Traffic4D: Single View Reconstruction of Repetitious Activity Using Longitudinal Self-Supervision

Fangyu Li[1], N. Dinesh Reddy[2], Xudong Chen[1] and Srinivasa G. Narasimhan[2]
http://www.cs.cmu.edu/ ILIM/projects/IM/TRAFFIC4D/

***Abstract*— Reconstructing 4D vehicular activity (3D space and time) from cameras is useful for autonomous vehicles, commuters and local authorities to plan for smarter and safer cities. Traffic is inherently repetitious over long periods, yet current deep learning-based 3D reconstruction methods have not considered such repetitions and have difficulty generalizing to new intersection-installed cameras. We present a novel approach exploiting longitudinal (long-term) repetitious motion as self-supervision to reconstruct 3D vehicular activity from a video captured by a single fixed camera. Starting from off-the-shelf 2D keypoint detections, our algorithm optimizes 3D vehicle shapes and poses, and then clusters their trajectories in 3D space. The 2D keypoints and trajectory clusters accumulated over long-term are later used to improve the 2D and 3D keypoints via self-supervision without any human annotation. Our method improves reconstruction accuracy over state of the art on scenes with a significant visual difference from the keypoint detector's training data, and has many applications including velocity estimation, anomaly detection and vehicle counting. We demonstrate results on traffic videos captured at multiple city intersections, collected using our smartphones, YouTube, and other public datasets.**

I. Introduction

Understanding vehicle motion in 3D space is useful for intelligent traffic systems. 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 [1], [2] 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 like KITTI [3] or Argoverse [4] 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 [5], [6], [7], [8], [9], they focus on short intervals such as over a few frames or seconds.

[1] are with NVIDIA,USA {fangyul,xudongc} @nvidia.com
[2] are with CMU, USA {dnarapur,srinivas}@cs.cmu.edu

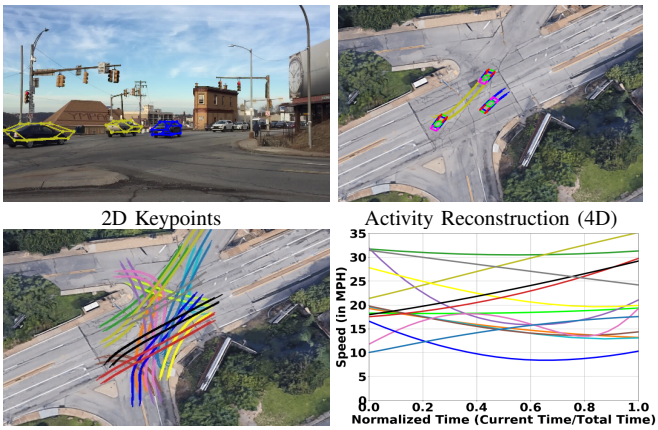


Fig. 1: Long term repetitious vehicular activity is used as self-supervision to compute accurate 2D and 3D keypoints, trajectories and velocities from a single fixed camera. Reconstruction accuracy improves significantly over 20 minutes at this intersection as compared to methods that enforce consistency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

In this work, we argue that 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 adapt a pre-trained keypoint detector [10] to new scenes it never saw before, and obtain higher accuracy 2D and 3D keypoints without any manual annotation. 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 contributions are summarized below and the entire framework is shown in Fig. 2:

(a) Joint optimization for longitudinal reconstruction (Sec IV-A): Consistent reconstruction of diverse motion and poses from single-view by joint optimization over all vehicles in long-term videos. This improves 3D keypoint reconstruction accuracy by 29% relatively over state of the art [11].

(b) Scene-specific repetitious activity clustering (Sec IV-B):

