Real-Time Traffic Analytics at Intersections

Srinivasa Narasimhan[1], Robert Tamburo[2], Dinesh Narapureddy

FINAL RESEARCH REPORT

CONTRACT #69A3551747111

PROJECT #335

[1]https://orcid.org/0000-0003-0389-1921
[2]https://orcid.org/0000-0002-5636-9443

1 Problem

More and more city planners are taking into consideration smart mobility solutions to address transportation needs and central to smart transportation systems is access to real-time data. For example, self-driving cars offer independence for seniors and people with disabilities, greater road safety, cost savings through ride sharing, increased productivity, reduced congestion, and reduced fuel use and carbon emissions. But what type of information and data is needed for city planners to accommodate smart transportation systems? Visual data is extremely rich in information and algorithms can process the data and extract the information. However, bandwidth is limited and too much time is needed to transfer visual data to remote computers for analysis. This work focused on developing computer vision algorithms for analyzing visual data and computing and sharing the resulting analytics and summary data in real-time. Algorithms were developed to understand vehicle motion in 3D space and time, and to track the pose of people in 3D. These algorithms are vital to computing analytics on real data in the presence of occlusions, cluttered scenes, and varied lighting conditions.

Vehicle Reconstruction in 3D Space and Time: The shapes, positions and velocities of vehicles in 3D reveal instantaneous traffic information, which can be aggregated to automate traffic monitoring and facilitate driver assistance systems. Depth sensors have been used to reconstruct 3D information, but are too expensive to deploy at city scale. In contrast, video surveillance cameras are already widely installed, but most surveillance systems are only able to collect 2D information such as 2D bounding boxes, re-identification and 2D trajectories. Due to the ambiguity between 3D location and 2D image projection, it is impossible to reconstruct 3D vehicles from these cameras directly without any priors. Recently, many deep learning-based reconstruction methods have been proposed to estimate 3D shape and position from visual appearance, but they are sensitive to training data and hard to transfer to new scenes. For example, models trained on egocentric views perform poorly on traffic surveillance cameras because of differences in view angle and background. Unstable and inaccurate detections cause 3D trajectory reconstruction to fail over time. Although many works attempt to enforce temporal consistency in reconstruction and video analysis, they focus on short intervals such as over a few frames or seconds. For this work, we recognized that the key to accurate vehicular 4D reconstruction (i.e. recovering 3D shape and motion) is exploiting the consistency in long-term (several minutes or greater) repetitious activity, i.e. vehicles passing an intersection clustered into groups with similar motion patterns. Using longitudinal consistency as self-supervision, we adapted a pre-trained keypoint detector to new scenes it never saw before, and obtain higher accuracy 2D and 3D keypoints without any manual annotation.

Multi-Person Articulated 3D Pose Tracking: We address the problem of tracking and reconstructing 3D articulated poses of multiple individuals captured in an arbitrary number of camera. This task requires identifying the number of people in the scene, reconstructing their 3D body joints into consistent skeletons, and associating 3D body joints over time. We do not make any assumption on the number of available camera views and focus on real-world scenarios that often include multiple close-by interacting individuals, fast motions, self- and person-person occlusions. A key challenge in such scenarios is that people might strongly overlap and expose only a subset of body joints due to occlusions or truncations by image boundaries, which makes it more difficult to reliably reconstruct and track articulated 3D human poses. Most multi-view strategies rely on multi-stage inference to first estimate 2D poses in each frame, cluster same person poses across views, reconstruct 3D poses from clusters based on triangulation, and finally link 3D poses over time. Solving each step in isolation is sub-optimal and prone to errors that cannot be recovered in later stages. This is even more true for monocular methods where solving each step in isolation often represents an ill-posed problem. Our method is a top-down approach that simultaneously addresses 3D body joint reconstructions and associations in space and time of multiple persons.

2 Approach

2.1 Vehicle Reconstruction in 3D Space and Time

Starting from off-the-shelf 2D keypoint detections and camera intrinsics, our method reconstructs 3D keypoints with an active shape model, fits an analytic trajectory model to each vehicle's 3D poses over time, and applies a novel method to cluster the vehicle trajectories in 3D. Later, the accurate 2D keypoints and 3D mean trajectories of each cluster (denoted as 2D and 3D experts) accumulated over the entire video are used to improve 2D and 3D keypoints in a self-supervised manner as shown in Fig. 1. We refer to this process as **longitudinal self-supervision**. Our main approach is summarized below and the entire framework is shown in Fig. 2:*(a) Joint optimization for longitudinal reconstruction*: Consistent reconstruction of diverse motion and poses from single-view by joint optimization over all vehicles in long-term videos. *(b) Scene-specific repetitious activity clustering*: Projecting 3D trajectories to subspaces with strong separability to suppress noise from imperfect detection and reconstruction, and then clustering the trajectories into fine-grained motion groups. *(c) 2D/3D longitudinal self-supervision*: Selecting and accumulating accurate 2D keypoints via geometry consistency to refine erroneous keypoints; Learning geometric correspondence between 3D mean trajectories and individual poses as a posterior to improve 3D reconstruction.

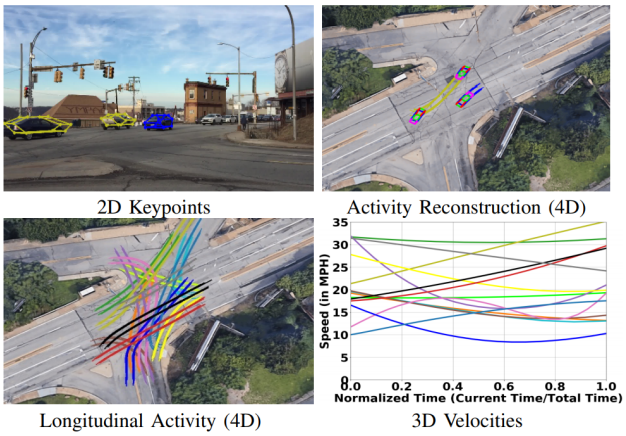

Figure 1: Long term repetitious vehicular activity is used as self-supervision to compute accurate 2D and 3D keypoints, trajectories and velocities from a single fixed camera. Reconstruction accuracy improves significantly over 20 minutes at this intersection as compared to methods that enforce consistency over short periods (a few frames to seconds).

The versatility and generalizability of our approach was demonstrated using traffic videos of 78k frames captured by 18 single view fixed cameras at city intersections. The datasets are from a variety of sources: (a) live YouTube cameras, (b) our iPhone cameras, and (c) the AI City Challenge dataset. Our method was also applied to traffic tasks such as velocity estimation, anomaly detection and vehicle counting.

