CARNEGIE MELLON UNIVERSITY

MASTER THESIS

Incorporating Tracking and Detection for Dynamic Reconstruction from Unsynchronized Camera

Author:
N Dinesh Reddy

Supervisor:
Prof. Srinivasa Narasimhan

A thesis submitted in fulfillment of the requirements
for the degree of Master in Robotics

in the

Robotic Institute
School of Computer Science

April 18, 2018

CARNEGIE MELLON UNIVERSITY

Abstract

Prof. Srinivasa Narasimhan
School of Computer Science

Master in Robotics

Incorporating Tracking and Detection for Dynamic Reconstruction from Unsynchronized Camera

by N Dinesh Reddy

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 Multi-view tracking of vehicles from multiple views. We also present a novel camera synchronization algorithm to improve the performance of the reconstruction over long trajectories. We demonstrate our framework at a busy traffic intersection by reconstructing over 62 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.

Acknowledgements

I would like to express my sincere gratitude to my advisors Prof. Srinivasa Narasimhan for his continuous support of my Masters study and research. I would like to thank my thesis committee for thier patience, motivation, guidance and knowledge. It has been invaluable in my work culminating into the writing of this thesis. I am also extremely grateful to Minh Vo, for mentoring me through the thick and thin of this journey, as well as for helping me and sharing his experience at all stages of my work.

I specifically thank Siddharth, Shashank, Nishitha, Deepthi, Ishan, Sudeep, Yashasvi, Kalli, Adithya, Gaurav, Lerrel for their support during the course of my stay at CMU. Finally, I would like to thank my family and all my friends who have stood with me and have served as a source of motivation and moral support.

Contents

List of Figures

List of Tables

Dedicated To my Family

Chapter 1

Introduction

FIGURE 1.1: 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.

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 [13, 16, 45, 20], tracking [66, 9, 61, 58] and reconstruction [69, 26, 18, 8, 3] of vehicles. Their performances are progressively improving thanks to recent advances in deep learning. In particular, detection of parts of vehicles (wheels, headlights, doors, etc.) across multiple views is becoming increasingly reliable [41, 31, 59]. However, the detected part locations are still not precise enough to directly apply

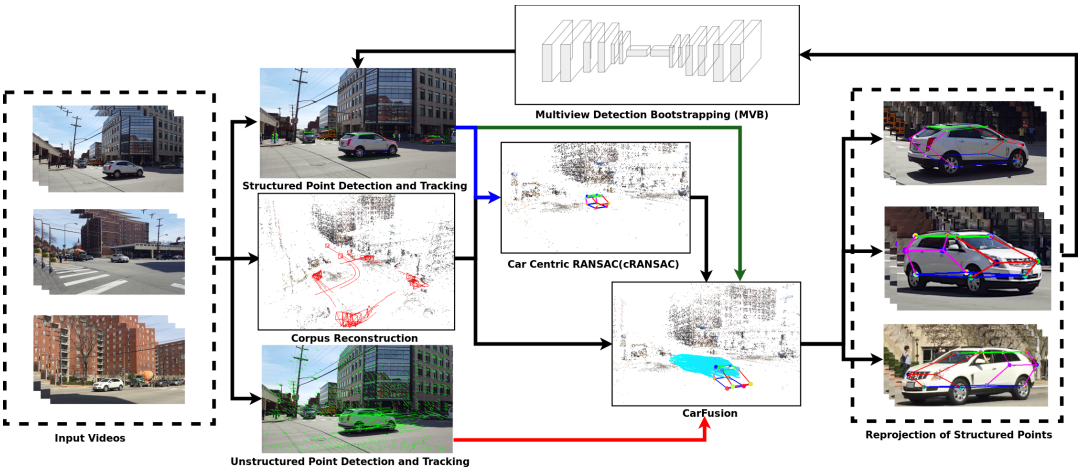


FIGURE 1.2: CarFusion: Our overall pipeline for dynamic 3D reconstruction of multiple cars from uncalibrated and unsynchronized video cameras. We fuse the structured points (detected vehicle parts) and the tracks of the unstructured feature points to obtain precise reconstruction of the moving vehicle. The reconstructions are reprojected into all the views and are used to bootstrap and improve the detectors.

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 [55, 1] in structure-from-motion approaches applied to a video from a single moving camera [27, 40, 11, 39]. 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 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 algorithm that fully exploits the complementary strength of the structured and unstructured points via rigidity constraints (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 to self-calibrate and synchronize the cameras by adapting recent prior works [57]. A detailed overview of our system is illustrated in Fig. 1.2.

We demonstrate reconstruction of vehicles at a busy intersection shown in Fig. 1.1. About 62 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 structured point trajectories 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.

