Multi-Object Tracking Using Depth - Group 10

Allan Cheboiwo ID: 73661290
Tarek Alkabbani ID: 84930239
Haoyu Wang ID: 42343871
Vanessa Laurel Hariyanto ID: 72484546

I. Introduction

Computer vision is an interdisciplinary scientific field focused on enabling computers to automatically extract meaningful information from images or videos to understand or represent the real world. It aims to develop systems that can achieve high-level visual understanding, similar to human perception [1]–[3]. Computer vision is crucial because it enables machines to understand and analyze visual data, thereby facilitating advancements in fields such as healthcare, autonomous driving, and industrial automation.

Multiple Object Tracking (MOT) plays an important role in computer vision. The task of MOT is largely partitioned into locating multiple objects, maintaining their identities, and yielding their individual trajectories given an input video [4]. By tracking multiple objects in the video accurately, MOT can enhance the ability to analyze movement patterns, improve safety in transportation systems, optimize traffic flow management, and strengthen security monitoring. These abilities make it a foundation in a variety of real-world scenarios that require dynamic object analysis. Fig. 1 [5] shows a general MOT process. The detector begins by detecting objects in the current frame. Then, when the detected object is fed into the MOT algorithm, it is tracked and given a unique ID. These detected objects are then tracked in the next frame, as shown in Fig. 2 [5].

When designing MOT methods, two key aspects must be considered: 1) determining an effective way to measure the similarity between objects within frames and 2) developing a method to restore identity information by utilizing similarity measurement

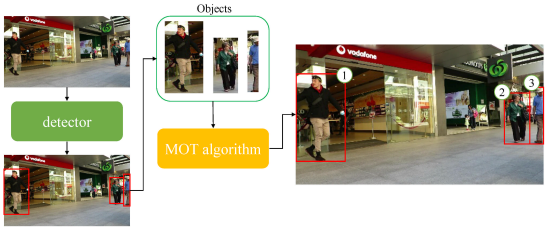


Fig. 1: A general MOT process.

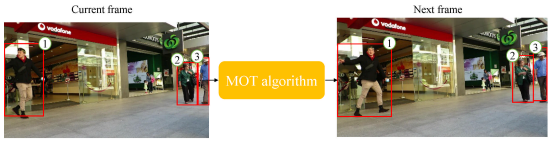


Fig. 2: The visualization of the object tracking for the next frame using the MOT algorithm.

algorithms across consecutive frames [5]. This introduces two significant challenges of MOT, which are occlusion and ID switching.

Occlusion occurs when one object partially or fully obstructs another in the same video frame, making it difficult for the tracker to maintain a continuous trajectory, especially when using only camera-based tracking without additional sensor data. An example of occlusion is shown in Fig. 3 [5], the objects are clearly separated with no overlap in frame 1. In frame 2, the objects begin to overlap slightly, a situation known as occlusion. In frame 3, the green object is almost completely hidden behind the orange one due to significant occlusion.

ID switching happens when the tracker incorrectly assigns a new identity to an object that should

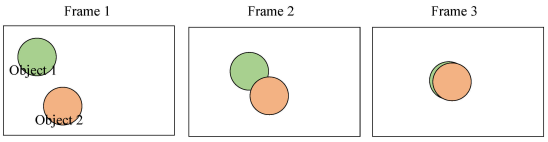


Fig. 3: Example of the occlusion for two objects (orange and green).

retain its original ID in the entire video, often due to temporary disappearance from the frame or inaccurate motion prediction [6]. Fig. 4 is an example of an ID switching [5]. As shown in the figure, when objects in frame 1 and frame 2 are on the predicted tracklet, the system recognizes that this is the same object and maintains the same ID. But in frame 3, the same object is not included in the tracklet predicted in the current frame, at which point it is considered to be the missing object in the current frame and is assigned a new ID.

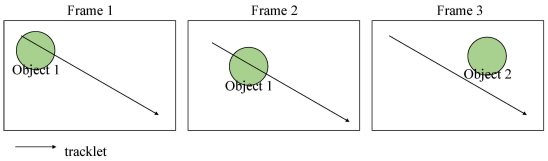


Fig. 4: Example of ID switch problem.

