Self-Supervising Occlusions For Vision

N Dinesh Reddy

CMU-RI-TR-22-72

December, 2022

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Srinivasa G. Narasimhan, Carnegie Mellon University (Chair)
Deva Ramanan, Carnegie Mellon University
Kris Kitani, Carnegie Mellon University
Yaser Sheikh, Carnegie Mellon University and Meta Reality labs
Jan-Michael Frahm, University of North Carolina, Chapel hill and Meta Reality labs

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Robotics.

Keywords: Occlusions, Self-Supervision, Self-Occlusion, Occluded-by-others

Dedicated to Family

Abstract

Virtually every scene has occlusions. Even a scene with a single object exhibits self-occlusions - a camera can only view one side of an object (left or right, front or back), or part of the object is outside the field of view. More complex occlusions occur when one or more objects block part(s) of another object. Understanding and dealing with occlusions is hard due to the large variation in the type, number, and extent of occlusions possible in scenes. Even humans cannot accurately segment or predict the contour or shape of the occluded region when the object is occluded. Current large human-annotated datasets cannot capture such a wide range of occlusions. In this thesis, we propose learning amodal .i.e both visible and occluded regions of objects in a self-supervised fashion in densely populated scenes.

Occlusions in a scene can be broadly categorized into either self-occlusion, occluded-by-others, and/or truncation. For learning in self-occluded regions, We use multi-view priors in a bootstrapping framework to infer the content of occluded regions of the image. We show that such supervision helps the network learn better image representations even with large occlusions. We extend this using temporal cues from a stationary camera to learn accurate 3D shapes of self-occluded objects. For Occlusion by others, we explored using longitudinal data i.e. videos captured over weeks, months, or even years to supervise occluded regions in an object. We exploit this real data in a novel way to first automatically mine a large set of unoccluded objects and then composite them in the same views to generate occlusion scenarios. This self-supervision is strong enough for an amodal network to learn the occlusions in real-world images.

Finally, We show two methodologies for learning different types of occlusions. First, We combine the previous two paradigms of learning Self-Occluded and Occlusion by others for predicting the 3D amodal reconstruction of objects. Here, we show by learning and exploiting different occlusion categories like Self-occluded, and occluded by others and truncation can enhance the accuracy of the reconstruction. On the other hand, we show learning of 3D reconstruction and tracking of objects in an end-to-end learning framework using multi-view video input. We will discuss and analyze the pros and cons of the different approaches and representations for the amodal representation of objects.

Acknowledgments

I am truly lost for words on how to start the acknowledgement section. It was truly a pleasure to be surrounded by some of the most dedicated researchers during my journey at CMU and it was a fun and exciting ride. Nonetheless, I should start with the the person who has played a major role in helping me shape this thesis, my advisor and mentor, Srinivas. I still remember the first meeting with him as a master student and the day he decided to take me on as his student. From that day till now he has constantly been there to help me improve at every step in my research pursuits. This dedication to research as well as supervision is truly commanding and is greatly appreciated. You have shown me how to channel all my enthusiasm for research into solving a meaningful research problem. I could not have asked for anything more from my advisor.

I am grateful to the thesis committee for taking out their valuable time to help improve on my presentation and research capabilities. yaser's constant inquiries into the details in my thesis both during my masters and PHD have played a major role for me to be equally inquisitive. Deva's patience, collaborative spirit and research direction have shown me how to think out of the box into different dimensions of the same problem. Big shout out to kris for his detailed comments on my thesis and for organizing smith hall events. Both of these things played an important role in finishing up the thesis. Finally, I would like to thank Jahn-micheal-frahm for taking his time out from his busy schedule for being my external committee member. Your comments in our conversations showed how to make the solutions more relevant and applicable.

Research progresses based on your interactions with other researchers and staying in smith hall gave me access to multiple researchers with plethora of research experience. Constant interactions with yannis, Matt o' Toole, Fernando la Torre, abhinav gupta, David held, Keenan crane, shubham thulsani, Deepak Pathak, jun Yan Zhu and Katerina Fragkiadaki gave me a good insight into multiple research directions and am thankful for the fruitful discussions. Many thanks to – Suzanne Muth, Brian hutchinson, Jess Butterbaugh, Stepheny Matvey, Christine Downey, for making everything smooth and seamless.

I am fortunate to have spent time as an intern at amazon go lab in Seattle. Collaborating with Leonid and Laurent was exciting for me to see how research impacts the company and general public in numerous ways. A big shout out to all the people who I collaborated with at CMU. Specifically the ILIM lab and the generic imaging group was very instrumental and helpful in many ways. To name a few, Minh Vo, Robert tamburo, Mark sheinin, Aditya pediredla, khiem vyong, gaurav pathak, siddharth ancha, tiancheng zhi, choa liu, shumian xin, gaurav parmar, aditya garlapati, fangyu li, xudong chen, zhiyu, shafali srivatsava, neha boloor, and many more.

Most of my time at CMU was spend in the first floor of smith hall. I formed some everlasting friendships in this place, which will be cherished and missed- Aditya murali, raaj yaadhav, rawal khirodkar, aayush bansal, sudeep dasari, yufei, gengshan yang, chaoyang wang, Tarasha Khurana, Roberto shu, Gines hidalgo, shikhar ball, pragna mannam, Raunaq bhirangi, dheeraj gandhi, neehar peri.

Half my PHD was remote due to the covid stay-at-home orders. Thanks to senthil purushwalkam, Achal Dave, Kenny, anti Bhatia and others for keeping me company during these times with rocket league tournaments. Constant contact from the nameless uundergrad group helped me escape the craziness of the lockdowns. Many others made my stay special during this time - NAMELESS Group from my undergrad batchmates, navyatha Sanghvi, John shi, navya yerrabelly, nishtha jain, sharvani chandu and many more for the fun discussions. I specifically thank Siddharth malreddy, Shashank jujjuvarapu, Deepthi hedge, Ishan nizam, Sudeep, Yashasvi, Kalli, Lerrel pinto for their support during the course of my stay at CMU during my masters.

Finally, I would like to dedicate this thesis to my family for there constant support during my both my up and downs. This thesis would not be possible without my mother(nirmala) and father(janardhana reddy) who stuck with my craziness to pursue a path new to them. My sister bhavya and my cousins swetha and madhu and their family for constantly looking out for me and letting me stay at their homes during covid. I dedicate this thesis to all of them and 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