Chapter 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, 30], our work is purely based on RGB cameras and thus, we only review methods using RGB sensors.

2.1 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 [46, 5, 4]. With the onset of better recognition algorithms [28, 60, 29], recent works utilize state-of-the-art object detectors [16] and instance segmentation [20, 65] algorithms to isolate an object, and follow various recipes to extract 3D information [2, 56, 38]. 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 [26, 37, 42]. Notably, Xiang et al. [62] 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[17, 67]. 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.

2.2 Active-Shape Model Reconstruction:

Many works have been motivated by using active shape models [10] for vehicle reconstruction [69, 68]. These algorithms exploit strong part detectors learned from CNNs [41, 59, 31, 6, 33] 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.

2.3 Multi-view stereo reconstruction:

Multiview stereo systems are widely used in the context of vehicle reconstruction for both dense shape and velocity estimation [18, 35, 34, 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. MVS leverages multiple views to overcome the problem of occlusion in generic reconstruction pipelines [70][63] [53]. So view selection play a major role in reconstruction. While most of the methods use heuristically[25] or probabilistically [50][51] select the views. Our method is more similar to view selection for depth map fusion, which integrates multiple depth maps into a unified and augumented scene representation. Most of these depth map fusion techniques select views either heuristically[14] [15] or by evaluating the visibility in 3D space [21]. we propose a learning based next-best view selection for reconstruction. This is indeed useful as you can achieve similar results to the state-of-the art with far fewer number of views. Our multiview detector bootstrapping is similar in spirit to Simon et al.[49] 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.

2.4 Object Recognition:

In the domain of computer vision and machine learning, multiple methods have been proposed for multi-view object recognition. Most of the recent papers in this direction use convolutional neural networks (CNN) for this task. 3D ShapeNets [60] proposes to use 3D features as input to a 3D CNN and showed state-of-the-art object recognition using 3D objects. They further showed next best view selection for 3D object recognition as an auxiliary task of their pipeline. Multi-view CNN [52][43] have generalized multiview recognition by learning from images that cover the full sphere of viewpoints over an object. Multiple methods have exploited the of multi-view recognition in 3D and 2D to improve the accuracy [44]. Some recent works also proposed a reinforcement learning based method to solve the problem of active vision by looking ahead [22] and have shown its applications to object recognition. Further a pairwise decomposition method has been proposed to learn the best pairs to recognize the object [23]. Although most of the methods have addressed the problem with different approaches, they introduce prior knowledge and bias in how to search for the next best view to recognize an object. We believe navigation with reinforcement learning as the best method to learn the next best view selection. There is work proposed in object detection with visual attention [36], which uses reinforcement learning over glimpses of a image to predict the label.

2.5 Multi-View Pose Estimation:

Most of the recent pose estimation algorithms use end-to-end learning based methods to lift pose of objects from 2D images[32][54][47]. In parallel to the Image based methods multiple multi-view methods for pose estimation have been proposed[24] [12]. But most of these methods use all the images to compute the pose of the objects. The computation cost for predicting pose is large for these methods. Our method uses a search based algorithm on the views based on the object view to find the best view to estimate the 3D pose.

Chapter 3

Multiview Reconstruction of Moving Objects

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 the 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 $s_m^c(f)$ and $u_m^c(f)$, 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. The 3D locations of the unstructured points can be computed using SfM algorithms [48]. 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 [19], 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_m^c. We want to find a set $\boldsymbol{g_m} = \{g_m^1, ..g_m^c\}$, where $g_m^i \in h^i$, 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 triangulate the hypothesis 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:

$$E_C = \alpha_I E_I + \alpha_S E_S + \alpha_L E_L \tag{3.1}$$

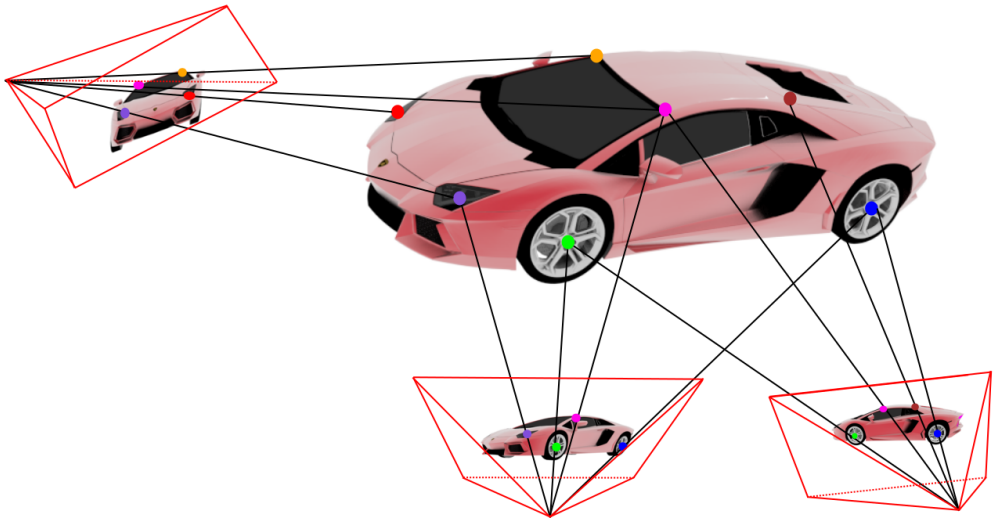


FIGURE 3.1: Illustration of cRANSAC, fitting of cars from multiple views using keypoints.

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.

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. 3.1 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 ;
    Optimize Eq.3.2 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*;

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:

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

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.The cost function promotes fixed distances between the structured and unstructured points (definition of rigid motion) during the course of motion. 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:

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

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

$$E = \min_{\mathbf{S}_m(t_0),\bar{\mathbf{L}}_m,\{R_m(f),T_m(f)\}} E_{I2} + E_S + E_L + E_R, \tag{3.2}$$

where, $\bar{\mathbf{L}}_m$ 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 3.2 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. 1.2). 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.

3.4 Camera Synchronization

To reconstruct the environment from multiple cameras, the most important aspect is the synchronization of the C cameras. The problem boils down to temporal reconstruction of moving objects from multiple cameras. We can use the structured and unstructured points simultaneously to compute the synchronization. We consider smooth motion of cameras and moving objects simultaneously to synchronize the cameras. When there is a motion performed on the scene, we can assume the following costs to be minimized

Rotation Averaging Cost: Since the rotation of the car is constant over time. we can assume the below motion to be minimized.

$$E_R = \sum_{m=1}^{M}\sum_{t=1}^{T} log\left(\frac{R_m^T(t)R_m(t+1)}{\sigma_r}\right)^2,$$

Constant Translation Cost: Since the motion of the cars is constant, we can penalize the overall kinetic energy of the car to be minimized.

$$E_T = \sum_{m=1}^{M} \sum_{t=1}^{T} \left(\frac{T_m(t) - T_m(t+1)}{\sigma_t} \right)^2,$$

We finally optimize the following terms, where δ is defined as the time synchronization of each of the camera.

$$E = \min_{\mathbf{X}_m(t_0), \{R_m, T_m\}, \{R_c, T_c\}, \delta} E_I + E_R + E_T, \tag{3.3}$$

Where E_I is the image evidence cost defined in the previous section and is computed for both the structured and unstructured points. While E_R is the constant rotation of the moving object defined over time. While E_T is defined as the constant motion over time.

Chapter 4

Experimental Analysis

4.1 Comparison with single video methods

Figure 4.1 shows a comparison between our method and a traditional single-video based SFM reconstruction of structured points on the intersection dataset. Single video based reconstruction fails as can been seen from the reprojection of structured points onto a different viewpoint (Where single video methods have 35 pixels RMS reprojection error versus ours of 2.5 pixels).

FIGURE 4.1: [Top-Left] Single-video based SFM(SVB-SFM) reconstruction of structured points. [Top-Right] The reprojection of the reconstruction onto a different view. [Bottom] Projection of reconstructed car using CarFusion to the above corresponding views.

We evaluate our method extensively on nearly 210,000 frames (3 minutes of 21 videos, each with 10000 frames), which is about 10 times the data (6 minute single video) evaluated by state-of-the-art SLAM pipelines (e.g., LSD-SLAM of Engel et al.). We show RMS reprojection errors in Tables 1 and 2 (in paper). Here we assume the structured points are accurately predicted by the network and evaluate the reconstruction. Tab. 3 compares the predicted structure points against ground truth labels. In lieu of actual 3D ground truth locations, RMS reprojection error in completely different views has been the most widely used evaluation metric in SFM and SLAM pipelines. We also show 3D reconstructions of many cars registered to a satellite photograph of the intersection.

	$\alpha = 0.1$	$\alpha = 0.2$
Pretrained	87.1	91.8
MVB	91.4	94.5

TABLE 4.1: Comparing the structured point detectors using the Percentage of Correct Keypoint (PCK) metric. Our multiview bootstrapping (MVB) shows clear improvement over the state-of-art baseline detector [41].

		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 4.2: 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.

4.2 Comparison with CAD/Active Shape Model Fitting

Figure 4.2 shows a comparison between a CAD dictionary fitting algorithm of ref [33] applied to a single image and our approach. CAD fitting approaches are sensitive to errors in 2D keypoint localization, especially in the presence of occlusions. In unconstrained settings as ours, CAD fitting orientation error is approximately 22 degrees (while our method shows only 1 degree error). Further, since current methods were trained on a small range of CAD models, they cannot generalize to arbitrary vehicles in the wild.

