Seg2Pose: Pose Estimations from Instance Segmentation Masks in One or Multiple Views for Traffic Applications

Martin Ahrnbom[a], Ivar Persson[b] and Mikael Nilsson[c]
Centre for Mathematical Sciences, Lund University, Sweden

Keywords: Pose Estimation, Instance Segmentation, Convolutional Neural Network, Traffic Safety, Road Users, Tracking, Stereo Camera, Trinocular Camera Array, Traffic Surveillance.

Abstract: A system we denote Seg2Pose is presented which converts pixel coordinate tracks, represented by instance segmentation masks across multiple video frames, into world coordinate pose tracks, for road users seen by static surveillance cameras. The road users are bound to a ground surface represented by a number of 3D points and does not necessarily have to be perfectly flat. The system works with one or more views, by using a late fusion scheme. An approximate position, denoted the normal position, is computed from the camera calibration, per-class default heights and the ground surface model. The position is then refined a novel Convolutional Neural Network we denote Seg2PoseNet, taking instance segmentations and cropping positioning as its input. We evaluate this system quantitatively both on synthetic data from CARLA Simulator and on a real recording from a trinocular camera. The system outperforms the baseline method of only using the normal positions, which is roughly equivalent of a typical 2D to 3D conversion system, in both datasets.

1 INTRODUCTION

Object detection and tracking in world coordinates is a core computer vision task that has received a significant amount of attention in recent years, in particular for robotics and autonomous driving scenarios (Brazil et al., 2020; Kumar et al., 2021; Muller et al., 2021; Yin et al., 2021). A somewhat less explored topic is for surveillance scenarios where the camera is stationary and a full extrinsic and intrinsic camera calibration is available. A part of the reason for this is that decent approximate solutions for converting pixel coordinate positions to world coordinate positions of objects exist and are quite trivial, but obtaining high precision world coordinate positions of road users in a surveillance video remains a challenging task.

A common approach for obtaining world coordinate locations of objects is to first locate them in pixel coordinates, and using camera calibration to convert the 2D position into 3D. The commonly used Axis-Aligned Bounding Box (AABB) does not capture the details of the position of objects very well. Therefore, we utilize an object detector which provides instance segmentations, containing more spatial information than AABBs, that can be useful when converting to world coordinates. Detectron2's implementation of Mask R-CNN (Wu et al., 2019; He et al., 2017) was chosen for this task. It also produces high quality detections for many scenes even when not fine-tuned to the particular dataset used, including the videos used in Section 4.

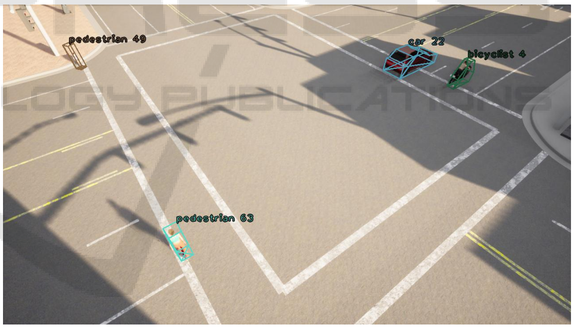

Figure 1: Result of our method, with the T-style Trinocular Seg2PoseNet configuration, on a CARLA Simulator image.

Furthermore, reasonably good tracking methods exist that use instance segmentations directly (Ahrnbom et al., 2021b; Yang et al., 2020). Thus, in order to obtain world coordinate tracks, the remaining task is to convert segmentation masks into world coordinates. Such a system has important applications for

[a] https://orcid.org/0000-0001-9010-7175
[b] https://orcid.org/0000-0003-4958-3043
[c] https://orcid.org/0000-0003-1712-8345

Ahrnbom, M., Persson, I. and Nilsson, M.
Seg2Pose: Pose Estimations from Instance Segmentation Masks in One or Multiple Views for Traffic Applications.
DOI: 10.5220/0010777700003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 777-784
ISBN: 978-989-758-555-5; ISSN: 2184-4321