Chapter 1

Introduction

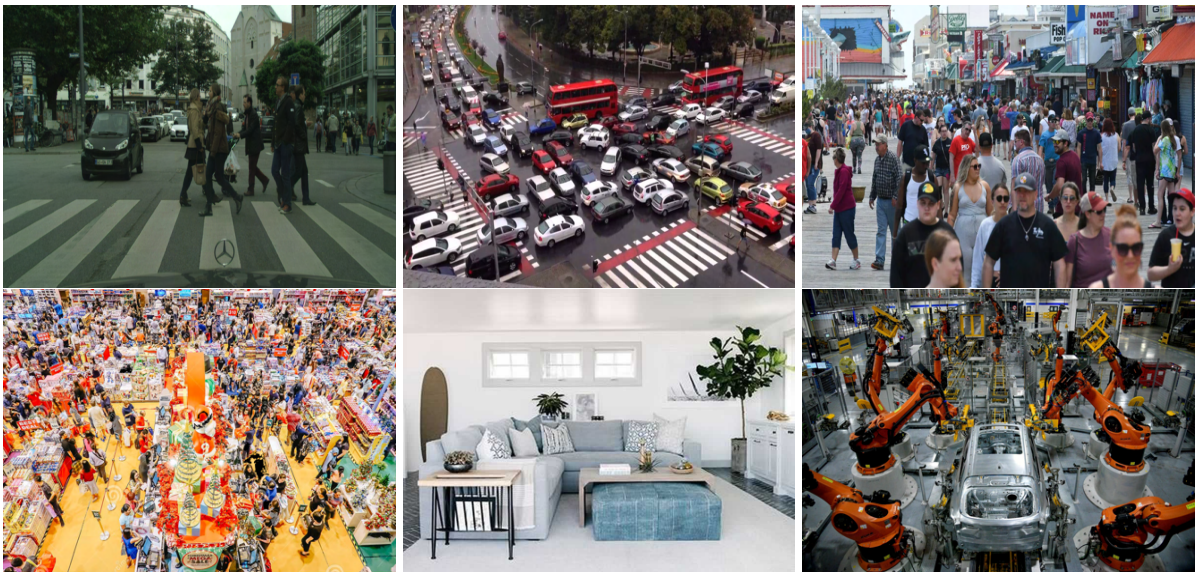


Figure 1.1: We illustrate images captured in the wild from a wide variation of camera locations like dashboard cameras, traffic cameras, mobile phones, Shopping CCTV footage, Home robots, and manufacturing. Observe that virtually all images captured have occlusions. We observe different kinds of occlusions like people and vehicles occluding objects, stationary objects, and robotics occluding scenes. Understanding complete 3D scenes from such data will play a major role in scene understanding and 3D Reconstruction which can be utilized by different downstream tasks like robot perception, planning, and automation. In this thesis, we will discuss the methodology to tackle vision problems like detection, segmentation, and reconstruction under occlusions and propose methods to generate an accurate holistic representation of objects using self-supervised frameworks.

Virtually every scene has occlusions as shown in Fig 1.1. Even a scene with a single object exhibits self-occlusions - a camera can only view one side of an object (left or right, front or back), or part of the object is outside the field of view. More complex occlusions occur when one or more objects block part(s) of another object. Understanding and dealing with occlusions is hard due to the large variation in the type, number, and extent of occlusions possible in scenes. Understanding occlusions and modeling vision algorithms to be robust to occlusions is the major emphasis of this thesis. While there has been strong progress in data-driven methods for object detection, tracking, segmentation and reconstruction with limited occlusions, most methods

under-perform in severely occluded scenarios. Severe occlusions are common in busy intersections and crowded places. Even in less dense scenes, pedestrians and vehicles often pass each other or pass behind other objects. As a result, objects are either not detected at all, or the 2D bounding boxes and segments are truncated and produce errors in downstream processes such as 3D reconstruction. Much of this state of affairs can be attributed to the fact that occlusions are treated as noise that must be overcome by robust measures. There are several challenges that make this strategy hard to succeed. First, it is much harder to label object bounding boxes or segments that are occluded, even for humans. Thus, even large datasets like COCO[11] and ImageNet[12] have relatively few objects labeled that are severely occluded. This creates a strong bias against learning robustness to occlusions. Further, the evaluation metrics are often reported on the entire datasets [11, 13, 14] that could hide problems in occluded scenarios. This thesis

Figure 1.2: We show annotation of a different object in an image using multiple human expert annotators to tackle the problem of bounding box, pose, and segmentations. We observe very high agreement when the full object is visible(First row). But in the example of an occluded object (the car behind the yellow cab), where the annotators have different thoughts about extending the object beyond the visible regions. This is naturally much harder to annotate than if the object had been fully visible in the image. Further such occlusions can be seen to extend to truncated objects(the black car in the third row) as well. We observe the huge variance in annotations across all representations .i.e Bounding boxes, keypoints etc. This illustrates the need for using self-supervision for reasoning about occlusions.

1.1 Why Human Annotations for occlusions are Imprecise?

Obtaining an accurate representation of objects when occluded is challenging. Even humans cannot accurately segment or predict the contour or shape of the object when occluded. To emphasize this point we did a simple experiment with human annotators. we had the same 200 camera images being annotated by 14 different professional annotators, all with high production quality. The images were typical for different applications both in terms of content and technical standard, and bounding boxes were annotated according to a well-defined annotation guideline. Altogether the images contained some 2500 objects, and with 14 annotations of each, we recorded some 35 000 bounding boxes in this experiment. We show some sample annotations in 1.2. To get the best possible reference annotation, we used "the wisdom of the crowd" and averaged the 14 possible different annotations to define the ground truth for each object. We thereafter studied how many pixels each boundary of each individual annotation deviated from this reference ground truth.

With well-trained professional annotators, it would seem reasonable to expect a high level of agreement between the annotators. And most of the time, that seems to be the case since most pixel deviations are close to 0. However, there is also a significant (and perhaps surprising) share of deviations that are not particularly close to 0 (the most extreme deviations, far beyond the shown range of the histogram, were in fact about 400 pixels). In statistical terms, the distribution has a so-called heavy tail. Generally, such distribution corresponds to occluded objects.

1.2 Self-Supervising occlusions

Such errors in human annotations propel us to learn occlusions using self-supervised methods with no human annotations. Occlusions in a scene can be broadly categorized into either self-occlusion, occluded-by-others, and/or truncation. For learning in self-occluded regions, We use multi-view priors in a bootstrapping framework to infer the content of occluded regions of the image. We show that such supervision helps the network learn better image representations even with large occlusions. We extend this using temporal cues from a stationary camera to learn accurate 3D shapes of self-occluded objects. For Occlusion by others, we explored using longitudinal data i.e. videos captured over weeks, months, or even years to supervise occluded regions in an object. We exploit this real data in a novel way to first automatically mine a large set of unoccluded objects and then composite them in the same views to generate occlusion scenarios. This self-supervision is strong enough for an amodal network to learn the occlusions in real-world images.

Finally, We show two methodologies for learning different types of occlusions. First, We combine the previous two paradigms of learning Self-Occluded and Occlusion by others for predicting the 3D amodal reconstruction of objects. Here, we show by learning and exploiting different occlusion categories like Self-occluded, and occluded by others and truncation can enhance the accuracy of the reconstruction. On the other hand, we show learning of 3D reconstruction and tracking of objects in an end-to-end learning framework using multi-view video input. We will discuss and analyze the pros and cons of the different approaches and representations for the amodal representation of objects.

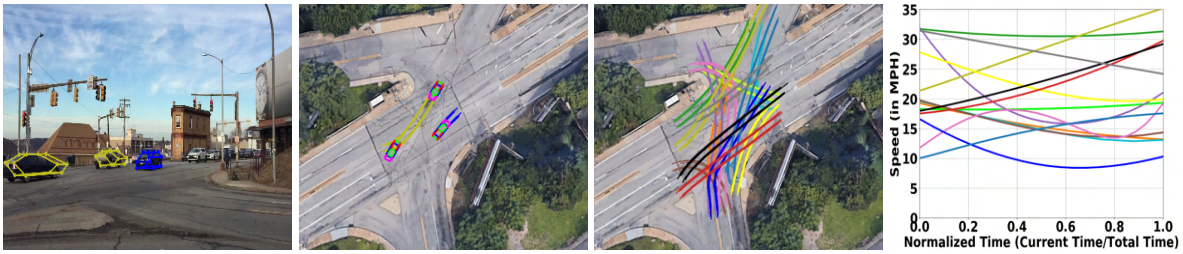
Figure 1.3: Self-Occlusion Supervision Emerges from multi-view data. The scene depicts multiple people playing a game of tag and the green-shirt lady is occluded in the view of the initial frame. To predict the exact pose of the occluded objects in the current image, we search in multi-view data(next frames) without occlusions and project into the initial image as ground truth supervision for occluded regions. This multi-View supervision helps in automatically generating large occlusion data using cameras in the wild.

Supervision for self-occlusion: In chapter 2, We present Occlusion-Net, a framework to predict 2D and 3D locations of occluded key points for objects, in a largely self-supervised manner. We use an off-the-shelf detector as input (e.g. MaskRCNN [15]) that is trained only on visible key point annotations. This is the only supervision used in this work. A graph encoder network then explicitly classifies invisible edges and a graph decoder network corrects the occluded keypoint locations from the initial detector. Central to this work is a trifocal tensor loss that provides indirect self-supervision for occluded keypoint locations that are visible in other views of the object. The 2D keypoints are then passed into a 3D graph network that estimates the 3D shape and camera pose using the self-supervised reprojection loss. At test time, Occlusion-Net successfully localizes keypoints in a single view under a diverse set of occlusion settings. We validate our approach on synthetic CAD data as well as a large image set capturing vehicles at many busy city intersections. As an interesting aside, we compare the accuracy of human labels of invisible keypoints against those predicted by the trifocal tensor.

