Comparative Study of Existing Tracking Algorithms

Tarushee Kumar

Department of Computer Science, Bharati Vidyapeeth's College of Engineering, New Delhi

---

Abstract—*Road conditions in Developing countries like India are chaotic as they are, which results in more than 1.2 million deaths globally per year. Relying on vehicular sensory data alone do not suffice for the turbulence cause by irregular driving behaviors and meandering pedestrian jaywalking. These draw a need to employ various methods to track the behavior of these dynamic road objects such as the Pedestrians and Vehicles. While computer vision algorithms perform fairly well on high density conditions, they are comparatively slower on embedded platforms than non-vision motion and intersection based approaches. Deep Learning based methods also incorporate the behavior of the pedestrians but demands higher computing capabilities. We compare methodologies for tracking in this paper and survey their results on MOT16 dataset for SORT, IOU Tracker and DeepSORT algorithms. We find different use cases of employment of these algorithms according to density and tumultuous conditions in the camera feed.*

Keywords—SORT; IOU Tracker; Deep Sort; Tracking algorithms

1. INTRODUCTION

Object tracking is a decisive task within the field of computer vision. The availability of high definition quality and inexpensive video cameras, and the rising demand for automatic video analysis has generated a great deal of significance in object tracking.

There are three key steps in video analysis:

- Detection of moving objects
- Tracking of objects from frame
- Perceiving their behavior or performance.

Therefore, the use of object tracking is important in the tasks of: motion-based recognition, automatic object detection, etc.

Multi-object tracking (MOT) is however, categorized as a broader topic involves being able to locate objects in successive frames to produce intact trajectories. However, MOT is particularly challenging, since frequent interactions, anomalous motion, and similar appearances of tracked objects are common in many real-world scenarios.[13]

Pedestrian detection and tracking are important in a wide range of applications in our daily lives, such as surveillance systems, airport or bank security, and human–robot interactions, to name just a few. Pedestrian occlusion is a very common phenomenon owing to the limitation of camera views and angles. This problem is challenging since it is unknown where the pedestrian of interest will appear or whether it will reappear or not. In addition, it is difficult to ensure the pose of the pedestrian without any change after occlusion.

The MOT archetype is a powerful tool and has a great potent on the nature of objects. However, surprisingly a very little is known about the neural system underlying MOT performance.

Multifarious approaches for object tracking have been proposed and contemplated. These fundamentally differ from each other by the way they approach the following questions: Which object representation is suitable for tracking? Which image features should be taken into account? How should the motion, shape, and appearance of the object be sculpted? The answers to these questions depend on the environment in which the tracking is being performed. A large number of tracking methods have been proposed which attempt to answer these questions for a variety of scenarios.

In this paper, we have targeting following things which make it novel.

1. We have trained all three algorithms (Intersection over union, Sort, Deep sort) on MOT16 dataset.
2. All the training and testing was done on CPU hence; our main goal was to establish comparative results for tracking algorithms on CPU.
3. We compared them using 8 benchmarked parameters.
4. Deep sort was tested on Indian dataset.

2. RELATED WORK

There is a vast literature on multi-object tracking.

A. Using Image sequences and video

Moving object detection and tracking (MODAT)[1] have been traditionally studied using image sequences and video. Objects are detected in the camera reference frame or 2D world coordinate system (Milan et al., 2014). Stereo matching enables us to detect and reconstruct objects in 3D, then their 3D trajectories can be reconstructed (Schindler et al., 2010).

B. Region-Level Tracking

For each pair of successive frames, the region-level tracking process computes region tracks by associating the new foreground regions (destination regions) with the existing regions (source regions). These associations can be characterized by a binary correspondence matrix Θ where rows and columns correspond to source and destination regions respectively. An entry is 1 if there is an association between the corresponding regions and 0 otherwise. As a region evolves, the following association events might occur: Continuation. A region continues from the frame at t − 1 to t. The corresponding column and row in Θ have only one non-zero element.[15]

- Appearing. A new region appears in the frame at t. The corresponding column has all zero elements.
- Disappearing. An existing region in the frame at t−1 disappears. The corresponding row has all zero elements.
- Merging. Two or more regions in the frame at t − 1 merge into one region in the frame at t. The corresponding column has more than one non-zero entries.
- Splitting. One region in the frame at t − 1 separates into two or more regions in the frame at t. The corresponding row has more than one non-zero entries.