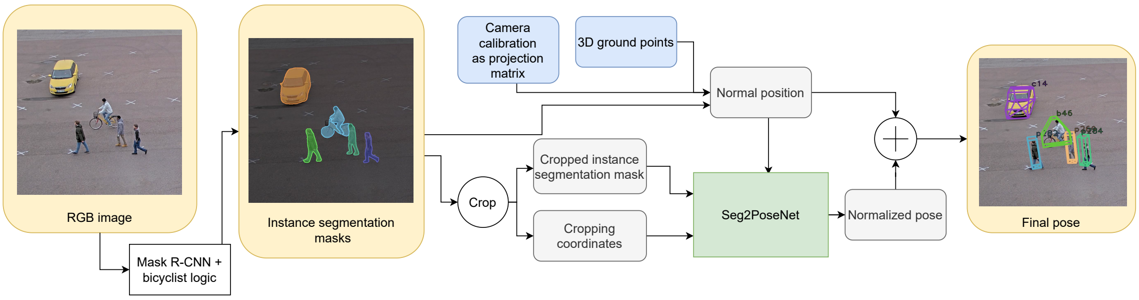
Camera calibration as projection matrix
3D ground points
Normal position
RGB image
Mask R-CNN + bicyclist logic
Instance segmentation masks
Crop
Cropped instance segmentation mask
Cropping coordinates
Seg2PoseNet
Normalized pose
Final pose

Figure 2: An overview of Seg2PoseNet and how it fits into our suggested pipeline. An instance segmentation mask is cropped and sent into Seg2PoseNet, alongside the normal position computed from the mask (See Section 3.2). The normalized output from Seg2PoseNet is added to the normal position to obtain the final pose in world coordinates. Best viewed in color.

traffic safety research. Obtaining high-precision positions of road users in videos allows for applications like safety estimation and traffic monitoring.

For this purpose, we propose a framework, **Seg2Pose**, including a Convolutional Neural Network (CNN), **Seg2PoseNet**, which takes as input an instance segmentation mask and produces the world coordinate position of the road user. By looking at multiple positions in a track, we estimate the orientation of the road user by assuming that they are facing the same direction as they are moving. Cases like when a car goes in reverse or drifts sideways are thus not covered in this work, although extending our approach to handle such cases should be possible.

A second goal of our work is to evaluate if using segmentation masks from more than a single camera view improves the positioning of road users. We therefore test our system with monocular, stereo and several types of trinocular (three-view) cameras, and evaluate how much the positioning improves when seeing the road user from different angles. We only consider camera designs where the lenses are close, allowing a single device to contain all the cameras.

We train and evaluate our Seg2PoseNet system on purely synthetic data from CARLA Simulator (Dosovitskiy et al., 2017), and then perform tests on recordings from a scene captured by a three-lens camera prototype device in reality, training on and comparing the results with manually created ground truth poses created with T-Analyst software (Lund University, Transport and Roads, 2018). An overview of our system is shown in Figure 2.

Our main contributions are as follows:

- We demonstrate how segmentation masks can be used to accurately find world coordinate positions in calibrated cameras.
- We evaluate different camera configurations against the baseline of using only normal positions, on both synthetic and real data.

2 RELATED WORK

2.1 Pose Estimation

6D pose estimation in general refers to the problem of estimating both the x, y and z coordinates as well as three rotational parameters in world coordinates, given some representation in pixel coordinates in one or more images. It is often assumed that accurate 3D models of the objects are known and used as input to such a system, and sometimes these models even have textures. Examples of work in this field are (Hodan et al., 2020; Zakharov et al., 2019). This task is in many ways similar to the road user pose estimation problem explored in this paper, but they differ in a number of critical ways:

- Our method does not require high quality 3D models of the road users.
- Road users only move along a ground surface, represented by a set of ground points, described in Section 3.1. Thus z is a function of x, y, so only two positional parameters need to be estimated.
- It is assumed that the rotation of a road user is determined by the changes in position over time.
- Seg2PoseNet does not require RGB images, as that is handled by Mask R-CNN.

2.2 Traffic Surveillance

One example of world coordinate traffic surveillance with modern computer vision tools is the STRUDL system (Jensen et al., 2019). It uses an approximate method for converting 2D into 3D, by picking a point in the AABB and assuming it to be on a fixed height above the ground, and computing a single 3D point from there. This method is essentially equivalent to the normal positions described in Section 3.2. Their focus were on making the whole system easily usable though, and not on the particular task for 2D to 3D conversion.