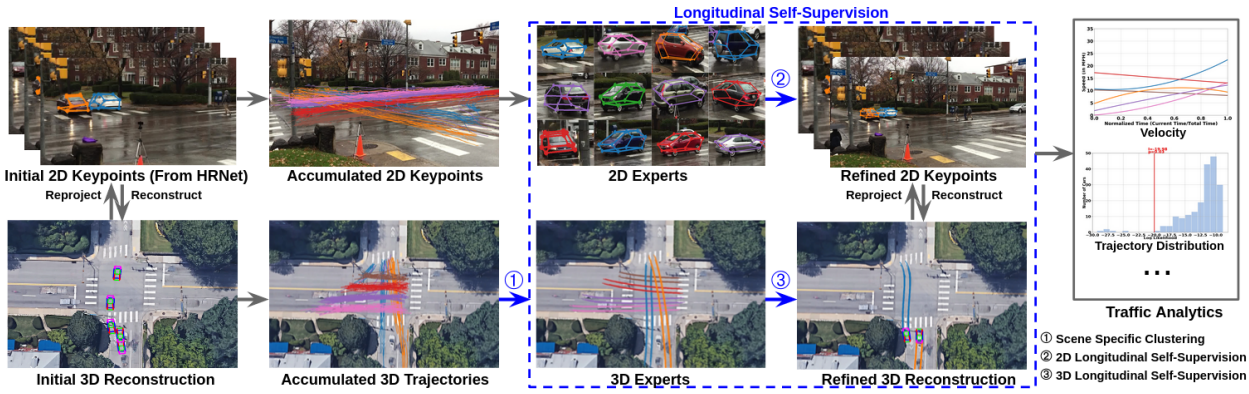

Figure 2: Framework for self-supervised 4D reconstruction of repetitious activity. Our method takes off-the-shelf 2D keypoint detections as input, reconstructs 3D keypoints with an active shape model, fits an analytic trajectory model to each vehicle's 3D poses along with frames, and accumulates them over time. Then, for 2D self-supervision, good keypoints from initial detections are selected as "2D experts" to refine bad 2D keypoints. For 3D, the accumulated 3D trajectories are clustered and the mean trajectories are used as "3D experts" to refine 3D poses. The reconstruction could be applied to traffic analysis such as velocity estimation and anomaly analysis.

2.2 Multi-Person Articulated 3D Pose Tracking

Our top-down approach simultaneously addresses 3D body joint reconstructions and associations in space and time of multiple persons. At the core of our approach is a novel spatio-temporal formulation that operates in a common voxelized feature space obtained by casting per-frame deep learning features from single or multiple views into a discretized 3D voxel volume. First, a 3D CNN is used to localize each person in the voxel volume. Then, a fixed spatio-temporal volume around each person detection is processed by a 4D CNN to compute short-term person-specific representations. Overlapping representations at neighboring time steps are further scored based on attention aggregation and linked using a differentiable matcher. Finally, 3D body joints of the same person are consistently predicted at each time step based on merged person-specific representations. Notably, all components are implemented as layers in a single feed-forward neural network and are thus jointly learned end-to-end.

Our approach relies on a novel spatio-temporal formulation that allows simultaneous 3D body joint reconstruction and tracking of multiple individuals. In contrast to multi-person 3D pose estimation approaches who similarly aggregate per frame information in 3D voxel space, we address a more challenging problem of multi-person 3D pose tracking and propose end-to-end person-specific representation learning. Our method does not make assumptions on the available number of camera views and performs reasonably well even in the purely monocular setting. Remarkably, using only a single view allows achieving similar MPJPE 3D joint localization error compared to five-view settings. In contrast to multi-person 2D pose tracking methods that rely on short-term spatio-temporal representation learning, our approach operates on the aggregated spatio-temporal voxel volume and provides a richer hypothesis comprising of tracked 3D skeletons.

The approach also formulates a novel learnable tracking formulation that allows extending person-specific spatio-temporal representation learning to arbitrary-long sequences. In contrast to methods that use a heuristic pairwise tracking score based on pose distance and perform matching using the Hungarian method, our method relies on an attention aggregation layer and a differentiable representation matching layer based on the Sinkhorn algorithm. Importantly, we match person-specific representations instead of the determined body pose tracklets, which allows learning of more expressive representations. This approach improves tracking accuracy but also improves joint localization.

3 Methodology

3.1 Vehicle Reconstruction in 3D Space and Time

3.1.1 Background

We use three coordinate systems, i.e. camera, world and map coordinates as shown in Fig. 3. The camera coordinate is defined with the origin at the focal point, parallel to image plane; while in world coordinates the ground plane and axis points upwards. The two coordinate systems are associated by a rigid transform. In world coordinates each object's trajectory is represented on the ground plane. Finally, we have a map coordinate system consistent with Google maps. The transform from world coordinates to map coordinates involves rotation, translation, and scaling that are estimated using annotated landmarks on input image and Google map (represented as yellow crosses in Fig. 3). Each new camera only needs these annotations for our 4D automatic self-supervision pipeline. We refer to each object's appearance in one frame as an *instance*. For a video of frames, a total of unique objects are captured

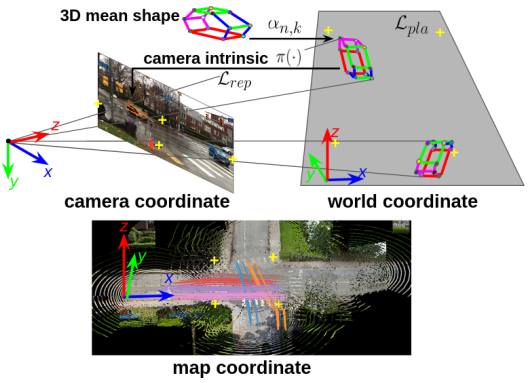

Figure 3: 3D reconstruction coordinate frames. Vehicle 3D keypoints are computed in camera coordinates. The world coordinate is defined with XY as the ground plane, in which we perform analytic model fitting and repetitious activity clustering. Map coordinates are defined based on Google maps, whose XY plane is also the ground. This is used to estimate real-world location and speed. Yellow cross landmarks transform world to map coordinates.

with keypoints for each instance. The 3D position is in camera coordinates and the 2D position is in image coordinates of the corresponding keypoint of an instance in the captured frame.

3D Shape Model: The object 3D keypoints are parameterized by an active shape model to regularize shape optimization. The mean shape of all object models, and their principle components are computed from an object CAD model dataset. Then each object's actual shape is formulated as linear combination of mean shape with the top principal components. For each object, we track it over time and enforce the shape parameter to be constant for its instances in different frames.

3D Trajectory Model: We use an h-th order polynomial as analytic model to fit each object's 3D motion. For simplicity, we convert all the poses into world coordinate so only the motion in x, y direction needs to be considered. We observe that in most of the experiments, $h = 3$ fits the model well (turns, including U-turns, and lane changes) but higher order may be necessary for rare complex motions. The reconstructed object poses are used to solve the parameters by minimizing loss. For a given frame, the coordinate and tangent is predicted by the solved parameters and should be close to the reconstructed pose. Both the tangent and rotation matrix are converted into a direction vector. A regularizing term is added for third order coefficients.

3.1.2 Self-Supervised 4D Reconstruction

In this section, we explain our approach to utilize longitudinal consistency in repetitious vehicular activity for accurate 4D reconstruction. Fig. 2 shows the overall pipeline with the three stages described below.