The problem of occlusion and ID switching is the key obstacle that restricts the wide application of multi-object tracking in the real world. These issues are critical in the high-precision tasks such as pedestrian tracking and automatic driving, as these scenarios require robust solutions to ensure robust tracking performance [5]. Occlusion will cause the object to lose correlation between multiple frames, which seriously affects the tracking stability [7]. The frequent occurrence of ID switches greatly reduces the reliability of the MOT systems, especially in visual perception tasks (such as autonomous driving, security monitoring) [6]. Therefore, accurately tracking each object in the video is important for subsequent tasks such as behavioral analysis, security decisions, and more. If these problems can be effectively solved, it will significantly improve the stability and reliability of MOT systems, especially in resource-constrained equipment.

In response to these two major problems, researchers have made many attempts and proposed some potential solutions. For occlusion, some researchers use motion models to predict the location of occluded objects [8]–[10]. Other researchers extract appearance information using the convolutional neural network (CNN) [11]–[13] or using the graph information to find global attributes [14], [15]. Others are working on models and methods that can recover the trajectory of the occluded objects [16]–[18]. For ID switching problem, the main method is to improve the data association mechanism between frames [19], [20]. At the same time, some researches focus on integrating object Re-ID mechanisms into the tracking process so that the identity of the object can be correctly recovered after it is obscured or briefly disappeared [21], [22]. In addition, end-to-end unified detection and tracking frameworks, such as the Transformer-based Transtrack model, have also been proposed to avoid ID switching problems in traditional multi-stage processes [23].

However, although deep learning methods (such as CNN, Long Short-Term Memory (LSTM), and transformer) have shown outstanding performance in MOT tasks [15], their drawbacks like high computational cost, reliance on large-scale datasets, and slow real-time inference speed make traditional methods like Kalman filtering still valuable for research. For example, in resource-constrained environments, such as embedded systems and edge computing devices, these methods can provide stable tracking performance with significantly lower computational overhead. Therefore, gaining a thorough understanding of these classical methods is crucial for recognizing their limitations and guiding the development of more accurate and efficient tracking solutions.

In this project, the specific application scenario we have chosen is pedestrian tracking. There are three key reasons for choosing this application scenario. First, unlike many other objects around us, pedestrians are non-rigid and dynamic, which makes them a representative application scenario

for studying MOT problems with many aspects that can be studied and improved [4]. Second, pedestrian videos are widely used in the real world, which makes them of great commercial value and practical significance. Third, pedestrian video datasets are relatively easy to obtain, which provides rich data for the development and evaluation of tracking algorithms. Building on this focus, we specifically chose the use of Kalman Filter (KF) and Hungarian Algorithm for pedestrian tracking. KF is a widely used approach in MOT due to its efficiency in state estimation and motion prediction. However, challenges such as occlusion and ID switching persist, making it necessary to explore enhancements [24].

In conclusion, this project aims to improve the performance of multiple object tracking (MOT) systems that suffer from ID switching due to occlusion. Specifically, when a target is partially or completely obscured by other targets in a video, existing tracking algorithms often fail to maintain the original identity of the target, resulting in it being mistakenly assigned a new ID. This error is mainly due to the fact that the current tracking algorithm is based on two-dimensional image information and lacks the modeling ability of occlusion relationship in three-dimensional environment [25]–[27], so it cannot correctly track the real motion trajectory and identity continuation of the occluded object. Besides the research work, there are many other applications of using depth information for tracking and perception in the real world, one of the most famous examples is the autonomous driving. Because a self-driving car's light detection and ranging (LiDAR) system is able to get depth information directly, the self-driving car can accurately identify and detect objects around it. Therefore, inspired by these works and examples, we hope to enhance and improve the overall accuracy and robustness of the tracking process in the occlusion scene by introducing the depth information into the existing algorithms.

The rest of this report is organized as follows: Section II will provide a literature review of the relevant research work in the field of MOT using Kalman Filter, and analyze the advantages and disadvantages of existing methods. Section III will explain the design and implementation of our system. Section IV will present the results of the experiment, including test inputs and outputs, key screenshots, and the necessary data charts to illustrate the performance of our system. In Section V, the advantages and disadvantages of the system will be discussed and the overall performance will be evaluated. Finally, Section VI will propose the possible improvement direction of the system in the future to further validate the approach and improve the accuracy and robustness of the tracking.