An example of a more detailed 3D pose estimation in a traffic environment is UTS (Bradler et al., 2021) which estimates simplified 3D models similar to the ones in this paper, but estimates them from AABBs and employs active edge detection to estimate the pose. Their approach is limited to motorized road users and only considers a monocular view. An unscented Kalman filter is used to produce physically reasonable tracks.

Another example is (Zhang et al., 2020) which has the benefit of being trained to be robust to intrinsic camera parameters, but requires both a 3D point cloud and known 3D models to estimate poses, is only tested on cars and only with a monocular camera.

Yet another example is (Zhang and Zhang, 2020), where 3D bounding boxes are estimated for only cars, buses and trucks, and they require seeing wheels to correctly place the pose, limiting the flexibility of their method. The scene used in Section 4.2 contains cars seen almost entirely from above, with no wheels visible. They also assume a perfectly flat ground surface, and only evaluated with monocular cameras, further limiting their flexibility.

In summary, while pose estimation of road users in traffic environments have been explored, to the best of our knowledge we are the first to publish a method which is flexible both in terms of which road user classes it works with, how many cameras are used, does not require detailed 3D models of the road users and uses segmentations as a detailed 2D description of the position, rather than the less detailed AABBs of other methods.

3 OUR SYSTEM: Seg2Pose

3.1 Ground Surface Representation

The ground surface is represented by n many 3D points (x_i, y_i, z_i). The ground height above any given top-down position (x, y) is given by

$$z(x,y) = \frac{\sum_i^n z_i e^{-\alpha\sqrt{(x_i-x)^2+(y_i-y)^2}}}{\sum_i^n e^{-\alpha\sqrt{(x_i-x)^2+(y_i-y)^2}}} \quad (1)$$

where α is a smoothing parameter, set to 0.3. This representation allows an accurate model for the ground for all typical traffic environments except those with multiple stacked ground levels, for example a bridge over a road. Such situations could be handled by treating the bridge and the road below as two separate ground surfaces. There exists devices designed for capturing highly accurate 3D points in real traffic environments, such as the Leica S06 (Leica Geosystems AG, 2009) used in our experiments.

3.2 Normal Positions

The *normal position* of a road user seen in one camera view is defined as the world coordinate point found by the following algorithm. Take the center point of the segmentation mask, in pixel coordinates, and compute the intersection of a camera ray going through that point and hitting the ground surface, here approximated by a plane, raised to the default height of that road user class. This position is fine-tuned by an optimization over the actual ground surface, again raised to the default height, until a world coordinate point is found that projects into the center point of the segmentation. The default heights for each road user class are computed by going through the training set, and testing different default heights with the bisection method until approximately half of their normal positions are closer than the true positions, and the other approximate half are further away.

3.3 Seg2PoseNet

The job of Seg2PoseNet is to fine-tune the normal positions by computing a position residual added to the normal positions of detected road users. The network takes two inputs:

- A cropped segmentation mask to the size 420×420, zero-padded if close to the border.
- A vector of four elements containing the pixel coordinate center position of as cropped segmentation mask, as well as the normal position.

This way, the network has access to both the appearance of the road user, as well as the region of the image it comes from. It should thus have all the information it needs to learn an accurate mapping between pixel and world coordinates for road users in a given scene. The network design is shown in Figure 3.

When estimating a road user's position from multiple views, a late fusion strategy is applied. Seg2PoseNet is applied to each view's mask independently, and the output is added to the normal position for each view to obtain several, independent estimates of the true position. The final estimate is simply the average of the estimates from the different views.

3.4 Estimating Rotations

When computing the poses for road user tracks, the world coordinate position of a given road user ID is

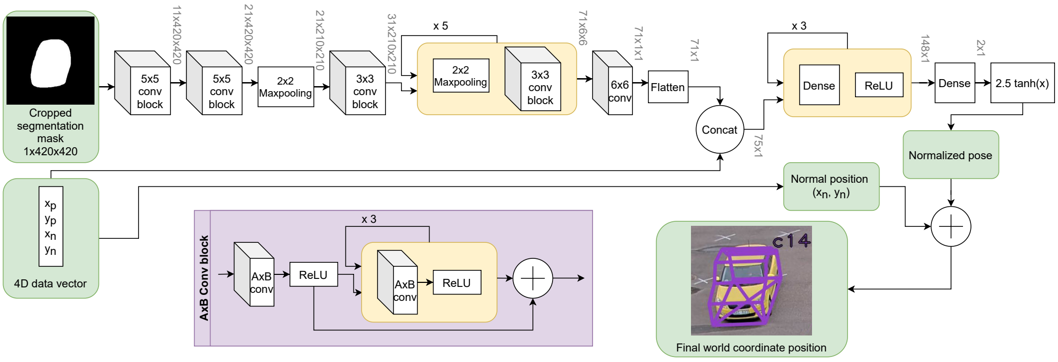

Figure 3: The design of Seg2PoseNet. The sizes of the feature maps and feature vectors are shown in gray. The purple box at the bottom explains what is meant by a “conv block”.