C. Multiple Hypothesis Tracking:

In MHT algorithm, several frames have been observed for better tracking outcomes MHT is an iterative algorithm. Iteration begins with a set of existing track hypotheses. Each hypothesis is a crew of disconnect tracks. For each hypothesis, a prediction of object's position in the succeeding frame is made. The predictions are then compared by calculating a distance measure. MHT is capable of tracking multiple object, handles occlusions and Calculating of Optimal solutions.[11]

3. IMPLEMENTATION

We implemented three algorithms, i.e., IoU, Sort and DeepSort on MOT16 dataset. MOT16 was launched in 2016. It consists of 11450 annotations with object density 15.3 per frame. It was filmed from a bus on a busy intersection at 25fps

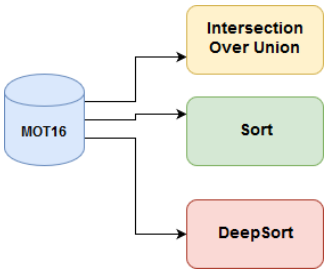

Figure 1. Overview of our model

Following are the Specifications of the system on which these algorithms were implemented:

Processor: Intel(R) Core (™) i5-4210U CPU@1.70GHz 2.40GHz

Installed Memory (RAM): 16.0GB

System type: 64-bit Operating System, x64-based processor.

C. Intersection over Union (IOU)

The IOU [2] tracker is a tracking algorithm which compares the area of intersection and area of union of different frames that the detection algorithm detects.

$$IOU(a,b) = \frac{Area(a) \cap Area(b)}{Area(a) \cup Area(b)} \qquad (1)$$

where a is the object detected in the previous frame and b is the object detected in the current frame.

It works considering the two assumptions. Firstly, the IOU tracker works assuming that the detection algorithm detects each and every object in every frame of the video that needs to be tracked. There are only a few or ideally none gaps in the

detections. Secondly, the IOU tracker assumes that the frames drawn by the detection algorithm in every frame have great area of intersection. This is usually achieved when the frame rate of the video is sufficiently high.

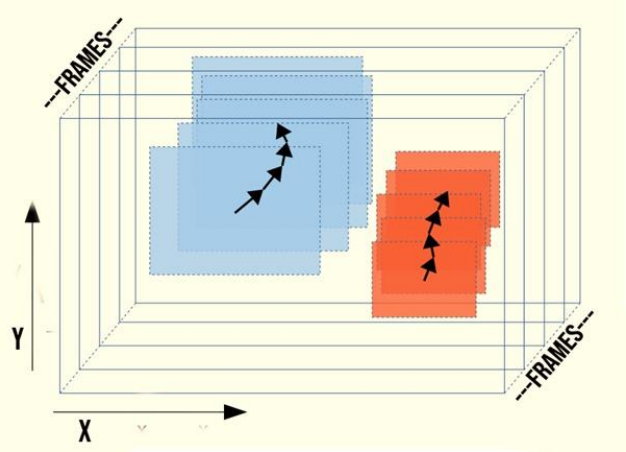

Figure 2. Basic principle of the IOU Tracker: with high accuracy detections at high frame rates, tracking can be done by simply associating detections by their spatial overlap between time steps.

If the above conditions are met, then the IOU tracker can track the object without using the image information and becomes trivial. A threshold value is set for the IOU. The IOU value is calculated by comparing and associating the detection of the current frame with the detection of the previous frame. If the IOU value calculated meets the threshold IOU set then the tracking algorithm tracks the object. If the detection algorithm misses a detection in a particular frame due to occlusion, the IOU tracker ends the assigned tracking in the previous frame. The IOU tracker assigns a new track for that object and gives it a new id.

