# Multi-modal Tracking using LIDAR and Visual Signals

Mohamed Hassan, Sanjay Kumar

***Abstract*—In recent years, Autonomous Vehicle obtain precise and real time information about objects in their vicinity which fully generates the safety of the passengers and vehicle in various environment. Moreover, Deep learning is the preferred analysis choice for scene understanding including applications such as Object detection and Tracking. Nowadays, multi-object tracking and segmentation (MOTS) depends on tracking-by-detection paradigm which is based on adopting convolutions for feature extraction. However, convolution-based feature extraction incontrovertibly mixes up the foreground and the background features which leads to ambiguities in resulting of subsequent instance association which, cause a fail in the long-term through either in crowded scenes due to 2D bounding boxes which might heavily overlap. In this project, we review on MOTS state-of-the-art approaches based on Tracking by-Segmentation which learns instance embeddings based on segments such as LIFTS [1], PointTrack [2], PointTrack++ [3]. Then, we focus on PointTrack approach which is a highly effective method for learning instance embeddings based on segments by extracting scattered 2D point-cloud from images. By analyzing these point-cloud it generates a new tracking-by-points paradigm that extracts instance embeddings from randomly selected points rather than images. PointTrack, surpasses all the previous state-of-the-art methods including 3D tracking methods by large margins (5.4% higher MOTSA and 18 times faster over MOTSFusion) with the near real-time speed (22 FPS) [2]. Furthermore, we verifed the effectiveness of this approach by testing it on various video sequances from KITTI dataset. To conclude, our research revealed that results obtained from quantitative and qualitative experiments using the KITTI dataset show the applicability and effectiveness of the proposed approach in various driving environments.**

## I. Introduction

Multi-object tracking and segmentation is significant in many applications ranging from autonomous vehicles, Robotics to video surveillance. The job of Multi-object tracking (MOT) is to locate multiple objects, maintaining their identities, and giving their trajectories of a given input video. Since recent years, object detection and association is getting difficult in very crowded scenes due to the frequent occlusions among objects. Although many approaches have been proposed to handle this problem, it still remains challenging. The recent approaches, to overcome these challenges operate by extending MOT by jointly taking into account instance segmentation and tracking [1]–[3] leading to Multi-object tracking and segmentation approaches (MOTS) a prominent research field in Computer Vision. MOTS provides pixel level segmentation; it is a technique used to categorize what is given in an image at pixel level. Beside pixel level segmentation, MOTS aims to acquire a more effective and scalable approach for object association established on segments compared to tracking by detection, which connects detected bounding boxes across frames via data association algorithms. In this report, we start by reviewing State-of-the-art methods for MOTS as we can see from igure [1]. Unfortunately, how to extract instance embedding features from segments that have rarely been tackled by current MOTS methods. TrackR-CNN [4] extends Region-based CNN [8] mask by 3D convolutions and adopts ROI Align to extract instance embeddings in bbox proposals. However, as affected by the receptive field of convolutions, the foreground features and the background features are still mixed up, which is harmful to learn discriminative features.

On the other side, as shown in the Figure [1], PointTrack is performing well because it overcome the discussed problem by using spatial embedding [6] approach by doing instance segmentation and using the unique training architecture inspired from PointNet [5] approach which allows feature accumulation directly from irregular formatted 3D point-cloud. But PointTrack is generating unordered 2D point-cloud from 2D image pixels and learning instance embedding and point-cloud processing. Expanding the bounding box (20%) to include more environment information that is used to learn context-aware embedding. After that uniformly samples 1000 points on the object and 500 points on the background then use color, position, offset from center, and class category to encode each 2D point in the point-cloud. We ended up by selecting with the PointTrack approach due to its high robustness and efficiency and learning instance embedding on segments, also its performance is almost near to real time [2].

The rest of the report is organized as follows: Section II explains the tools and software used during all the process. Section III makes a brief review of state-of-the-art and relevant papers used. Section IV briefly presents the method used to obtain the goal. Section V presents and discusses experimental results leading to a set of conclusions in section VI.