stored. If a previous position exists for that road user ID, the difference is computed. If that difference is larger than 2 cm, the direction is updated to be

$$D_{t+1} = 0.1D + 0.9D_t \tag{2}$$

where D is the normalized difference, D_{t+1} is the direction for the new frame and D_t is the direction of the previous frame. These are two-dimensional vectors in world coordinates.

3.5 Rough 3D Models

During training, every fourth training example was a rough, simple 3D model, as shown in Figure 4, placed in a random position and orientation in the scene. These models were rasterized into segmentation masks. This helps the network learn something about how to position road users in parts of the image that do not appear in the training set.

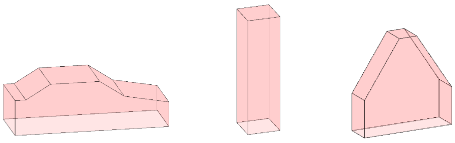

Figure 4: Rough 3D models for car, pedestrian and bicyclist, from left to right. These simple 3D models, are used in alongside the datasets’ training examples.

4 EXPERIMENTS

We performed two separate experiments for different goals. One experiment was done on purely synthetic data from the CARLA Simulator, where an exact ground truth position is known for every visible road user. The purpose of this experiment is to get an accurate numerical evaluation of our system, and how the different camera designs affect the performance of our positioning system. The other experiment was made using a real recording to verify that the system works in reality.

4.1 CARLA Simulator

A road junction in the default map for CARLA Simulator version 0.9.11 (Dosovitskiy et al., 2017) was selected, and 83 road users (43 pedestrians, 12 bicyclists and 28 cars) were placed into the environment and their movements were simulated. This scene was captured by five cameras, positioned to allow the recreation of different camera configurations according to Figure 5. These cameras were recording images every 0.075 seconds in sync, for 22.000 frames. The relatively low frame rate increases the variety in each frame without increasing the time to perform the experiment. This video was then divided into a training, validation and testing with 7000 frames in each set. Between the sets, 150 frames were discarded to reduce similarity between the sets.

In addition, 184 points were sampled from the ground surface around the area visible in the camera views. Particularly many were sampled near the edges of the sidewalks, where the height of the ground surface varies the most.

Mask R-CNN was applied to all the images, and the instance segmentations were matched to the road user IDs by projecting the true 3D positions into each camera and associating it with any segmentation mask that covered that point, if they were of the same road user class. Then, Seg2PoseNet was trained on these examples. For each camera and road user class pair, the network was trained with a “best of four” strategy: the network was trained four times and the epoch with the best validation loss across all these was chosen for use on the test set. Training multiple times decreases the importance of the random sampling for training examples. The same four initializations were used for all cameras. The learning rate was 5×10^{-5}, the batch size was 16, each epoch was 64 batches and

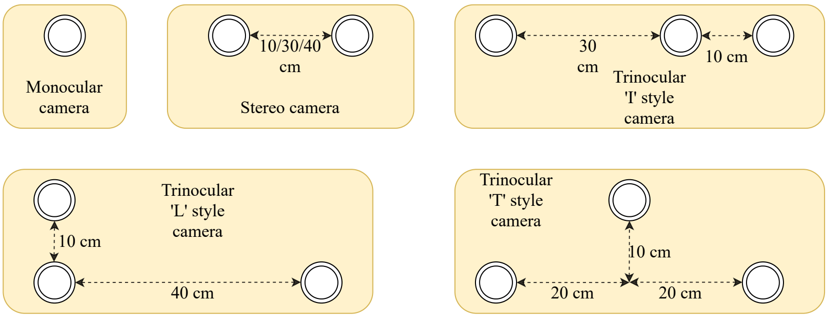
Monocular camera
10/30/40 cm
Stereo camera
30 cm
10 cm
Trinocular 'I' style camera
Trinocular 'L' style camera
10 cm
40 cm
Trinocular 'T' style camera
10 cm
20 cm
20 cm

Figure 5: Camera designs used with the CARLA dataset. Multiple stereo cameras with different baselines, as well as three trinocular designs denoted 'I', 'L' and 'T', were tested. This illustration is not to scale.