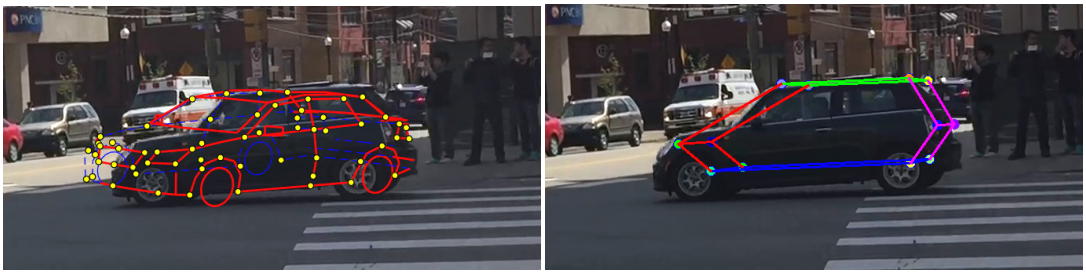


FIGURE 4.2: [Left] CAD based reconstruction [33] shows 22^o angular error with respect to the motion of the vehicle. [Right] In comparison, our result shows 1^o angular error. The CAD basis based methods fail due to inaccurate keypoint predictions, especially when only one side of the vehicle is visible.

4.3 Structured and Unstructured Points

The structured points are 14 car keypoints, obtained by training the Stacked hourglass CNN architecture [41] on the annotated dataset of Kitti dataset and applying this model to our videos. Multi-View Bootstrapping is the process of finetuning the Stacked hourglass CNN model on our videos using the reprojected locations of the 14 3D car keypoints to the all

FIGURE 4.3: 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.

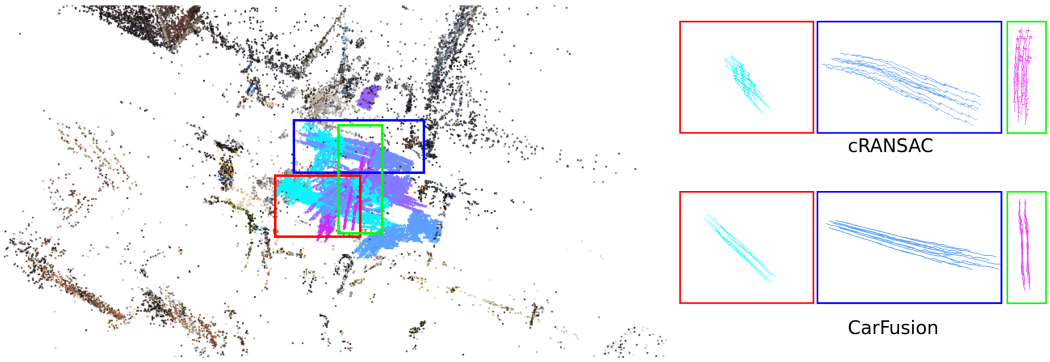


FIGURE 4.4: Visualization of the reconstructed trajectories of multiple cars using cRANSAC. The insets on the right show detailed comparisons of the trajectories stability between cRANSAC and CarFusion. CarFusion produces clearly more stable trajectories. Visually, they correspond well to the motion of a moving car.

cameras (views) as self-supervised labels. Tab. 3 in the paper shows the improvement in the CNN model when applied to our videos and the resulting improvement in 3D reconstruction of the structured points. The unstructured points are detected using Harris corner detection algorithm and tracked using a extended version of lucas-kanade tracker.

4.4 Ablation Analysis

Fig. 4.5 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 Fig. 4.3. 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 Fig. 4.3, 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.