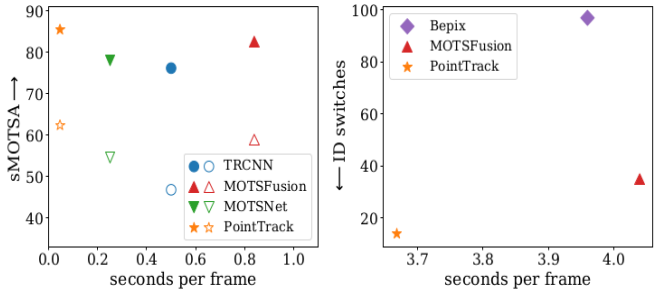


Fig. 1. The figure shows a comparsion between proposed PointTrack and stateof-the-art MOTS methods. We can see on the left figure sMOTSA and on the right the number of id switches. Different color shapes in the left figure denoted the performance of cars and for pedestrians respectively. On the right figure, all methods implement tracking on the same segmentation result. This figure is adapted from PointTrack [2].

## II. Preliminaries and Tools

Tracking becomes more challenging with the increase of instances. For this reason, PointTrack creates a new challenging dataset APOLLO MOTS [2] based on the public ApolloScape [9] dataset. APOLLO MOTS [2] provide a more challenging MOTS dataset for both 2D tracking and 3D tracking compering to KITTI [10] dataset. APOLO MOTS [2] dataset presents 68% higher instance density compared to KITTI dataset as shown in Table [7]. PointTrack uses APOLLO MOTS [2] for the training phase to increase the tracking challenge, Also it uses other MOTS datasets for evaluation such as KITTI [10] dataset. For evaluating the PoinTrack model, We selected five video sequences from KITTI MOTS [10] dataset contains various tracking challenges such as particle occlusion, full occlusion, scale variance and appearance variance.

We used Python3 and PyTorch framework to test the pre-trained model on the KITTI dataset and generating an output video for tracking. The hardware we used for deploying our model is NVDIA GPU (GTX1660) which allows us to use CUDA in PyTorch to accelerate the computational performance of our model. while retraining the model using different parameters on a kitti [10] dataset, we faced a technical issue, the laptop started to overheat because our hardware is not sufficient for retraining on this huge dataset. To overcome this issue, we used a pre-trained model from PointTrack as inference for doing the evaluation.

## III. Related work

### A. Tracking-by-detection:

Detection-based approaches detect objects and then link objects into trajectories via data association. However, the main problem is that 2D bounding boxes can overlap heavily in crowded scenes which can affect the tracking performance badly. Since instances may heavily overlap each other, the predictions are very likely to be mistakenly suppressed by Non- maximum suppression (NMS). Data association highly depends on the robustness of the similarity measurement. Hence, the main problem is that 2D bounding boxes can overlap heavily in crowded scenes which can affect the tracking performance badly. Consequently, the challenges remain in MOT especially for crowded scenes which is the case for autonomous driving applications as shown in figure 2.

### B. Tracking-by-segmentation:

After reviewing and understanding the state-of-the-art about MOTS [1]–[3] using Tracking by segmentation approach which overcome the heavily 2D bounding boxes overlapping which happen in crowded scenes in Tracking by detection approach. We ended up with three main papers which are showing pretty well performance, Firstly, Lidar and Monocular Image Fusion for Multi-Object Tracking and Segmentation (LIFTS) [1]. Secondly, Segment as Points for Efficient Online Multi-Object Tracking and Segmentation (PointTrack) [2]. Finally, PointTrack++ for

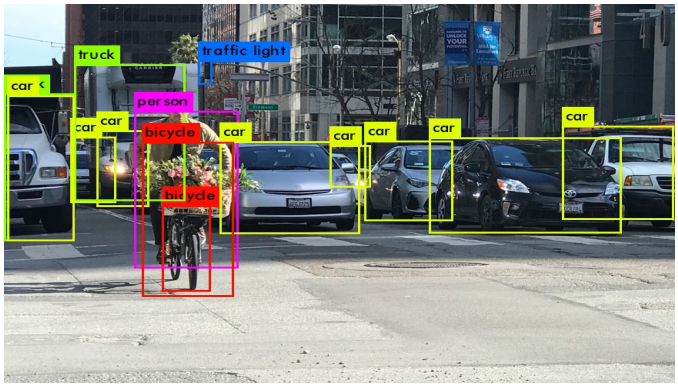


Fig. 2. Overlapping between bounding boxes [12]

Effective Online Multi-Object Tracking and Segmentation [3].