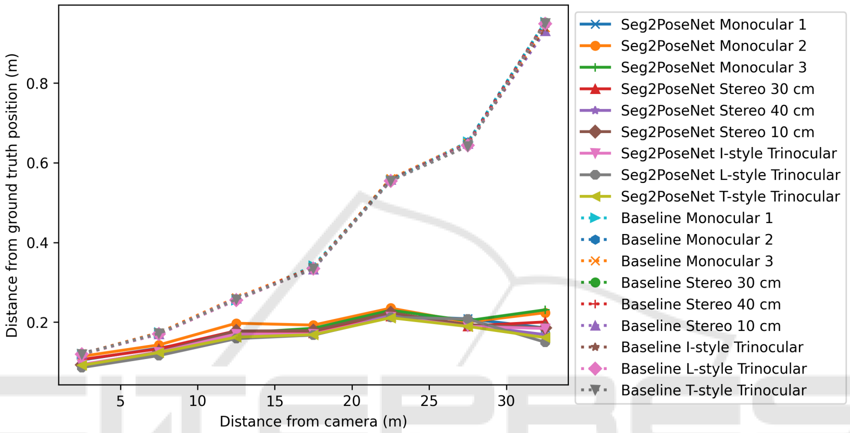

Figure 6: Results from the CARLA test set, with errors on the y-axis and distance from the cameras on the x-axis. The results are presented as averages of all road users 0 to 5 meters from the cameras, 5 to 10 meters from the cameras and so on. Seg2PoseNet performs better than the baseline methods, especially when further away from the cameras. The differences between the different camera configurations is small. Best viewed in color.

each training run lasted for 50 epochs. The validation set was used only for selecting the epoch with lowest validation loss for testing.

4.1.1 Results

In Figure 6, the positioning error on the test set for the different camera configurations are shown, as the average error for all those road users at 0-5 m distance from the camera, 5-10 m distance from the camera and so on, to show how well the different configurations work at different distances. We note that when Seg2PoseNet is applied, the errors decrease compared to the baseline method which only uses the normal positions, as expected. A somewhat unexpected result is that the different camera configurations have a small impact on the accuracy, and the best performing configuration is actually the largest stereo configuration. The total average distances are shown in the third column of Table 1. An example image from the CARLA dataset is shown in Figure 1.

4.2 Real World Scene

An outdoors area was recorded using an I-style trinocular camera when cars, bicyclists and pedestrians moved, instructed to act like in a traffic environment. The 3D positions of 86 ground points were collected using a Leica S06 (Leica Geosystems AG, 2009) device. A specialized camera calibration method for a trinocular camera (Ahrnbom et al., 2021a) was used, using the same ground points. Detectron2's Mask R-CNN (He et al., 2017; Wu et al., 2019) was used as an object detector, and SORTS (Ahrnbom et al., 2021b) was used to create tracks in pixel coordinates for each road user. Then, these tracks were converted to world coordinates by the Seg2Pose system.

The video clip, 22 650 frames long (at 24 FPS) was divided into a training, validation and test set with 7750 frames in the training set, 7500 frames in the validation set and 7400 frames in the test set. At least 150 frames were removed between the sets to reduce similarity. Seg2PoseNet was trained on the training set and the validation set was used only for

Table 1: Average errors of the different configurations, in metres, on the CARLA dataset and the real scene. The best values for each dataset are in bold. The configurations with “S2PN” are those that use Seg2PoseNet, while those without use the baseline method. For the stereo lengths, the left number applies to CARLA and the right is the real scene.

