resulting in cumulative spatial displacement up to 190 pixels. The primary object detection backbone employed in the experiments is YOLOv3 [2] (darknet53-coco). To ensure objective evaluation in the absence of manual bounding box annotations, pseudo-ground-truth trajectories were constructed using a higher-capacity ensemble model based on YOLOX-X [18] executed over full-resolution frames and overlapping spatial tiles. A total of

16 verified moving vehicle trajectories were established as ground truth, while static parked vehicles were excluded from error accounting.

Quantitative evaluation follows standard Multiple Object Tracking (MOT) metrics, including Multiple Object Tracking Accuracy (MOTA) [19], Identification F1-Score (IDF1) [20], Identity Switches (IDSW), Precision (P), Recall (R), and Track Initialization Latency. Comparative experiments were conducted against two baseline propagation strategies: zero-order hold (frame holding) and constant-velocity motion extrapolation across keyframe intervals.

5.1. Performance Comparison Across Keyframe Intervals

Table 1 summarizes the performance metrics across varying keyframe detection intervals K . At full-frame execution ($K = 1$), YOLOv3 exhibits significant tracking jitter and false identity splits, resulting in 49 identity switches due to bounding box instability and inter-object merging. Integrating the bidirectional KLT point tracker significantly stabilizes bounding box transitions. At an optimal interval of $K = 5$, the proposed method achieves the highest overall precision and identity tracking accuracy, yielding a MOTA of 0.74 and IDF1 of 0.78, while drastically reducing identity switches from 49 to 13.

Table 1. Quantitative performance comparison under varying keyframe intervals K (Ground Truth: YOLOX-X)

Tracking Strategy	Detector Executions	MOTA	IDF1	P	R	IDSW	Latency (frames)
Full-frame YOLOv3 ($K = 1$)	455	0.69	0.75	0.92	0.79	49	8.1
Hybrid KLT ($K = 3$)	152	0.73	0.76	0.95	0.79	16	13.3
Proposed Hybrid ($K = 5$)	91	0.74	0.78	0.96	0.79	13	11.5
Hybrid KLT ($K = 15$)	31	0.62	0.62	0.92	0.68	12	16.8
Frame Holding Baseline ($K = 15$)	31	0.40	0.38	0.88	0.49	40	14.6
Velocity Extrapolation ($K = 15$)	31	0.49	0.54	0.91	0.57	28	14.0

As the detection sparsity increases to $K = 15$, the superiority of feature-point optical flow tracking over kinematic baselines becomes pronounced. While frame holding and velocity prediction suffer catastrophic performance degradation (IDF1 dropping to 0.38 and 0.54, respectively), the proposed bidirectional KLT

tracker maintains an IDF1 of 0.62 by accurately accommodating non-linear vehicle trajectories and uncompensated camera rotation.

5.2. Component Ablation and Computational Efficiency

To evaluate individual design choices, an ablation study was performed at $K = 5$. Omitting the forward-backward error validation mechanism increases identity switches from 13 to 14 and decreases IDF1 from 0.778 to 0.773, confirming the utility of cyclic point pruning. Supplementary modules, such as background homography stabilization (IDF1 = 0.748) and appearance-based Re-ID feature memory (IDF1 = 0.741), led to degraded overall tracking continuity due to perspective distortion and headlight reflections, justifying their exclusion from the final architecture. A baseline using purely frame differencing without neural detection achieved a meager recall of 0.28 accompanied by a high false alarm rate of 0.44.

Table 2. Computational latency and throughput benchmark in MATLAB (Intel i7 CPU)

Pipeline Configuration	MOTA	IDF1	IDSW	Per-Frame Latency (ms)	Throughput (FPS)
Full YOLOv3 Execution ($K = 1$)	0.69	0.74	55	425	2.4
Proposed Hybrid Method ($K = 5$)	0.73	0.76	13	107	9.3
Hybrid + Homography + Re-ID	0.74	0.73	14	167	6.0

Computational benchmarks executed in MATLAB demonstrate a near 4-fold increase in processing throughput, elevating frame rates from 2.4 FPS to 9.3 FPS at $K=5$. A single neural inference pass consumes 325 ms, whereas the lightweight KLT propagation phase requires only 41 ms per frame. The primary operational trade-off is a moderate increase in target initialization latency (11.5 frames vs. 8.1 frames), corresponding to the keyframe sampling window required for newly entering vehicles.

6. Limitations

The evaluated hybrid tracking framework exhibits several limitations that outline key directions for future research:

Experimental Dataset Scope: Validation is currently constrained to a single benchmark sequence featuring 16 moving vehicles. Furthermore, the pseudo-ground-truth labels generated via YOLOX-X occasionally fail to resolve closely

space-adjacent targets. Comprehensive validation requires manual annotations on broader standardized benchmarks, such as UA-DETRAC [21].

Backbone Detection Upper Bound: Tracking recall is fundamentally upper-bounded by the primary detection network. Since full-frame YOLOv3 achieves a maximum recall of 0.79 on this dataset, missed detections directly limit overall target recovery regardless of KLT tracking accuracy.

Target Initialization Latency: Sparsifying the detection interval ($K < 1$) inherently introduces an initialization delay of up to $K-1$ frames for newly entering objects before their initial tracklet instantiation.

Comparative Baseline Coverage: The framework was evaluated strictly against kinematic motion baselines (zero-order hold and velocity extrapolation). Direct benchmarking against state-of-the-art multi-object trackers, such as SORT [6] and DeepSORT [8], remains to be conducted.

