CarFusion: Combining Point Tracking and Part Detection for Dynamic 3D Reconstruction of Vehicles

N Dinesh Reddy Minh Vo Srinivasa G. Narasimhan
Robotic Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, USA
{dnarapur,mpvo,srinivas}@cs.cmu.edu

Abstract

Despite significant research in the area, reconstruction of multiple dynamic rigid objects (eg. vehicles) observed from wide-baseline, uncalibrated and unsynchronized cameras, remains hard. On one hand, feature tracking works well within each view but is hard to correspond across multiple cameras with limited overlap in fields of view or due to occlusions. On the other hand, advances in deep learning have resulted in strong detectors that work across different viewpoints but are still not precise enough for triangulation-based reconstruction. In this work, we develop a framework to fuse both the single-view feature tracks and multi-view detected part locations to significantly improve the detection, localization and reconstruction of moving vehicles, even in the presence of strong occlusions. We demonstrate our framework at a busy traffic intersection by reconstructing over 40 vehicles passing within a 3-minute window. We evaluate the different components within our framework and compare to alternate approaches such as reconstruction using tracking-by-detection.

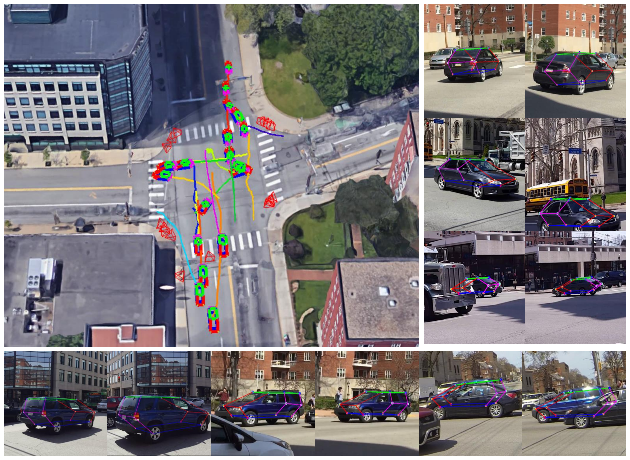

Figure 1: Reconstruction of vehicles crossing a busy intersection, making turns, going straight and changing lanes. A subset of vehicle skeletons (3D detector locations) and their 3D trajectories are augmented within the Google Earth view of the intersection. The reconstructions are reprojected into multiple views of two cars (a sedan and an SUV) demonstrating good performance under partial occlusions.

1. Introduction

Multiple video cameras are becoming increasingly common at urban traffic intersections. This provides us a strong opportunity to reconstruct moving vehicles crossing those intersections. The shapes (even sparse) and motions of the vehicles can be invaluable to traffic analysis, including vehicle type, speed, density, trajectory and frequency of events such as near-accidents. Infrastructure-to-Vehicle (I2V) communication systems can provide such analysis to other (semi-)autonomous vehicles approaching the intersection. That said, reconstructing moving vehicles in a busy intersection is hard because of severe occlusions. Furthermore, the cameras are often unsynchronized, provide wide-baseline views with little overlap in fields of view and need to be calibrated each frame as they are often not rigidly attached and sway because of wind or vibrations.

There has been a rich history of detection [12, 13, 34, 17], tracking [48, 9, 44, 41] and reconstruction [50, 19, 15, 8, 3] of vehicles. Their performances are progressively improving thanks to recent advances in deep learning. In particular, detection of parts of vehicles (front right wheel, back left wheel, right and left front doors, etc.) across multiple views is becoming increasingly reliable [32, 24, 42]. However, the detected part locations are still not precise enough to directly apply triangulation-based 3D reconstruction methods, and are incomplete in the presence of occlusions. For the same reason, tracking via per-frame detection is not stable enough to be useful for structure-from-motion approaches. We will refer to the detected part locations as ”structured points”.

On the other hand, there has also been significant work on tracking feature points [38, 1] in structure-from-motion approaches applied to a video from a single moving camera [20, 31, 11, 30]. But corresponding these features across wide-baseline views is near impossible given that each camera sees only parts of a vehicle (front, one side, or back) at

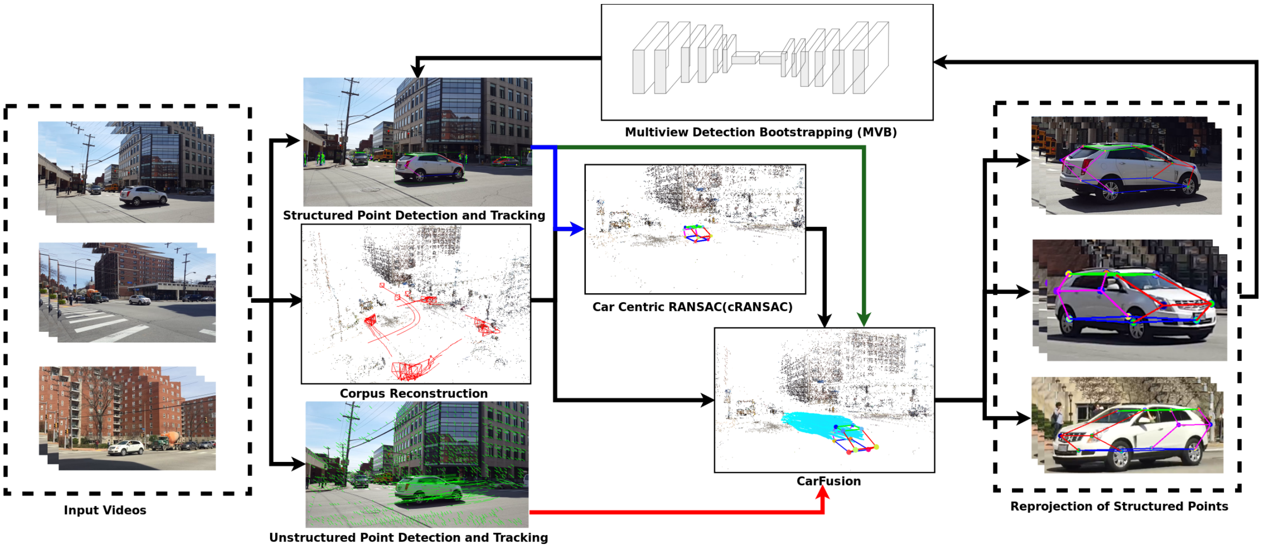
Multiview Detection Bootstrapping (MVB)
Structured Point Detection and Tracking
Car Centric RANSAC(cRANSAC)
Corpus Reconstruction
CarFusion
Input Videos
Unstructured Point Detection and Tracking
Reprojection of Structured Points