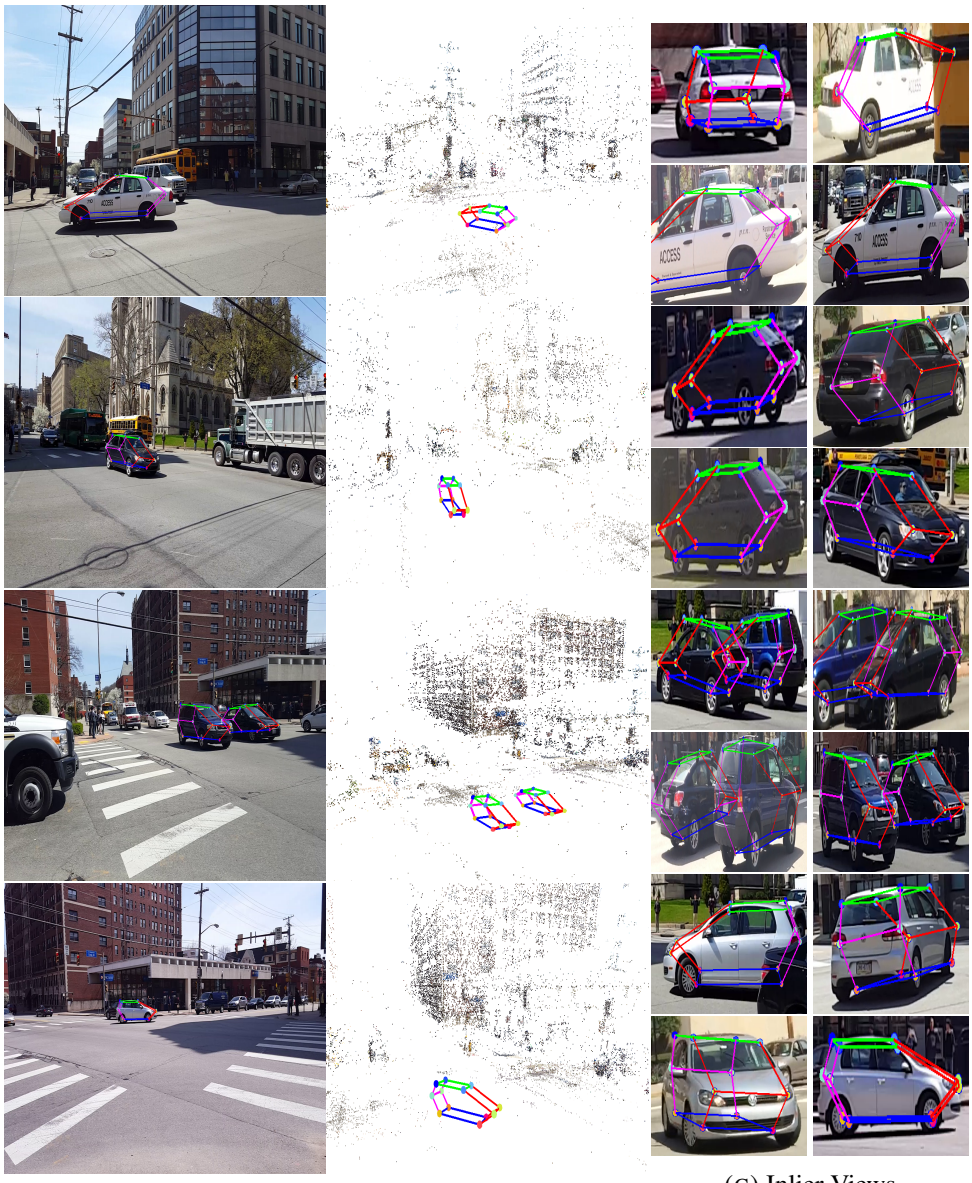
(A) Input Image (B) Manually Annotated (C) [41] (D) MVB @*Iter*1 (E) CarFusion

FIGURE 4.5: 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.[41]. 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.

We adopted the widely used PCK metric [64] 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. As showed in Table 1, our finetuned detector improves the accuracy of the baseline method by 4.3% with $\alpha = 0.1$ and 2.7% 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.

We analyze the improvement in the accuracy of reconstructed structured points with respect to the ground truth annotations according to the tracking length and the number of unstructured points in Fig. 4.8. As expected, the increase in visibility (track length) of structured points better stabilize the structured points which leads to higher quality reconstruction. We also find that the larger number of unstructured points improve the quality of the structured points due stronger rigidity constraints and the improvement is more evident for stricter threshold ($\alpha = 0.1$).

Fig. 4.4 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 compared to SFM on structured points. Assuming the detector is accurate, we quantify the accuracy of re-projected 2D point with



(A) Input View (B) Reconstructed (C) Inlier Views

FIGURE 4.6: We show the fitting of vehicles using our Multi-view RANSAC based object fitting

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 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 Fig. 4.4. 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.