In recent years, the computer vision society has made significant progress in multi-object tracking (MOT) and video object segmentation (VOS) respectively. Further progress can be achieved by effectively combining the detection, segmentation and tracking together. LIFTS (Lidar and monocular Image Fusion based multi-object Tracking and Segmentation) [1] is the multi-stage framework. The approach is mainly consisting of 3 stages. Firstly, detector Network for 3D Part-Aware and Aggregation on the point-cloud data for getting 3D object locations. The network uses both CNN appearance features and objects spatial information of detections for doing robust associate objects along time. Secondly, Region-based CNN [8] network involves a Cascade Mask on the input monocular data with PointRend [11] head for getting instance segmentation results. then, using Hungarian algorithm to bridge 3D world space and 2D image space. Thirdly, optical-flow guided instance segmentation used for generating the mask results. for ID switches recover object re-identification (ReID) is applied to overcome long-term occlusions. KITTI-MOTS [10] dataset was used for Evaluating it, LIFTS [1] can achieve a 79.6 sMOTSA for Cars and 64.9 for pedestrians. Moreover, LIFTS [1] approach for fusing lidar data with image data, may help us in the future improvements for PointTrack.

PointTrack [2] is an online MOTS framework. It beats the state of art methods by a large margin with practically real time speed 22 FPS and 5.4% higher MOTSA and 18 times faster over MOTSFusion as we can see in table 1 and 2 which are taken from Kitti Benchmark website [7]. The effectiveness and efficiency of PointTrack is evaluated on three different datasets. PointTrack introduced their own challenging crowded dataset called APOLLO based on the observation that previously used MOTS datasets were less crowded.

Unlike PointTrack [2], proposed PointTrack++ [3] introduce, three improvements. Firstly, in instance segmentation phase, in order to increase the instance selection features, they implement a semantic segmentation decoder which is trained using a focal loss. Secondly, PointTrack

TABLE I
RESULTS ON KITTI DATASET FOR CARS

| Method | sMOTSA | MOTSA | MOTSP | MODSA | MT | ML | IDS | Frag |
|---|---|---|---|---|---|---|---|---|
| LIFTS | 79.6 | 89.6 | 89 | 89.9 | 79.1 | 2.9 | 346 | 532 |
| PointTrack | 87.5 | 90.9 | 87.1 | 91.82 | 90.84 | 0.6 | 270 | 538 |
| PointTrack++ | 82.8 | 92.6 | 89.73 | 93.35 | 89.49 | 1.20 | 114 | 579 |

TABLE II
RESULTS ON KITTI DATASET FOR PEDESTRIANS

| Method | sMOTSA | MOTSA | MOTSP | MODSA | MT | ML | IDS | Frag |
|---|---|---|---|---|---|---|---|---|
| LIFTS | 64.90 | 80.90 | 81.00 | 81.90 | 61.5 | 8.90 | 206 | 277 |
| PointTrack | 61.74 | 76.5 | 80.96 | 77.36 | 48.9 | 9.26 | 176 | 716 |
| PointTrack++ | 68.13 | 83.67 | 82.22 | 84.88 | 66.67 | 4.81 | 250 | 536 |

copy and paste instances for data augmentation into training images, to increase the segmentation performances. Finally, in instance association stage, they propose best training approach in order to differentiate learned instance embedding.

Lidar sensor point-cloud can provide us with the depth information for each point, which is an advantageous for increasing the tracking robustness, by adding depth information provided by the Lidar to the feature vector that were extracted from the image using the PointTrack [2] method.

## IV. A MORE DETAILED DESCRIPTION POINTTRACK APPROACH

The approach proposed in PointTrack paper [2] is highly effective for learning unique instance embeddings on segments by extracting un-ordered 2D point-clouds from the 2D image. PointTrack [2] presented new tracking-by-points model, in which the learning is from scattered selected points instead of using the whole image. This approach is implemented in two main steps. Firstly, context-aware instance embeddings extraction. Secondly, instance segmentation with Temporal Seed Consistency