The IOU tracker used for our research has been modified by removing the tracks that are considered as not important. The tracks of objects that do not meet the minimum time of them being in the video are omitted, i.e., the objects that are present in the video for a short length of time (minimum time that is set) are filtered for improved performance. The tracks whose detections do not meet the minimum score of detection algorithm for even a single detection in the frames of the video are also removed for better performance.[14]

The IOU tracker used for our research has been modified by removing the tracks that are considered as not important. The tracks of objects that do not meet the minimum time of them being in the video are omitted, i.e., the objects that are present in the video for a short length of time (minimum time that is set) are filtered for improved performance. The tracks whose detections do not meet the minimum score of detection algorithm for even a single detection in the frames of the video are also removed for better performance.

Algorithm IOT:

1: Inputs:
D = {D0, D1, ..., DF −1} = {{d0, d1, ..., dN−1}, {d0, d1, ..., dN−1}, ...}
2: Initialize:
Ta = ∅, Tf = ∅
D = {{di |di ∈ Dj , di ≥ σl}|Dj ∈ D}
3: for f = 0 to F :
4: for ti ∈ Ta :
5: dbest = dj where max(IOU(dj , ti)), dj ∈ Df
6: if IOU(dbest, ti) ≥ σIOU :
7: add dbest to ti
8: remove dbest from Df
9: else
10: if highest score(ti) ≥ σh and len(ti) ≥ tmin:
11: add ti to Tf
12: remove ti from Ta
13: for dj ∈ Dt :
14: start new track t with dj and insert into Ta
15: for tj ∈ TA :
16: if highest score(ti) ≥ σh and len(ti) ≥ tmin :

17: add ti to Tf
18: return Tf

The IOU tracker does not require high computational cost as the complexity of the algorithm is very low as compared to other tracking algorithms. It does not require any kind of information from the frames for tracking the object and hence can be understood by everyone and is considered simple as compared to other state-of-the-art trackers. The IOU tracker has a great speed in generating tracks and can be very easily executed online with frames exceeding 100K fps. [12]

$$existence\_time(et) = time\ of\ obj(a)\ in\ video(v) \quad (2)$$

Algorithm OT:

1: Inputs:
et, T_{min}
2: Initialize:
object_tracking = false
3: if (et < T_{min}) :
4: object_tracking = false
5: else:
6: object_tracking = true
7: return object_tracking

D. Simple Online Real Time (SORT)

SORT[3] is one of the simplest algorithm used in the field of tracking. It is completely based on mathematical heuristics as it maximizes the Intersection over union metrics between the bounding boxes across the neighboring frames. All the objects in the frames are localized by the bounding box. These boxes are labeled with a number, i.e., each object is assigned a unique id.

Sort has a capability to detect multiple objects in real time but it hardly associates pre-detected objects across different frames. We can overcome this problem by using a good and an optimized object detector which helps sort tracker to match detected objects across the frame. [9]

E. DeepSort

For multiple object tracking deep sort[4] tracker is one of the best method. Before deep sort, the functionality of simple sort was to take the input from the detection in which it contained the information about the objects present in the current frame and compared it with the previous frame, if any similarity exists then it gives the same identity otherwise it treats as a new object, but sort tracker fails in occlusion scenario. To overcome this drawback deep sort developed with deep learning.

Deep Sort uses the matrices of weight and biases which is generated by the cosine metric learning network which is trained on mars data set. Such that it helps to generate the detection of an object on our data in such a way so it can reduce the problem of occlusion and in result increase the performance of the tracker.

Deep Cosine Metric Learning is the network which uses as training of deep sort model. Cosine metric learning is implemented for re-identification of object that will help in tracking the object over a long period of time. So the network is trained on MARS dataset.[8] MARS contains over one million images that have been annotated in a semi-supervised fashion. The data has been generated using a multi-target tracker that extracts short, reliable trajectory fragments that were subsequently annotated to consistent object trajectories. This annotation procedure not only leads to a larger amount of data, but also puts the dataset closer to real-world applications avoiding manual cropping. MARS is an extension of Market 1501 that contains 1,261 identities and over 1,100,000 images.