Figure 2: CarFusion: Our overall pipeline for dynamic 3D reconstruction of multiple cars from uncalibrated and unsynchronized video cameras. We fuse the structured points (detected vehicle parts) and the tracks of the unstructured feature points to obtain precise reconstruction of the moving vehicle. The reconstructions are reprojected into all the views and are used to bootstrap and improve the detectors.

any given time instant. These feature points do not often have a semantic meaning (like the structured parts) and we will call them "unstructured points".

In this paper, we present a comprehensive framework that fuses (a) incomplete and imprecise structured points (part detections) across multiple views with (b) precise but sparse single-view tracks of unstructured points, to reconstruct moving vehicles even in severe occluded scenarios. We call this framework as "CarFusion" and it consists of three main stages: (1) a novel object-centric (as opposed to feature-centric) RANSAC approach to provide a good initialization of the 3D geometry of the structured points of the vehicle (Sec. 3.1), (2) a novel approach that incorporates rigidity constraints between structured and unstructured points (Sec. 3.2), and (3) closing-the-loop by reprojecting the reconstructed structured points to all views to retrain the part detectors (Sec. 3.3). We implement a full end-to-end system that also includes a pre-processing stage (Sec. 4.1) to self-calibrate and synchronize the cameras by adapting recent prior works [40]. A detailed overview of our system is illustrated in Fig. 2.

We demonstrate reconstruction of vehicles at a busy intersection shown in Fig. 1. About 40 vehicles were detected, tracked and reconstructed within a 3-minute duration captured from 21 handheld cameras that are uncalibrated and unsynchronized and were panning to cover wider fields of view. A subset of vehicle trajectories and vehicle skeletons (structured points) are augmented within the Google Earth image of the intersection. They include cars of different types (sedans, SUVs, hatch-backs, jeeps, etc.) making left and right turns, going straight-ahead as well as changing lanes. Several views of two specific cars in various occluded scenarios are shown with the reprojections of the structured points.

We evaluate the performance of each stage of our framework. We also compare our approach to alternate methods that rely only on tracking-by-detection or feature based structure-from-motion. By treating them in a unified framework, we are able to show significant improvements in vehicle detection rates, vehicle trajectory lengths (or tracks) and reconstruction accuracies. Our approaches are designed to handle partial occlusions but fail when a vehicle is mostly occluded at all times. The estimated 3D vehicle tracks are accurate but slightly wobbly and will benefit from additional domain specific priors.

2. Related Work

The literature on 3D reconstruction of vehicles can be largely classified into two categories: coarse, object centric reconstruction using a single image or monocular video and dense reconstruction using multiview stereo. Unlike works that employ different sensor modalities [7, 23], our work is purely based on RGB cameras and thus, we only review methods using RGB sensors.

Single-View Reconstruction in the Wild: Reconstructing 3D information from a single view has been the subject of study for multiple decades. The earlier approaches assumed an isolated object for analysis similar to a projection of a CAD model on a plain background [35, 5, 4].

With the onset of better recognition algorithms [21, 43, 22], recent works utilize state-of-the-art object detectors [13] and instance segmentation [17, 47] algorithms to isolate an object, and follow various recipes to extract 3D information [2, 39, 29]. Multi-stage pipelines involve detecting and segmenting objects in the scene, estimating 3D poses, fitting shape models to the segment masks, enabling coarse to fine improvement [19, 28, 33]. Notably, Xiang et al. [45] estimated 3D voxels of the object directly from detection and segmentation results instead of estimating viewpoints and keypoints. Approaches that regress depth from monocular video have also been explored[14, 49]. In general, these approaches produce coarse and category specific reconstruction (e.g., car, chair). The reconstruction may potentially be geometrically inconsistent if re-projected to multiple views.

Active-Shape Model Reconstruction: Many works have been motivated by using active shape models [10] for vehicle reconstruction [50, 51]. These algorithms exploit strong part detectors learned from CNNs [32, 42, 24, 6, 25] and deform the shape model to fit the observations. Recent works have also combined SLAM with active shape priors for reconstruction of objects [8]. In general, these approaches produce more detailed 3D shape than those with category specific reconstruction. And despite mainly applied to monocular settings, the shape model is flexible enough for extension to multi-view system.