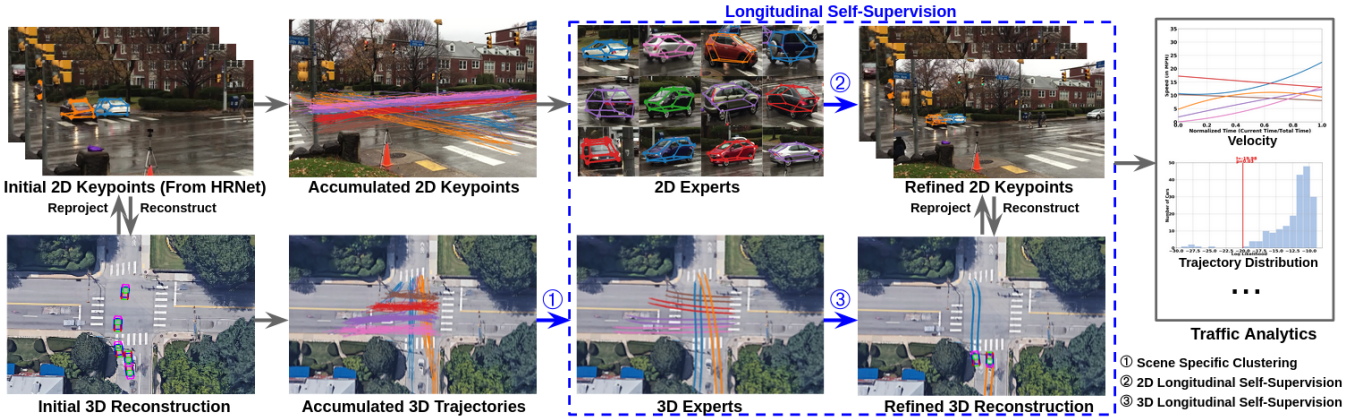


Fig. 2: Framework for self-supervised 4D reconstruction of repetitious activity. Our method takes off-the-shelf 2D keypoint detections as input, reconstructs 3D keypoints with an active shape model, fits an analytic trajectory model to each vehicle's 3D poses along with frames, and accumulates them over 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.

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. This method outperforms the state of the art clustering algorithm by 25% relatively.

(c) 2D/3D longitudinal self-supervision (Sec IV-C): 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. The continuous self-learning framework improves the accuracy of detection and reconstruction by 16% using self-supervision over long term videos.

We demonstrate the versatility and generalizability of our approach 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 [12]. We also apply our method to traffic tasks such as velocity estimation, anomaly detection and vehicle counting. **See supplementary video and the project webpage for better visualizations of our results**.

II. Related Work

Single View Reconstruction: Many methods utilize Lidar [13], [14], IMU [15], UAV [16] to acquire 3D information, or deploy deep networks to infer 3D geometric properties from RGB images, largely in a supervised manner [17], [18], [1], [19]. For the pure RGB methods, obtaining 3D ground truth in the wild is challenging. Further, deep models trained on the subset of data do not generalize well. To address these issues, shapes and poses are optimized with stronger geometrical constraints instead of 3D labels. Works [20], [2], [11], [21] build active shape models to optimize/retrieve 3D shapes from 2D images. Recent works [11], [20], [22] enforce coplanar or pairwise distance constraints for short term or local objects to resolve ambiguity in reconstruction. All of these methods do not study long term temporal consistency. As far as we know, our method is the first to perform trajectory reconstruction using long term self-supervision from a single 2D view.

Repetitious Activity Analysis: Multiple methods model repetitious activity using dimensionality reduction [23], [24] and clustering [25], [26]. Specific to modeling repetitious activity, [27] proposed clustering vehicle trajectories using kernel shrinkage. However, these previous methods are constrained to 2D image trajectories and are not robust to noise. In contrast, our method the first to uplift 2D vehicle trajectories to 3D, resulting in strong separability of clusters in higher dimensions and achieving state of the art accuracy.

Self-Supervision in the Wild: Supervised methods require large amount of labels and are sensitive to training data. To circumvent these issues, the community has collectively proposed many weakly supervised or self-supervised methods with automatic supervisory signals such as shape symmetry [28], [29], [30] and style consistency [31], [32]. In addition, many works [33], [34], [35] utilize alignment between frames to learn optical flow; [6], [36] detect and reconstruct objects based on their motion over frames. All these supervisions come from short intervals such as a few frames or seconds. But in this paper we argue long term consistency can be a strong supervisory signal and propose longitudinal self-supervision to improve the accuracy of detection and reconstruction simultaneously.

III. Background

Here we introduce the notation, 3D shape representation and motion model as preliminaries to our approach.

Notation used in the paper: We use three coordinate systems, i.e. camera, world and map coordinates as shown in Fig. 3. The camera coordinate is defined with origin at focal point, XY parallel to image plane; while in world coordinate

XY is the ground plane and Z axis points upwards. The two coordinate systems are associated by a rigid transform. In world coordinate each object's trajectory is represented with x, y as we assume coordinate z to be constant with a planar ground. 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 M frames, a total of N unique objects are captured with J keypoints for each instance. $\mathbf{P}^{(c)}_{n,m,j}$ and $\mathbf{p}_{n,m,j}$ denotes the 3D position (in camera coordinates) and 2D position (in image coordinates) of the j-th keypoint of the n-th instance in m-th frame, respectively. Each instance's rotation and translation vector ($\mathbf{r}^{(c)}_{n,m}$, $\mathbf{t}^{(c)}_{n,m}$) are in camera coordinates, while ($\mathbf{r}^{(w)}_{n,m}$, $\mathbf{t}^{(w)}_{n,m}$) are in world coordinates. $\boldsymbol{\pi}(\cdot)$ is the 3D-to-2D camera projection and $\boldsymbol{\eta}^{(c)} : \eta_1 x + \eta_2 y + \eta_3 z + \eta_4 = 0$ is the ground plane in camera coordinates.