II. Literary Review

A. Traditional Tracking Methods

1) Kalman Filter with Hungarian Algorithm (SORT): Bewley et al. [28] proposed Simple Online and Realtime Tracking (SORT). It is a tracking method that combines Kalman filtering and the Hungarian Algorithm. The Kalman filter is used to predict the state of a detected object in the next frame. The predictions are made on the assumption that the object will move in a linear way and at a constant velocity. Also, gaussian noise is added to the predictions to handle random deviations.

The Hungarian algorithm is then used to match the predictions from the Kalman filter with measurements (object detections) from the next frame. The Hungarian algorithm is a classic method for solving the optimal matching problem. It is often used in multi-target tracking to associate the detection results of the current frame with the tracking trajectory of the previous frame. The core idea is to use a "cost matrix" to represent the matching cost between each detection and each trajectory (such as location information, appearance difference, etc.), and then find a set of one-to-one matches in this matrix to minimize the total cost of all matches. In this way, the Hungarian algorithm can effectively find the optimal allocation scheme, thereby achieving stable and accurate object identity association in MOT.

These matches are then used to update the Kalman filter. This process is then repeated recursively until the last frame is completed.

- **Strengths:** Since SORT is simple and requires lower computation power compared to other methods, it is suitable for real-time tracking.
- **Weaknesses:** Since SORT assumes that the object will move in a straight line and with a constant speed, its predictions can be way off when objects deviate from the path abruptly as happens in the real world. Also, since it does not include depth information, it makes it harder for it to handle occlusions which results in label switching even if it is the same object when the object reappears.

2) Multiple Hypothesis Tracking (MHT): Multiple Hypothesis Tracking (MHT) introduced by Reid [29] uses the Kalman filter to predict the locations of objects in the next frame. It then tries to match the predictions with the detected objects from the next frame using a likelihood score. If it is unsure, with a low probability below a certain threshold, it delays the decision to match and creates a tree of possible scenarios. It then eliminates branches that are unlikely to occur in the tree to prevent it from becoming too big and computationally expensive to process.

- **Strengths:** MHT excels in handling occlusions and complex motion because it keeps track of all possible matches for a prolonged period.
- **Weaknesses:** Its combinatorial complexity grows exponentially as more and more objects and frames are processed especially in densely populated scenarios. This makes it impractical for real-time applications. Like SORT, it lacks depth information, which results in occlusions which lead to ID switches.

B. Deep Learning-Augmented Tracking Methods

1) Deep SORT with Appearance Features: Wojke et al. [30] extended SORT with DeepSORT. DeepSort integrates a CNN-trained appearance descriptor into the tracking framework. In addition to the Kalman Filter and Hungarian Algorithm, DeepSORT uses the distance between the appearance features to improve association. This complements the intersection of Union costs.

- **Strengths:** Appearance features reduce ID switches in occlusion-heavy pedestrian scenes. This improves its robustness over SORT.
- **Weaknesses:** The CNN descriptor increases computational overhead, and its 2D focus misses depth cues which might help reduce occlusion issues.

2) FairMOT: End-to-End Tracking: The research of Zhang et al. [22] presented FairMOT as a fully connected multi-object tracking (MOT) system which unifies detection with tracking through deep learning. The system implements CenterNet as its backbone network to discover object centers while generating re-identification feature outputs.

- **Strengths:** The end-to-end structure of FairMOT improves tracking precision in 3D pedestrian environments through successful detection in combination with association.
- **Weaknesses:** One major drawback of this system is its need for substantial computing power and large datasets which reduces its usefulness for edge devices. Also, its limitation to 2D dimension operations prevents this system from understanding depth which produces inaccuracies whenever an object becomes occluded.

C. Comparison and Gaps

The SORT method is simple and less computationally expensive compared to other methods. This makes it suitable for real-time tracking especially when compute is limited. MHT is better than SORT at handling occlusions but this comes at a cost of time and compute as it has to keep track of a lot of states and possibilities. The deep learning methods used in DeepSort and FairMOT make them more accurate in data association but you would require a large dataset and compute to train the models compared to the traditional approaches. The methods we have discussed all lack depth consideration in their execution. This makes them always susceptible to occlusion problems that arise from 3D environments. To better deal with this, we have proposed extending SORT, due to its simplicity and suitability to real-time tracking, to include depth information. This will help us better

handle occlusions and reduce ID switching in our pedestrian tracking application.