Multi-view stereo reconstruction: Multiview stereo systems are widely used in the context of vehicle reconstruction for both dense shape and velocity estimation [15, 27, 26, 3]. These approaches exploit cues from 2D bounding box detection, image instance segmentation or object category shape to regularize the stereo disparity for large displacement and textureless/glossy regions. Our work also employs multiple cameras but reconstructs both the car skeleton and sparse trajectories of the car body using 3D priors on symmetry, link length, and rigidity constraints.

Our multiview detector bootstrapping is similar in spirit to Simon et al.[37] for hand keypoint detection. However, their work is conducted in a laboratory studio equipped with massive number of cameras and the method can produce a good hand skeleton using multiview triangulation alone. Our work is "in the wild" where stable vehicle reconstruction is hard even with ground truth correspondences.

3. Multiview Reconstruction of Moving Cars

Consider C video cameras observing M rigidly moving cars over F video frames. At any time instance f, the car m has a fixed number of structured points $S_m(f)$ and an arbitrary number of unstructured points $U_m(f)$. The structured points are semantically meaningful 3D locations of parts on the car. They can be reliably but imprecisely detected and can be matched to different images at any time instance. The unstructured points are 3D locations of the local features (say, Harris corners) in the observed image. They can precisely be detected and reliably matched only within the same video. Their 2D locations are $s^c_m(f)$ and $u^c_m(f)$, respectively. The motion of an individual car is characterized by a rigid transformation $[R_m(f), T_m(f)]$ at frame f. Denote $x^c(f) = \pi^c(X(f))$ as the image projection of an arbitrary 3D point X to camera c at time f by the camera projection matrix $\pi_c(f)$. The visibility of X in camera c is given by $V^c(X(f))$. We assume all the cameras are calibrated and temporally synchronized as described in Sec. 4.1. The 3D locations of the unstructured points can be computed using SfM algorithms [36]. Our goal is to precisely estimate and track the 3D configurations of the structured points.

3.1. cRANSAC: Car-Centric RANSAC

To reconstruct the vehicle from multiple views, we must find correspondences across views first. We propose a car-centric RANSAC procedure for finding such correspondences. Compared to common point-based RANSAC [16], we consider the entire car as a hypothesis, which allows explicit physical constraints on the car link length and its left-right symmetry to be enforced. Due to the uncertainty in detecting the 2D location of the structured points from different views, these additional constraints are needed for reliable multiview correspondence estimation.

Concretely, consider a set of 2D car proposals $\boldsymbol{h}(f) = \{h^1(f), ..., h^c(f)\}$ available from all the cameras at frame f. Each proposal consists of a set of structured points s^c_m. We want to find a set $\boldsymbol{g_m} = \{g^1_m, ..g^c_m\}$, where $g^i_m \in h^i$, for every car m visible in the cameras. At every RANSAC iteration, we sample proposals within a triplet of cameras with sufficiently large baselines and solve DLT-based triangulation [16] to obtain $S_m(f)$. These points are back-projected to all cameras to find a better hypothesis g_m. We optimize a car-centric nonlinear cost function E_C and prune proposals with large error within g_m. This procedure is applied for fixed number of iterations. The hypothesis with the largest number of elements is taken as the inlier proposal for that car. These proposals are removed from the proposal pool h and the process is restarted until no good hypothesis is left. The car-centric cost function is defined as:

$$E_C = \alpha_I E_I + \alpha_S E_S + \alpha_L E_L \tag{1}$$

where, $\{E_I, E_S, E_L\}$ are the image evidence cost, car link length symmetry cost, and car link length consistency cost, respectively, and $\{\alpha_I, \alpha_S, \alpha_L\}$ are the weights balancing the contributions of each cost. The cost functions are described below.

Image evidence cost: This cost function penalizes the deviation between the 3D projection of a point and its detected 2D location:

$$E_I(f) = \sum_{c=1}^{C}\sum_{p=1}^{S_m} V_p^c(f)\rho\left(\frac{\pi^c(S_p(f)) - s_p^c(f)}{\sigma_I}\right),$$

where, ρ is the Tukey Biweight estimator and σ_I is the deviation in 2D localization of the structured point x_p^c.

Link length consistency cost: This cost incorporates prior information about the expected length of two structured points and penalizes the deviation of the estimated length with respect to its mean:

$$E_L(f) = \sum_{\{p,q\}\in\mathbf{L}} \rho\left(\frac{L_{p,q}(f) - \overline{L}_{p,q}}{\sigma_L}\right)^2,$$

where, $L_{\{p,q\}}$ is the Euclidean distance between two structured points $\{p, q\}$, in the connectivity graph $\mathbf{L}$, and its expected length $\overline{L}_{\{p,q\}}$, defined based on the vehicle type, e.g. sedan or truck, and σ_L is the expected variation in length.

Left-right symmetry cost: We penalize large differences between the left and right link length of the car. This constraint is useful in fusing detectors visible from the opposite side in other views. This cost function is given as:

$$E_S(f) = \sum_{\{l,r\}\in S} \left(\frac{L_l(f) - L_r(f)}{\sigma_s}\right)^2,$$

where, S is the set of corresponding left and right links, and σ_S is the expected variation in the left and right link lengths.

We rescale the SfM reconstruction into metric units and set $\{\sigma_I, \sigma_L, \sigma_S\}$ to $\{10, 1.5, 0.1\}$, $\{\alpha_S, \alpha_S, \alpha_L)\}$ to $\{1, 1, 0.5\}$, respectively for our experiments.

