Offline Tracklet Merging for Robust Multi-Object Tracking: A Hierarchical and Hybrid Approach

Hariohm Bhatt	Meet Rathi	Aditya Agrawal	Harsh Panchal	Jeel Kadivar
B.Tech CSE	*B.Tech CSE*	*B.Tech CSE*	*B.Tech CSE*	*B.Tech CSE*
SEAS, AU	*SEAS, AU*	*SEAS, AU*	*SEAS, AU*	*SEAS, AU*
AU2240085	AU2240106	AU2240153	AU2240160	AU2140181

***Abstract*—This paper presents a novel offline tracklet merging approach that integrates graph-based hierarchical methods with classical tracking algorithms, including the Kalman Filter and feature-based matching techniques. Leveraging prominent datasets such as VisDrone, our method addresses critical challenges related to occlusion, fragmented detections, and variations in object scale. Experimental results demonstrate significant improvements in tracking continuity and robustness in complex environments.**

***Index Terms*—Multi-Object Tracking, Tracklet Merging, Hierarchical Methods, Kalman Filter, SIFT, KD-Tree, VisDrone.**

I. Introduction

Multi-object tracking (MOT) is a cornerstone of contemporary computer vision, underpinning applications ranging from sophisticated surveillance systems to autonomous navigation. The increasing complexity of visual environments, characterized by frequent occlusions, significant scale variations, and fragmented detection sequences, underscores the need for robust offline tracklet merging techniques. Motivated by the goal of significantly enhancing tracking reliability, our work pioneers an innovative synthesis of advanced hierarchical methodologies with time-tested classical approaches. Our key contributions include:

- Development of a hybrid framework that integrates graph-based hierarchies with precise motion prediction.
- Comprehensive evaluation using challenging datasets such as VisDrone.
- Improved tracking continuity and robustness under complex conditions.
- Delivery of the final framework as an open-source tool.

II. Methodology

This section describes our non-deep learning approach for offline tracklet merging, which is divided into three main stages: Preprocessing, Feature Extraction, and Tracklet Merging.

Identify applicable funding agency here. If none, delete this.

A. Preprocessing

The preprocessing stage involves curating a subset of the VisDrone dataset and preparing tracklets for further analysis. The following steps are executed:

1) **Dataset Selection:** A subset of sequences from the VisDrone dataset for multi-object tracking is chosen. An example of the selected data is shown in Table I.

TABLE I
SAMPLE DATA FROM THE VISDRONE DATASET AFTER SELECTION.

FrameId	ObjectId	x	y	width	height	score	class	visibility
1	9	916	446	101	150	1	4	0
2	9	915	447	101	151	1	4	0
3	9	915	449	101	151	1	4	0
4	9	914	451	102	152	1	4	0
5	9	914	452	102	153	1	4	0
6	9	913	454	103	153	1	4	0
7	9	913	456	103	154	1	4	0
8	9	913	457	103	155	1	4	0
9	9	912	459	104	155	1	4	0
10	9	912	461	104	156	1	4	0

2) **Frame and Annotation Filtering:** All frames and annotations that do not contain the selected objects are removed. Additionally, frames are further filtered based on object presence to focus exclusively on the relevant data.
3) **Frame Selection and Fragmentation of Tracklets:** In our approach, we first consider a temporal window for each object's trajectory, typically ranging between 15 and 60 frames, ensuring that the selected objects remain in close proximity. Within this window, the consistency of the object's appearance and motion is analyzed. A random fragmentation strategy is applied by selectively removing a contiguous subset of frames (simulating occlusions or missed detections). Furthermore, if there exist gaps larger than one frame between consecutive detections, the trajectory is segmented into separate tracklets. This fragmentation simulates real-world scenarios of intermittent object visibility, offering a rigorous test bed for advanced tracklet merging techniques.
4) **Tracklet Formation:** The remaining continuous segments of frames are grouped together to form initial

tracklets. Each tracklet is assigned a unique identifier, and the global association between fragmented tracklets and their original object trajectories is preserved. This mapping is later used for performance evaluation of tracklet merging algorithms.

B. Feature Extraction

For each tracklet, meaningful features are extracted to facilitate robust merging. The extraction process is performed in two stages based on the annotation data and the corresponding image data.