### *A. Context-aware instance embeddings extraction*

Instance segmentation provide us with the instance vector and for each instance they surrounding it with smallest possible rectangle to cover the instance, then they enlarge the rectangle size in all the dimension by scale factor $(k)$ in border in all four directions $(top, down, left, right)$. After that, we consider the foreground and environment segments as 2D point-cloud. Each point consist of six dimensional data space $(u, v, R, G, B, C)$, the image plane coordinate are represented in $(u, v)$, the color of the pixel is represented in RGB and $(C)$ is representing the pixel class. For covering the foreground segment they used random point sample consists of 1000 points which allows to cover relative large instance, and they used same methodology for covering the environment by generating 500 point. For calculating center point $(uc, vc)$, they getting the mean of the foreground points, which is helping them in calculating the offset $(u, v)$ for each point by subtracting the point coordinates from the center point coordinate. Then for calculating the RGB, they just extract the RGB value from this point pixel. For increasing the environment context into point-wise feature they mention the background class in $(C)$. Finally, by combining the previous data they are extracting feature vector for each point. Multi-Layer perceptron network are used for learning foreground and environment embedding which based on feature vector that consist of the above four data modalities, context-aware instance embeddings extraction architecture can be seen in Figure 3.

### *B. Instance segmentation with Temporal Seed Consistency*

For instance segmentation, PointTrack uses the state of the art method name SpatialEmbedding [6], which does not use bounding box proposals for performing instance segmentation. In addition to that, SpatialEmbedding [6] shows fast performance due to using a one stage method. However, they mentioned that the segmentation between consecutive sequential frames results in many false positive and false negative. Therefore, PointTrack introduced temporal consistency loss by penalizing the difference between the warped seed from last frame with optical flow and the seed predicted from current frame during the training, which improves the predicted seed map quality and robustness.

## V. EXPERIMENTAL STUDY

For testing and evaluating the PointTrack model, KITTI [10] dataset contains a suite of vision tasks built using an autonomous driving platform. The full benchmark contains many tasks such as object detection and tracking. This dataset contains 7481 training images annotated with 3D bounding boxes, divided into 21 videos, each video contains various tracking challenges such as particle occlusion, full occlusion, scale variance and appearance variance. Moreover, moving objects in videos are a combination between cars, trucks and pedestrians. Hence, we are focusing on car tracking, we selected 5 videos that almost cover all the previous tracking challenges for car tracking in an urban environment are shown in table 5, for test and evaluate PointTrack [2] on them. We

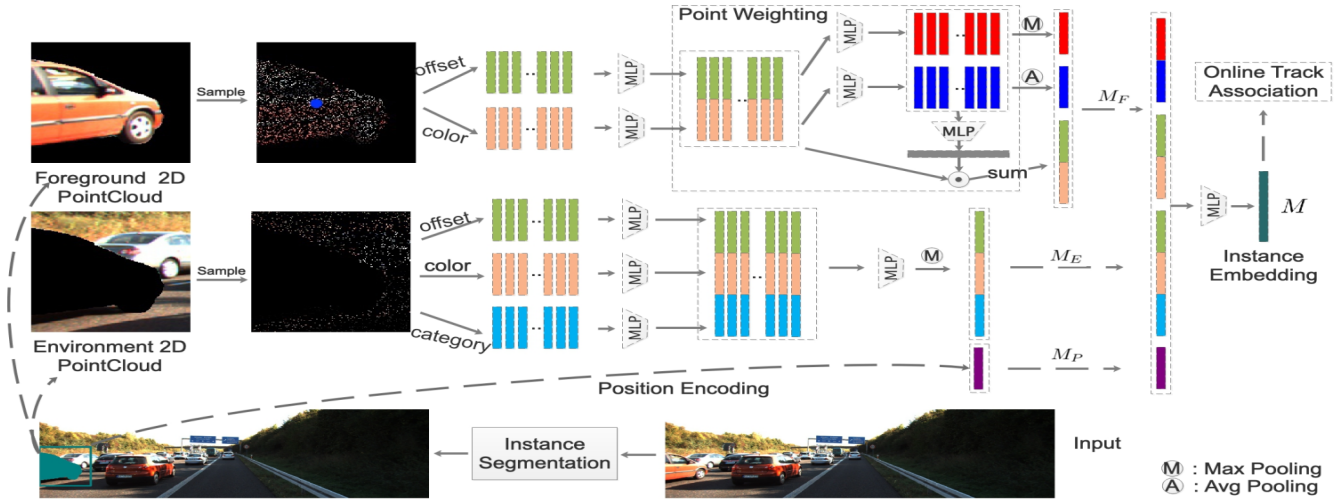


Fig. 3. Overview of PointTrack. For an input image, PointTrack obtains instance segments by an instance segmentation network. Then, PointTrack regards the segment and its surrounding environment as two 2d point-cloud and learn features on them separately. MLP stands for multi-layer perceptron with Leaky ReLU. This Figure is adapt from PointTrack [1]