Figure 1.4: Example results of Supervision for self-occlusion. We accurately localize occluded keypoints under a variety of severe occlusions Specifically in self-occluded cases. Different colors depict different vehicles.

Supervision for Temporal occlusions: 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. In chapter 3, We present a novel approach exploiting longitudinal (long-term) repetitious motion as self-supervision to reconstruct 3D activity from a video captured by a single fixed camera. Starting from off-the-shelf 2D keypoint detections, our algorithm optimizes 3D shapes and poses, and then clusters their trajectories in 3D space. The 2D key points and trajectory clusters accumulated over the long

term are later used to improve the 2D and 3D key points 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 videos captured at multiple city intersections, collected using our smartphones, YouTube, and other public datasets.



Keypoints Reconstruction Mean Traj. Mean Vel.

Figure 1.5: The keypoints and 3D reconstructions overlaid on Google map for a camera. We show 3D mean trajectories and velocities of the mean trajectories. These mean trajectories represent typical vehicle motions.

Figure 1.6: Occlusion by Others Supervision Emerges from Longitudinal data. The scene depicts a parking lot at a busy intersection with continuous motion. We use time lapse videos of the camera over long duration to extract unoccluded object .i.e the black car in the image as supervision at instances when the car is occluded by mutiple other vehicles.

Supervision for occlusion by others: Current methods for object detection, segmentation, and tracking fail in the presence of severe occlusions in busy urban environments. Labeled real data of occlusions is scarce (even in large datasets) and synthetic data leaves a domain gap, making it hard to explicitly model and learn occlusions. In chapter 4, we present the best of both the real and synthetic worlds for automatic occlusion supervision using a large readily available source of data: time-lapse imagery from stationary webcams observing street intersections over weeks, months, or even years. We introduce a new dataset, Watch and Learn Time-lapse (WALT), consisting of multiple (4K and 1080p) cameras capturing urban environments over a year. We exploit this real data in a novel way to first automatically mine a large set of unoccluded objects and then composite them in the same views to generate occlusion scenarios. This self-supervision is strong enough for an amodal network to learn the object-occluder-occluded layer representations. We show how to speed up the discovery of unoccluded objects and relate the confidence in this discovery to the rate and accuracy of training of occluded objects. After watching and automatically

learning for several days, this approach shows significant performance improvement in detecting and segmenting occluded people and vehicles, over human-supervised amodal approaches. Reconstructing 4D activity (3D space and time) from cameras is useful for autonomous vehicles, commuters, and local authorities to plan for smarter and safer cities.

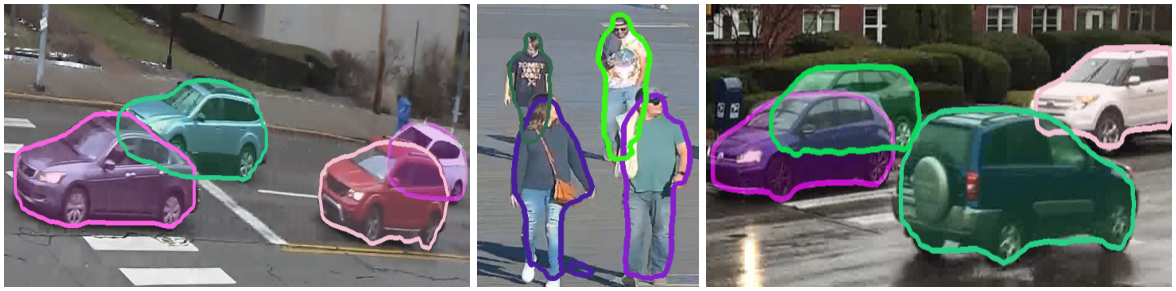


Figure 1.7: The amodal representation of vehicles and people under severe occlusions by others showing significant improvement in amodal detection and segmentation.

Exploiting Occlusion Categories: Objects are occluded in scenes in numerous complex ways. For example, they may be partially occluded by other static or dynamic objects, truncated by the camera's field of view, or be self-occluded, i.e., the camera-facing side of the object is occluded by the opposing side of the object. In Chapter 5, We present a holistic approach to handle such occlusions for amodal 3D shape reconstruction. The approach starts with learning occlusion categories with human supervision. Then, these learned categories are exploited in a novel framework that uses a mixed representation (keypoints, segmentations and shape basis) for objects to automatically generate a large physically realistic dataset of occlusions using freely available time-lapse imagery. This dataset provides strong 2D and 3D self-supervision to a network that jointly learns amodal 2D keypoints and segmentations, which are then used in optimization to reconstruct 3D shapes. Automatically estimated visibility is used to supervise the entire pipeline. Our system demonstrates significant improvements in amodal 3D reconstruction of heavily occluded objects captured at any time of the day from traffic, hand-held, and vehicle cameras.

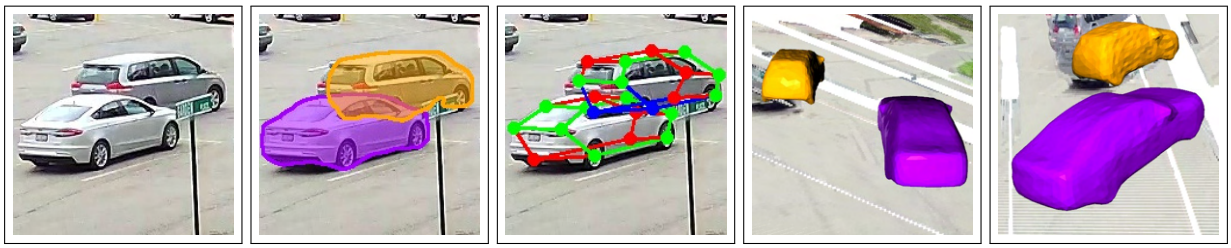


Figure 1.8: We show additional qualitative results of our method on multiple sequences of the WALT dataset. The input image **(col 1)** to the pipeline produces amodal segmentation mask **(col 2)** and keypoint locations **(col 3)**. in **(col 4 and 5)**, We visualize the 3d reconstruction from multiple views.

End-to-End occlusion learning: In chapter 5, We consider the task of 3D pose estimation and tracking of multiple people seen in an arbitrary number of camera feeds. We propose a novel top-down approach that simultaneously reasons about multiple individuals' 3D body joint reconstructions and associations in space and time in a single end-to-end learnable framework. At the

core of our approach is a novel Spatio-temporal formulation that operates in a common voxelized feature space aggregated from single- or multiple-camera views. After the detection step, a 4D CNN produces short-term person-specific representations which are then linked across time by a differentiable matcher. The linked descriptions are then merged and deconvolved into 3D poses. This joint Spatio-temporal formulation contrasts with previous piece-wise strategies that treat 2D pose estimation, 2D-to-3D lifting, and 3D pose tracking as independent sub-problems that are error-prone when solved in isolation. Furthermore, unlike previous methods, our method is robust to changes in the number of camera views and achieves very good results even if a single view is available at inference time. Quantitative evaluation of 3D pose reconstruction accuracy on standard benchmarks shows significant improvements over the state of the art.

Figure 1.9: We illustrate the output of Tessetrack on the Tagging sequence. The top two row potray the projections of keypoints on two views, while the bottom row shows the 3D pose tracking. Observe smooth tracking of people in the wild with moving cameras for long duration of time.

1.3 Datasets

Figure 1.10: The top column consists of Google map view of the intersection used to capture the data.