Configuration	S2PN	CARLA	Real
Monocular 1	Yes	0.1738	0.1822
Monocular 2	Yes	0.1890	0.1822
Monocular 3	Yes	0.1768	0.1931
Stereo 30/33 cm	Yes	0.1734	**0.1775**
Stereo 40/43 cm	Yes	0.1661	0.1812
Stereo 10 cm	Yes	0.1742	0.1837
I-style Trinocular	Yes	0.1682	0.1791
L-style Trinocular	Yes	**0.1649**	N/A
T-style Trinocular	Yes	0.1654	N/A
Monocular 1	No	0.3653	0.4191
Monocular 2	No	0.3655	0.4042
Monocular 3	No	0.3682	0.4181
Stereo 30/33 cm	No	0.3641	0.4069
Stereo 40/43 cm	No	0.3652	0.4070
Stereo 10 cm	No	0.3654	0.4064
I-style Trinocular	No	0.3643	0.4045
L-style Trinocular	No	0.3623	N/A
T-style Trinocular	No	0.3631	N/A

early stopping. The same hyperparameters as in Section 4.1 were used. Like in Section 4.1, the network was trained four times per camera and road user class, and the weights from the epoch with the lowest validation loss were used when testing.

The ground truth was collected by manually finding the world coordinate positions by first clicking on where the road users appear in some frames, and then manually moving the pose until it looked right. Interpolation was applied to save time. T-Analyst software (Lund University, Transport and Roads, 2018) was used to create these annotations.

4.2.1 Results

The average errors are shown as a function of distance from the cameras in Figure 7. The total average errors are shown in the rightmost column of Table 1.

A number of images showing results of Seg2Pose on the test set are shown in Figure 8. It should be noted that the images shown are not hand-picked and are reasonably representative of the general performance of our system. The system is capable of accurately position road users in the scene in most cases, and when it fails, it seems to often be due to failures by Mask R-CNN and SORTS.

5 DISCUSSION

In the CARLA experiments, Seg2PoseNet clearly improves upon the baseline of only using normal positions, especially at longer distances from the camera(s). When comparing different camera configurations, we note that using more cameras is better than using fewer, and having them further apart improves performance for the stereo configurations, as expected. The differences between camera configurations are, however, small.

In the experiments done on real camera recordings, it is worth pointing out that the ground truth annotations have some noise, and possibly bias, caused by the human annotation process. In particular, road users far away from the cameras may suffer from T-Analyst’s less accurate camera calibration being used, and it is difficult to determine how much this affects the results. In addition, the low number of participants mean that the training and test sets have significant similarity. The CARLA dataset was used to avoid precisely these types of issues. We believe these issues contribute to add noise that partially hides that more using camera views should improve positioning. Randomness in training could also contribute, despite the effort taken to reduce it.

In this work, we have treated using only normal positions as our baseline method, and this could use some clarification and motivation. Most previous traffic surveillance research uses simple methods for converting pixel coordinates to world coordinates because it is not the focus of their work, and it is known that these approximate methods work well enough for some applications. There is no “standard method”, making it difficult to select a representative baseline. We believe the normal positions are better than most existing systems, because they do not assume that the ground is perfectly flat, and since they use the center point of an AABB rather than the bottom point, commonly assumed to lie on the ground. This means, for example, that it works for road users seen mostly from above, something that occurs in both the datasets.

One of the benefits of using segmentation masks as the input of Seg2PoseNet, as opposed to RGB images, is that Seg2PoseNet could be more easily trained on synthetic data placed into real scenes, as explained in Section 3.5. This could be further improved by using more realistic and varied 3D models, such as the ones used in the CARLA dataset. We would like to experiment with such an approach in the future. Perhaps it is possible to fully train Seg2PoseNet without any dataset annotations, using only camera calibration, ground points and synthetic road user models.

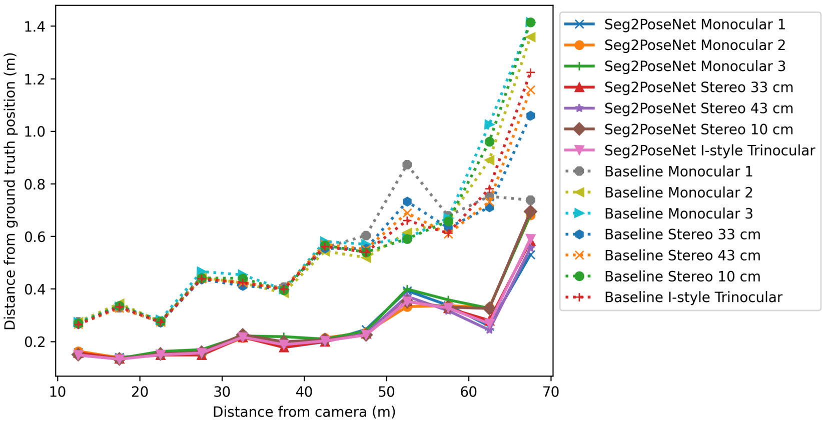

Figure 7: Results from the real video's test set. The format is the same as in Figure 6. The configurations using Seg2Pose outperform the baselines significantly, by a factor of about two. With Seg2PoseNet, the average error remains close to 20 cm up to almost 50 m distance from the cameras. The differences between different camera configurations are small.