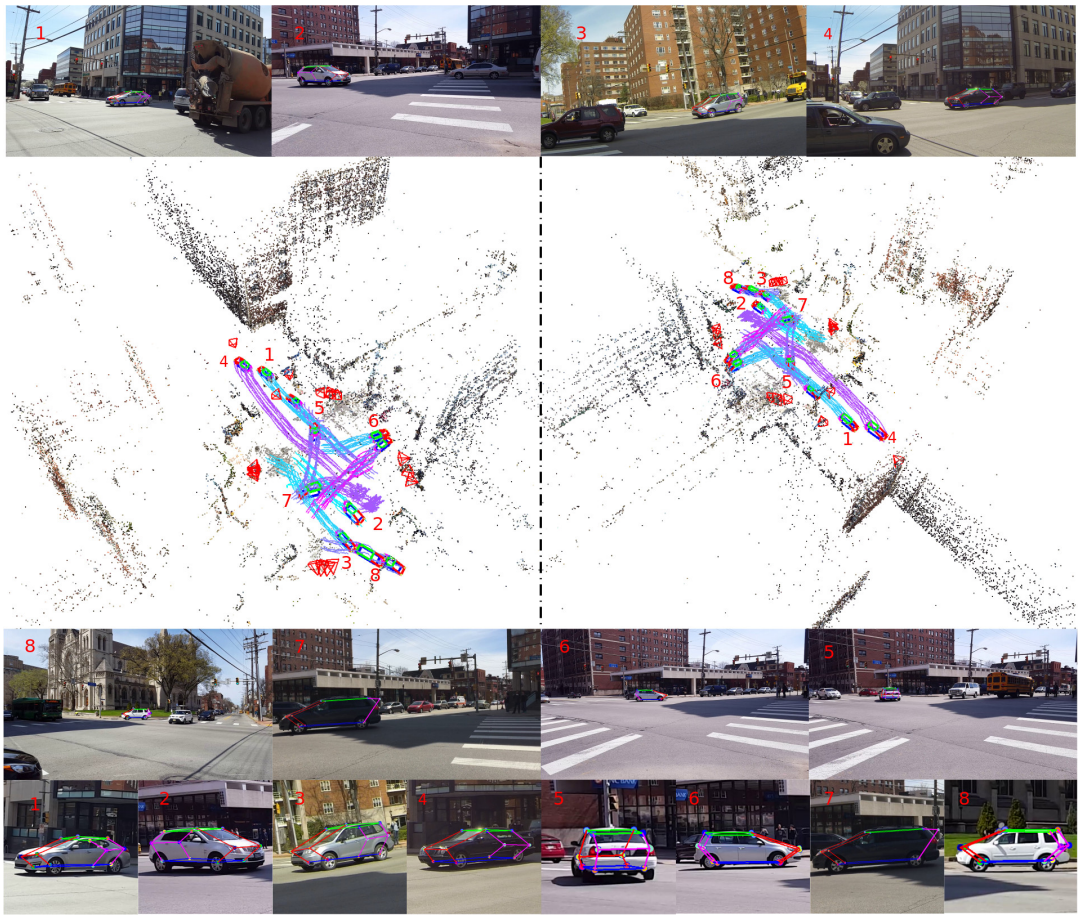


FIGURE 4.7: Visualization of the 8/43 reconstructed cars using CarFusion. We show the 2D re-projection of the reconstructions onto sample frame containing those cars. All the re-projected points fit the cars well.

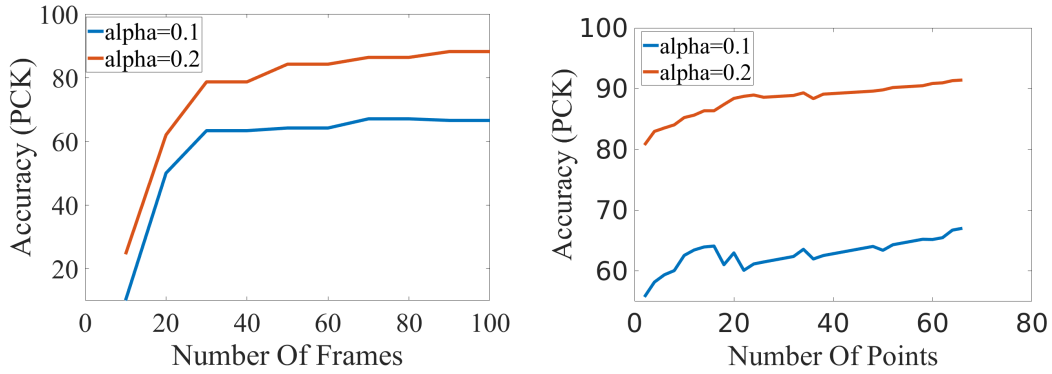


FIGURE 4.8: Analysis of accuracy with respect to increase in number of frames (left) and increase in number of unstructured points (right) used in the CarFusion algorithm.

4.5 Qualitative Analysis

We provide fine-grain analysis of the methods in Table 4.2, 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 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. 4.7 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.

The method applies to any rigidly moving objects. The formulation can be further generalized to piece-wise rigid or articulated objects. We do not correspond the unstructured points across multiple wide-baseline videos but only track them within a single video. Only structured points are corresponded across videos.

Chapter 5

Conclusion

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.