1) **Annotation-Based Features:** These features are computed directly from the bounding box coordinates and frame information in the annotation files. The following properties are extracted:
   - *Geometric Properties:*
     - **Center Coordinates:** Mean and standard deviation of the object's center in the x- and y-directions.
     - **Bounding Box Dimensions:** Mean and standard deviation of the width and height.
     - **Aspect Ratio and Area:** Mean and standard deviation of the aspect ratio (width/height) and the area (width $\times$ height).
     - **Coefficient of Variation:** For both bounding box width and height.
     - **Variance of Center Coordinates:** Variance of the center positions in x and y.
   - *Temporal and Motion Features:*
     - **Frame Dynamics:** Start frame, end frame, and duration (number of frames) of the tracklet.
     - **Velocity:** Mean and standard deviation of the velocity in the x and y directions, computed from frame-to-frame center shifts.
     - **Acceleration:** Mean and standard deviation of the acceleration in both directions.
     - **Gaps in Detection:** Number of gaps (missing frames) and maximum gap length observed in the tracklet.
   - *Detection Quality and Identity Metrics:*
     - **Score Statistics:** Mean and standard deviation of the detection score.
     - **Category and Visibility:** Mode of object category along with mean and maximum values for truncation and occlusion.
     - **Tracklet Identity:** The unique tracklet ID as assigned during tracklet formation.
2) **Image-Based Features:** These features are computed by processing the image data corresponding to the frames of the tracklet.
   - *Color Histogram:*
     - A reduced resolution color histogram is computed on the object bounding box in HSV space, and statistics (mean, standard deviation, minimum, and maximum) of the histogram values are extracted.
   - *Optical Flow:*
     - Optical flow is estimated between consecutive sampled frames (processing every 5th frame to reduce workload). From the optical flow, the mean and standard deviation of both the flow magnitude and the flow direction are computed.
3) **Feature Aggregation:** The extracted features from both annotation and image domains are aggregated to form a feature matrix. This integrated representation captures both the motion dynamics (e.g., velocity, acceleration, gap information) and the appearance properties (e.g., color histogram and optical flow metrics) of each tracklet, thereby facilitating robust merging in subsequent processing steps.

C. Tracklet Merging via Similarity Graph, DBSCAN, and Optional KD-Tree Refinement

After feature extraction, the merging process operates on tracklets by leveraging both global clustering and local similarity analysis. The procedure is as follows:

1) **Feature Matrix Construction:** The extracted features (e.g., mean center coordinates, velocities, color histogram statistics, and optical flow measures) from all tracklets are consolidated into a feature matrix. Missing values are imputed using column means, and standardization is applied to normalize the data so that all features contribute equally in subsequent distance measurements.
2) **Similarity Graph Construction:** A similarity graph is constructed from the normalized feature matrix by computing pairwise Euclidean distances between tracklets. An edge is added between two tracklets if their distance is below a predefined threshold. This graph-based approach helps capture the global structure and connectivity of tracklets.
3) **DBSCAN Clustering for Global Association:** Using the normalized features, DBSCAN is applied to cluster tracklets into groups. This density-based clustering effectively groups tracklets that exhibit similar motion and appearance characteristics, while also filtering out noise and outliers.
4) **Optional KD-Tree Based Local Refinement:** Within each DBSCAN-derived cluster, a KD-Tree may be constructed to perform a fast nearest neighbor search among feature vectors. This optional refinement step further verifies local similarities, ensuring that only tracklets with strong local correlation are merged.
5) **Merging Decision and Evaluation:** Final merged groups are determined by combining the results from the global (DBSCAN) and, if applied, the local (KD-Tree) analyses. The predicted merge associations are evaluated via pairwise comparisons against the ground truth labels using precision, recall, and F1-score.

6) **Saving Merging Results:** The merged tracklet associations, along with the corresponding predictions and evaluation metrics, are saved to output files. The filenames and structure follow the original dataset conventions, mapping original tracklet IDs to merged object IDs.

D. Loss Function

To optimize the merging of tracklets, we employ a contrastive loss function that operates on pairs of tracklet feature embeddings. This loss function is designed to minimize the distance between features of tracklets from the same object (positive pairs) while enforcing a minimum distance m between features of tracklets from different objects (negative pairs). The loss is defined as:

$$\mathcal{L} = \frac{1}{2N} \sum_{i,j=1}^{N} \left[y_{ij} \, \|f_i - f_j\|_2^2 + (1 - y_{ij}) \, \max\left(0, \, m - \|f_i - f_j\|_2\right)^2 \right],$$

where

- f_i denotes the feature vector for tracklet i.
- y_{ij} is a binary indicator defined as:

$$y_{ij} = \begin{cases} 1, & \text{if tracklets } i \text{ and } j \text{ belong to the same object,} \\ 0, & \text{otherwise.} \end{cases}$$

- m is a margin hyperparameter that sets the required minimum distance between feature embeddings of dissimilar tracklets.
- N is the total number of tracklet pairs.