III. System Design

Our proposed 3D SORT algorithm integrates depth information into the baseline SORT algorithm to do multi-object tracking of pedestrians. Our system architecture is divided into two parts, namely: object detection and depth estimation, and Tracking with SORT using 3D Descriptors. The aim of our system is to reduce to ID switching that results from occlusion due to lack of distinction of objects with different depths but almost the same x, y coordinates in a 2D scenario.

A. Object Detection and Depth estimation

For object detection, we used YoloV8s model, a deep neural network, to get the 2D bounding boxes of pedestrians detected in each frame and the confidence of each detection. The format of the bounding boxes was $[x_1, y_1, x_2, y_2]$ which represent its top left corner and bottom right corner. We used the MiDaS model as our depth estimator, which calculated the depth map of every pixel in the frame. We then extracted depth values within the bounding box from the object detector and retrieved the max and min values in that area as indicated in Fig. 5. This was then used to form the 3D descriptor, $[x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}, confidence]$, which was passed to our sort algorithm.

B. Tracking with SORT using 3D Descriptors

We implemented the 3D SORT using two different methods: one using IoU and the other using Euclidean distance for association

1) Using IoU: The SORT algorithm creates a tracking object based on the descriptor information, $[x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}, confidence]$, from our detectors. The SORT tracking objects have a state matrix to capture the state information and its changes overtime with regards to change in time using a constant velocity model as is the case for Kalman filters. The following is the information captured in the object state:

- x, y, z: Center coordinates of the bounding box in the image plane (x, y) and depth (z).
- s: Scale, defined as the average of the bounding box width and height, i.e., $s = (w * h)$.
- r: Aspect ratio, computed as the ratio of width to height, i.e., $r = w/h$.
- d: Depth range, representing the difference between maximum and minimum depth, i.e., $d = z_{\text{max}} - z_{\text{min}}$.
- v_x, v_y, v_z, v_s, v_d: Respective velocities of the center coordinates (x, y, z), scale (s), and depth range (d).

Similarly, the measurements from the detectors are also captured in the state measurement matrix as follows:

- x, y, z: Center coordinates of the bounding box, representing position in the image plane (x, y) and depth (z).
- s: Scale, defined as the area of the bounding box, i.e., $s = (w * h)$.
- r: Aspect ratio, computed as the ratio of width to height, i.e., $r = w/h$.
- d: Depth range, representing the difference between maximum and minimum depth, i.e., $d = z_{\text{max}} - z_{\text{min}}$.

Each object also has covariance matrices to properly reflect the uncertainties/noise in our measurements, state information as we predict and processes. Processes in this case mean unexpected behavior from objects, such as increasing speed or path deviation. These covariance measures are then used to weight our final prediction output based on whether we trust our predictions or measurements more. Our associated measurements and predictions are matched using 3D IoU. The cost matrix is obtained by the following operation: $1 - IoU$. The Hungarian algorithm then finds matches based on this. Trackers that remain unmatched for a set number of consecutive frames are deleted. Unmatched measurements are used to create new tracker objects after a certain minimum number of frames . These operations repeat until the final frames are passed to our SORT algorithm.

2) Using Euclidean Distance: The tracking logic(Kalman filter with Hungarian algorithm) remains the same from the IoU section. The main difference is that we are using Euclidean distance between the center point of each measurement and

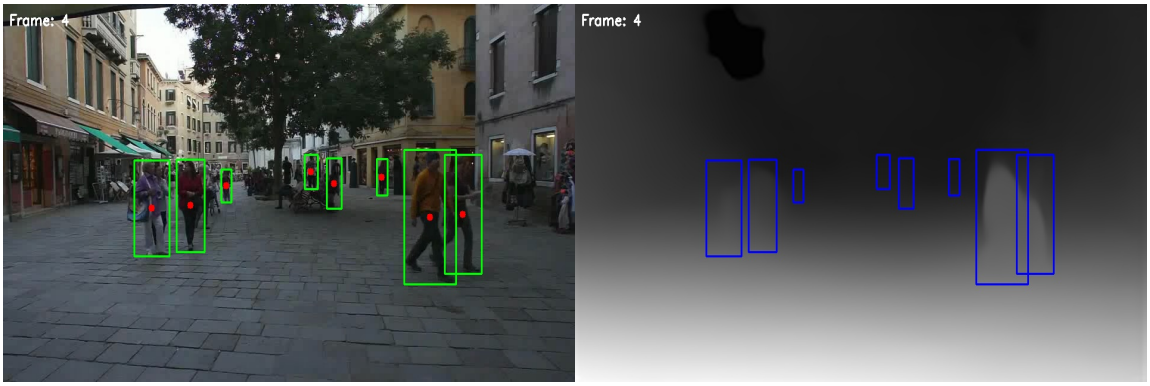


Fig. 5: A frame from the dataset with detections alongside depth map. The bounding boxes in the depth map represent the regions we are using to get the minimum and maximum depths used in forming the 3D descriptors fed to our tracker.