Bibliography

[1] Simon Baker and Iain Matthews. “Lucas-kanade 20 years on: A unifying framework”. In: *IJCV* (2004).

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

[3] Aseem Behl et al. “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] Irving Biederman. “Recognition-by-components: a theory of human image understanding.” In: *Psychological review* (1987).

[5] Thomas O Binford. “Visual perception by computer”. In: *IEEE Conf. on Systems and Control*. 1971.

[6] Florian Chabot et al. “Deep MANTA: A Coarse-to-fine Many-Task Network for joint 2D and 3D vehicle analysis from monocular image”. In: *arXiv preprint arXiv:1703.07570* (2017).

[7] Xiaozhi Chen et al. “Multi-view 3d object detection network for autonomous driving”. In: (2017).

[8] Falak Chhaya et al. “Monocular reconstruction of vehicles: Combining slam with shape priors”. In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE. 2016, pp. 5758–5765.

[9] Wongun Choi. “Near-online multi-target tracking with aggregated local flow descriptor”. In: *ICCV*. 2015.

[10] Timothy F Cootes et al. “Active shape models-their training and application”. In: *CVIU* (1995).

[11] Andrew J Davison et al. “MonoSLAM: Real-time single camera SLAM”. In: *TPAMI* (2007).

[12] Ahmed Elhayek et al. “Efficient ConvNet-based marker-less motion capture in general scenes with a low number of cameras”. In: *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE. 2015, pp. 3810–3818.

[13] Pedro F Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. In: *TPAMI* (2010).

[14] Yasutaka Furukawa et al. “Towards internet-scale multi-view stereo”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 1434–1441.

[15] David Gallup, Marc Pollefeys, and Jan-Michael Frahm. “3d reconstruction using an n-layer heightmap”. In: *Joint Pattern Recognition Symposium*. Springer. 2010, pp. 1–10.

[16] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[17] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. “Unsupervised Monocular Depth Estimation with Left-Right Consistency”. In: *CVPR*. 2017.

[18] Fatma Guney and Andreas Geiger. “Displets: Resolving stereo ambiguities using object knowledge”. In: *CVPR*. 2015.

[19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[20] Kaiming He et al. “Mask r-cnn”. In: *ICCV*. 2017.

[21] Michal Jancosek and Tomás Pajdla. “Multi-view reconstruction preserving weakly-supported surfaces”. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 3121–3128.

[22] D. Jayaraman and K. Grauman. “Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion”. In: *ECCV*. 2016.

[23] Edward Johns, Stefan Leutenegger, and Andrew J Davison. “Pairwise decomposition of image sequences for active multi-view recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3813–3822.

[24] Hanbyul Joo et al. “Panoptic Studio: A Massively Multiview System for Social Motion Capture”. In: *The IEEE International Conference on Computer Vision (ICCV)*. 2015.

[25] Sing Bing Kang, Richard Szeliski, and Jinxiang Chai. “Handling occlusions in dense multi-view stereo”. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2001, pp. I–I.

[26] Abhishek Kar et al. “Category-specific object reconstruction from a single image”. In: *CVPR*. 2015.

[27] Georg Klein and David Murray. “Parallel tracking and mapping for small AR workspaces”. In: *ISMAR*.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *NIPS*.

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444.

[30] Bo Li, Tianlei Zhang, and Tian Xia. “Vehicle detection from 3d lidar using fully convolutional network”. In: *Robotics: Science and Systems*. 2016.

[31] Chi Li et al. “Deep Supervision with Shape Concepts for Occlusion-Aware 3D Object Parsing”. In: *arXiv preprint arXiv:1612.02699* (2016).

[32] Sijin Li and Antoni B Chan. “3d human pose estimation from monocular images with deep convolutional neural network”. In: *Asian Conference on Computer Vision*. Springer. 2014, pp. 332–347.

[33] Yen-Liang Lin et al. “Jointly optimizing 3d model fitting and fine-grained classification”. In: *European Conference on Computer Vision*. Springer. 2014, pp. 466–480.

[34] Moritz Menze and Andreas Geiger. “Object Scene Flow for Autonomous Vehicles”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[35] Moritz Menze, Christian Heipke, and Andreas Geiger. “Joint 3D Estimation of Vehicles and Scene Flow”. In: *ISPRS Workshop on Image Sequence Analysis*. 2015.

[36] Volodymyr Mnih et al. “Recurrent Models of Visual Attention”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2204–2212. URL: http://papers.nips.cc/paper/5542-recurrent-models-of-visual-attention.pdf.

[37] Roozbeh Mottaghi, Yu Xiang, and Silvio Savarese. “A coarse-to-fine model for 3D pose estimation and sub-category recognition”. In: *CVPR*. 2015.