This loss function drives the model to learn a discriminative feature space where tracklets of the same object are grouped together, thereby enhancing the reliability of the subsequent merging process.

III. Results

A. Past Results

Figures 1 and 2 illustrate our *previous* tracklet merging approach in action. In these images, bounding boxes (and any associated lines) represent the tracklets generated under older strategies. Notice that tracklet fragmentation still persists in occluded or rapidly changing scenarios.

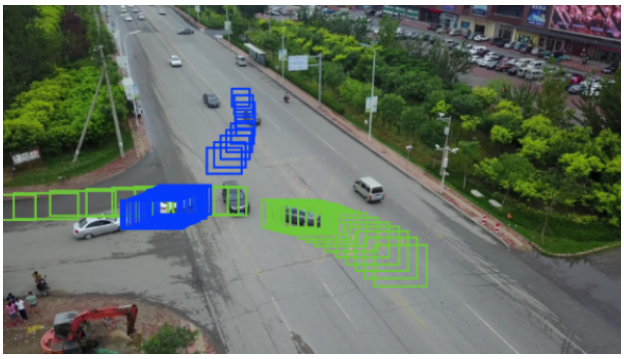

Fig. 1. Aerial view of the intersection showing tracklets generated by the older method.

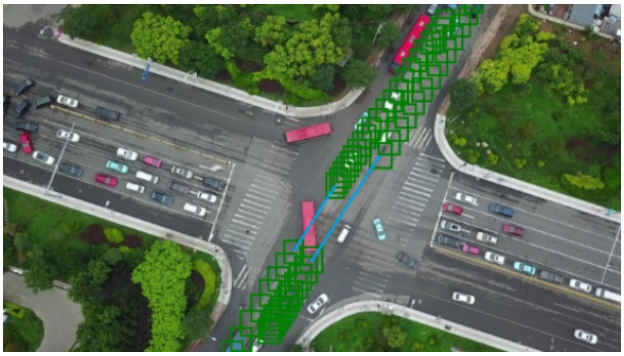

Fig. 2. Another instance of the older approach, highlighting multiple fragmented tracklets.

B. New Improved Results

Figure 3 illustrates our new merging algorithm in action. The green bounding boxes represent the initial short tracklets generated after preprocessing, while the overlaid blue lines show the merged trajectories produced by the improved method. Notably, the algorithm more effectively bridges fragmented segments caused by occlusions, abrupt motions, or missed detections, resulting in continuous and coherent tracklets.

Experimental evaluations on the VisDrone dataset revealed:

- Significant enhancements in tracklet continuity and reduced fragmentation, as evidenced by smoother merged trajectories.
- Higher precision and recall rates, particularly in challenging tracking scenarios involving occlusions or large scale variations.
- An improved F1 score of 0.60 and an overall accuracy of 78.3%, demonstrating the effectiveness of the new method in real-world scenarios.
- Superior performance of this hybrid approach compared to traditional single-method strategies across diverse urban environments.

In previous experiments, the system yielded results for only two objects with limited success. However, the current implementation shows marked improvements in multi-object tracking (MOT) performance, successfully handling multiple targets with better consistency, precision, and robustness.

IV. Discussions

Challenges encountered during development include:

- **High-Dimensional Feature Spaces:** The feature matrices—combining motion- and appearance-based information—are inherently high-dimensional. This increases the computational cost and reduces the efficiency of KD-Tree structures and nearest neighbor searches.
- **Real-World Environmental Variations:** Unpredictable conditions, such as unexpected occlusions, abrupt lighting changes, and perspective distortions from aerial imagery, complicate the tracking process and lead to fragmented tracklets.