3.2. Fusion of Structured and Unstructured Points

By exploiting the physical constraints on link length and left-right symmetry, we can estimate plausible 3D configurations of S_m from multiple wide baseline cameras at any time instances. Yet, these estimations remain spatially and temporally unstable due to large uncertainty in detected locations of structured points. On the other hand, the unstructured points can be detected and tracked precisely for every camera. However, it is difficult to reliably establish correspondence between unstructured points across cameras due to large viewpoint changes.

Our fusion cost combines the complementary strengths of the structured and unstructured points using rigidity constraints. It enables precise and spatio-temporally stable estimation of the 3D configuration of the structured points. This cost function is formulated as:

$$\begin{aligned} e(f) \quad &= \left(\frac{\|R_m(f)S_i^c(f_s)+T_m(f)-U_j^c(f))\|_2-\|(S_i^c(f_s)-U_j^c(f_s))\|_2}{\sigma_R}\right)^2, \\ E_R \quad &= \textstyle\sum_c^C \sum_f^F \sum_j^{U_m^c} \sum_i^{S_m^c} e(f), \end{aligned}$$

Algorithm 1: CarFusion

```
Input: ${s_m^c(f), u_m^c(f)}$, $\pi^c(f)$, $\boldsymbol{h}(f)$
Output: ${S_m(f), U_m(f)}$,${R_m(f), T_m(f)}$
repeat
    while No more cars available do
        while Inliers < Min Inliers do
            repeat
                $g_m \leftarrow$ Sample $h$ from three cameras;
                $S_m \leftarrow$ DLT($g_m$);
                $g_m \leftarrow$ Project $S_m$ to all cameras;
                $g_m \leftarrow$ Optimize Eq. 1 and prune $g_m$;
                if $g_m > g_{mbest}$ then
                    $g_{mbest}$ = $g_m$;
                end
                iter++;
            until iter < Max Iteration;
        end Sec 3.1
        Remove $g_m$ from $h$;
        Reconstruction $U_m(f)$ objects (Sec.4.1);
        Optimize Eq.2 for $S_m(f)$, ${R_m(f), T_m(f)}$;
    end
    Project $S_m$ and retrain the detector (Sec.3.3);
until iter < Max Iteration;
```

where, σ_R, set to 0.1, is the expected deviation from rigid deformation of the car 3D configurations over time, and f_s is the frame where the car is first reconstructed (with sufficient inliers) using our RANSAC algorithm. This formulation links structured and unstructured points between all the visible cameras seamlessly over space and time. No spatial constraints are needed for unstructured points. No temporal constraints are needed for structured points.

Since the car motion is a rigid transformation, we explicitly enforce this constraint into the image evidence cost and integrate it over all time instances:

$$\begin{aligned} e(f) \quad &= \rho\left(\frac{\pi^c(R_m(f)(f)S_m(0)+T_m(f))-s_p^c(f)}{\sigma_I}\right) \\ E_{I2} \quad &= \textstyle\sum_{c=1}^C \sum_f^F \sum_{p=1}^{S_m} V_p^c(f)e(f), \end{aligned}$$

We then optimize the following total cost for precise 3D reconstruction of each car:

$$E = \min_{\mathbf{S}_m(t_0),\overline{\mathbf{L}}_m,\{R_m(f),T_m(f)\}} E_{I2} + E_S + E_L + E_R, \quad (2)$$

where, $\overline{\mathbf{L}}_m$ is set of mean link lengths and is initialized using mean of the 3D configurations S_m estimated in Sec. 3.1.

For efficiency, we start the reconstruction of each vehicle progressively, starting from the first time when our RANSAC detects the 3D object, and optimize Equation 2 for its structured point trajectories. The reconstructed cars are removed from the hypotheses pool. We iterate this process until no more cars can be reconstructed. Please refer to Algorithm 1 for the entire process.

3.3. Multiview Detection Bootstrapping

Precise and temporally stable 3D reconstruction of the car from multiple views can bootstrap the 2D detection of the structured points (loop-back shown in Fig. 2). In turn, better 2D localization of the structured points enable more precise 3D estimation of the car. Given the 3D locations of structured points and their visibilities, we project the 3D points onto all the views. We use the reprojected points as automatically computed labels for fine-tuning the car detector. We recompute the reconstruction using the improved detectors for better fitting of the structured points and further minimization of the reprojection error. The emphasis is to improve detections using reconstruction and vice-versa from cameras captured in the wild.

4. Experimental Evaluation

We evaluate our framework on a traffic scene captured with six Samsung Galaxy 6, ten iPhone 6, and six Gopro Hero 3 cameras at 60fps in a busy intersection for 3 minutes. These videos were captured by 13 people, some of whom carried two cameras. The sequence is challenging as there are no constraints on the camera motion or the vehicle motion in the scene. We call this the "Intersection Dataset".

We evaluate our reconstruction pipeline at its progressive stages: car-centric RANSAC (cRANSAC), temporal integration using only the structured points (TcRANSAC), and the fusion of both structured points and unstructured track (CarFusion). The TcRANSAC is the result of optimizing the cost function 2 but without the fusion term E_R. This method can be considered as reconstruction using tracking-by-detection. We also compare the evolution in accuracy of the 2D structured point detector before and after the multiview bootstrapping. We use the Stacked Hourglass architecture [32] referred to as "pretrained", to detect the structured points. The same architecture is used for finetuning with the point labels obtained using our multiview reconstruction. The finetuned detector is referred to as MVB (multiview bootstrapping).