3D Shape Model: We parameterize the object 3D keypoints by an active shape model [21] to regularize shape optimization. The mean shape $\bar{\mathbf{Q}}$ of all object models, and their principle components $\mathbf{Q}_1, ..., \mathbf{Q}_K$ are computed from an object CAD model dataset [37]. Then each object's actual shape $\mathbf{X}_n$ is formulated as linear combination of mean shape with the top K principal components: $\mathbf{X}_n = \bar{\mathbf{Q}} + \sum_{k=1}^{K} \alpha_{n,k} \mathbf{Q}_k$, where α_n is the shape coefficient vector that needs to be estimated in the later optimization stage. For each object, we track it over time and enforce the shape parameter α_n 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. The trajectory of the n-th object $\hat{\mathbf{t}}^{(w)}_n = [\hat{t}^{(w)}_{n,x}, \hat{t}^{(w)}_{n,y}]^T$ is parameterized as

$$\hat{t}^{(w)}_{n,x}(t) = a_h t^h + ... + a_2 t^2 + a_1 t + a_0 \tag{1}$$

$$\hat{t}^{(w)}_{n,y}(t) = b_h t^h + ... + b_2 t^2 + b_1 t + b_0 \tag{2}$$

where $\mathbf{c}_n = [a_h, \ldots, a_0, b_h, \ldots, b_0]^T$ denotes the parameters to solve and t represents the time-stamps in video. $t = m - m_0$ for the object in frame m with first appearance in frame m_0, so all objects are aligned temporally. We observed 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 (from Sec IV-A) are used to solve $\mathbf{c}_n$ by minimizing ℓ_2 loss. In frame m, the coordinate $\hat{\mathbf{t}}^{(w)}_{n,m}$ and tangent $\nabla\hat{\mathbf{t}}^{(w)}_{n,m}$ predicted by $\mathbf{c}_n$ should be close to the reconstructed pose ($\mathbf{t}^{(w)}_{n,m}, \mathbf{r}^{(w)}_{n,m}$) in XY plane. We convert both $\nabla\hat{\mathbf{t}}^{(w)}_{n,m}$, $\mathbf{r}^{(w)}_{n,m}$ into direction vector denoted as $\mathbf{u}(\cdot)$. We also add regularizing terms for third order coefficients.

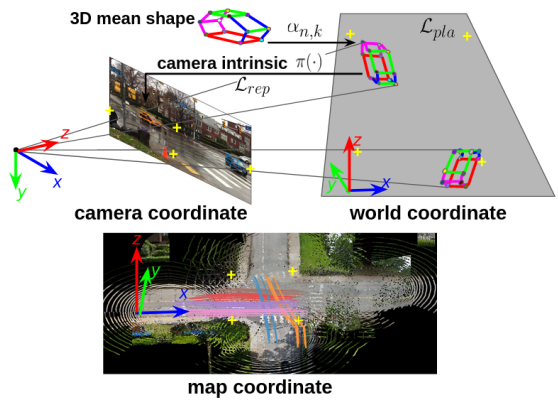


Fig. 3: 3D reconstruction coordinate frames. Vehicle 3D keypoints are computed in camera coordinates. The world coordinate is defined with XY as the ground plane, in which we perform analytic model fitting and repetitious activity clustering. Map coordinates are defined based on Google maps, whose XY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

$$\mathscr{L}_{fit,n} = \sum_m \Bigg(\left\| \hat{\mathbf{t}}^{(w)}_{n,m} - \mathbf{t}^{(w)}_{n,m} \right\|^2 + \beta_1 \left\| \mathbf{u}(\nabla\hat{\mathbf{t}}^{(w)}_{n,m}) - \mathbf{u}(\mathbf{r}^{(w)}_{n,m}) \right\|^2 + \beta_2 a_3^2 + \beta_3 b_3^2 \Bigg) \tag{3}$$

where $\beta_1, \beta_2, \beta_3$ are weight coefficients for the loss terms.