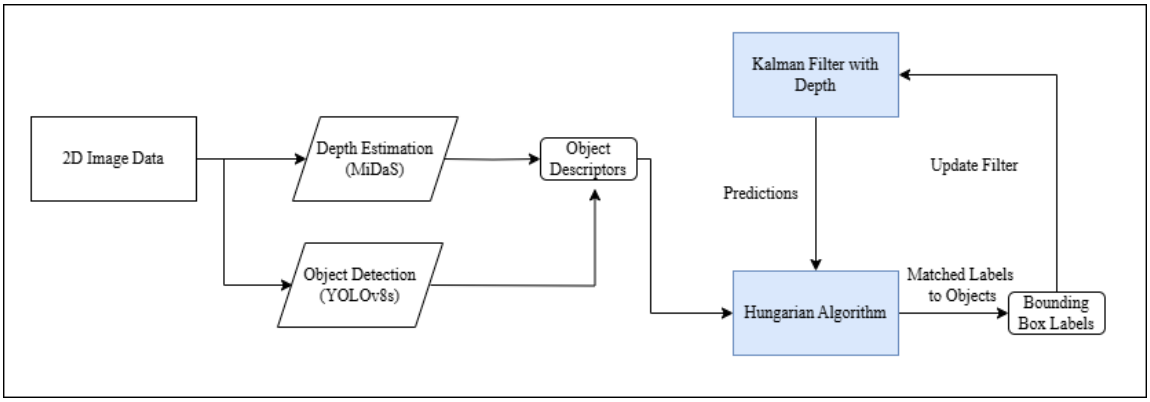


Fig. 6: System flow chart showing how our program is structured. In the first phase, a 2D image frame from the video is passed to the Depth Estimator, which builds a depth map of every pixel in the image and the object detector, which outputs the bounding box of each detected pedestrian. This information is then combined to form a descriptor, which is then sent to the tracking part of our algorithm. The tracking is done using an extended SORT algorithm which captures depth information. The predictions and matches are done using the Kalman filter and Hungarian algorithm, respectively.

our predictions to make matches, instead of the IoU method. We also keep track of fewer state information. The following is the state matrix information we keep track of in the method:

- x, y, z: are the center co-ordinates of the object
- v_x, v_y, v_z: Respective velocities of the center coordinates (x, y, z).

Similarly, the measurements we keep track of are as follows:

- x, y, z: Center coordinates of the bounding box, representing position in the image plane (x, y) and depth (z).

Fig. 6 is the flow chart of our overall system.

IV. Results

We have used MOT-17-SDP-02 and MOT-17-SDP-04 sequences from the MOT Challenge

datasets for the evaluation of our 3D SORT algorithm implementations. We then compared them with the baseline SORT algorithm and its variants DeepSORT, and OC-SORT. FairMOT, a very competitive Multi-Object tracking algorithm, is also included. The evaluation Metrics used were HOTA, MOTA, IDF1 and IDs. HOTA evaluates both detection and association accuracy giving a sense of the overall quality of the tracker. MOTA measures and combines detection accuracy and number of ID switches. IDF1 measures how well algorithms can track identities across frames. IDs count how many times a tracker assigns a different ID to the same ground-truth object during its trajectory. For MOTA,HOTA and IDF1, the higher the value the better the tracker while for IDs, the lower the better.

Our algorithm underperforms across the board as shown in Table I with the exception being on the IDs metric. Fig. 7 is a more intuitive comparison of the performance of different algorithms.

TABLE I: Performance comparison of SORT-3D variants with SORT, DeepSORT, OC-SORT, and FairMOT on MOT17.