Joint Optimization For Longitudinal Reconstruction We propose to jointly optimize for the shape and pose of objects moving in the scene over long durations of time. We show clear improvement in reconstruction accuracy compared to previous proposed methods, which either optimize for shape or pose over short durations (few consecutive frames). Specifically, exploiting rigidity over consecutive frames and a constant ground plane constraint show that our joint reconstruction outputs are more accurate and consistent compared to previous state of the art methods.

Pose Initialization: We use HRNet to detect 2D bounding boxes and keypoints for objects in each frame. We pass these detections into a Visual Intersection-Over-Union (V-IOU) multi-object tracker. We enforce each object is rigid over frames using the tracking ids. Then, the 3D rotation and translation is initialized using RANSAC based EPnP to account for inaccurate keypoints from detector.

Joint Optimization over all Objects: The 3D keypoint locations can be computed from the shape model by optimizing the shape coefficients vector and pose jointly for all the vehicles in all the frames. We exploit the following geometric constraints to enforce the joint consistency in reconstruction over long term. *(1) Reprojection loss:* the error between the projection of each object's 3D keypoints and its respective 2D detections. *(2) Joint planar loss:* This loss constrains all the vehicles in the long-term video to be as close as possible to a ground plane. We formulate this error as the squared distance in camera coordinates between the vehicle's bottom center and the ground plane.

3.1.3 Scene-Specific Repetitious Activity Clustering

Capturing repetitious motion patterns over a long duration plays an important role in deciphering higher level semantics of the environment. We observe and demonstrate using experiments that such higher order semantics are much more distinguishable in 3D compared to 2D. Thus, we first fit a polynomial model to each object's 3D poses to suppress noise and reduce data dimension. Then, the trajectory parameters are clustered hierarchically and projected to sub-spaces with good cluster-separability using a novel scene-specific clustering approach.

Hierarchical Scene-Specific Clustering: Repetitious activity, like vehicles moving in the same lanes every day, can be used as a signal for supervision. The method proposes using additional scene specific constraints for clustering such activity. We illustrate this with an example of separating the vehicles into lane-specific activity as shown in Fig. 4.

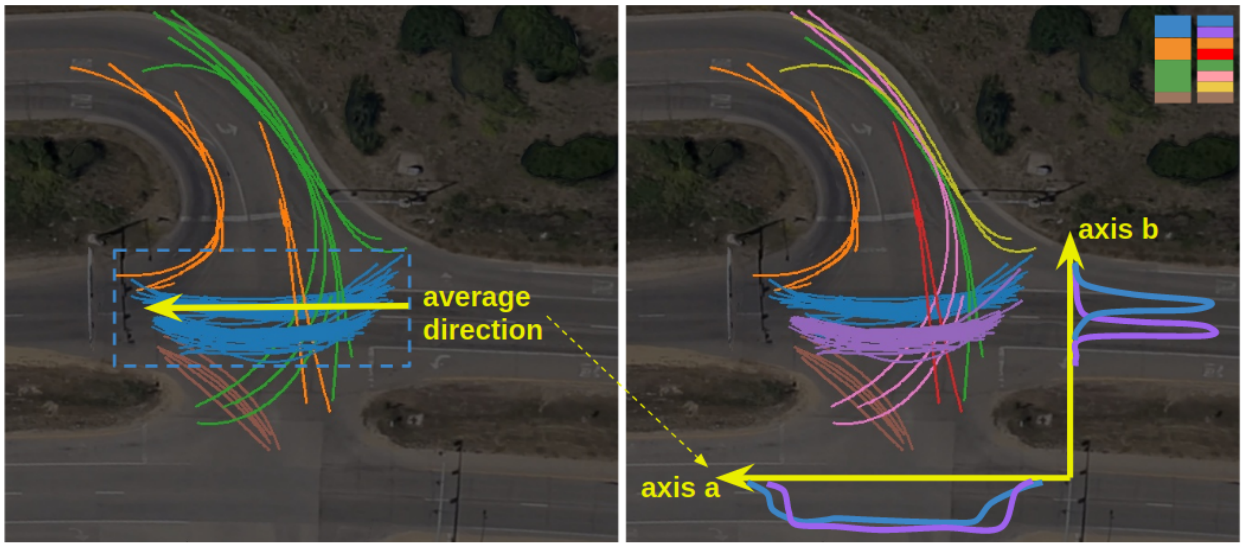

Figure 4: Demonstration of our hierarchical clustering in birds-eye view. **Left:** First stage clusters and the average direction of the blue cluster. **Right:** Second stage clustering. Trajectories are projected along their average direction, maximizing the spatial difference between near clusters. The blue trajectories from left are projected onto axis **b** and are distinguished very well into two clusters, while they are almost overlapped on axis **a**.

We face two challenges here: (a) vehicles on different lanes can be close to each other (see blue and purple lines in Fig. 4) and (b) trajectories of the same lane have different shapes and positions. The issues are further exaggerated by imperfect tracklets and keypoints. We solve these issues with a hierarchical approach. First, we directly cluster trajectory parameters using a Gaussian Mixture Model. We observe vehicles in different directions are in different clusters (orange in Fig. 4), but lanes in the same direction (blue and purple) cannot be distinguished. Thus, in the second stage of the hierarchy, our observation is that each sub activity will have a scene-specific dominant direction that can be used to cluster. For this, we find a direction to project trajectories belonging to the same initial cluster from 2D to 1D. Then each trajectory is projected along the average direction. In Fig. 4, axis **a** is the average direction. Blue and purple trajectories are projected along axis **a** to axis **b**. We notice the overlapping between the two lanes is mostly eliminated, so they become easily distinguishable. Our method is unsupervised and takes scene-specific information (say, the geometry of traffic lanes) into account to maximize the separation between similar clusters (lanes). For each fine-grained cluster, we then save the average of the parameters of all trajectories.

3.1.4 2D and 3D Longitudinal Self-Supervision

Humans generally improve their cognitive skills from observations and repetitious behaviors generally reinforce inference. Inspired from human cognition, we propose self-improvement in detection both in 2D and 3D using the clustered mean shapes. These mean shapes act as anchors for any new observation and show a clear improvement in detection in 2D and 3D over passage of time as shown later in the results.

2D Longitudinal Self-Supervision: Learning-based detectors produce precise as well as erroneous keypoints. We would like to use the accurate detections to improve the badly localized keypoints. We distinguish the good ones from the erroneous by using a threshold on the reprojection error. All the inliers below the threshold are considered as *2D experts* and integrated into a 2D expert pool. Each instance above the threshold is considered erroneous and needs to be refined. To refine each erroneous instance, it is necessary to retrieve a 2D expert from the expert pool with a similar shape as the instance. Since the camera is fixed and object motion is constrained, we can assume that objects with bounding boxes of similar size and location tend to have similar 3D shapes and pose, so we extract temporal bounding boxes as the feature for matching. For an instance at frame m, we concatenate its 2D bounding box's 4 corner coordinates

from frame $m-k$ to $m+k$ as the feature for retrieval. Similar features for all 2D experts are stored for matching. The erroneous instance finds its guiding 2D expert from the expert pool by minimizing ℓ_2 distance of bounding box features using the nearest neighbor algorithm. Two vehicles having similar bounding box features need not be perfectly aligned in 3D, so we transform the bounding box and keypoints to overlap between instance and the 2D expert. We optimize for scale and translation from the 2D expert bounding box to the instance bounding box. Then the optimized transformations are applied to the 2D expert's keypoints. If the distance between the transformed expert keypoint and the instance keypoint is above a threshold, the instance keypoint is considered as misclassified and updated with the expert keypoint.