1.3.1 CarFusion:

The dataset captured multiple traffic scenes with six Samsung Galaxy 6, ten iPhone 6, and six Gopro Hero 3 cameras at 60 fps in a busy intersection for 10 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. To model a wide range of real occlusions, we collect an extensive dataset captured simultaneously by multiple mobile cameras at 60fps at 5 crowded traffic intersections . This extended dataset consists of 2.5 million images out of which 53000 images were sampled at uniform intervals from each video sequence. Approximately, 100000 cars detected in these images were annotated with 12 keypoints each. Each annotation contains the visible and occluded keypoint locations on the car. We do not use the occluded keypoints for training. We selected four annotated intersections to train the network while using one intersection to test it, which split the annotation data into 36000 images for training and 17000 for testing. We further compute a 90-10 train validation split on the training data to validate our training algorithm. The dataset was completely captured "in the wild" and contains numerous types and severity of occlusions.he data for this "CarFusion Dataset" is available for research purposes[1]. We show the Google map view of the intersections, which were used for capturing the data in Figure 1.10. We released the dataset for further research in the direction of Multi-View data for different tasks like keypoint detection, segmentation etc.

Dataset	Image source	Appearance diversity in terms of				# images	# car instances	Occ. keypoint annotations	Per-keypoint occ. type
		Cities	Times of Day	Weathers	Viewpoints				
PASCAL3D+	Natural	Yes	Yes	Yes	No	6,704	7,791	No	No
KITTI-3D	Self-driving	No	No	No	No	2,040	2,040	No	No
ApolloCar3D	Self-driving	No	No	No	No	5,277	60,000	No	No
Carfusion	Handheld	No	No	No	No	53,000	100,000	Yes	No
WALT-Annotated	Handheld Self-driving Traffic cameras	Yes	Yes	Yes	Yes	7,018	42,547	Yes	Yes

Table 1.1: Summary and comparison of our datasets to other publicly available datasets with vehicle keypoint annotations.

[1]http://www.cs.cmu.edu/∼ILIM/projects/IM/CarFusion/

1.3.2 Watch And Learn Time-lapse (WALT) Dataset:

The dataset consists of 25 4K resolution cameras setup by us and 75 1080p YouTube public live streams. The cameras overlook public urban settings analyzing the flow of traffic and people with severe occlusions. We used 4 cameras from our setup and 6 cameras from YouTube for training. Data captured from 2 cameras are used for testing. The data is captured for 3-second bursts at 30 FPS every few minutes. Only the images with notable changes from the previous image are stored. This results in storing approximately 5000 images per day for a year. We will be releasing months of data captured from cameras set up by us and publish a live stream video of the cameras on YouTube for research purposes. The code to automatically capture and process data from YouTube live streams will be released.

Chapter 2

Supervision For Self-Occlusions

Virtually any scene has occlusions. Even a scene with a single object exhibits self-occlusions - a camera can only view one side of an object (left or right, front or back), or part of the object is outside the field of view. More complex occlusions occur when one or more objects block part(s) of another object. Understanding and dealing with occlusions is hard due to the large variation in the type, number and extent of occlusions possible in scenes. As such, occlusions are an important reason for failure of many computer vision approaches for object detection [15, 16, 17, 18], tracking[19, 20, 21, 22], reconstruction [23, 24] and recognition, even today's advanced deep learning based ones.

The computer vision community has collectively attempted numerous approaches to deal with occlusions [25, 26, 27, 28] for decades. Bad predictions due to occlusions are dealt with as noise/outliers in robust estimators. Many methods provide confidence or uncertainty estimates to downstream approaches that need to sort out whether the uncertainty corresponds to occlusion. But it is hard to predict performance as they usually do not take occlusions explicitly into account.

On the other hand, occlusions are explicitly treated as missing parts in model fitting methods [29, 30]. These approaches have had better success as they exploit a statistical model of a particular type of object (e.g. car, human, etc.). But much remains to be done. For instance, severe occlusions, such as when a large part of an object is blocked, can result in poor fitting[31]. Further, often these approaches do not explicitly know which parts of an object are missing and attempt to simultaneously estimate the model fit as well as the missing parts.

In this work, we present an approach to explicitly predict 2D and 3D keypoint locations of the occluded parts of an object using graph networks, in a largely self-supervised manner. Our method receives as input, the output of any detector (e.g., using the MaskRCNN architecture [15]) that has been trained on a particular category of object with human supervision of *only visible keypoints* and their types (e.g., front, back, left, right). Implicitly, then, the key points that are not labeled are assumed to be invisible. This is the only human supervision used in this work. The detector usually provides an uncertainty of all key point locations. We first show that the distribution of the uncertainties for visible and occluded points overlap significantly, making it hard to predict which key points are occluded at test time. To address this issue, we design an encoder-decoder graph network that first predicts which edges have an occluded node, and then localizes the occluded node in 2D in the decoder. Visible or invisible edge classification is trained using the implicit non-labeled supervision of occluded points.

We then train the decoder graph network to localize invisible keypoints using multiple wide-

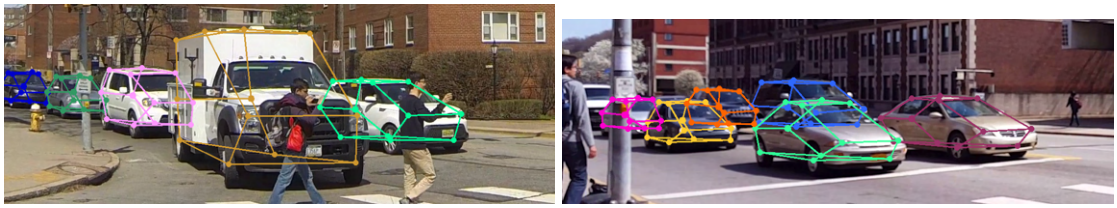


Figure 2.1: Accurate 2D keypoint localization under severe occlusion in our CarFusion dataset. Different colors depicts different objects in the scene.

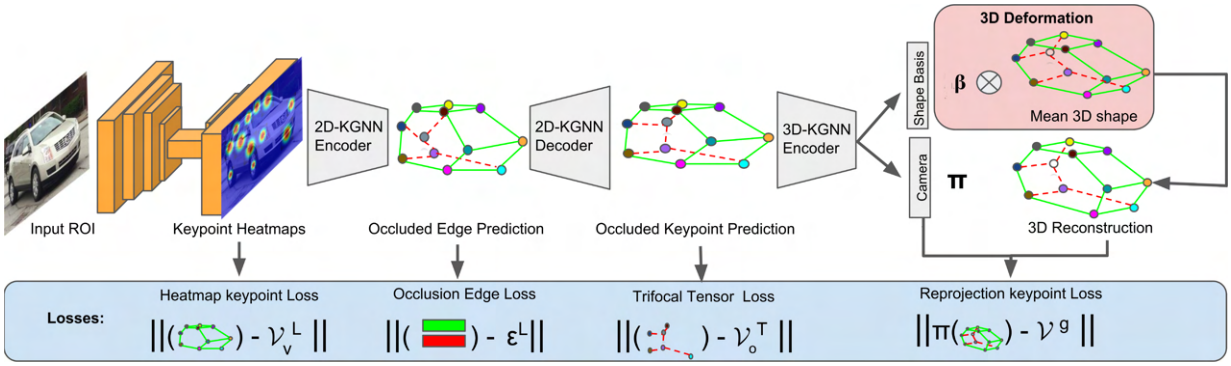


Figure 2.2: Occlusion-net: We illustrate the overall approach to training a network to improve the localization of occluded key points. The input is an ROI region from any detector, which is passed through multiple convolutional layers to predict the heatmaps with a confidence score. These confidences are passed through a graph encode-decoder network and trained using multi-view trifocal tensor loss for localization of occluded 2D keypoints. The output from the decoder is passed through a 3D encoder to predict the shape basis and the camera orientation. This network is a self-supervised graph network and is trained using reprojection loss with respect to the 2D decoder output.

baseline views of objects. Our observation is that while some parts may be missing in one view, they are visible and labeled in another view. But how do we provide supervision for a hidden point location in a view? We use two views where a keypoint is seen (and labeled by humans) and compute the trifocal tensor using camera matrices to predict its location in the view where the keypoint is occluded. We call this the **Trifocal tensor loss**, which is minimized to correct the 2D keypoint positions from the initial detector. Compared to other approaches that use multiple views [32, 33, 34], our approach explicitly predicts occluded keypoints.