4.1. Preprocessing

Structured points detection and tracking: We used the FCIS model [47] to obtain the car proposal hypotheses. For each hypothesis, we detect its structured points using the Stacked Hourglass model [32] trained on the KITTI dataset [24, 18]. We generate tracklet of each proposal by examining the overlapping area of the bounding boxes in consecutive frames. We split the tracklet if there are other bounding boxes with 70% overlapping area in one frame.

Camera calibration and 3D background estimation: We estimate the camera intrinsics and extrinsics at keyframes and reconstruct the stationary background points using ColMap [36]. The camera poses are propagated from the keyframes to all other frames using the affine Lucas-Kanade tracking and PnP pose estimation. The time offsets between cameras are estimated using the approach in [40].

3D reconstructing of the unstructured points: For every car proposal, we detect the Harris features and track them using the affine Lucas-Kanade algorithm. We initialize the detection every 30 frames and the track them for 120 frames in both backward and forward directions. We estimate their 3D locations using single view SFM.

4.2. Ablation Analysis

Figure 3 compares detection of the structured points before and after multiview bootstrapping with respect to the ground truth labels for two cars observed in three different views. We visualize only detections with more than 50% confidence. Our multiview bootstrapping shows clear improvements over the baseline method as more confident points are accurately detected. Using CarFusion, the reprojected points accurately localize the structured points and provide plausible prediction for occluded locations, as showed for twelve snapshots of another car in Figure 4. We attribute this property to the use of symmetry, link length, and rigidity constraints in the reconstruction stage. Although some of the structured points are not visible from any of the views, for example the left front wheel of the car in Figure 4, we are still able to accurately reconstruct the point in 3D due to our left-right symmetry and link length constraints. Without these constraints the reconstruction of the structured points, even fully visible from multiple views, often explodes due to erroneous detection hypothesis.

We use the Percentage of Correct Keypoints (PCK) metric [46] to evaluate the accuracy of 2D structured point detection. Under this metric, a 2D prediction is deemed correct when it lies within specified radius $\alpha * B$ of the ground truth label, where B is the larger dimension of the car bounding box. The analysis is conducted using manual annotations of 100 images of different cars from different viewing angles in the Intersection dataset. As showed in Table 1, our finetuned detector improves the accuracy of the baseline method by 5.8% with $\alpha = 0.1$ and 1.9% with $\alpha = 0.2$ just by finetuning the detector from the 2D re-projection of the reconstructed structured points. This result clearly demonstrates the benefit of CarFusion for accurate 3D structured points reconstruction and multiview bootstrapping for more accurate structured point detection.

Figure 5 shows a comparison between the quality of the reconstructed trajectories of the structured points using cRANSAC and the complete CarFusion pipeline. The trajectories are smoother by incorporating the Fusion of points