IV. 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.

A. 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) [11], [21]. 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 [38] 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 [39]. 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 n at frame m can be computed from the shape model parameterized by α_n with object pose ($\mathbf{r}^{(c)}_{n,m}, \mathbf{t}^{(c)}_{n,m}$) as:

$$\mathbf{P}^{(c)}_{n,m} = \mathbf{R}^{(c)}_{n,m}(\bar{\mathbf{Q}} + \sum_{k=1}^{K} \alpha_{n,k}\mathbf{Q}_k) + \mathbf{t}^{(c)}_{n,m} \tag{4}$$

where $\mathbf{R}^{(c)}_{n,m}$ is the rotation matrix from $\mathbf{r}^{(c)}_{n,m}$. We need to optimize the shape coefficients vector α_n and pose $(\mathbf{r}^{(c)}_{n,m}, \mathbf{t}^{(c)}_{n,m})$ 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.

$$\mathscr{L}_{rep} = \sum_{n,m,j} \|\boldsymbol{\pi}(\mathbf{P}^{(c)}_{n,m,j}) - \mathbf{p}_{n,m,j}\|^2 \tag{5}$$

(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 $\mathbf{P}^{(c)}_{n,m,j_b} = [P^{(c)}_{n,m,j_b,x}, P^{(c)}_{n,m,j_b,y}, P^{(c)}_{n,m,j_b,z}]^T$ (center of the rectangle formed by joining wheel centers) and the ground plane $\boldsymbol{\eta}^{(c)}$.

$$\mathscr{L}_{pla} = \sum_{n,m} \frac{(\eta_1 P^{(c)}_{n,m,j_b,x} + \eta_2 P^{(c)}_{n,m,j_b,y} + \eta_3 P^{(c)}_{n,m,j_b,z} + \eta_4)^2}{\eta_1^2 + \eta_2^2 + \eta_3^2} \tag{6}$$

We solve α_n, $\mathbf{r}^{(c)}_{n,m}$, $\mathbf{t}^{(c)}_{n,m}$ and $\boldsymbol{\eta}^{(c)}$ by minimizing the two losses via Levenberg-Marquardt optimization: $\mathscr{L}_{rec} = \gamma_1 \mathscr{L}_{rep} + \gamma_2 \mathscr{L}_{pla}$, where γ_1, γ_2 are the weights of corresponding loss terms.

B. 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 [40], [27]. Thus, we first fit a polynomial model to each object's 3D poses to suppress noise and reduce data dimension as described in Section III. Then, the trajectory parameters are clustered hierarchically and projected to subspaces 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. 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. We define the direction of a trajectory as the vector between its starting and ending points

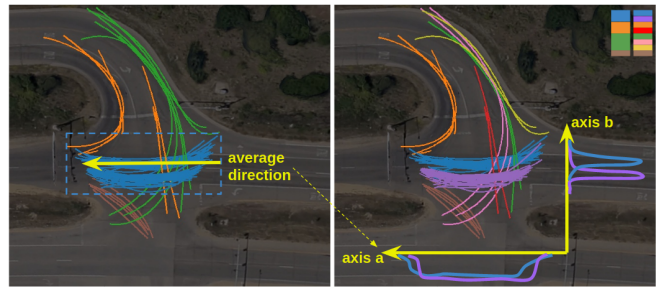


Fig. 4: Demonstration of our hierarchical clustering in birds-eye view. **Left:** First stage clusters and the average direction of the blue cluster. **Right:** Second stage clustering. Trajectories are projected along their average direction, maximizing the spatial difference between near clusters. 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**.

$\Delta\mathbf{t}^{w}_{n,\min t_n,\max t_n} = \mathbf{t}^{w}_{n,\max t_n} - \mathbf{t}^{w}_{n,\min t_n}$, and the average direction in each cluster i from the first stage is computed among all N_i objects as $\mathbf{p}_i = \frac{1}{N_i}\sum_n \Delta\mathbf{t}^{w}_{n,\min t_n,\max t_n} = [p_{i,x}, p_{i,y}, 0]^T$. Then each trajectory is projected along the average direction as:

$$\hat{t}'_n(t) = \frac{\hat{t}^{(w)}_{n,x}(t)p_{i,x} + \hat{t}^{(w)}_{n,y}(t)p_{i,y}}{\|\mathbf{p}_i\|} \tag{7}$$

which is still an h-order polynomial with $\mathbf{c}'_n = [a_h p_{i,x} + b_h p_{i,y}, \ldots, a_0 p_{i,x} + b_0 p_{i,y}]^T$ as coefficients. 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.