The predicted 2D keypoints (both occluded and visible) are then used in a graph network to estimate the 3D object shape and the camera projection matrix. Similar to previous work [31, 35], we will estimate the parameters of a shape basis computed a priori of the object of interest. The training is performed in a self-supervised way by minimizing the reprojection loss i.e. error between the reprojection and the predicted 2D keypoint locations. We train the entire pipeline, called Occlusion-net, end-to-end with the aforementioned losses.

We evaluate our approach on images of vehicles captured at busy city intersections with numerous types and severity of occlusions. The dataset extends the previous CarFusion dataset [33] to include many more city intersections, where 18 views of the intersection are simultaneously recorded. A MaskRCNN car detector is trained using 100000 cars, with human labeled visible keypoints to produce a strong baseline for our method to compare to and build upon.

Our Occlusion-net significantly outperforms (about 10%) this baseline across many metrics and performs well even in the presence of significant occlusions (see Figure 5.1). As an interesting exercise, we also show a comparison of the trifocal loss against human labeling of the 2D occluded point locations and observe that humans label around 90% of the points to lie within the acceptable range of error. We also evaluate our approach on a large synthetic CAD dataset, showing similar performance benefits and improvements of up to 20% for occluded keypoints. Our network is efficient to train and can localize keypoints in 2D and 3D in real-time (more than 30 fps) at test-time. While we have demonstrated our approach on vehicles, the framework is general and applies to any object category.

2.1 Related Work

Occlusion Detection: While there has been significant progress in predicting the visible keypoints by using part detectors learned from CNNs [27, 36, 37, 38, 39, 40], most of these methods fail short to precisely localize occluded keypoints. Using synthetic data, Moreno et al. [41] show that such occlusion modeling is crucial. To address this problem, many methods employ active shape models [42] for vehicle detection under occlusion [31, 43, 44]. However, these methods only model self-occlusions and omit often seen occlusions by other objects. Recently, [33, 34] propose a multi-view bootstrapping approach to generate accurate CNN training data when precise human labeling is not possible. However, their methods are trained in stages and do not explicitly model the interaction between visible and occluded points. Most related to our work, [2] only incorporates intermediate keypoint supervisions from CAD model during training. Interestingly, they show that training such a model on synthetic images can generalize to real images. We train our model on real images and incorporate multiview constraints to propagate ground truth visible keypoints from multiple views to supervise occluded points.

Graph Neural Networks: Modeling keypoints as a graph problem can be dated back to the first attempt at scene understanding [45, 46]. Multiple works have built on this graph representation and solved pose using belief propagation [47, 48]. Recently, [49, 50, 51, 52, 53] have extended classical graphical modeling to a deep learning paradigm and showed better modeling capability for unstructured data. Based on the success of these methods on the graph classification tasks, multiple recent works have extended the methods to address multiple 3D problems like Shape segmentation [54], 3D correspondence [55] and CNN on surfaces [56]. We model keypoint prediction as a deformable graph that is learned using multi-view supervision.

2.2 Occlusion-Net

Occlusion-Net consists of three main stages - visible keypoints detection, occluded 2D keypoint localization and 3D keypoint localization networks - as shown in Figure 6.2. The 2D-Keypoint Graph Neural Network deforms the graph nodes to infer the 2D image locations of the occluded keypoints. The 3D-Keypoint Graph Neural Network localizes the 3D keypoints of the graph using a self-supervised training procedure. We combine these networks to accurately predict the 3D and 2D keypoint locations. Each of these stages is described in the following sections.

2.2.1 2D-Keypoint Graph Neural Network

The 2D-Keypoint Graph Neural Network(2D-KGNN) consists of three components: initial keypoint heatmap prediction, a graph encoder to model the occlusion statistics of the graph, and a graph decoder infering the 2D locations of the occluded keypoints. We use the heatmap based methods [15][36] to compute the location of all the keypoints in an image. The input to the graph network consists of k keypoints, which are further categorized as v visible keypoints and o invisible/occluded keypoints. We denote the vertex of the graph as $\mathcal{V} = (\mathcal{V}_1, ..., \mathcal{V}_k)$ for k keypoints. The relationship between all nodes is encoded in the edge $\mathcal{E}_{ij} = \{\mathcal{V}_i, \mathcal{V}_j\}$, where

$$\mathcal{E}_{ij} = \begin{cases} 1, & \text{if } i \in v \; and \; j \in v \\ 0, & \text{otherwise} \end{cases}$$

We also denote $\mathcal{V}^l$ as labeled keypoint annotations and $\mathcal{V}^g$ as keypoints predicted from 2D-KGNN, respectively.

2D-KGNN Encoder: Occluded Edge Predictor

The 2D keypoint graph network (2D-KGNN) needs to infer the locations of the occluded keypoints (or, edges $\mathcal{E}_{ij}$) from the keypoint heatmaps. We convert the heatmap into a graph by encoding the location and confidence of each keypoint into a node feature. The feature for keypoint i, can be more formally represented as $\mathcal{V}_i = \{x_i, y_i, c_i, t_i\}$, where (x_i, y_i) is the location, c_i is the confidence and t_i is defined as the type of the keypoint. Since, we do not know the underlying graph, we use the GNN to predict the latent graph structure.
The encoder is modeled as $q(\mathcal{E}_{ij}|\mathcal{V}) = softmax(f_{enc}(\mathcal{V}))$ where $f_{enc}(\mathcal{V})$ is a GNN acting on the fully connected graph produced from the heatmaps. Given the input graph our encoder computes the following message passing operations to produce the occlusion statistics:

$$h_j^1 = f_{enc}(\mathcal{V}_j) \tag{2.1}$$

$$v \rightarrow e : h_{(i,j)}^1 = f_e^1([h_i^1, h_j^1]) \tag{2.2}$$

$$e \rightarrow v : h_j^2 = f_v(\sum_{i \neq j} h_{(i,j)}^1) \tag{2.3}$$

$$v \rightarrow e : h_{(i,j)}^2 = f_e^2([h_i^2, h_j^2]) \tag{2.4}$$

In the above equations, h^t denotes the t^{th} hidden layer of the network, while v and e denote the vertex and edge of the graph. Here, $v \rightarrow e$ shows a convolution operation from vertex to edge, while $e \rightarrow v$ represents the operation from edge to vertex. The functions $f()$ are implemented as fully connected layers. The edge loss for this encoder is the cross-entropy loss between the predicted edges and the ground truth edges, given as:

$$L_{Edge} = -\sum_{i,j \in k} \mathcal{E}_{ij} log(\mathcal{E}_{ij}^l) \tag{2.5}$$

The $\mathcal{E}_{ij}^l$ is the visibility statistics for each edge computed from the labeled keypoints.

2D-KGNN Decoder: Occluded Point Predictor The decoder predict consistent 2D keypoint locations of the occluded keypoints from the erroneous initial graph and the edges predicted

from the encoder. This can mathematically be represented as estimating $P_\theta(\mathcal{V}^g|\mathcal{V},\mathcal{E})$, where $\mathcal{V}^g$ represents the output graph from the decoder and $\mathcal{E}$ is the input from encoder, while $\mathcal{V}$ is the graph from the initial heatmap. The following message passing steps are computed on the graph network:

$$v \rightarrow e : h_{(i,j)} = \sum_p \mathcal{E}_{ij,p} f_e^p([\mathcal{V}_i, \mathcal{V}_j]) \tag{2.6}$$

$$e \rightarrow v : \mu_j^g = \mathcal{V}_j + f_v(\sum_{i \neq j} h_{(i,j)}) \tag{2.7}$$

$$P_\theta(\mathcal{V}^g|\mathcal{V},\mathcal{E}) = \mathcal{N}(\mu_j^g, \rho^2 I) \tag{2.8}$$

Here $\mathcal{E}_{ij,p}$ denotes the p-th element of the vector $\mathcal{E}_{ij}$. An important thing to observe is the current state is added into Eq. 2.7, so inherently the model is learning to deform the keypoints i.e predict the difference $\Delta\mathcal{V} = \mathcal{V}^g - \mathcal{V}$. Further in Eq. 2.7, μ is the mean location predictor and $\mathcal{N}$ produces the probability of the locations. We only minimize the distance between the predicted and ground truth occluded points in this network using a trifocal tensor loss.