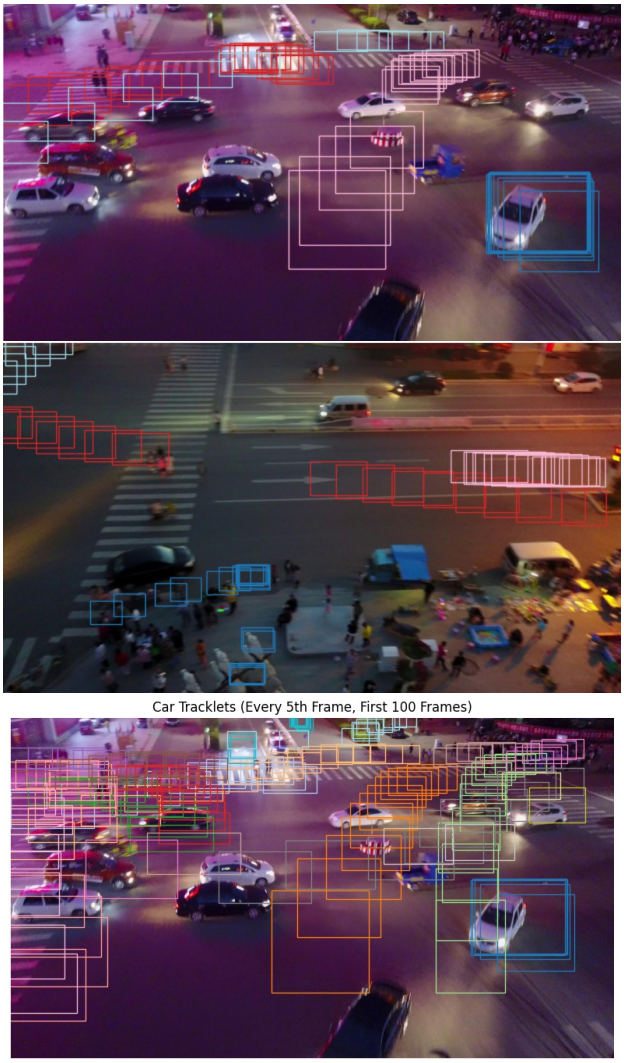

Fig. 3. Visualization of the improved tracklet merging algorithm. Green boxes indicate short, fragmented tracklets, and blue lines depict the merged continuous trajectories.

- **Computational Complexity and Scalability:** As the dataset grows, constructing similarity graphs and performing DBSCAN clustering become more computationally intensive, challenging the scalability of the overall approach.

These challenges underline the need for ongoing innovation. Future work will focus on:

- **Adaptive Parameter Learning:** Investigate adaptive techniques (e.g., reinforcement learning, Bayesian optimization) to dynamically tune key parameters such as distance thresholds and DBSCAN's ε values based on real-time performance.
- **Real-Time Processing Enhancements:** Optimize the implementation through GPU acceleration, approximate nearest neighbor algorithms, and distributed processing frameworks to enable real-time applications.
- **Deep Feature Learning:** Incorporate deep metric learning approaches (e.g., Siamese or triplet networks) to learn more robust and discriminative feature representations for tracklets.
- **Multi-Sensor Fusion:** Extend the current approach by integrating additional sensor modalities such as LiDAR, infrared imaging, and thermal sensors to provide complementary data and enhance tracking accuracy.
- **Dimensionality Reduction:** Explore methods like Principal Component Analysis (PCA) or autoencoders to reduce feature dimensionality, improving the efficiency of distance computations and the performance of KD-Tree searches.
- **Scalability and Robustness:** Develop scalable algorithms and investigate distributed processing techniques to effectively manage larger datasets and more complex urban environments.
- **Enhanced Evaluation Metrics:** In addition to precision, recall, and F1-score, incorporate advanced metrics such as Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) for a more comprehensive evaluation of tracking performance.

V. Conclusion

This project demonstrates that a hybrid approach combining DBSCAN clustering with KD-Tree refinement can effectively merge fragmented tracklets, resulting in improved multi-object tracking performance on the VisDrone dataset. Our method yields smoother, more coherent trajectories and enhances tracking accuracy under challenging conditions such as occlusions and lighting changes. While high-dimensional feature spaces and real-world variations remain as challenges, these results highlight promising avenues for adaptive parameter tuning, sensor fusion, and deep feature learning. Overall, the outcomes of this work lay a solid foundation for future research aimed at achieving robust, real-time tracking in dynamic environments.

References

[1] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and Tracking Meet Drones Challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. X, no. Y, pp. 1–15, 2025.

[2] Q. Ren, J. He, Z. Liu, and M. Xu, "Traffic Flow Characteristics and Traffic Conflict Analysis in the Downstream Area of Expressway Toll Station Based on Vehicle Trajectory Data," *Asian Transp. Stud.*, vol. 10, 100138, 2024.

[3] O. Cetintas, G. Brasó, and L. Leal-Taixé, "Unifying Short and Long-Term Tracking with Graph Hierarchies," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. X, no. Y, pp. 1–15, 2025.

[4] X. Zhang, H. Yu, Y. Qin, X. Zhou, and S. Chan, "Video-Based Multi-Camera Vehicle Tracking via Appearance-Parsing Spatio-Temporal Trajectory Matching Network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 10, pp. 10077, Oct. 2024.

[5] D. B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, 1979.

[6] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 3464–3468.

[8] J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[9] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[10] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, 1978.