C. 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 δ_{rep} 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 $\mathbf{s}_{n,m}^{(b)}$ and translation $\mathbf{t}_{n,m}^{(b)}$ from the 2D expert bounding box $\hat{\mathbf{b}}_{n,m}$ to the instance bounding box $\mathbf{b}_{n,m} = \mathbf{s}_{n,m}^{(b)}\hat{\mathbf{b}}_{n,m} + \mathbf{t}_{n,m}^{(b)}$. Then the optimized transformations $\mathbf{s}_{n,m}^{(b)}$, $\mathbf{t}_{n,m}^{(b)}$ 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 $\mathbf{t}_{n,m}^{(c)}$, $\mathbf{r}_{n,m}^{(c)}$ in frame m and the 3D expert of one specific cluster, we find a point $\hat{\mathbf{t}}_{n,m}^{(c)}$ on the 3D expert minimizing its distance to the object position $\mathbf{t}_{n,m}^{(c)}$. 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: $d_{n,cham} = \sum_m \|\mathbf{t}_{n,m}^{(c)} - \hat{\mathbf{t}}_{n,m}^{(c)}\|$. 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 δ_{ch}, it is used to refine the object pose. For the pose $\mathbf{t}_{n,m}^{(c)}$, $\mathbf{r}_{n,m}^{(c)}$ in frame m, we find its closest point $\hat{\mathbf{t}}_{n,m}^{(c)}$ on the 3D expert when calculating Chamfer distance. $\hat{\mathbf{r}}_{n,m}^{(c)}$ is the tangent direction of the 3D expert at $\hat{\mathbf{t}}_{n,m}^{(c)}$. We propose the 3D longitudinal loss to learn correspondence between individual pose and 3D experts by minimizing

$$\mathscr{L}_{long} = \beta_4 \|\mathbf{t}_{n,m}^{(c)} - \hat{\mathbf{t}}_{n,m}^{(c)}\|^2 + \beta_5 \|\mathbf{r}_{n,m}^{(c)} - \hat{\mathbf{r}}_{n,m}^{(c)}\|^2 \tag{8}$$

where β_4 and β_5 are coefficients. We add this longitudinal loss term and refine the 3D reconstruction by optimizing $\mathscr{L}_{refine} = \gamma_1 \mathscr{L}_{rep} + \gamma_2 \mathscr{L}_{pla} + \gamma_3 \mathscr{L}_{long}$.

V. Experimental Evaluation

We evaluate our approach on two datasets captured at intersections by stationary cameras with various view angles, vehicle motions, and scene appearances. A new dataset captured by us named TRAFFIC4D, and a public dataset AI City Challenge [12] have been used in all experiments. We compare our method with other benchmarks and analyze how 2D and 3D longitudinal self-supervision improve reconstruction accuracy. We compare our repetitious activity clustering accuracy with the state of the art to show the advantage of using scene-specific clustering. We also demonstrate application to traffic tasks such as velocity estimation, anomaly analysis and vehicle counting.

A. Datasets

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 [12] 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.

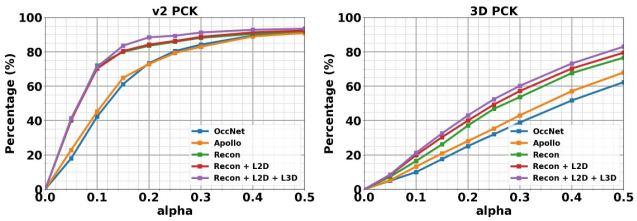


Fig. 5: Accuracy of reconstruction with respect to varying window size (α) on TRAFFIC4D stereo pairs. **Left** and **right** are keypoints projected to the second view of stereo and reconstructed in 3D respectively. "Recon" indicates using our joint optimization for reconstruction. Note that longitudinal 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.

B. Evaluation Metrics and Baseline Methods

CarFusion dataset [41] is used to pretrain our 2D keypoint detector [38]. Then we run the detector and perform reconstruction, clustering, and longitudinal self-supervision on the two evaluation datasets without using any ground truth annotations. Note that the appearance and view angle of the evaluation datasets and Carfusion are quite different.

Fig. 6: 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.