3D Longitudinal Self-Supervision: We use 3D mean trajectories learned from repetitious activity clustering as our *3D experts*. Since 3D experts represent the typical motion over a long duration, they act as a strong regularization to refine erroneous 3D poses. To refine each 3D pose, we find a correspondence between the estimated 3D pose and the 3D experts for supervision. For each object, we first find out from all the 3D experts, the one most similar to the object's motion. Considering the object's pose in a frame and the 3D expert of one specific cluster, we find a point on the 3D expert minimizing its distance to the object position. We compute the Chamfer distance from this object's trajectory to the 3D expert as the sum of such distance over all frames where this object appears. From 3D experts of different clusters, we select the one with the minimal Chamfer distance to the object's trajectory. If the selected 3D expert's Chamfer distance is less than a threshold, it is used to refine the object pose. For the pose in a frame , we find its closest point on the 3D expert when calculating Chamfer distance.

3.2 Multi-Person Articulated 3D Pose Tracking

To learn person tracking and pose estimation in 3D we build multiple differentiable layers with intermediate supervisions. The pipeline is illustrated in Fig 5. Our network is made up of three main blocks, each one with an associated loss. The first block is a person detection network in 3D voxel space. Given person detections, a 4D CNN extracts a spatio-temporal representation of each detected person over a short period of time. In order to track people, we then solve an assignment problem between the set of descriptors for two frames. All matched descriptors which overlap are then merged into a single descriptor which is finally deconvolved into a 3D pose for the person tracked at central frame.

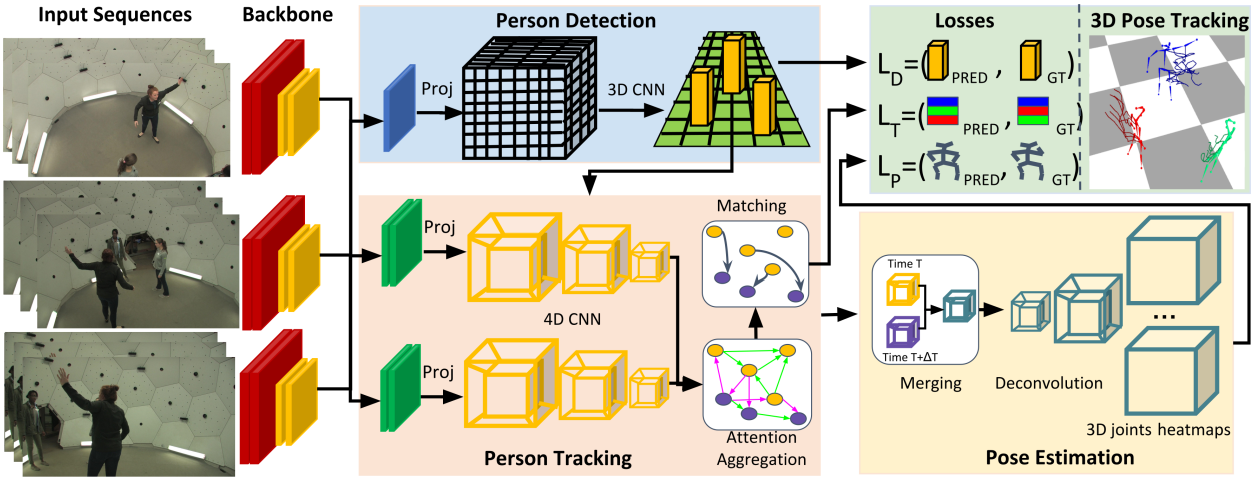

Figure 5: The complete pipeline of tessetrack has been illustrated. Initially, the video feed from multiple cameras is passed through shared HRNet to compute the features required for detection and 3D pose tracking. The final layer of the HRNet is passed through a 3D convolution to regress to the center of the human 3D bounding boxes. Each of the hypotheses is combined with the HRNet final layer to create a spatio-temporal Tube called tesseract. We use a learnable 3D tracking framework for a person association over time using spatio-temporal person descriptors. Finally, the associated descriptors are passed through deconvolution layers to infer the 3D pose. Note that the framework is end-to-end trainable except for the NMS layer in the detection network.

3.2.1 Person Detection Network

Our approach starts with a multi-view person detection network (PDN) trained to detect people in 3D at a specific time instance. We use HRNet as our backbone for extracting image-based features at each frame. We use the pre-final layer of the network and pass it through a single convolution layer to convert it into a feature. The feature maps coming from all the camera views are then aggregated into a 3D voxelized volume by an inverse image projection method without fusing the 2D joint heatmaps in 3D but with the richer feature vectors picked from the pre-final layer of HRNet. The voxel grid is initialized to encompass the whole space observed by the cameras. Using the camera calibration data, each voxel center is projected into the camera views.

We aggregate all the feature vectors picked in image space by concatenating them and passing through a shallow network with a softmax layer. This produces a unique feature vector. We thus end up with a data structure of the size of the feature vector times the dimensions of the voxel grid. We then apply 3D Convolutions to this volume to generate detection proposals. For each person, we train the network to detect its "center", which is defined as the midpoint between neck and center of the hips. The loss at each time is expressed directly as a distance between the expected heatmap and the output heatmap. We apply non-maximum suppression (NMS) on the 3D heatmaps and only retain the detections with large score.

3.2.2 Spatio-Temporal Descriptors and Tracking

For each detected person we create a spatio-temporal volume of fixed dimension centered on the person and use a 4D CNN to produce a short time description of the person around the detection frame. We call this spatio-temporal volume a *tesseract* as it is a 4D volume of size R×T×X×Y×Z, where T represents temporal window size and X,Y,Z are the dimensions of the cuboid centered on the detected person. The goal of extending the volume in time around the detection frame is twofold. First, using a temporal context allows to better estimate the joint positions in the central frame, and especially to extrapolate/interpolate occluded joints or to handle pose or appearance ambiguities in a single frame. Second, extending a person's description in time generates a descriptor which overlaps with adjacent frames, hence producing descriptors that can be matched by similarity for tracking purposes.

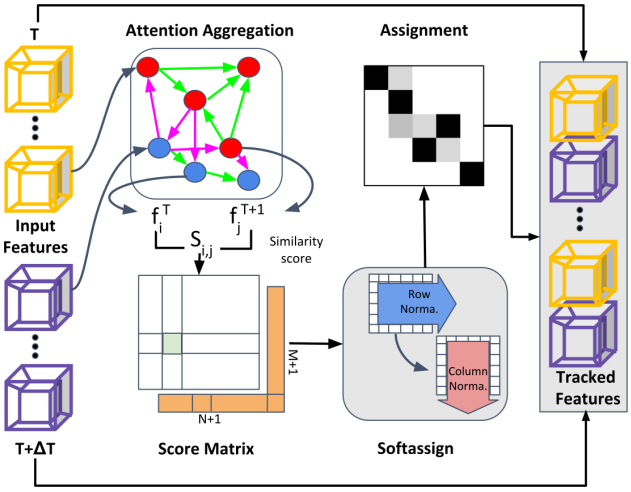

Figure 6: The learnable tracking framework. The input is the tesseract features for multiple detected humans at two different time instances. The output is an assignment matrix providing the correspondence between the detected persons at different times.