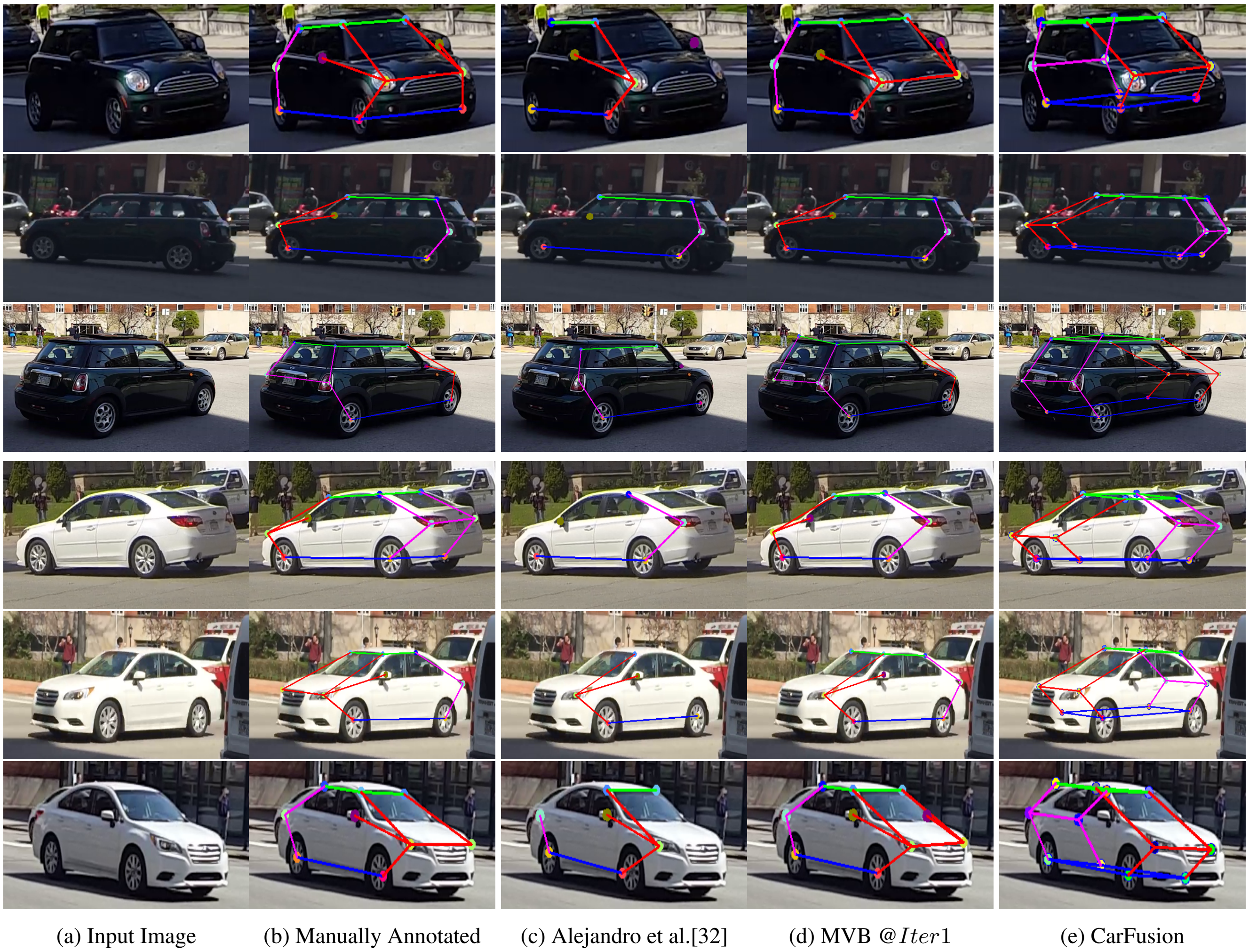

Figure 3: Qualitative analysis of the structured point detector before and after multiview bootstrapping (MVB), shown for two cars in three different views. Initial detectors were trained using Alejandro et al.[32]. The CarFusion approach was used to reconstruct the cars. Then the resulting 3D structured points were re-projected to all the views and used to retrain/bootstrap the detectors. The MVB approach shows clear visual improvement over the baseline, even in the presence of occlusions.

		cRANSAC			T-cRANSAC			CarFusion		
	Length of Traj	No of Traj	RMSE		No of Traj	RMSE		No of Traj	RMSE	
			Pretrained	MVB		Pretrained	MVB		Pretrained	MVB
Straight	234	14	12.24	8.52	14	17.8	7.1	112	16.8	2.5
Turning	172	14	8.94	6.95	14	12.5	5.83	101	15.5	3.1
Multi	202	42	7.45	5.3	42	14.3	4.47	414	17.4	2.2

Table 1: Reprojection error of the reconstructed tracks at different stages of the pipeline. The rows refer to cases where one car is moving straight, turning left or right and multiple cars in the intersection. The number of trajectories using cRANSAC and T-cRANSAC is fixed to the number of parts, while with point fusion we have a combination of structured and unstructured tracks. The full pipeline (CarFusion + MVB) performs best, reducing the error of cRANSAC and T-cRANSAC by 4 and 2 times, respectively.

compared to SFM on structured points. Assuming the detector is accurate, we quantify the accuracy of re-projected 2D point with respect to the detections. A 2D re-projection of the 3D structured point is correct when it lies within specified radius $\beta * B$ of the corresponding detected visible point in the image. We set $\beta = 0.1$. We report the percentage of

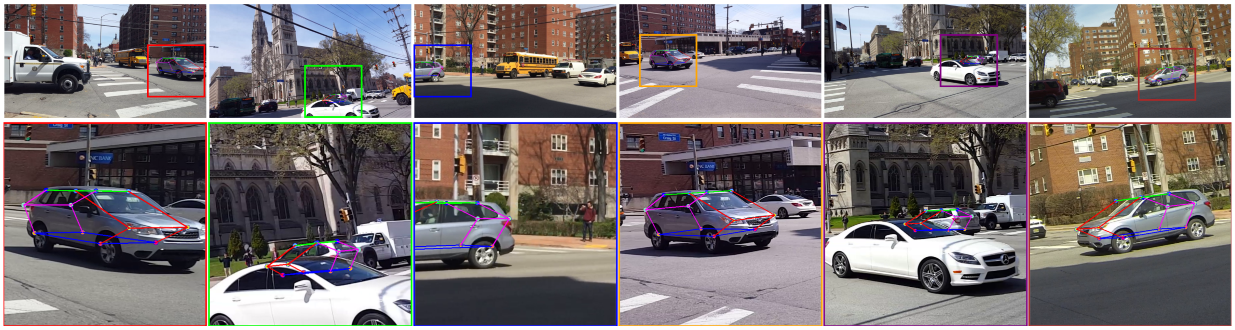

Figure 4: The 2D re-projections of the steel gray colored car in many occluded configurations. The CarFusion method can accurately reconstruct the 3D configuration of car despite strong occlusion. The top row shows the full field of views and the bottom row shows zoomed in insets.

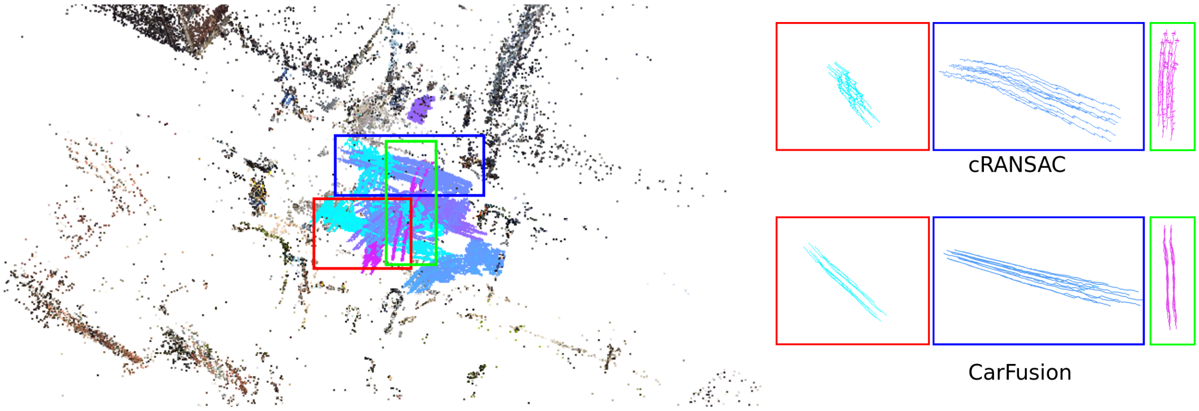

Figure 5: Visualization of the reconstructed trajectories of multiple cars using cRANSAC. The insets on the right show detailed comparisons of the trajectories stability between cRANSAC and CarFusion. CarFusion produces clearly more stable trajectories. Visually, they correspond well to the motion of a moving car.