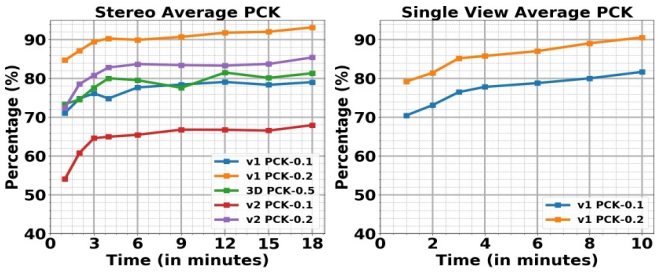


Fig. 7: The plot depicts PCK-α accuracy improving over time by using longitudinal self-supervision. We observe 11% absolute and 16% relative improvement in average accuracy of 3D reconstruction and detections over stereo cameras (**left**) in TRAFFIC4D dataset with 18 minutes of continuous learning. 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.

We analyze the accuracy of our reconstruction by using metrics both in 2D and 3D. We use 3D-PCK (Percentage of Correct Keypoints) [42] between our 3D reconstructed keypoints and 3D ground truth keypoints for evaluating the reconstruction. We further evaluate the reconstruction by comparing the reprojection of keypoints onto the stereo pair with ground truth using 2D-PCK. According to the PCK metric, a keypoint is considered correct if it lies within the radius αL of the ground truth. Here L is defined as the maximum length and width of the bounding box and $0 < \alpha < 1$. For data without stereo, we compare 3D poses with the annotated ground truth trajectory using the A3DP metric [11]. For each reconstructed pose, we find its nearest point on the ground truth trajectory. This nearest point's location and the tangent direction are used as ground truth translation and rotation. As in [11], the criteria for judging a true positive is that both the rotation and translation differences lie within a threshold.

For reconstruction comparisons, we use two state of the art methods i.e. Apollo3D [11] and Occnet [21]. To make a fair comparison, we use HRNet as the common backbone for all the approaches. These methods act as strong baselines to evaluate the 3D and 2D pose reconstruction of objects.

For clustering, we compare with multiple state of the art 2D trajectory clustering methods i.e. AMKS [27], MS [40], MBMS [43]. We further extend these methods to 3D for a fair comparison with our method. For 2D we keep the algorithms unchanged and use each vehicle's bounding box center trajectory as input; For 3D we feed 3D trajectories given by Sec IV-A to all the algorithms. We report the proportion of correctly clustered trajectories metric to evaluate our method as proposed in [27].

C. Accuracy Analysis

Reconstruction Analysis: Fig. 5 compares reconstruction on the stereo pairs of TRAFFIC4D. We observe higher PCK accuracy compared to [21] and [11] 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.

Fig. 6 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 I shows improvement on A3DP for our method compared to baselines on S01 and S02 sets of AI City dataset. Similar to Fig. 5, 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 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. Fig. 7 left illustrates the effect on reconstruction accuracy for varying sub-sequence length

TABLE I: 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 [11]. Traffic4D shows an average improvement of 14.62%(in absolute terms) and 34.2% (in relative terms) compared to [11] on both sequences, without any manual supervision.

Method	L2D	L3D	S01 A3DP-Rel mean(in %)	S01 A3DP-Rel c-l(in %)	S01 A3DP-Rel c-s(in %)	S02 A3DP-Rel mean(in %)	S02 A3DP-Rel c-l(in %)	S02 A3DP-Rel c-s(in %)
OccNet [21]			9.30	45.44	8.90	12.21	51.54	6.98
Apollo [11]			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**

TABLE II: Comparing the accuracy of TRAFFIC4D clustering algorithm with previous clustering methods MS [40], MBMS [43], AMKS [27]. 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).