Tesseract Convolutions: The input to this subnetwork is still the output of the HRNet pre-final layer which is cast in 3D at each time stamp. We follow the same procedure as for the person detection network to generate the features for each time instance of the tesseract. The tesseract is then passed through multiple 4D convolutions and max pooling layers to produce a reduced size tesseract feature. These features represent a spatio-temporal descriptor of a person centered around a detection. This bottleneck descriptor is used in both the tracking and pose estimation modules.

Attention Aggregation: Before temporal matching, as illustrated in Fig 6, we pass the features into a Graph Neural Network to integrate contextual cues and improve the features distinctiveness. We use two types of undirected edges: self edges, connecting features belonging to the same time instance and cross edges, connecting features from adjacent time instances. We use a learnable message passing formulation

to propagate the information in the graph. The resulting multiplex network starts with a high-dimensional state for each node and computes at each layer an updated representation by simultaneously aggregating messages across all incident edges for all nodes.

Temporal Matching Layer: The final features of the attention module are passed through a trained matching layer, which produces an assignment matrix. For a given time instance, we consider the features of people at the current time and another time instance. As in the standard bipartite graph matching formulation, an optimal assignment is a permutation matrix which maximizes a total score. We compute the similarity between the descriptor at the two time instances. As opposed to learned visual descriptors, the matching descriptors are not normalized, and their magnitude can change as per the feature during training to reflect the prediction confidence. To let the network suppress some predicted persons (false detections) and to handle changes in the number of persons in the scene, we augment each set with a dustbin so that matching is always computed on a fixed length feature vectors. This leads to optimal assignments for each available detection and the rest unassigned dustbins always correspond one-to-one with the next time instance.

Following recent end-to-end learning approaches which include an optimal assignment step, we use the Softassign algorithm to solve the assignment problem by a differentiable operator. The Softassign algorithm is based on Sinkhorn iterative matrix balancing, which projects an initial score matrix into a doubly stochastic matrix by iteratively normalizing the matrix along rows and columns. The Softassign algorithm can be efficiently implemented on GPUs by unrolling a fixed number of Sinkhorn iterations. After a fixed number iterations, the final score matrix and the association for the detection is then extracted. Since all of the above layers are differentiable, we train the tracking module in a supervised manner with respect to the ground truth.

3.2.3 3D Pose Estimation

The last module of the network computes the persons' 3d poses using the persons descriptors and their tracking.

Spatio-temporal descriptors merging: If T is the tesseract temporal window size, then after tracking a person for T frames, we obtain T spatio-temporal descriptors of this person which overlap at a common time and encode the person's pose and motion over a total time interval of length $2T-1$. We thus merge all these descriptors to estimate the person's pose at their common time. As previously described, we use a softmax-based merging strategy and the result is a single tesseract description for the central frame.

Tesseract deconvolution: The merged tesseract is finally passed through multiple 4D deconvolution layers to produce 3D heatmaps of person's joints at time t. The predicted joint position k^q_{Pred} is obtained by a soft-argmax operator, i.e. by a heatmap scores-weighted average of the voxel centers. We then combine two loss functions for the pose estimation task: a L1 distance computed on the keypoints positions and a loss on the response of the heatmap at the ground truth joint position. The gradient is propagated back to the initial images, including through the HRNet backbone which is shared by the detection module and the tracking + pose estimation modules.

4 Data

4.1 Vehicle Reconstruction in 3D Space and Time

TRAFFIC4D Dataset: This is a novel dataset proposed in the paper to analyze data at intersections over a long duration. It includes 10 videos (70k frames) obtained from multiple sources: 3 live YouTube streams from static cameras and 7 views captured by iPhone 6 fixed on tripods. This dataset is divided into 3 stereo pairs and 4 single view videos. The stereo pairs were captured to evaluate the accuracy of 3D reconstruction. We sampled frames from the stereo pairs and computed 3D keypoints locations using the triangulation of manually annotated 2D keypoints. We also annotate the ground truth trajectory clusters.

AI City Challenge Dataset: There are few public datasets for fixed camera reconstruction. Track 1 of AI City Challenge 2019 has 5 monocular camera sets, two of them taken at intersections with enough traffic, so we choose these two sets having 8 cameras, 8k frames in total, each captured for around 5 minutes. The ground truth trajectories are manually annotated and projected on to 3D ground plane using homography. The reconstructed vehicles should lie on or close to these annotated trajectories and are used as metric for evaluating the reconstruction.

4.2 Multi-Person Articulated 3D Pose Tracking

We selected the following standard 3D human pose estimation datasets for experimental evaluation. All datasets provide calibrated camera poses.

Human3.6M was captured from 4 cameras with a single human performing multiple actions. The dataset contains 8 actors performing 16 actions captured in controlled indoor settings. Motion capture was used to create ground truth 3D poses. We use 6 sequences to train and 2 sequences (S09, S11) to test our algorithm.

TUM Shelf was captured indoors using 5 stationary cameras, with 4 people disassembling a shelf. The dataset provides sparse 3D pose annotations. Severe occlusions and random motion of the persons are the key challenges.

TUM Campus was captured outdoors using 3 stationary cameras, with 3 people interacting on campus grounds. Similar to *Shelf*, it provides sparse 3D pose annotations. The dataset is challenging for 3D pose estimation due to a small number of cameras and wide baseline views.

CMU Panoptic was built to understand human interactions in 3D. It contains 60 hours of data with 3D poses and tracking information captured by 500 cameras. We follow [**?**] and sample the same 5 cameras for evaluation, and use the same sequences for training. We split the training and testing sequences following [**?**].

Tagging was captured in unconstrained environments where people are interacting in a social setting. There are no constraints on the motion of the cameras or the number of persons during the capture. This "in the wild" setting makes this dataset particularly interesting for 3D pose tracking. However, since no GT pose annotations are available, we only use this dataset for qualitative evaluation.