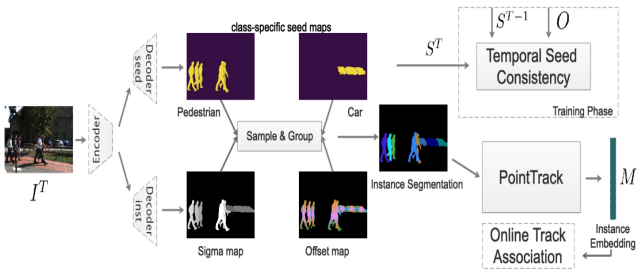


Fig. 4. The Figure shows Segmentation network of PointTrack. this Figure is adapt from PointTrack [1].

follow the same order of the videos name from KITTI [10] dataset.

### A. Dataset and challenges:

Below we are discussing the details and challenges in the five videos sequences from the KITTI dataset that we used for testing, the main challenges in each video sequence are shown in table 5.

*Video*2: In this video sequence, the camera is moving towards the four-way junction. The main challenges in this sequence are partial and full occlusion among cars that are moving against each other, another challenge is the scale of the object change while the camera moves.

*Video*6: In this video sequence, the moving camera is stopped at the red traffic light and other cars are moving further away, after some second car start and turn left. The main challenges in this sequence are partial occlusion between cars and street light beams, another challenge is the scale size of the car.

*Video*8: In this video sequence, the moving camera is crossing the highway road. The main challenges in this sequence are detecting close, far and the car coming from opposite direction.

*Video*13: In this video sequence, the moving camera is going across the street where there are lot of pedestrian but few amount of parking cars.

*Video*16: In this video sequence, the moving camera is stopped at crosswalk point. The sequence contains parking cars and many pedestrians are crossing the road. The main challenges are pedestrian making occlusion with parking cars that we want to track.

### B. Experiment Strategy:

We are evaluating the tracker based on 2 evaluation criteria. Firstly, the resulting evaluation matrix as shown in table 3 and 4. Secondly, observing the failure behaviour in the video sequence.

### C. Evaluation Matrix Explanation:

In this section, we discuss the performance of each video depend on the evaluation matrix in detail. we focus on these parameters $(MOTSA, FP, MT, PT, F1)$ which are discussed in table 6, for the evaluation parameters.

Among the five videos that we select for testing and evaluation, *video*6 perform outclass than the rest of the four videos. We can see from tables III and IV, MOTSA accuracy of *video*6 is 97.% which is higher than other videos. Also, we observed that as compared to other videos, the false-positive parameter rate was low $(FP = 4)$ in video 6, the reason is few trucks are detected as cars. In terms of, mostly tracked (MT) and partially tracked (PT) parameters all videos showed 100% and

TABLE III
EVALUATION MATRIX

| Videos | sMOTSA | MOTSA | MOTSP | MOTSAL | MODSA | MODSP | Recall | Prec | F1 | FAR | MT | PL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Video2 | 77.26 | 91.14 | 85.22 | 91.61 | 91.69 | 87.16 | 93.91 | 97.70 | 95.77 | 8.58 | 100.00 | 0.00 |
| Video6 | 89.75 | 97.77 | 91.89 | 97.77 | 97.77 | 92.75 | 98.98 | 98.88 | 98.88 | 2.22 | 100.00 | 0.00 |
| Video8 | 89.06 | 97.41 | 91.48 | 97.57 | 97.60 | 91.32 | 97.98 | 99.61 | 98.79 | 1.03 | 90.48 | 9.52 |
| Video13 | -9.34 | -2.78 | 92.84 | -2.78 | -2.78 | 99.30 | 91.67 | 49.25 | 64.08 | 10.00 | 100.00 | 0.00 |
| Video16 | 69.80 | 88.37 | 79.62 | 89.00 | 89.09 | 79.58 | 91.13 | 97.81 | 94.35 | 8.13 | 100.00 | 0.00 |