7. Conclusion

This paper presented a novel hybrid vehicle tracking framework designed for real-time CPU execution by coupling sparse neural network detections with bidirectional KLT feature point tracking. The proposed method requires neither supplementary offline training nor prior video stabilization, rendering it inherently robust to camera ego-motion in mobile or handheld recording scenarios.

Experimental validation on a challenging nighttime video benchmark demonstrated that operating at an optimal keyframe interval of $K=5$ achieves a nearly 4-fold throughput enhancement (rising from 2.4 FPS to 9.3 FPS) compared to full-frame detection. Concurrently, the framework yields superior tracking fidelity: MOTA increases from 0.69 to 0.73, IDF1 improves from 0.74 to 0.76, and identity switches are reduced by over 75% (from 55 down to 13).

Future research will focus on evaluating the framework across broader standardized datasets (such as UA-DETRAC) with manual ground-truth annotations, alongside direct comparative benchmarking against state-of-the-art multi-object trackers like SORT and DeepSORT.

Acknowledgement

This work is supported by the Bulgarian National Science Fund, Project title “Innovative Methods and Algorithms for Detection and Recognition of Moving Objects by Integration of Heterogeneous Data”, KP-06-N 72/4/05.12.2023.

Reference

1. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016). <https://doi.org/10.1109/CVPR.2016.91>.

2. Garvanova, M., Ivanov, V.: Quality assessment of defocused image recovery algorithms. In: 3rd International Conference on Sensors, Signal and Image Processing (SSIP 2020), ACM International Conference Proceeding Series, pp. 25–30 (2020). <https://doi.org/10.1145/3441233.3441242>.
3. Garvanova, M., Ivanov, V.: Quality assessment of image deburring algorithms. IOP Conference Series: Materials Science and Engineering, vol. 1031(1), 012051 (2021). <https://doi.org/10.1088/1757-899X/1031/1/012051>.
4. Redmon, J., Farhadi, A.: YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018).
5. Shishkov, B., Garvanova, G.: A review of pilotless vehicles. In: Shishkov, B., Lazarov, A. (eds.) Telecommunications and Remote Sensing (ICTRS 2023), Communications in Computer and Information Science, vol. 1990, pp. 138–151. (2023). https://doi.org/10.1007/978-3-031-49263-1_11.
6. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: IEEE International Conference on Image Processing (ICIP), pp. 3464–3468 (2016). <https://doi.org/10.1109/ICIP.2016.7533003>.
7. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: IEEE International Conference on Image Processing (ICIP), pp. 3645–3649 (2017). <https://doi.org/10.1109/ICIP.2017.8296962>.
8. Garvanov, I., Kabachiev, C., Behar, V., Garvanova, M.: The experimental study of possibility for pulsar signal detection. In: 2nd International Conference on Engineering & Telecommunications (En&T 2016), Moscow-Dolgoprudny, Russia, pp. 68–71 (2016). <https://doi.org/10.1109/EnT.2016.0213>.
9. Tsonkov, G., Garvanova, G., Garvanov, I., Garvanova, M.: Software architecture for object detection in images based on color features with integrated artificial intelligence. In: Shishkov, B. (ed.) Business Modeling and Software Design (BMSD 2024), Lecture Notes in Business Information Processing, vol. 523, pp. 267–282. (2024). https://doi.org/10.1007/978-3-031-64073-5_18.
10. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1–2), 83–97 (1955). <https://doi.org/10.1002/nav.3800020109>.
11. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2349–2358 (2017). <https://doi.org/10.1109/CVPR.2017.252>.
12. Kabachiev, C., Garvanov, I., Doukovska, L., Kyovtorov, V., Rohling, H.: Data association algorithm in multiradar system. In: IEEE Radar Conference (RADAR 2008), Rome, Italy, pp. 1771–1774 (2008). <https://doi.org/10.1109/RADAR.2008.4721043>.
13. Garvanov, I., Garvanova, M., Borissova, D., Garvanova, G.: A model of a multi-sensor system for detection and tracking of vehicles and drones. In: Shishkov, B. (ed.) Business Modeling and Software Design (BMSD 2023), Lecture Notes in Business Information Processing, vol. 483, pp. 299–307. (2023). https://doi.org/10.1007/978-3-031-36757-1_21.

14. Luo, H., Xie, W., Wang, X., Zeng, W.: Detect or track: Towards cost-effective video object detection/tracking. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33(01), pp. 8803–8810 (2019). <https://doi.org/10.1609/aaai.v33i01.33018803>.
15. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), vol. 2, pp. 674–679 (1981).
16. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: Automatic detection of tracking failures. In: 20th International Conference on Pattern Recognition (ICPR), pp. 2756–2759 (2010). <https://doi.org/10.1109/ICPR.2010.675>.
17. Shi, J., Tomasi, C.: Good features to track. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 593–600 (1994). <https://doi.org/10.1109/CVPR.1994.323794>.
18. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: Exceeding YOLO series in 2021. arXiv preprint arXiv:2107.08430 (2021).
19. Bernardin, K., Stiefelwagen, R.: Evaluating multiple object tracking performance: The CLEAR MOT metrics. EURASIP Journal on Image and Video Processing, vol. 2008, 246309 (2008). <https://doi.org/10.1155/2008/246309>.
20. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: Hua, G., Jégou, H. (eds.) Computer Vision – ECCV 2016 Workshops, Lecture Notes in Computer Science, vol. 9914, pp. 17–35. (2016). https://doi.org/10.1007/978-3-319-48881-3_2.
21. Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.-C., Qi, H., Lim, J., Yang, M.-H., Lyu, S.: UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. Computer Vision and Image Understanding 193, 102907 (2020). <https://doi.org/10.1016/j.cviu.2020.102907>.