5 Analyses and Results

5.1 Vehicle Reconstruction in 3D Space and Time

Figure 7 compares reconstruction on the stereo pairs of TRAFFIC4D. We observe higher PCK accuracy compared to other methods in 2D and 3D. Specifically, when no longitudinal self-supervision is used, our second view (v2) and 3D PCK are significantly higher than the others, indicating our reconstruction is more consistent in 3D. We emphasize that the global co-planar loss contributes to the improvement in reconstruction accuracy as it regularizes all the vehicles' poses in the video for better spatial consistency. Moreover, our method achieves better accuracy after 2D and 3D longitudinal self-supervision.
Figure 8 plots keypoint refinement results of 2D longitudinal self-supervision. The heatmaps illustrate that 2D experts supervise most frequently at image borders, occluded places, or positions far from the camera as expected from failures from the initial detector. For each instance, the three nearest neighbor experts (vehicles with accurate keypoints predicted from original detectors) are visualized. We notice the same vehicle correctly detected at neighbor frames or a different vehicle with a similar appearance from a different time instance are used as experts. Observe that the retrieved experts have accurate shape ensuring the success of longitudinal learning. Table 1 shows improvement on A3DP for our method compared to baselines on S01 and S02 sets of AI City dataset. Similar to Fig. 7, adding 2D and 3D longitudinal self-supervision improves A3DP as well.
Accuracy vs. Video Length: The key idea of longitudinal self-supervision is to accumulate information over time, so the duration of the video being used is a critical parameter affecting keypoint accuracy. For

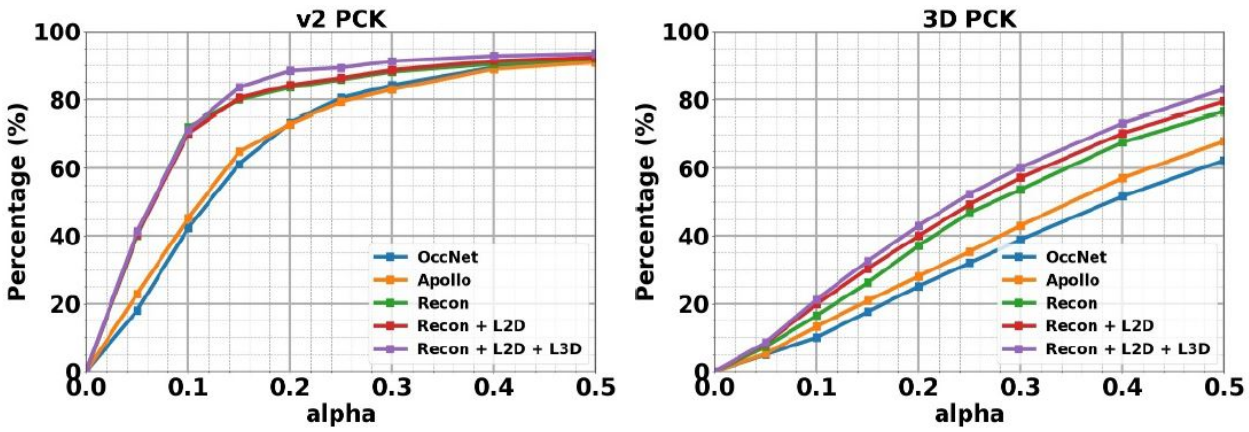

Figure 7: Accuracy of reconstruction with respect to varying window size (α) on TRAFFIC4D stereo pairs. **Left** and **right** are keypoints projected to the second view of stereo and reconstructed in 3D respectively. "Recon" indicates using our joint optimization for reconstruction. Note that longitudinal self-supervision (denoted L2D, L3D) consistently outperforms other baselines. Averaging over $\alpha = [0.05, 0.3]$, v2/3D PCK shows 35%/53% relative and 16%/12% absolute improvement over the nearest baseline.

Table 1: Comparing to state of the art trajectory reconstruction methods on AI City dataset using A3DP metric. "Mean", "c-l", and "c-s" denote mean, loose and strict criteria with different thresholds relative ("Rel") to depth. Traffic4D shows an average improvement of 14.62%(in absolute terms) and 34.2% (in relative terms) in comparison on both sequences, without any manual supervision.

			S01			S02		
Method	L2D	L3D	A3DP-Rel			A3DP-Rel		
			mean(in %)	c-l(in %)	c-s(in %)	mean(in %)	c-l(in %)	c-s(in %)
OccNet			9.30	45.44	8.90	12.21	51.54	6.98
Apollo			24.91	43.14	25.72	31.14	53.72	31.00
Traffic4D			28.03	47.55	24.84	41.04	63.86	44.68
Traffic4D	✓		33.11	57.49	30.96	44.27	63.90	46.99
Traffic4D	✓	✓	**39.42**	**63.88**	**40.16**	**45.86**	**65.59**	**47.11**

each sub-sequence split based on time specified, we construct the 2D expert pool and 3D experts from it and use them to refine over keypoints on the complete sequence. Figure 9 left illustrates the effect on reconstruction accuracy for varying sub-sequence length on TRAFFIC4D dataset stereo cameras. We observe a clear increase in accuracy with an increase in sub-sequence length illustrating that longitudinal supervision enhances the reconstruction accuracy. The accuracy converges after a specific duration of time emphasizing that the activity clustering for the sequence has been learned. We observe similar improvements in PCK accuracy on single view cameras as shown on the right in Fig. 9.

Repetitious Activity Clustering Analysis: Table 2 reports the proportion of correctly clustered trajectories in each video of TRAFFIC4D dataset. Notice that 3D clustering outperforms 2D in all the videos and our method achieves the highest accuracy in most sequences. The reason is trajectories in the same direction but belonging to different lanes look quite near each other if they are distant or the camera looks straight forward, while 3D clustering eliminates the view angle and perspective effect by converting them to 3D.

5.2 Multi-Person Articulated 3D Pose Tracking

Most recent works on multi-person articulated 3D pose tracking focus on evaluation of 3D pose reconstruction accuracy using MPJPE or 3D-PCK. However, this is not clear how existing methods advance actual body joint tracking accuracy in multi-person scenarios. We thus intend to fill in this gap and propose a

Figure 8: Examples of keypoint refinement via 2D longitudinal self-supervision. **First row:** Visualization of 2D experts. The heatmaps show frequency of 2D experts being used to refine other instances. 2D experts are used mostly at image border, occluded or far away places. The vehicle patches show the top three nearest neighbors retrieved from expert pool (good keypoints predicted by initial detector), which have very similar shape and pose to the refined instance; **Second row:** Initial erroneous keypoints from detector; **Third row:** Refined keypoints after 2D longitudinal self-supervision.

set of novel evaluation metrics for multi-person articulated 3D pose tracking. To that end, we build on the popular Multiple Object Tracking (MOT) and articulated 2D pose tracking metrics and extend them to the 3D pose use case. The proposed metrics require predicted 3D body poses with track IDs. First, for each pair of (predicted pose, GT pose) 3D-PCK is computed. Predicted and GT poses are matched to each other by a global matching procedure that maximizes per pose 3D-PCK. Finally, Multiple Object Tracker Accuracy (MOTA), Multiple Object Tracker Precision (MOTP), Precision, and Recall metrics are computed.

Evaluation details: Evaluation is performed on the Panoptic dataset using the proposed 3D MOTA metric. In the following we compare *FTDL* to *FTGL* and *FIG*.