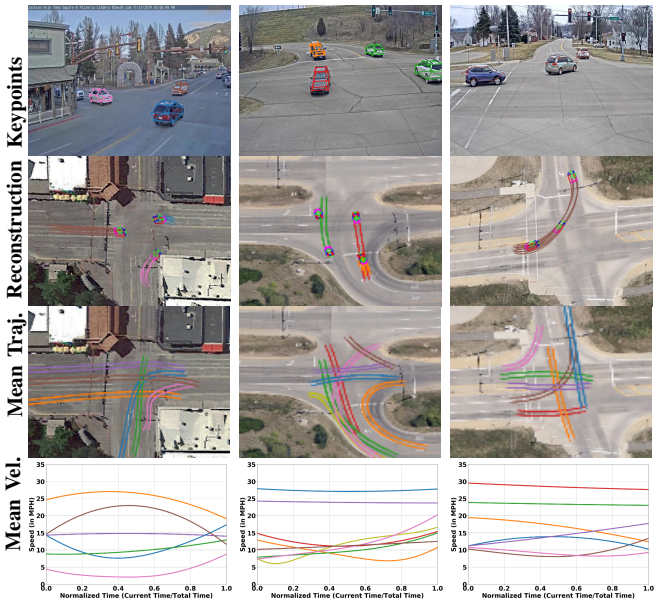
Seq No.	2D MS	2D MBMS	2D AMKS	3D MS	3D MBMS	3D AMKS	Traffic4D
001	57.32	63.59	66.10	75.31	66.10	73.22	**90.37**
002	60.68	59.83	60.68	64.10	76.92	**83.76**	82.05
003	48.18	52.27	49.54	62.27	61.36	66.81	**90.90**
004	59.32	41.04	66.04	68.28	79.85	75.74	**93.28**
005	51.73	53.06	54.40	56.00	56.53	68.00	**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**

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. 7.

Repetitious Activity Clustering Analysis: Table II 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.

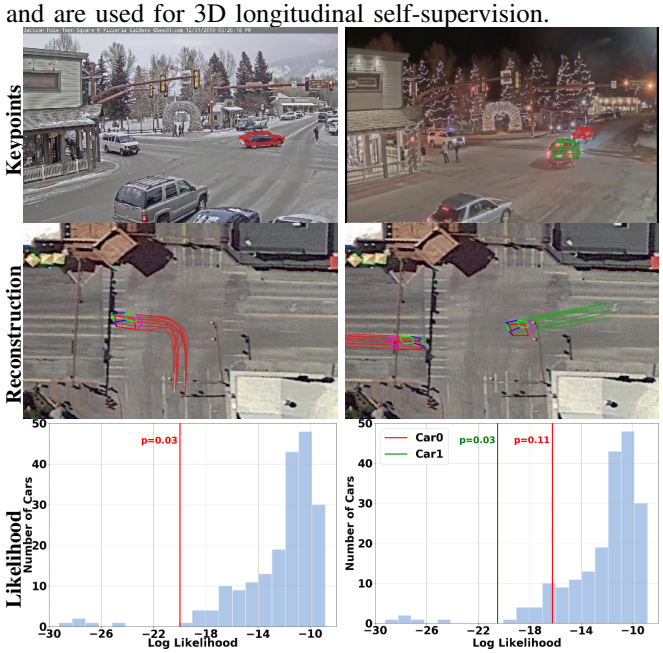
D. Applications

(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 taking time derivative: $v_{n,x}^{(w)}(t) = \frac{d\hat{t}_{n,x}^{(w)}(t)}{dt}$, $v_{n,y}^{(w)}(t) = \frac{d\hat{t}_{n,y}^{(w)}(t)}{dt}$. Fig. 8 shows the accurate reconstruction results of individual vehicles, 3D mean trajectories and speed profile after longitudinal self-supervision.



TRAFFIC4D Seq 009 AI City S02_c007 AI City S01_c001

Fig. 8: 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.



Forbidden Left Turn Sudden Stop in Near Collision

Fig. 9: 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.

(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. 9 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.

VI. Conclusion and Future Work

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.

VII. Acknowledgments

This paper was supported in parts by NSF Grants IIS-1900821 and CNS-2038612, DOT RITA Mobility-21 Grant 69A3551747111, and a PhD fellowship from Amazon Go.