TABLE IV
EVALUATION MATRIX

| Videos | ML | TP | FP | FN | IDS | Frag | GT Obj | GT Trk | TR Obj | TR Trk | Ig TR Tck |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Video2 | 0.00 | 848 | 20 | 55 | 5 | 26 | 903 | 15 | 1201 | 11 | 333 |
| Video6 | 0.00 | 1021 | 4 | 21 | 2 | 10 | 1042 | 21 | 1216 | 15 | 191 |
| Video8 | 0.00 | 531 | 6 | 6 | 0 | 3 | 537 | 11 | 756 | 6 | 219 |
| Video13 | 0.00 | 33 | 34 | 3 | 0 | 1 | 36 | 2 | 124 | 5 | 57 |
| Video16 | 0.00 | 760 | 17 | 74 | 6 | 38 | 834 | 4 | 783 | 9 | 6 |

Fig. 5. This figure shows snapshots from videos $(2, 6, 8, 13, 16)$ respectively from top to down.

0% accuracy respectively, except *video*8 where the accuracy of MT and PT is 90.48% and 9.52% respectively. However, *video*13 showed the worst performance with MOTSA accuracy $-2.78\%$ and false positive rate were also high ($FP = 34$), the reason for low performance is that all the trucks detected as car, hence performance dropped significantly. Regarding the F1 parameter which is calculated from precision and recall, *video*6 showed greater accuracy at 98.88% than others, while *video*13 showed low accuracy 64.08%. In conclusion, the PointTrack pre-trained model shows acceptable performance for overcoming the tracking challenges that are shown in table 5. However, in the next section, we discussed the reasons that lead to low performance in some videos.

### *D. Failure Behaviour Explanation:*

*Video*2: In this video sequence, we observed that the far away cars which have small scale are not detected at all as shown in Figure 6. Moreover, when cars crossed each-other in opposite direction model failed to keep the same instance association, because camera detected the car from different angle after the full occlusion that happened, as shown in Figure 7 and 8.

Fig. 6. video 2: Not detecting far away objects

Fig. 7. video 2: Changing identity after occlusion

Fig. 8. video 2: Changing identity after occlusion

*Video*6 : In this video sequence, we observed that initially cars are detecting by the camera. However, when they start to move away, the model failed to detect them as shown in Figure 8. In addition to, truck which is coming from opposite direction is detected as car (False positive) as shown in Figure 9.

Fig. 9. video 6: Not detecting far away objects

Fig. 10. video 6: detecting truck as a car

*Video*8: In this video sequence, we observed that model is detecting the truck as a false positive for small amount of time, and also failed to detect the far cars as shown in Figure 11.

Fig. 11. video 8: Not detecting far away objects and detecting truck as a car

*Video*13: In this video sequence, we observed that model is detecting the truck as a false positive for small amount of time, as shown in Figure 12.

*Video*16: In this video sequence, we observed that after the full occlusion that happened from the moving pedestrian on the parking cars, the model detects one of the parking car as a new car (first it shows label as car 3 then it changes it's label to car 9) as shown in Figures 13 and 14.

Fig. 12. video 13: detecting truck as a car


Fig. 13. video 16: Changing identity after occlusion

Fig. 14. video 16: Changing identity after occlusion

### *E. Discussion:*

From the analysis of video sequences, we observed that the model performed pretty well in most of the video sequences as shown in tables 3 and 4. However, model does not perform well in some situations. We can see far away objects are not detected as shown in Figures 1 and 11, the model sometimes can't keep the same instance id after fully occlusion specially if the car appears from different angle after the occlusion as shown in Figure 7 and 8. Also we observed that model is detecting the truck as a false positive for small amount of time, as shown in Figure 10 and 11. To overcome the problem that we mentioned before during the tracking, for that we need to train the model on a small scale car size and increase the truck datasets. Moreover, we are looking forward to increase the tracking performance by using lidar sensor point-cloud and try to fuse them with the generated point-cloud from the image which will allow us to propose the depth in the feature vector.