Trifocal Tensor Loss. We exploit multiple views of the object captured "in the wild" to estimate the occluded keypoints. The assumption is that the keypoints occluded in one view are visible in two or more different views. Thus, the trifocal tensor [57] can transfer the locations in the two visible views to the occluded view. Then, the loss for each occluded keypoint is computed as:

$$L_{Trifocal} = \sum_{j \in o} [\mathcal{V}_j^g]_\times (\sum_i (\mathcal{V}')_j^i T_i) [\mathcal{V}_j'']_\times, \tag{2.9}$$

where i represents the three views considered for the trifocal tensor T, $\mathcal{V}_j^g$ is the prediction from the decoder for the occluded keypoint j in the current view, and $\mathcal{V}'_j$ and $\mathcal{V}''_j$ are the annotated keypoints j in two different views. We computed T using the camera poses in the object reference frame. In our setting, since the object (vehicle) is rigid, the two visible views could come from any camera viewing the same object at any other time instants.

2.2.2 3D-Keypoint Graph Neural Network

Given the graph from the 2D-KGNN decoder, the 3D-keypoint graph neural network encoder predicts a 3D object shape W and the camera projection matrix π. This encoder takes as input the graph and predicts the 3D location of the all the keypoints using a self-supervised projection loss. Mathematically, this is formulated as $q(\beta, \pi|\mathcal{V}) = f_{enc}(\mathcal{V})$, where, β are the deformation coefficients of PCA shape basis of the object and π is the camera projection matrix.

Shape Basis: We model the shape as a set of 3D keypoints corresponding to the predicted 2D keypoints. We compute the mean shape b_0 and n principal shape components b_j and corresponding standard deviations σ_j, where $1 \leq j \leq n$, using the 3D repository of the object [58] with annotations of 3D keypoints from [27]. Given the shape bases, any set of deformable

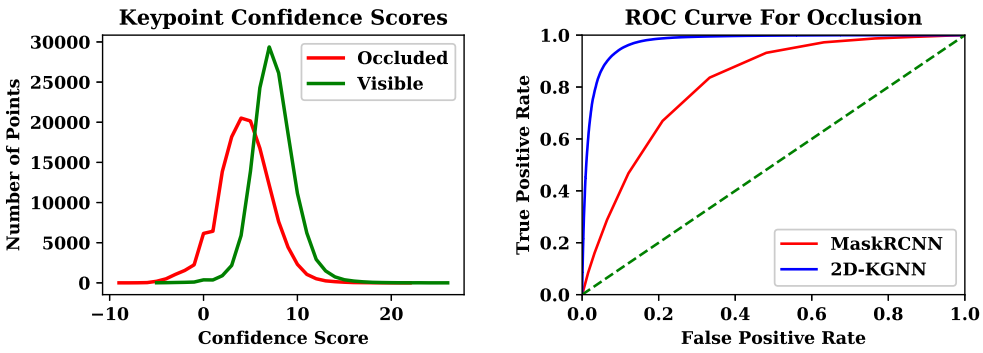


Figure 2.3: We analyze the need for a 2D-KGNN encoder. The left image shows the confidence score of the heatmaps from the baseline method (the distribution is colored based on Ground Truth visibility). The right image shows the ROC curve of the predictions from graph encoder and baseline. At 0.1 false positive rate, the baseline returns 0.5 true positive rates compared to 0.8 of the 2D-KGNN.

3D keypoints can be represented as a linear combination of the n principal components β as $W = b_0 + \sum_{k=1}^{n} \beta_k * \sigma_k * b_k$.

Camera Projection Matrix: Let $\pi(W)$ be the function that projects a set of 3D keypoints W onto the image coordinates. We use the perspective camera model and describe π as a function of the camera focal length f, the rotation q, represented as quaternion, and translation t of the object in the camera coordinate frame [57]. We assume the principle point of the camera is at the origin. To account for the normalization of the image to a square matrix from the original dimensions, we re-scale the projected 2D points by $s = w/h$, where w and h denote the width and height of the input image (see [59] for further details).

Keypoint Reprojection Loss: We train the 3D-Keypoint Graph network in a self-supervised manner using the reprojection loss, i.e. the difference between the projected 3D keypoints and the keypoints computed from the 2D-KGNN:

$$L_{Reproj} = \sum_{j \in k} ||\pi(W_j) - \mathcal{V}_j^g||^2 \tag{2.10}$$

The use of the 3D basis shape allows explicit enforcement of 3D symmetry which provides further constraints for the 2D keypoint estimation via the reprojection loss.

2.2.3 Total Loss

Our Occlusion-Net is trained to minimize the sum of the aforementioned losses:

$$L = L_{Keypoints} + L_{Edge} + L_{Trifocal} + L_{Reproj}, \tag{2.11}$$

where, $L_{Keypoints}$ is the cross-entropy loss over a t^2-way softmax output between the predicted keypoints and the ground truth labels [15]. Here, t is the number of keypoints.

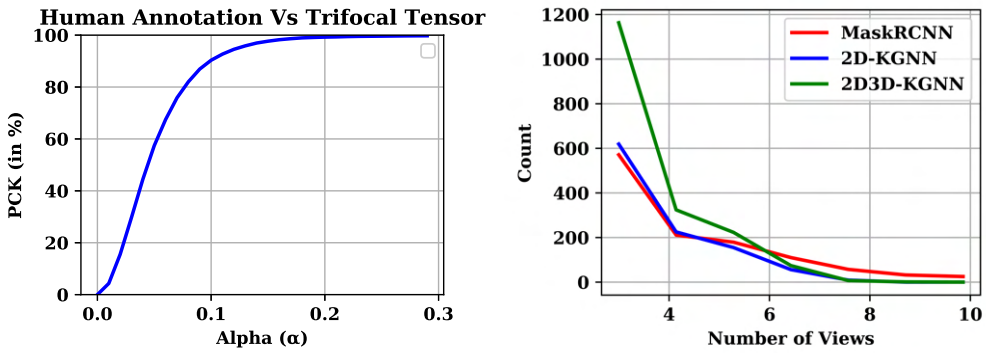


Figure 2.4: On the left, we show accuracy of human annotations with respect to geometrically obtained keypoints. We observe that most of the keypoints are labeled within $\alpha = 0.1$ PCK error. On the right, count of multi-view correspondences of keypoints predicted using different methods. When few views are available, the occluded points predicted by Occlusion-Net provide much more correspondences to improve multi-view reconstruction.

2.3 Experimental Results

We demonstrate the ability of our approach to infer occluded keypoints and 3D shape from a single view on the new and challenging CarFusion dataset. We first describe this dataset in section 2.3.1. We then perform ablative analysis of the algorithm in Section 2.3.2. Finally, we show qualitative comparisons against the state of art Mask-RCNN [15] detector in section 2.3.3. For a fair comparison, we retrain this baseline model on our dataset. In the evaluation metrics, 2D-KGNN refers to the output after the decoder layer and 3D-KGNN refers to the projections of predicted 3D keypoints onto the image.

2.3.1 Datasets

Car-render Self-occlusion dataset: We use the 472 cars sampled from shapenet [60] and 3D annotated by [27]. We select 12 keypoints from the annotated 36 keypoints and render them from different viewpoints. The viewpoints are randomly selected on a level 5 Icosahedron, at varying focal lengths and distances from the object. We use 300 synthetic CAD models for training, 72 for validation and 100 for testing. We project the 3D keypoint annotations of the CAD model with visibility. we trace a ray toward the object from a pixel and check if the first intersection is close to the ground truth location to determine visibility.