Tracker	HOTA(%)↑	MOTA(%)↑	IDF1(%)↑	IDs↓
SORT-3D Eucl.	28.35	21.18	35.96	**252**
SORT-3D IoU	7.92	-26.27	3.35	1581
SORT	43.1	43.1	39.8	4,852
DeepSORT	55.0	61.4	62.2	781
OC-SORT	54.2	63.2	62.1	522
FairMOT	59.3	73.7	72.3	330

From the results, our extended SORT implementation using euclidean distance significantly outperforms the implementation using IoU on all the scores across the board. This is likely because this approach aligns better with the MOT17 2D dataset and is more robust to noise from the depth values generated by MiDaS as compared to IoU approach. They both however underperform baseline SORT and the other algorithms on the HOTA, MOTA and IDF1 metrics. This likely due to 3D implementation not aligning well the 2D benchmark dataset and noise from MiDaS. However our algorithms perform better on the IDs metric with the Euclidean implementation outperforming all other algorithms.

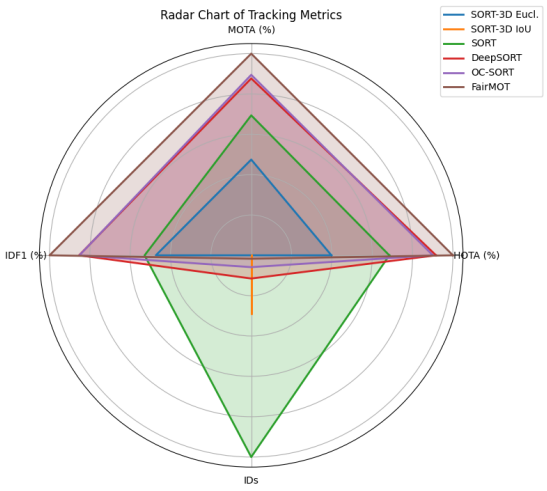


Fig. 7: Radar map showing the performance of the algorithms across the four categories.

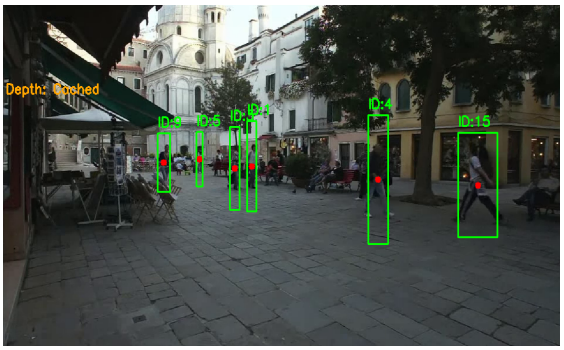
The under performance of our algorithms as stated above is likely due to the noise in the MiDaS depth information which affects the quality of our descriptors. This is consistent with the sentiment in the SORT [28] where they found that the quality of the tracking performance is underpinned by the quality of the detector.

Fig. 8 shows the results of our system running on different test videos. As can be seen from the figure, our system was able to run successfully and incorporate depth information into the detection.

V. Discussions

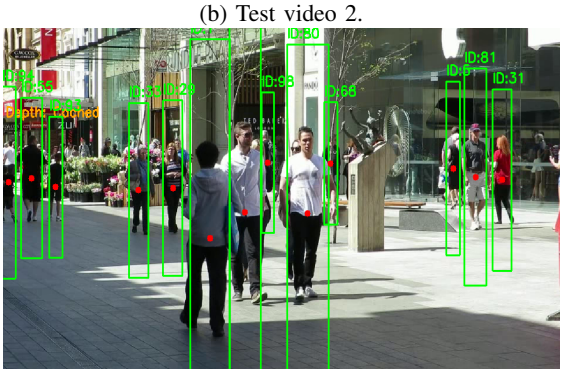
Our goal was to enhance pedestrian tracking using Multiple Object Tracking (MOT) by utilizing depth information to reduce occlusion-induced ID switching. However, currently the MOT17 evaluation test reports that our method slightly under performs in most metrics in comparison with SORT. It also under performs some of the newer deep learning approaches by a significant margin. However, our ID switch metric was significantly better than even the state of the art.

Through qualitative analysis, we expect such results. Our current pipeline is suffering most sig-



(a) Test video 1.

(b) Test video 2.



(c) Test video 3.

Fig. 8: Running results on different test videos.

nificantly from the instability and variance of the deep learning methods' outputs. Currently, our implementation of our detector struggles with detecting all the subjects in a frame. This effect is significantly worse when dealing with high traffic environments and samples. Additionally, currently the detector's bounding boxes are overestimating the size of the subjects significantly as seen in Fig. 9. This severely impacts the HOTA and MOTA scores as they severely punish incorrect bounding boxes.