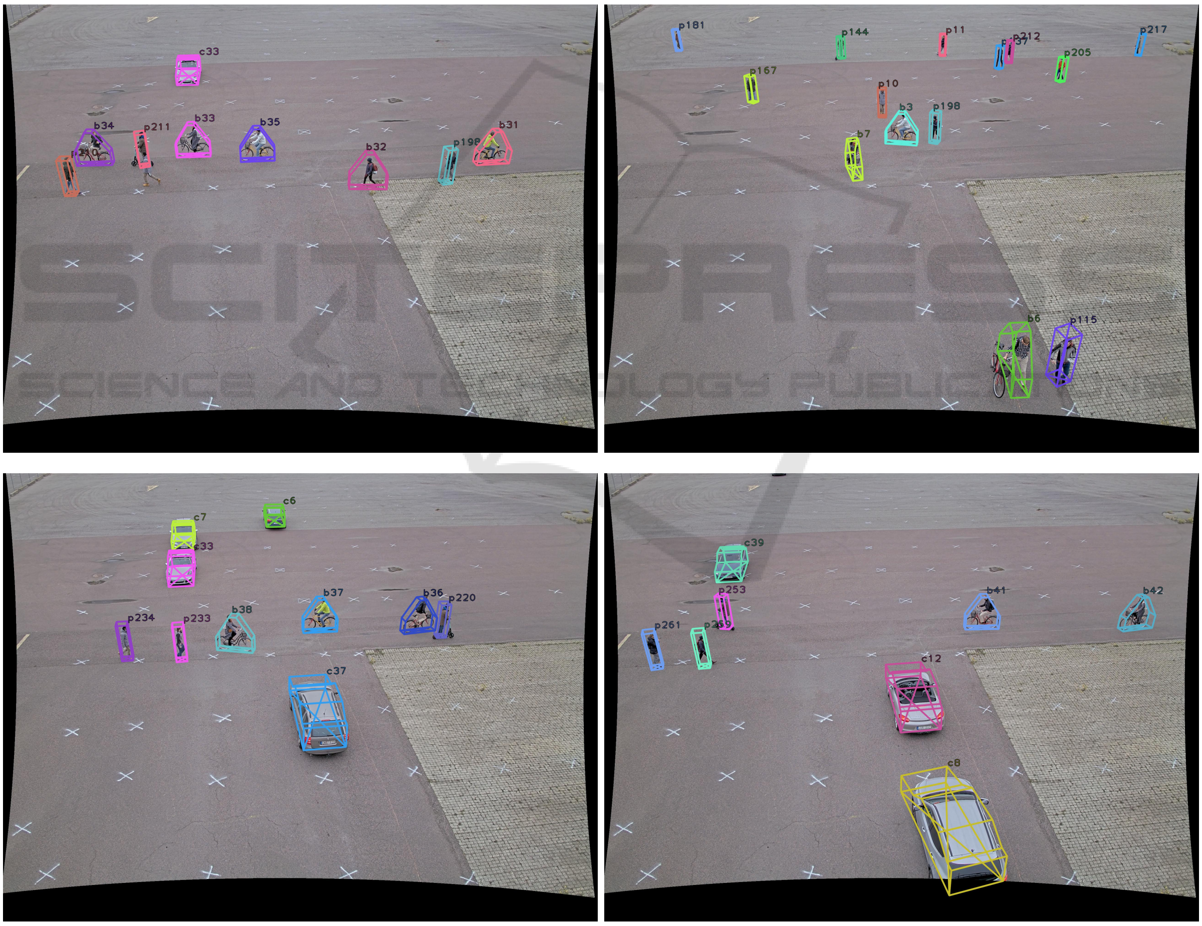

Figure 8: Results of Seg2PoseNet on the test set of the real video, when using an I-style Trinocular camera configuration. World coordinate tracks are shown by simplified 3D shapes, with unique IDs for each track, shown alongside a 'c' (car), 'p' (pedestrian) or 'b' (bicyclist). Sometimes, Mask R-CNN or SORTS fails to locate a road user, like one of the pedestrians in the top left image. Rarely, classification fails like "bicyclist" 32 in the top left image, which is likely due to an error in the logic that combines a bicycle and person detected by Mask R-CNN into a bicyclist. The Seg2Pose system works as intended for most road users in most frames. Note that scooters, like "pedestrian" 220 in the bottom left image, are currently classified as pedestrians, as Mask R-CNN trained on MS COCO (Lin et al., 2014) does not recognize scooters. Best viewed in color.

6 CONCLUSION

We present a system called Seg2Pose for converting instance segmentation tracks into world coordinate pose tracks for road users in static surveillance cameras. The system uses our novel CNN, Seg2PoseNet, which we show outperforms the baseline of only using normal positions on both synthetic data from CARLA Simulator and a real world video, approximately cutting the positioning errors in half. We further show that stereo and trinocular cameras improve accuracy on the CARLA dataset slightly, but this trend is not clearly shown in our experiments with real data.

ACKNOWLEDGEMENTS

This research was funded by VINNOVA project 2017-05510 “The Third Eye”.

REFERENCES

Ahrnbom, M., Nilsson, M., Ardö, H., Åström, K., Yastremska-Kravchenko, O., and Laureshyn, A. (2021a). Calibration and absolute pose estimation of trinocular linear camera array for smart city applications. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 103–110.

Ahrnbom, M., Nilsson, M., and Ardö, H. (2021b). Real-time and online segmentation multi-target tracking with track revival re-identification. In *VISIGRAPP (5: VISAPP)*, pages 777–784.

Bradler, H., Kretz, A., and Mester, R. (2021). Urban traffic surveillance (uts): A fully probabilistic 3d tracking approach based on 2d detections. In *IEEE Intelligent Vehicles Symposium, IV 2021, Nagoya, Japan, July 11 - July 17, 2021*. IEEE.