CarFusion dataset: To model a wide range of real occlusions, we collect an extensive dataset captured simultaneously by multiple mobile cameras at 60fps at 5 crowded traffic intersections (extending previous work [33]). This extended dataset consists of 2.5 million images out of which 53000 images were sampled at uniform intervals from each video sequence. Approximately, 100000 cars detected in these images were annotated with 12 keypoints each. Each annotation contains the visible and occluded keypoint locations on the car. We do not use the occluded keypoints for training the Occlusion-Net. We selected four annotated intersections to train the network while using one intersection to test it, which split the annotation data into 36000 images for training and 17000 for testing. We further compute 90-10 train validation split on the

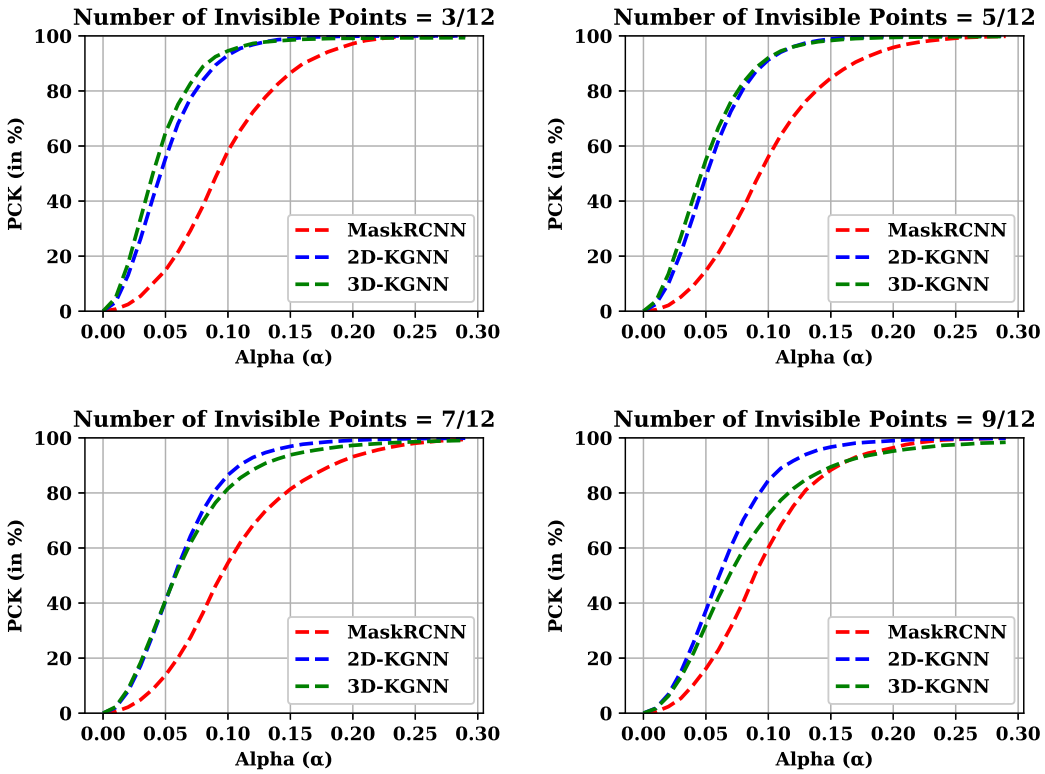


Figure 2.5: Accuracy with respect to different alpha values of PCK for the Car-render dataset. Graph based methods (2D/3D) outperform the MaskRCNN trained keypoints for all the occlusion types. Specifically at alpha=0.1 we observe an increase of 22% for cases with 3 invisible points and 10% in case of 9 invisible points (out of 12 keypoints).

training data to validate our training algorithm. The dataset was completely captured "in the wild" and contains numerous types and severity of occlusions.

Preprocessing: Computing the trifocal loss requires the virtual camera poses in the object frame. For every image, the virtual pose is estimated by solving a PnP [61] between the visible keypoints and the 3D points computed from [33].

2.3.2 Quantitative Evaluation

We compare our approach with other state-of-the-art keypoint detection networks. We use the PCK metric [62] to analyze both the 2D and the 3D occluded keypoint locations. 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 of length and width of the bounding box and $0 < \alpha < 1$. To evaluate the 3D reconstruction, we project the reconstructed keypoints into their respective views and compute the 2D PCK error.

Occlusion Prediction: We demonstrate that the confidence scores computed using MaskRCNN is insufficient to predict occlusions. The left image in Fig 2.3 shows the distributions of confi-

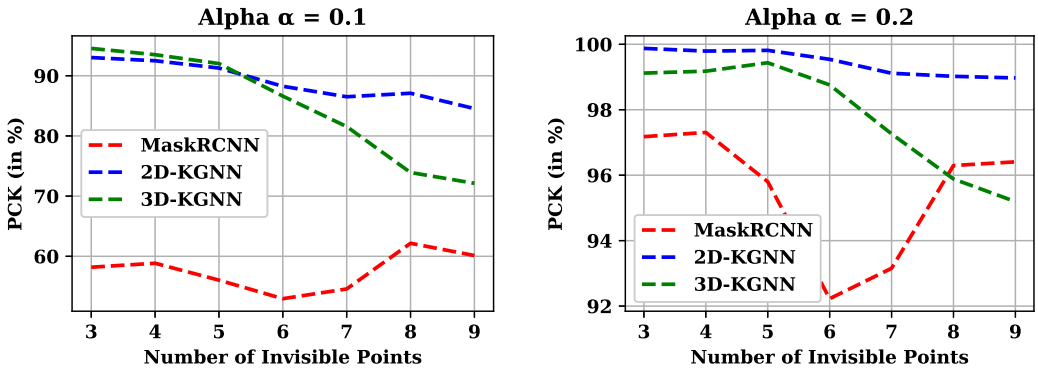


Figure 2.6: Accuracy plots with varying number of occluded keypoints on the Car-render dataset. Graph based methods (2D/3D) outperform the baseline (in red) in the case of $\alpha = 0.1$. For a more conservative alpha, the performances are comparable. The 2D KGNN plots in both the alpha scenarios have a variance of 5% and are robust to occlusion, compared to the 3D KGNN plot (15%) and the baseline MaskRCNN plot (25%).

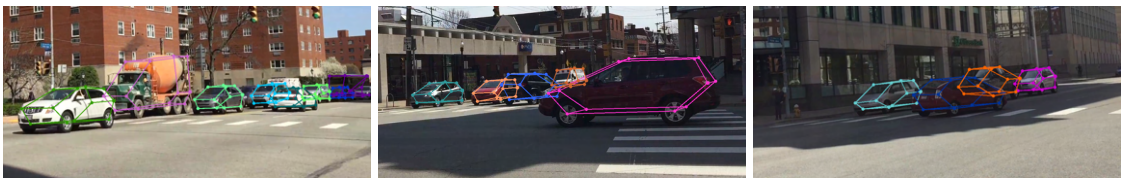


Figure 2.7: Example results of occlusion-net on sample images of the CarFusion dataset. We accurately localize occluded keypoints under a variety of severe occlusions. See supplementary for additional results. Different colors depict different vehicles in the scene.

dence scores of occluded and visible points. These distributions overlap significantly making it hard to distinguish occluded points from visible points. In contrast, by modeling a graph network to exploit relative locations of the keypoints, we observe a significant boost in the accuracy of occlusion prediction as seen from the right image in figure 2.3. We observe an AUC of 0.83 with MaskRCNN, whereas 2D-KGNN gives an AUC of 0.95.

Robustness Analysis: We analyze the effect of adding error to input locations of the graph to analyze the robustness of the learned model. Figure 2.10 shows the accuracy with respect to different Gaussian error added to the input graph. We observe that 3D-KGNN is more stable with increasing error while 2D-KGNN performs well for highly occluded points but falls steeply with increasing error in input.