	$Iter1$	$Iter2$
cRANSAC	31.2	36.1
T-cRANSAC	71.6	73.7
CarFusion	76.3	79.4

Table 2: Percentage of inlier detections at every stage of the pipeline after iterations 1 and 2 of multi-view bootstrapping. The full CarFusion pipeline shows significant performance improvement over the first stage, cRANSAC, in the pipeline, while showing modest improvements over the T-cRANSAC stage of the pipeline.

	$\alpha = 0.1$	$\alpha = 0.2$
Pretrained	85.6	94.6
MVB	91.4	96.5

Table 3: Comparing the structured point detectors using the Percentage of Correct Keypoint (PCK) metric. Our multi-view bootstrapping (MVB) shows clear improvement over the state-of-art baseline detector [32].

inlier points at different stages before and after multiview bootstrapping in Table 2. Regardless of the finetuning step, cRANSAC performs poorly, as confirmed visually in Figure 5. This is due to erroneous detection that leads to frequent failure of cRANSAC . We observe a significant boost in the accuracy by temporal smoothing of the cRANSAC results over time. Our full CarFusion algorithm with multiview bootstrapping performs best, with 79.4% inliers detected.

We provide fine-grain analysis of the methods in Table 1, using three sub-sequences: car moving straight in single lane, car turning, and a three cars scene. The first sub-sequence is observed for 234 frames, the second sub-sequence is observed for 172 frames, and last sub-sequence is observed for 202 frames. We report the root mean square error (RMSE) of the difference between the re-projected points and the detected points. We observe that the RMSE of the cRANSAC algorithm is large because of many detections with high variation in part localization. This error