Brazil, G., Pons-Moll, G., Liu, X., and Schiele, B. (2020). Kinematic 3d object detection in monocular video. In *In Proceeding of European Conference on Computer Vision*, Virtual.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Hodan, T., Barath, D., and Matas, J. (2020). Epos: Estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jensen, M., Ahrnbom, M., Kruithof, M., Åström, K., Nilsson, M., Ardö, H., Laureshyn, A., Johnsson, C., and Moeslund, T. (2019). A framework for automated analysis of surrogate measures of safety from video using deep learning techniques. In *Transportation Research Board. Annual Meeting Proceedings*, pages 281–306. Transportation Research Board National Cooperative Highway Research Program. Conference date: 13-01-2019 Through 17-01-2019.

Kumar, A., Brazil, G., and Liu, X. (2021). Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Nashville, TN.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing.

Muller, N., Wong, Y.-S., Mitra, N. J., Dai, A., and Nießner, M. (2021). Seeing behind objects for 3d multi-object tracking in rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6071–6080.

Leica Geosystems AG (2009). Leica S06 Specification. https://w3.leica-geosystems.com/downloads123/zz/tps/FlexLine%20TS06/brochures-datasheet/FlexLine_TS06_Datasheet_en.pdf.

Lund University, Transport and Roads (2018). T-analyst. https://bitbucket.org/TrafficAndRoads/tanalyst/wiki/Manual.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2. https://github.com/facebookresearch/detectron2.

Yang, F., Chang, X., Dang, C., Zheng, Z., Sakti, S., Nakamura, S., and Wu, Y. (2020). Remots: Self-supervised refining multi-object tracking and segmentation. *arXiv preprint arXiv:2007.03200*.

Yin, T., Zhou, X., and Krahenbuhl, P. (2021). Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793.

Zakharov, S., Shugurov, I., and Ilic, S. (2019). Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1941–1950.

Zhang, B. and Zhang, J. (2020). A traffic surveillance system for obtaining comprehensive information of the passing vehicles based on instance segmentation. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–16.

Zhang, S., Wang, C., He, Z., Li, Q., Lin, X., Li, X., Zhang, J., Yang, C., and Li, J. (2020). Vehicle global 6-dof pose estimation under traffic surveillance camera. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:114–128.