References

[1] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *CVPR*, 2017.
[2] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in *CVPR*, 2017.
[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.
[4] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *CVPR*, 2019.
[5] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine, "Sfv: Reinforcement learning of physical skills from videos," *ACM Trans. Graph.*, vol. 37, no. 6, 2018.
[6] C. Lin, O. Wang, B. C. Russell, E. Shechtman, V. G. Kim, M. Fisher, and S. Lucey, "Photometric mesh optimization for video-aligned 3d object reconstruction," in *CVPR*, 2019.
[7] S. Tulsiani, A.Efros, and J. Malik, "Multi-view consistency as supervisory signal for learning shape and pose prediction," in *CVPR*, 2018.
[8] J. Gwak, C.Choy, M.Chandraker, Garg, and Savarese, "Weakly supervised 3d reconstruction with adversarial constraint," in *3DV*, 2017.
[9] N. D. Reddy, P. Singhal, V. Chari, and K. M. Krishna, "Dynamic body vslam with semantic constraints," in *IROS*, 2015.
[10] G. D. P. G. R. He, Kaiming Gkioxari, "Mask r-cnn," in *ICCV*, 2017.
[11] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang, "Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving," in *CVPR*, 2019.
[12] M. Naphade, Z. Tang, M.-C. Chang, D. C. Anastasiu, A. Sharma, R. Chellappa, S. Wang, P. Chakraborty, T. Huang, J.-N. Hwang, and S. Lyu, "The 2019 ai city challenge," in *CVPR Workshops*, 2019.
[13] A. Scheel, C. Knill, S. Reuter, and K. Dietmayer, "Multi-sensor multi-object tracking of vehicles using high-resolution radars," in *IVS*, 2016.
[14] P. Martinek, G. Pucea, Q. Rao, and U. Sivalingam, "Lidar-based deep neural network for reference lane generation," in *IVS*, 2020.
[15] B. Roessle and S. Gruenwedel, "Vehicle localization in six degrees of freedom for augmented reality," in *IVS*, 2020.
[16] C. J. de Frías, A. Al-Kaff, F. M. Moreno, Madridano, and J. M. Armingol, "Intelligent cooperative system for traffic monitoring in smart cities," in *IVS*, 2020.
[17] N. Gählert, J. J. Wan, N. Jourdan, J. Finkbeiner, and J. Denzler, "Single-shot 3d detection of vehicles from monocular rgb images via geometrically constrained keypoints in real-time," in *IVS*, 2020.
[18] N. Gählert, J. Wan, M. Weber, J. M. Zöllner, U. Franke, and J. Denzler, "Beyond bounding boxes: Using bounding shapes for real-time 3d vehicle detection from monocular rgb images," in *IVS*, 2019.
[19] P. Li, H. Zhao, and F. Cao, "RTM3D: real-time monocular 3d detection from object keypoints for autonomous driving," in *ECCV*, 2020.
[20] F. Chhaya, D. Reddy, S. Upadhyay, V. Chari, M. Z. Zia, and K. M. Krishna, "Monocular reconstruction of vehicles: Combining slam with shape priors," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5758–5765.
[21] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Occlusion-net: 2d/3d occluded keypoint localization using graph networks," in *CVPR*, 2019.
[22] Y. Chen, L. Tai, K. Sun, and M. Li, "Monopair: Monocular 3d object detection using pairwise spatial relationships," in *CVPR*, 2020.
[23] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Trajectory space: A dual representation for nonrigid structure from motion," *TPAMI*, 2011.
[24] T. Zhang, H. Lu, and S. Z. Li, "Learning semantic scene models by object classification and trajectory clustering," in *CVPR*, 2009.
[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996.
[26] Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato, and Yozo Fujino, "Learning motion patterns and anomaly detection by human trajectory analysis," in *ICSMC*, 2007.
[27] H. Xu, Y. Zhou, W. Lin, and H. Zha, "Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage," in *ICCV*, 2015.
[28] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised learning of probably symmetric deformable 3d objects from images in the wild," in *CVPR*, 2020.
[29] R. Yeh, Y.-T. Hu, and A. Schwing, "Chirality nets for human pose regression," in *NeurIPS*, 2019, pp. 8163–8173.
[30] A. Kanazawa, S. Tulsiani, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *ECCV*, 2018.
[31] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *CVPR*, 2018.
[32] J.-Y. Zhu, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *CVPR*, 2017.
[33] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *CVPR*, 2019, pp. 2566–2576.
[34] A. W. Harley, S. K. Lakshmikanth, F. Li, X. Zhou, H.-Y. F. Tung, and K. Fragkiadaki, "Learning from unlabelled videos using contrastive predictive neural 3d mapping," in *ICLR*, 2020.
[35] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *CVPR*, 2018.
[36] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele, "Kinematic 3d object detection in monocular video," in *ECCV*, 2020.
[37] C. Li, M. Z. Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with shape concepts for occlusion-aware 3d object parsing," *arXiv preprint arXiv:1612.02699*, 2016.
[38] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *TPAMI*, 2020.
[39] E. Bochinski, T. Senst, and T. Sikora, "Extending iou based multi-object tracking by visual information," in *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018.
[40] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *TPAMI*, vol. 24, no. 5, pp. 603–619, 2002.
[41] N. Dinesh Reddy, M. Vo, and S. G. Narasimhan, "Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicle," in *CVPR*, 2018.
[42] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR*, 2011.
[43] W. Wang and M. A. Carreira-Perpinán, "Manifold blurring mean shift algorithms for manifold denoising," in *CVPR*, 2010.