Impact of temporal representations on tracking: Results are shown in Tab. 3. Using temporal person descriptors (*FTDL* and *FTGL*) significantly improves tracking accuracy compared to instantaneous person descriptor (*FIG*). Using a end-to-end learnable tracking framework (*FTDL*) instead of a Hungarian matching

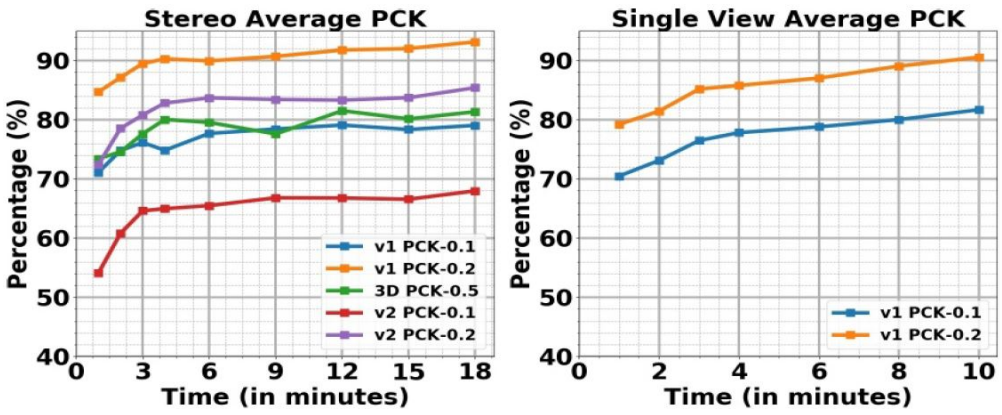

Figure 9: The plot depicts PCK-α accuracy improving over time by using longitudinal self-supervision. We observe 11% absolute and 16% relative improvement in average accuracy of 3D reconstruction and detections over stereo cameras (**left**) in TRAFFIC4D dataset with 18 minutes of continuous learning. Here, at time zero we use an off-the-shelf detector, while at 18 minutes we use a retrained detector from longitudinal self-supervision. We observe similar accuracy boost in the single view cameras (**right**) of TRAFFIC4D dataset.

Table 2: Comparing the accuracy of TRAFFIC4D clustering algorithm with previous clustering methods MS, MBMS , AMKS. The metric used is proportion of correctly clustered trajectories (higher is better). "2D" means clustering on trajectories using bounding box centers in image; "3D" means clustering on 3D trajectories reconstructed by our approach. We observe that using our hierarchical clustering algorithm improves the accuracy of clustering by 14.79% (in absolute terms) and 19.76% (in relative terms) with respect to current state of the art (3D AMKS).

Seq No.	2D MS	2D MBMS	2D AMKS	3D MS	3D MBMS	3D AMKS	Traffic4D
001	57.32	63.59	66.10	75.31	66.10	73.22	**90.37**
002	60.68	59.83	60.68	64.10	76.92	**83.76**	82.05
003	48.18	52.27	49.54	62.27	61.36	66.81	**90.90**
004	59.32	41.04	66.04	68.28	79.85	75.74	**93.28**
005	51.73	53.06	54.40	56.00	56.53	68.00	**86.67**
006	68.07	67.60	69.95	64.78	63.85	67.14	**85.44**
007	62.20	64.56	66.14	75.59	71.65	84.25	**91.34**
008	41.44	47.75	49.55	45.05	45.95	58.55	**91.89**
009	57.89	63.90	67.66	73.30	78.19	83.08	**86.09**
010	60.16	62.60	65.85	75.61	73.17	77.24	**85.36**

algorithm (*FTGL*) further improves tracking accuracy. This can be attributed to the fact that the learnable descriptors matching can distinguish interacting people much better than graph-based tracking methods.

Robustness to number of cameras: We analyze the accuracy of 3D pose tracking with respect to a varying number of cameras. Results are shown in Fig. 10 (right). While an increasing number of cameras allows improving the accuracy of all variants, we observe that relying on spatio-temporal representation learning results in significant tracking accuracy improvements specifically in the few cameras mode (*FTDL*

Method	Neck	Head	Shou.	Elbow	Wrist	Hip	Knee	Ankle	Avg
1*FIG	89.7	87.4	90.8	88.0	82.2	92.7	89.1	92.4	87.6
1*FTGL	93.9	91.7	93.0	92.1	87.4	94.4	**93.9**	94.6	92.1
1*FTDL	**94.6**	**93.6**	**93.4**	**92.7**	**88.2**	**94.7**	93.8	**95.0**	**94.1**

Table 3: 3D MOTA evaluations on the Panoptic dataset. Using an end-to-end learnable framework (*FTDL*) systematically improves the accuracy of 3D pose tracking across all keypoints.

and *FTGL* vs. *FIG*). Furthermore, using a learnable tracklet matcher (*FTDL*) results in consistent increase in tracking accuracy over a wide range of number camera views. Both observations underline the advantages of the proposed formulation when only a few cameras are available. Finally, in the pure monocular setting, *FTDL* achieves a reasonable 76% 3D MOTA accuracy, despite not being specifically tuned in this setting. We envision that incorporating scene constraints and performing spatio-temporal articulated model fitting should significantly boost the accuracy of in monocular setting.

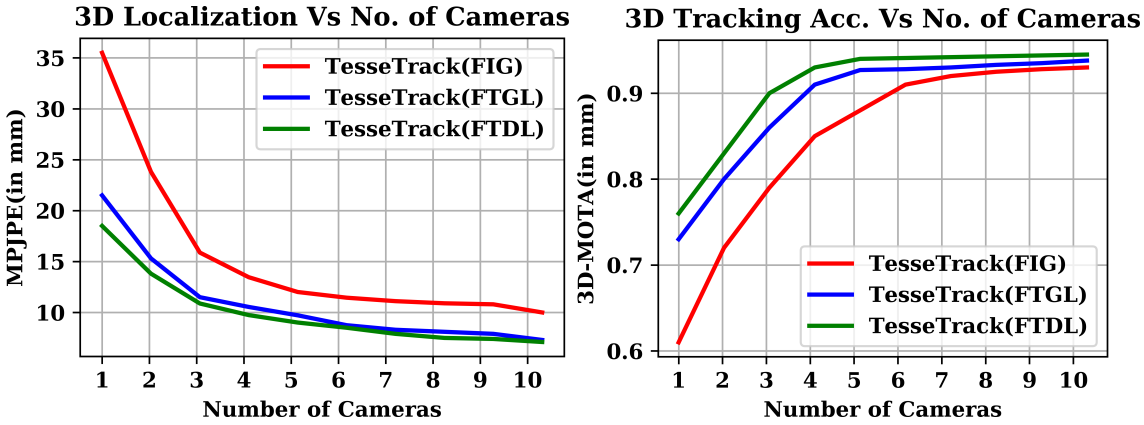

Figure 10: Impact of number of cameras on body joint localization error (MPJPE) (left) and pose tracking accuracy (3D MOTA) (right). Tessetrack (FTDL) shows the greatest advantage with lower number of cameras.