[38] Arsalan Mousavian et al. “3D Bounding Box Estimation Using Deep Learning and Geometry”. In: *CVPR*. 2017.

[39] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. “ORB-SLAM: a versatile and accurate monocular SLAM system”. In: *T-RO* (2015).

[40] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. “DTAM: Dense tracking and mapping in real-time”. In: *ICCV*. 2011.

[41] Alejandro Newell, Kaiyu Yang, and Jia Deng. “Stacked hourglass networks for human pose estimation”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 483–499.

[42] Georgios Pavlakos et al. “6-dof object pose from semantic keypoints”. In: *ICRA*. 2017.

[43] Charles R Qi et al. “Volumetric and multi-view cnns for object classification on 3d data”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5648–5656.

[44] Weichao Qiu and Alan Yuille. “Unrealcv: Connecting computer vision to unreal engine”. In: *Computer Vision–ECCV 2016 Workshops*. Springer. 2016, pp. 909–916.

[45] Jimmy Ren et al. “Accurate Single Stage Detector Using Recurrent Rolling Convolution”. In: *arXiv preprint arXiv:1704.05776* (2017).

[46] Lawrence G Roberts. “Machine perception of three-dimensional solids”. PhD thesis. Massachusetts Institute of Technology, 1965.

[47] Grégory Rogez and Cordelia Schmid. “Mocap-guided data augmentation for 3d pose estimation in the wild”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 3108–3116.

[48] Johannes L Schonberger and Jan-Michael Frahm. “Structure-from-motion revisited”. In: *CVPR*. 2016.

[49] Tomas Simon et al. “Hand Keypoint Detection in Single Images using Multiview Bootstrapping”. In: *arXiv preprint arXiv:1704.07809* (2017).

[50] Christoph Strecha, Rik Fransens, and Luc Van Gool. “Combined depth and outlier estimation in multi-view stereo”. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2006, pp. 2394–2401.

[51] Christoph Strecha et al. “On benchmarking camera calibration and multi-view stereo for high resolution imagery”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. Ieee. 2008, pp. 1–8.

[52] Hang Su et al. “Multi-view convolutional neural networks for 3d shape recognition”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 945–953.

[53] Jian Sun et al. “Symmetric stereo matching for occlusion handling”. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE. 2005, pp. 399–406.

[54] Bugra Tekin et al. “Direct prediction of 3d body poses from motion compensated sequences”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 991–1000.

[55] Carlo Tomasi and Takeo Kanade. “Detection and tracking of point features”. In: (1991).

[56] Shubham Tulsiani and Jitendra Malik. “Viewpoints and keypoints”. In: *CVPR*. 2015.

[57] Minh Vo, Srinivasa G. Narasimhan, and Yaser Sheikh. “Spatiotemporal Bundle Adjustment for Dynamic 3D Reconstruction”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[58] Shaofei Wang and Charless C Fowlkes. “Learning optimal parameters for multi-target tracking with contextual interactions”. In: *International Journal of Computer Vision* 122.3 (2017), pp. 484–501.

[59] Shih-En Wei et al. “Convolutional pose machines”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4724–4732.

[60] Zhirong Wu et al. “3d shapenets: A deep representation for volumetric shapes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1912–1920.

[61] Yu Xiang, Alexandre Alahi, and Silvio Savarese. “Learning to track: Online multi-object tracking by decision making”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4705–4713.

[62] Yu Xiang et al. “Data-driven 3d voxel patterns for object category recognition”. In: *CVPR*. 2015.

[63] Qingxiong Yang et al. “Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.3 (2009), pp. 492–504.

[64] Yi 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. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1385–1392. ISBN: 978-1-4577-0394-2. DOI: 10.1109/CVPR.2011.5995741. URL: http://dx.doi.org/10.1109/CVPR.2011.5995741.

[65] Jifeng Dai Xiangyang Ji Yi Li Haozhi Qi and Yichen Weil. “Fully Convolutional Instance-aware Semantic Segmentation”. In: *CVPR*. 2017.

[66] Li Zhang, Yuan Li, and Ramakant Nevatia. “Global data association for multi-object tracking using network flows”. In: *CVPR*. 2008.

[67] Tinghui Zhou et al. “Unsupervised Learning of Depth and Ego-Motion from Video”. In: *CVPR*. 2017.

[68] M Zeeshan Zia, Michael Stark, and Konrad Schindler. “Towards scene understanding with detailed 3d object representations”. In: *IJCV* (2015).

[69] M Zeeshan Zia et al. “Detailed 3d representations for object recognition and modeling”. In: *TPAMI* (2013).

[70] C Lawrence Zitnick and Takeo Kanade. “A cooperative algorithm for stereo matching and occlusion detection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.7 (2000), pp. 675–684.