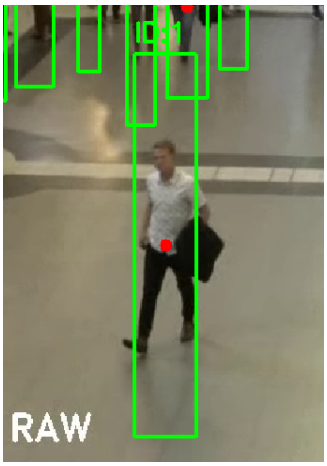


Fig. 9: Example of inaccurate bounding box.

Also, the depth estimator is not optimized for giving consistent outputs between continuous frames. Even if the relative depth estimation is somewhat accurate, the absolute depths can vary significantly. This is also more apparent when using the smaller version of MiDaS, where the number of ID switching increases significantly. As our method relies on the absolute depths, these variation can severely impact the quality of the tracking. This was also verified by the significant improvement of quality in tracking when we moved to the Euclidean distance criterion. Due to the variance introduced by MiDaS, the bounding box interpretation of our descriptors can vary wildly from frame to frame. Using centroid averages those variations in z. This makes the criterion more robust to variance in the depth prediction.

One issue we ran into when evaluating our method is the format of existing evaluation tools.

Existing tools work exclusively on 2D data. This required us to truncate our descriptor when evaluating the method. This dimension reduction would further decrease the performance of our method as significant 3D data is lost. Additionally, we were unable to utilize some of the evaluation metrics that concern ID tracking and ID switching. This was due to the fact that these metrics usually isolate the detection method and only test the tracker. This is done by passing correct detections only to the tracer and doesn't deal with the image data. This was not possible in our method as the tracker lies on 3D detections that were unavailable.

With all of these issues aside, the significant decrease in ID switching compared with the other methods, insinuates that the use of depth information might be a valid idea for implementation in MOT. With the current implementation, we are unable to validate this approach. This is due to the significant variance and errors caused by our depth estimator and detector. A better approach is to evaluate this tracking method with a dataset that already includes depth information in the image data. However, currently there isn't a public dataset that fits this criteria. All depth datasets we explored cannot be used for MOT. And all MOT datasets we explored don't contain depth information. These results are consistent with earlier studies that highlight how crucial detection quality is to tracking systems' efficacy.

This method might still be viable for applications where the depth and image information are available. Modern cars carry a suite of sensors that include LiDaR sensors and image sensors. This would be a great application where the 3D data can be utilized more efficiently for tracking. This also can be extended for a lot of robotic applications. Humanoid robots dexterity can be improved as it allows for a new dimension for tracking different objects.

The dependence on the MiDaS for depth estimate was a significant cause of the poorer performance. Although this model is computationally efficient, it lacks the depth accuracy necessary for accurate IoU-based matching, which results in more ID switches and fewer valid associations. Further ID mismatches resulted from the need to transform our results back to 2D for benchmarking, which further reduced the use of 3D knowledge in addressing occlusions.

VI. Future Work

In our present approach, a depth estimator applied to a 2D dataset was used to extract depth information. Due to the possibility of the depth estimator's unreliability, this adds possible noise and irregularities to our tracking procedure. Furthermore, because our tracker's accuracy is affected by both the detector and the depth estimator, it is challenging to separate the tracker's efficacy from the detector, which further complicates performance measurement. Using annotated 3D datasets, which would offer a standardized benchmark comparable to the ground truth values in the MOT benchmark datasets, is a more trustworthy method of evaluating the performance of our algorithm.

However, there are not many of these datasets available right now. In order to create a consistent platform for assessing and contrasting 3D Multi-Object Tracking algorithms in pedestrian scenarios, future research should concentrate on expanding the availability of high-quality 3D pedestrian tracking datasets. More consistent detections would be possible with a larger dataset pool, making it easier to evaluate various tracking techniques objectively.

Furthermore, the shortage of extensive real-world 3D datasets may be lessened by utilizing synthetic 3D datasets and data augmentation methods. In order to decrease reliance on depth estimators, future studies should also look into ways to enhance depth estimation methods or incorporate other depth acquisition strategies. These enhancements would make our tracking system more reliable, accurate, and applicable in real-world settings including smart cities application, surveillance, and autonomous driving.