## VI. Conclusions

To conclude, tracking by segmentation shows better performance than tracking by detection which fails during overlapping in the crowded scene. After reviewing the state of the art papers for MOTS [1]–[3], we ending up with PointTrack [1] approach that propose a new tracking by point model which shows high efficiency for instance embedding learning by breaking the image into un-ordered 2D point-cloud. PointTrack [1] uses APOLLO MOTS dataset for training the model in crowded scene. Moreover, in testing the model produces a

TABLE V
EVALUATION PARAMETERS

| Parameter | Definition |
|---|---|
| MOTSA | Multiple Object Tracking Accuracy combines four different paramteres Id switch , false negatives, false positives and ground truth |
| FP | False positive is an error in which a test result incorrectly indicates the presence of a condition such as object when the object is not present |
| MT | Mostly tracked which is the Percentage of GT trajectories which are mostly covered by tracker output |
| PT | Partially tracked which is the Percentage of GT trajectories which are partically covered by tracker output |
| F1 | a measure of a test's accuracy. It is calculated from the precision and recall of the test |

TABLE VI
MAIN CHALLENGES IN TEST DATASET

| Video | Particle Occlusion | Full Occlusion | Scale Variance | Appearance Variance |
|---|---|---|---|---|
| Video 2 | ✓ | ✓ | ✓ | ✓ |
| Video 6 | ✓ | ✗ | ✓ | ✓ |
| Video 8 | ✗ | ✗ | ✓ | ✓ |
| Video 13 | ✓ | ✗ | ✓ | ✗ |
| Video 18 | ✓ | ✓ | ✗ | ✗ |

TABLE VII
RESULTS ON KITTI DATASET FOR PEDESTRIANS

| | Frames | Tracks | Annotations | Car density | Crowd cars | Frames per second |
|---|---|---|---|---|---|---|
| APOLLO MOTS | 11488 | 1530 | 64930 | 5.65 | 36403 | 10 |
| KITTI MOTS (Car) | 8008 | 582 | 26899 | 3.36 | 14509 | 7 |

good tracking performance in most of the MOT challenges however fails in some situations such as not detecting the far away objects.

Future work: To overcome the problem that we mentioned before during the tracking such as tracking scale objects, id switches after full occlusion. For tracking small-scale objects there are various solutions, such as extracting a point of interest (POI) using flexible detector-descriptor like SURF [13] which can extract the POI in different scale levels. We can apply that on PointTrack instead of, extracting random points, we can extract these POI instead. On the other side, that may decrease PointTrack speed. Lidar data could significantly improve the tracking robustness [1], which allows reducing the id switches that happened after full occlusion. However, the state-of-the-art methods [1] for fusing lidar data with visual data, deal with each data separately then fuse the results, which increases the processing time. Our idea is how to match the extracted 2d point-cloud from an image using PointTrack, with the lidar points which allows to include the depth information to the embedded feature vector which increases the ability to learn discriminative features. However, the matching process is still a challenge, we could project the 3d point-cloud on the 2d image then matching the closest image points with lidar points. Noticed that the previously mentioned proposals still under study and prototyping.

## REFERENCES


[1] Zhang, Haotian, et al. "LIFTS: Lidar and Monocular Image Fusion for Multi-Object Tracking and Segmentation." BMTT Challenge Workshop, IEEE Conference on Computer Vision and Pattern Recognition. 2020.

[2] Xu, Zhenbo, et al. "Segment as points for efficient online multi-object tracking and segmentation." European Conference on Computer Vision. Springer, Cham, 2020.

[3] Xu, Zhenbo, et al. "PointTrack++ for Effective Online Multi-Object Tracking and Segmentation." arXiv preprint arXiv:2007.01549 (2020).

[4] Voigtlaender, Paul, et al. "Mots: Multi-object tracking and segmentation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[5] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[6] Neven, Davy, et al. "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[7] cvlibs.net/datasets/kitti.

[8] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R- ´ CNN. In ICCV, 2017. 2, 3, 5

[9] Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., Yang, R.: The apolloscape dataset for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 954–960 (2018)

[10] Geiger, Andreas, et al. "Vision meets robotics: The kitti dataset." The International Journal of Robotics Research 32.11 (2013): 1231-1237.

[11] Kirillov, Alexander, et al. "Pointrend: Image segmentation as rendering." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[12] https://bit.ly/3bKxFt9.

[13] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." European conference on computer vision. Springer, Berlin, Heidelberg, 2006.