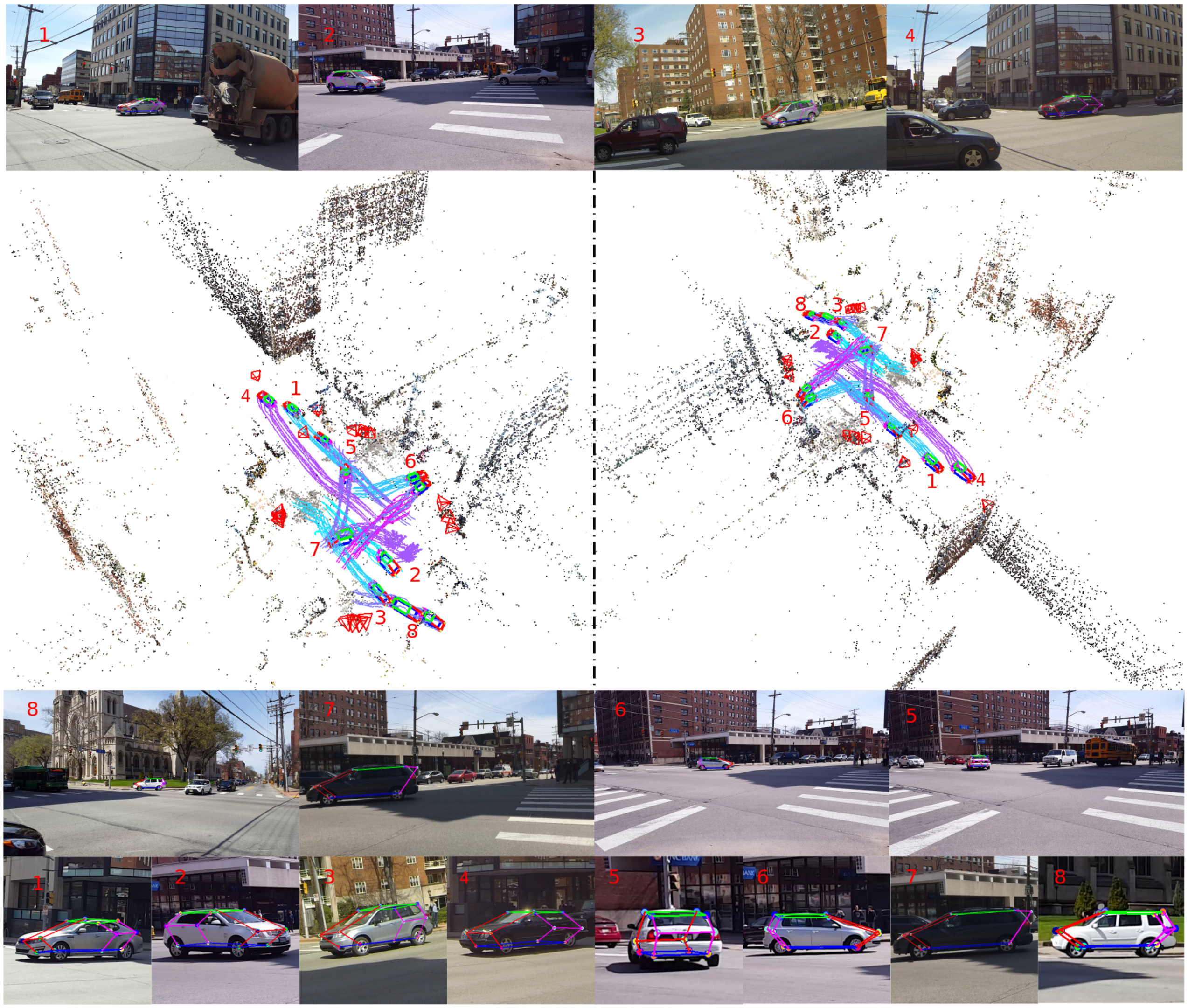

Figure 6: Visualization of the 8/43 reconstructed cars using CarFusion. We show the 2D re-projection of the reconstructions onto sample frame containing those cars. All the re-projected points fit the cars well.

is reduced by finetuning (MVB) and can be attributed to the fact that better detection produces a more consistent 3D model. Interestingly, without multiview bootstrapping the error increases for T-cRANSAC. This could be because the detections are not temporally consistent. As expected, this error drops after detector finetuning. Using the unstructured tracks reduces the overall reprojection error of the 3D tracks by at least 5 times (12.24 to 2.5 or 7.45 to 2.2). However, the finetuned network gives modest improvement over the reconstruction of the structured tracks. This could be due to the limitation of the CNN architecture where the training image is down sampled substantially. The length of the trajectory of the car is the max length of the bounding box tracks over all the inlier videos.

In Figure. 6 we illustrate the complete 3D reconstruction of trajectories of structured points on moving cars using CarFusion and the 2D projection to inlier views for several cars. As can be seen from the results we are able to accurately reconstruct the trajectories of the cars over time captured from unsynchronized videos.

5. Summary

We have presented a method to fuse imprecise and incomplete part detections of vehicles across multiple views and the more precise feature tracks within a single view to obtain better detection, localization, tracking and reconstruction of vehicles. This approach works well even in the presence of strong occlusions. We have quantified improvements due to the different stages of the end-to-end pipeline that only uses videos from multiple uncalibrated and unsynchronized cameras as input. We believe this approach can be useful for stronger traffic analytics at urban intersections. In the future, we will extend our methods to identify and fit vehicle CAD models to the videos for better visualization.

6. Acknowledgments

This paper was supported in parts by a Heinz Foundation grant, an NSF award CNS-1446601, an ONR grant N00014-15-1-2358, a CMU University Transportation Center T-SET grant and a PhD fellowship from Qualcomm.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 2004.

[2] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5965–5974, 2016.

[3] A. Behl, O. H. Jafari, S. K. Mustikovela, H. A. Alhaija, C. Rother, and A. Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *International Conference on Computer Vision (ICCV)*, 2017.

[4] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 1987.

[5] T. O. Binford. Visual perception by computer. In *IEEE Conf. on Systems and Control*, 1971.

[6] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. *arXiv preprint arXiv:1703.07570*, 2017.

[7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. 2017.

[8] 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 *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 5758–5765. IEEE, 2016.

[9] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *ICCV*, 2015.

[10] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *CVIU*, 1995.

[11] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *TPAMI*, 2007.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010.

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

[14] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[15] F. Guney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *CVPR*, 2015.

[16] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[18] S. S. J. Krishna Murthy and K. M. Krishna. Shape priors for real-time monocular object localization in dynamic environments. In *IROS*, 2017.

[19] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015.

[20] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR*.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*.

[22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[23] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3d lidar using fully convolutional network. In *Robotics: Science and Systems*, 2016.

[24] 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.

[25] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis. Jointly optimizing 3d model fitting and fine-grained classification. In *European Conference on Computer Vision*, pages 466–480. Springer, 2014.

[26] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[27] M. Menze, C. Heipke, and A. Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis*, 2015.

[28] R. Mottaghi, Y. Xiang, and S. Savarese. A coarse-to-fine model for 3d pose estimation and sub-category recognition. In *CVPR*, 2015.

[29] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017.

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *T-RO*, 2015.

[31] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011.

[32] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[33] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *ICRA*, 2017.

[34] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. *arXiv preprint arXiv:1704.05776*, 2017.

[35] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1965.

[36] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[37] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. *arXiv preprint arXiv:1704.07809*, 2017.

[38] C. Tomasi and T. Kanade. Detection and tracking of point features. 1991.

[39] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *CVPR*, 2015.

[40] M. Vo, S. G. Narasimhan, and Y. Sheikh. Spatiotemporal bundle adjustment for dynamic 3d reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[41] S. Wang and C. C. Fowlkes. Learning optimal parameters for multi-target tracking with contextual interactions. *International Journal of Computer Vision*, 122(3):484–501, 2017.

[42] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[43] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[44] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4705–4713, 2015.

[45] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. Data-driven 3d voxel patterns for object category recognition. In *CVPR*, 2015.

[46] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1385–1392, Washington, DC, USA, 2011. IEEE Computer Society.

[47] J. D. X. J. Yi Li, Haozhi Qi and Y. Weil. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017.

[48] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.

[49] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[50] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3d representations for object recognition and modeling. *TPAMI*, 2013.

[51] M. Z. Zia, M. Stark, and K. Schindler. Towards scene understanding with detailed 3d object representations. *IJCV*, 2015.