The model might be made more scalable by optimizing it for deployment on edge devices, which would allow real-time processing without requiring expensive GPUs. Advanced methods such as transformer-based models or graph-based tracking might be investigated to reduce ID switching and

boost performance in congested environments, thus increasing tracking accuracy.

REFERENCES

[1] R. Klette, *Concise computer vision*, vol. 233. Springer, 2014.

[2] G. Stockman and L. G. Shapiro, *Computer vision*. Prentice Hall PTR, 2001.

[3] T. Morris, *Computer vision and image processing*. Palgrave Macmillan Ltd, 2004.

[4] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103448, 2021.

[5] Y. Park, L. M. Dang, S. Lee, D. Han, and H. Moon, "Multiple object tracking in deep learning approaches: A survey," *Electronics*, vol. 10, no. 19, p. 2406, 2021.

[6] S. Hassan, G. Mujtaba, A. Rajput, and N. Fatima, "Multi-object tracking: a systematic literature review," *Multimedia Tools and Applications*, vol. 83, no. 14, pp. 43439–43492, 2024.

[7] Z. Sun, G. Wei, W. Fu, M. Ye, K. Jiang, C. Liang, T. Zhu, T. He, and M. Mukherjee, "Multiple pedestrian tracking under occlusion: A survey and outlook," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 2, pp. 1009–1027, 2025.

[8] G. Tian, X. Zhang, S. Guo, Y. Liu, X. Liu, and K. Wang, "Occlusion handling based on motion estimation for multi-object tracking," in *2021 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 1031–1036, 2021.

[9] J.-M. Li, C.-W. Chen, and T.-H. Cheng, "Motion prediction and robust tracking of a dynamic and temporarily-occluded target by an unmanned aerial vehicle," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1623–1635, 2021.

[10] M. Zolfaghari, H. Ghanei-Yakhdan, and M. Yazdi, "Real-time object tracking based on an adaptive transition model and extended kalman filter to handle full occlusion," *The Visual Computer*, vol. 36, pp. 701–715, 2020.

[11] W. Huo, J. Ou, and T. Li, "Multi-target tracking algorithm based on deep learning," in *Journal of Physics: Conference Series*, vol. 1948, p. 012011, IOP Publishing, 2021.

[12] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 31, 2017.

[13] Y. Tian, A. Dehghan, and M. Shah, "On detection, data association and segmentation for multi-target tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2146–2160, 2018.

[14] M. Ullah and F. Alaya Cheikh, "A directed sparse graphical model for multi-target tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1816–1823, 2018.

[15] Z. Shagdar, M. Ullah, H. Ullah, and F. A. Cheikh, "Geometric deep learning for multi-object tracking: A brief review," in *2021 9th European Workshop on Visual Information Processing (EUVIP)*, pp. 1–6, IEEE, 2021.

[16] A. W. Harley, Z. Fang, and K. Fragkiadaki, "Particle video revisited: Tracking through occlusions using point trajectories," in *European Conference on Computer Vision*, pp. 59–75, Springer, 2022.

[17] D. Stadler and J. Beyerer, "Improving multiple pedestrian tracking by track management and occlusion handling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10958–10967, June 2021.

[18] A. Nayak and A. Eskandarian, "Cooperative probabilistic trajectory prediction under occlusion," *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2024.

[19] J. Jin, J. Zhang, K. Zhang, Y. Wang, Y. Ma, and D. Pan, "3d multi-object tracking with boosting data association and improved trajectory management mechanism," *Signal Processing*, vol. 218, p. 109367, 2024.

[20] J. Lee, M. Jeong, and B. C. Ko, "Graph convolution neural network-based data association for online multi-object tracking," *IEEE Access*, vol. 9, pp. 114535–114546, 2021.

[21] X. Zhang, X. Wang, and C. Gu, "Online multi-object tracking with pedestrian re-identification and occlusion processing," *The Visual Computer*, vol. 37, no. 5, pp. 1089–1099, 2021.

[22] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International journal of computer vision*, vol. 129, pp. 3069–3087, 2021.

[23] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[24] M. Khodarahmi and V. Maihami, "A review on kalman filter models," *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 727–747, 2023.

[25] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10359–10366, 2020.

[26] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[27] J. Masci, E. Rodolà, D. Boscaini, M. M. Bronstein, and H. Li, "Geometric deep learning," in *SIGGRAPH ASIA 2016 Courses*, pp. 1–50, 2016.

[28] 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.

[29] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[30] 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.