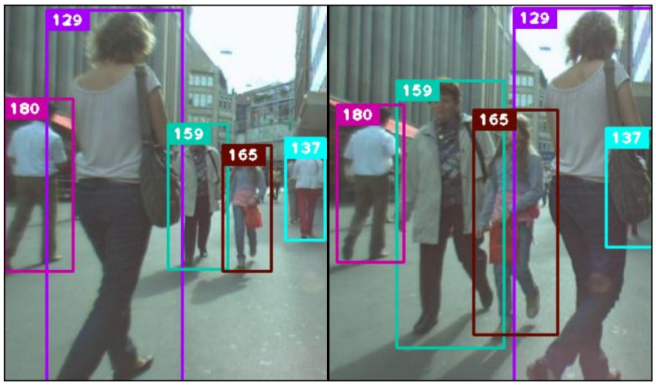

Figure 3. Exemplary output of our method on the MOT challenge dataset in a common tracking situation with frequent occlusion.

The network architecture used in the model is relatively shallow to allow for fast training and inference. Input images are rescaled to 128 × 64 and are presented to the network in RGB color space. A series of convolutional layers reduces the size of the feature map to 16 × 8 before a global feature vector of length 128 is extracted by layer Dense 10. The final `2 normalization projects feature onto the unit hypersphere for application of the cosine SoftMax classifier. The network contains several residual blocks that follow the pre-activation layout. The design follows the ideas of wide residual networks. All convolutions are of size 3 × 3 and max-pooling is replaced by convolutions of stride 2. When the spatial resolution of the feature map is reduced, then the number of channels is increased accordingly to avoid a bottleneck. Dropout and batch normalization are used as means of regularization. Exponential linear units are used as activation function in all layers. Note that within 15 layers (including two convolutional layers in each residual block) the network is relatively shallow when compared to the current trend of ever deeper architectures. This decision was made for the following two reasons. First, the network architecture has been designed for the application of both person re-identification and online people tracking.[6]

Two losses were used which minimized the loss function, i.e., Triplet Loss and Magnet Loss. The triplet loss is defined over tuples of three examples ra, rp, and rn that include a positive pair ya = yp and a negative pair ya 6= yn. For each such triplet the loss demands that the difference of the distance between the negative and positive pair is larger than a predefined margin m ∈ R.

$$L_t\,(ra, rp, rn) = \{||ra - rp\,||_2 - ||ra - rn||_2 + m\}_+ \quad (3)$$

The magnet loss has been proposed as an alternative to siamese loss formulations that works on entire class distribution rather than individual samples.[7] The loss is a likelihood ratio measure that forces separation in terms of each sample's distance away from the means of other classes. In its original proposition the loss takes on a multi-modal form. Here, a simpler, unimodal variation of this loss is employed as it better fits the single-shot person re-identification task.[16]

$$L_m(y, r) = \{-log\frac{e^{-\frac{1}{2e^2}||r-\mu_y||_2^2-m}}{\sum_{k\epsilon C(y)} e^{-\frac{1}{2e^2}||r-\mu_k||_2^2}}\}_+ \quad (4)$$

Algorithm DST:

Input: Track indices T = {1, . . . , N}, Detection indices D = {1, . . . , M}, Maximum age Amax
1: Compute cost matrix C = [ci,j] using weighted sum
2: Compute gate matrix B = [bi,j] using gating region
3: Initialize set of matches M ← ∅
4: Initialize set of unmatched detections U ← D
5: for n ∈ {1, . . . , Amax} do
6: Select tracks by age Tn ← {i ∈ T | ai = n}
7: [xi,j] ← min cost matching(C, Tn, U)
8: M ← M ∪ {(i, j) | bi,j · xi,j > 0}
9: U ← U \ {j | P i bi,j · xi,j > 0}
10: end for
11: return M, U

4. RESULTS

DeepSORT is the best tracker implemented as it takes the visual information of the object. The detections generated from the detection algorithm draws the bounding box on each object in every frame. The DeepSORT algorithm learns the behavior of the object while tracking and takes the information of the detection while tracking it. Hence, the DeepSORT algorithm is the best algorithm implemented out of the three algorithms as it visualizes the movement of the object being tracked and learns it.[10]

SORT algorithm is worse than the DeepSORT algorithm. Though it works upon the baseline of the IOU tracker but it is better than it. The SORT algorithm uses the velocity of the object that needs to be tracked. The SORT algorithm implements the Kalman Filter[5] that distinguishes the continuous changes between the frames of the video. The SORT Algorithm also implements the Hungarian Algorithm