6 Findings

6.1 Vehicle Reconstruction in 3D Space and Time

(1) Vehicle velocity estimation and activity visualization: Vehicle activity reconstruction provides insights into driving behavior by estimating real world speeds. Each vehicle's velocity vector in world coordinates is obtained from trajectory by taking time derivatives. Fig. 11 shows the accurate reconstruction results of individual vehicles, 3D mean trajectories and speed profile after longitudinal self-supervision. *(2) Anomaly analysis:* As an application of our model, vehicular anomalies can be identified. The log likelihood of a trajectory belonging to a specific cluster is obtained by sampling from the corresponding Gaussian component in the clustering model. The trajectory is considered as an anomaly if its likelihoods are lower than a threshold in all the clusters. Compared to previous anomaly detection methods purely in 2D, the 3D anomaly trajectory also reveals the anomaly vehicle's position and velocity in 3D real world. Fig. 12 shows the trajectories and likelihood of anomalies.*(3) Vehicle counting:* The number of vehicles in each direction and lane is counted based on cluster ids. The supplementary video and webpage show the results.

6.2 Multi-Person Articulated 3D Pose Tracking

Shown in Figure 13 is the output of Tessetrack on the Tagging sequence. The top two row portray the projections of keypoints on two views, while the bottom row shows the 3D pose tracking. Observe smooth tracking of people in the wild with moving cameras for long duration of time.

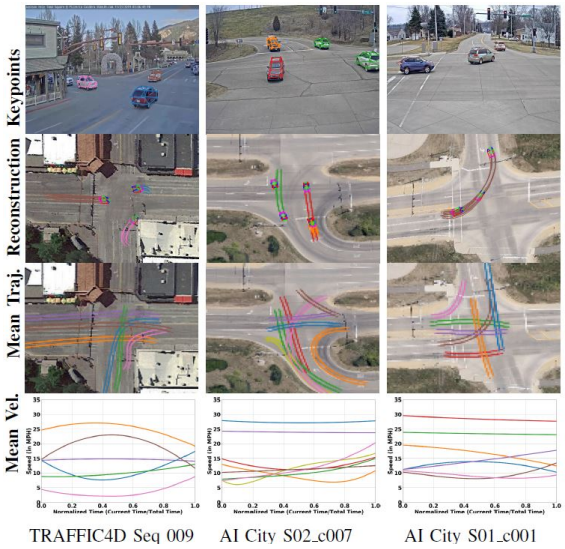

Figure 11: The keypoints **(first row)** and 3D reconstructions overlaid on Google map **(second row)** at different times, as well as 3D mean trajectories **(third row)** and velocities of the mean trajectories **(fourth row)** for three intersections. These mean trajectories represents typical vehicle motions and are used for 3D longitudinal self-supervision.

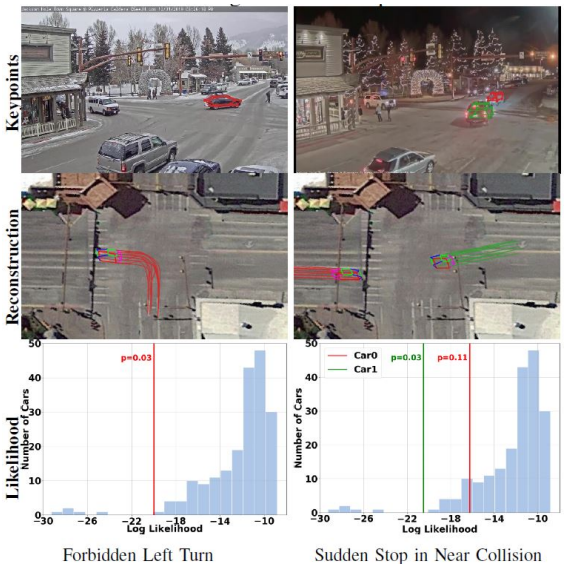

Figure 12: Automatic anomaly detection. The plot shows different anomalies like vehicles making forbidden left turn **(Left column)**, sudden stop in near collision **(Right column)** using our method. **Last row** shows the anomaly's log likelihood (red/green lines, p represents the probability) is much lower than the normal trajectories (blue bars) in the cluster.

Figure 13: We illustrate the output of Tessetrack on the Tagging sequence. The top two row portray the projections of keypoints on two views, while the bottom row shows the 3D pose tracking. Observe smooth tracking of people in the wild with moving cameras for long duration of time.

7 Conclusions

Vehicle Reconstruction in 3D Space and Time: We proposed a novel approach to reconstruct repetitious vehicular activity in 4D from a single view using longitudinal self-supervision. Our algorithm takes as input off-the-shelf 2D keypoint detections, optimizes 3D vehicle poses and clusters their motion in 3D space. The accumulated 2D keypoints and trajectory clusters are then used to refine the 2D and 3D keypoints without any human annotation. Experimental results show our self-learning framework greatly improves the accuracy of detection and reconstruction on long term testing videos unseen by the detector. In the future, longitudinal self-supervision could be extended to people or robot activity reconstruction with analogous keypoint detectors and geometric constraints.

Multi-Person Articulated 3D Pose Tracking: Reliably reconstructing and tracking the 3D poses of multiple persons in real-world scenarios using calibrated cameras is a challenging problem. In this work, we address it by proposing a novel formulation, which jointly solves the tasks of tracking and 3D pose reconstruction within a single end-to-end learnable framework. In contrast to previous piece-wise strategies which first reconstruct 3D poses based on geometrical optimization algorithms and then subsequently linking the poses over time, our method infers the number of persons in a scene and jointly reconstructs and tracks their 3D poses using a novel 4D spatio-temporal CNN and a learnable tracking framework using differentiable matching. Experimental evaluation on five challenging datasets show significant improvements not only in multi-person 3D pose tracking but also in multi-person 3D pose reconstruction accuracy.

8 Recommendations

With the current and impending focus on improving traffic mobility through smart cities technologies, more and more sensors are being deployed to the infrastructure. Sensors like cameras provide a wealth of knowledge and information but require heavy computing to distil the information. Time-sensitive information and applications also require low latency in terms of compute times but also transmission of information. This work set out to address the technical challenges of detecting, tracking, and reconstructing vehicles and people while addressing these challenges. Computer vision algorithms developed reliably detect and track vehicles in 3D space and time, but also do the same for people and their pose. The algorithms enable computation of various analytics that can be of interest to city planners, connected vehicles, etc. To demonstrate the utility of these algorithms, they were applied to data from camera's installed in the Pittsburgh area (with remote and edge processing), as well as cameras installed throughout the world– those available through live web streams. The process of transforming image data to analytic information is a compressive process that transforms data from thousands of pixels every second to a small fraction of that, e.g., vehicle counts, people counts, etc. The road infrastructure deserves attention, not just in standard maintenance, but in moving towards a future where sensing and computing become a part of the infrastructure. This idea will enable the potential for real-time information about activity on the road, which in turn can inform planning decisions or even instantaneous actuation, e.g., transmitting information to driver assist path planning systems.

9 Publications

- "Traffic4D: Single View Longitudinal 4D Reconstruction of Repetitious Activity using Self-Supervised Experts," Fangyu Li, N. Dinesh Reddy, Xudong Chen and Srinivasa G. Narasimhan. IEEE Intelligent Vehicles Symposium (IV), 2021. **Best Paper Award**.
- "TesseTrack: End-to-End Learnable Multi-Person Articulated 3D Pose Tracking," N Dinesh Reddy, Laurent Guigues, Leonid Pischulini, Jayan Eledath and Srinivasa G. Narasimhan. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.