The IOU Tracker is the worst tracker of the 3 algorithms as it does not uses any visual information of the object from the detections generated by the detection algorithms. The IOU Tracker only compares the area of the detections in subsequent frames to track an object.[17]

Tracker	MOTA	MOTP	MT	ML	ID	FM	FP	FN
IOU	57.1	77.1	14.80%	18.20%	1743	1846	12627	136078
SORT	59.8	79.6	25.40%	22.70%	1423	1835	8689	63245
DeepSORT	61.8	79.1	32.80%	18.20%	781	2008	12858	56668

The columns in the table represent:

- **MOTA:** Multi Object Tracking Accuracy combines three errors, i.e. false positives, false negatives and ID switches.

$$MOTA = 1 - \frac{\sum_i (FN_i + FP_i + ID_i)}{\sum_i Gnd} \quad (5)$$

  where i is the frame index and Gnd is the number of ground truth objects.
- **MOTP:** Multi Object Tracking Precision is the average dissimilarity between all true positives and their corresponding ground truth targets.

$$MOTP = \frac{\sum_{i,j} d_{i.j}}{\sum_i c_i} \quad (6)$$

  where c_i denotes the number of matches in frame i and $d_{i,j}$ is the bounding box overlap of target j with its assigned ground truth object.
- **MT:** Mostly Tracked Targets. A target is mostly tracked if it is successfully tracked for at least 80% of its life span. Note that it is irrelevant for this measure whether the ID remains the same throughout the track.
- **ML:** Mostly Lost Targets. If a track is only recovered for less than 20% of its total length, it is said to be mostly lost (ML). All other tracks are partially tracked.
- **ID:** ID Switches. If a ground truth object i is matched to hypothesis j at time t − 1 and the distance (or dissimilarity) between i and j in frame t is below td, then the correspondence between i and j is carried over to frame t even if there exists another hypothesis that is closer to the actual target. A mismatch error (or equivalently an identity switch, IDSW) is counted if a ground truth target i is matched to track j and the last known assignment was k ≠ j.
- **FM:** Fragmentation. The number of track fragmentations counts how many times a ground truth trajectory is interrupted (untracked). In other words, a fragmentation is counted each time a trajectory changes its status from tracked to untracked and tracking of that same trajectory is resumed at a later point.
- **FP:** False Positive is a false alarm of the output.
- **FN:** False Negative is a target that is missed by any hypothesis.

5. REFERENCES

[1] Wiman Nur Ibrahim, Aryo & Pang, Wee Ching & Seet, G & Lau, Michael & Czajewski, Witold. (2010). Moving Objects Detection and Tracking Framework for UAV-based Surveillance. Image and Video Technology, Pacific-Rim Symposium on. 456-461. 10.1109/PSIVT.2010.83.

[2] E. Bochinski, V. Eiselein, 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, 2017.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft. Simple Online and Realtime Tracking. In arXiv:1602.00763, 2016.

[4] N. Wojke, A. Bewley, D. Paulus. Simple Online and Realtime Tracking with a Deep Association Metric. In arXiv:1703.07402, 2017.

[5] G. Welch. G. Bishop. An Introduction to the Kalman filter in 1995.

[6] V. Eiselein, E. Bochinski, and T. Sikora. Assessing post-detection filters for a generic pedestrian detector in a tracking-by-detection scheme. In Analysis of video and audio "in the Wild" workshop at IEEE AVSS17, Lecce, Italy, Aug. 2017.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained partbased models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587, 2014.

[9] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler. ´ MOT16: A benchmark for multi-object tracking. arXiv:1603.00831 [cs], Mar. 2016. arXiv: 1603.00831.

[10] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globallyoptimal greedy algorithms for tracking a variable number of objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1201–1208. IEEE, 2011.

[11] 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, pages 91–99, 2015.

[12] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan. The clear 2006 evaluation. Multimodal Technologies for Perception of Humans, pages 1–44, 2007.

[13] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multi people tracking with lifted multicut and person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[14] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue. Evolving boxes for fast vehicle detection. Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 2017.

[15] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. DETRAC: A new benchmark and protocol for multi-object detection and tracking. arXiv CoRR, abs/1511.04136, 2015.

[16] 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 (CVPR), pages 1282–1289, 2014.

[17] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2129–2137, 2016