Evaluations of visible points: We show evaluation of our network with respect to existing visible keypoint estimation methods. Both 3D-KITTI[64] and PASCAL3D+ [65] datasets have annotations only for *visible keypoints* and do not contain occluded point annotations or multiple views to directly evaluate our method. The 2D keypoint predictions in [64] are evaluated only on visible keypoints and the 3D model is evaluated by fitting only visible keypoints on objects that are not truncated or occluded by other objects ("Full" in their table). Our model has *not* been trained on either of these datasets or the CAD dataset from [64]. Table 4.2 compares our method against those on the annotated 2D *visible* points in 3D-KITTI. Table 4.2 also shows the evaluation against the ground truth 3D model for the "Full" (unoccluded) case - the only case

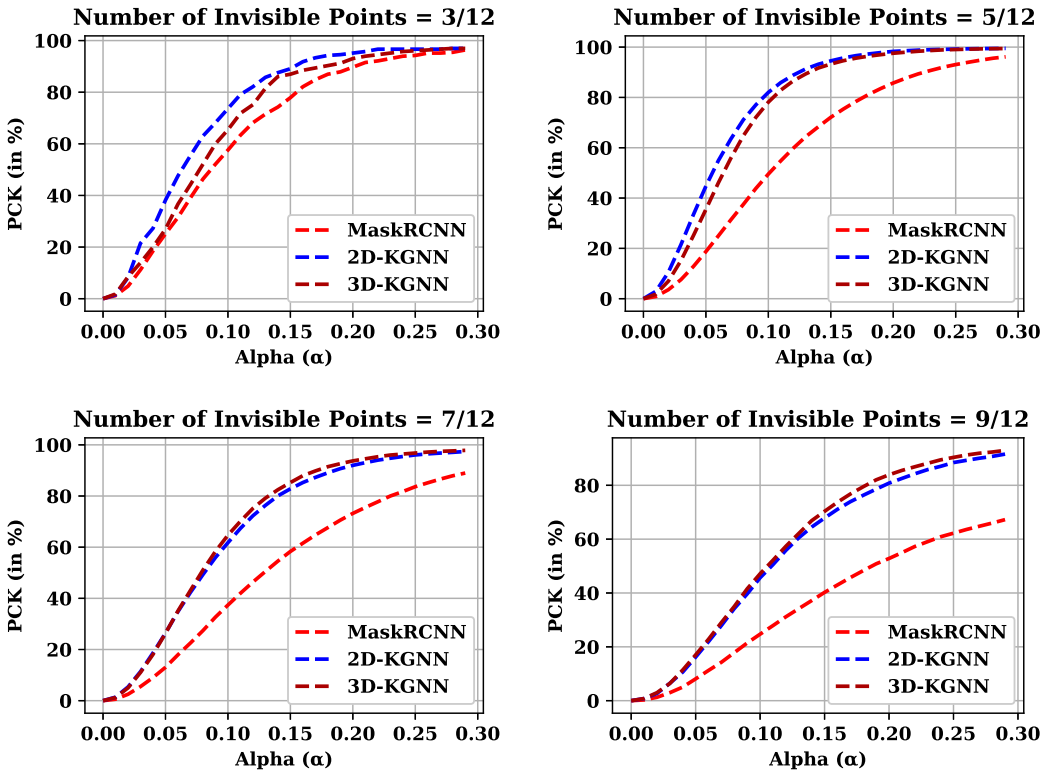


Figure 2.8: Accuracy vs Alpha on the CarFusion dataset. Focusing on Alpha=0.1 across the plots, graph based methods show an improvement of 6% for cases where only 3 (out of 12) points are occluded and nearly 10% or more improvement for more severe occlusion, justifying the usage of graph networks for occlusion modeling.

mentioned in [64]. We observe that our approach outperforms the other methods for two categories .i.e. Truncation and oth-Occlusion. This can be attributed to the fact that our dataset models a range of occlusion types and severity.

Importance of 3D-KGNN: The 3D pose computed is useful for traffic analysis (speed, flow) and understanding/visualizing activity at busy city intersections. 3D-KGNN can also be used to find correspondence across views for multi-view reconstruction, especially when there are very few views available and the keypoints may be occluded. Figure 2.4 demonstrates that 3D-KGNN finds significantly more inliers for multiview correspondence compared to 2D-KGNN or MaskR-CNN.

Human Annotation vs Geometric Prediction: The CarFusion dataset has annotated keypoints for occluded points as well as the visible points across multiple views. Thus, as an interesting aside, we evaluate the accuracy of hand-labeled occluded points with respect to those obtained using the trifocal tensor, as shown in Figure 2.4. We observe that at $\alpha = 0.1$, nearly 90% of the hand-labeled keypoints lie within the region of the geometrically consistent keypoints.

Accuracy Analysis: Figure 2.5 depicts the change in accuracy with respect to Alpha on Car-

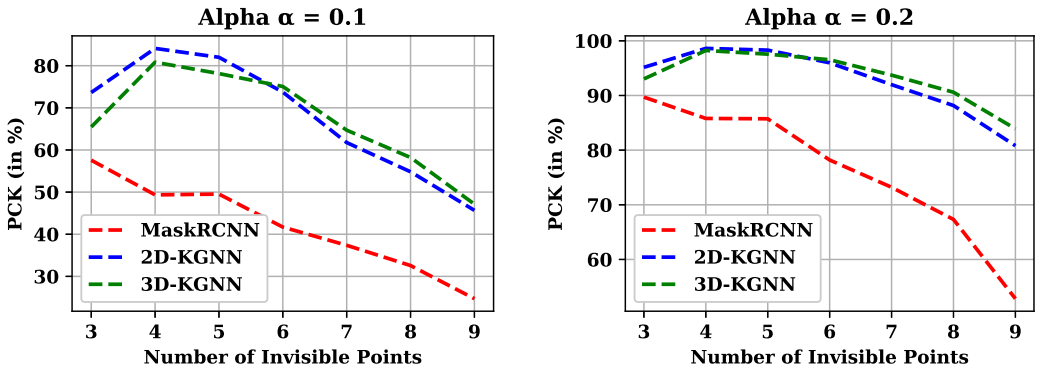


Figure 2.9: Accuracy analysis with varying occlusion configurations. Notice for occlusions with 4 (out of 12) visible points, our approach is nearly 25% higher compared to the baseline for occluded points.

Method	2D					3D	yaw(Error)
	Full	Truncation	Car-Occ	Oth-Occ	All	Full	Full
[63]	88.0	76.0	81.0	82.7	82.0	NA	
[31]	73.6	NA				73.5	7.3
[2]	**93.1**	78.5	**82.9**	85.3	85.0	**95.3**	2.2
Ours	89.73	**87.41**	81.68	**86.45**	**88.8**	93.2	**1.9**

Table 2.1: PCK Evaluation[α=0.1] and comparison of Occlusion-Net on 2D *visible* keypoints annotated in KITTI-3D. Full denotes unoccluded cars, Truncation denotes cars not fully contained in the image, Car-Occ denotes cars occluded by cars, and Oth-Occ denotes cars occluded by other objects. All represents combining the statistics for all the occlusion categories. Our method outperforms in most of the occlusion categories. The 3D keypoint localization (last two columns) in [2] is only evaluated on Full.

render dataset. We show four different plots with different occlusion configuration, ranging from 3 (very less occluded) to 9 (highly occluded) invisible points out of 12 keypoints in total. We observe that our method outperforms the baseline method in all configurations for occluded keypoints. At α=0.1 we observe a boost of 22% for 3 invisible points and 10% for 9 invisible points. Figure 2.6 shows the change in accuracy with respect number of occlusions for Car-render dataset. We plot the graph for two different value of α and observe that 2D graph method is more stable with increasing occlusion compared to the 3D-KGNN. We show similar accuracy vs. alpha plots on CarFusion dataset in Figure 2.8. We observe that with increasing occlusions our method shows higher accuracy improvement compared to the baseline MaskRCNN. At $\alpha = 0.1$ we nearly gain a boost of at least 6% in all the occlusion categories and nearly 12% boost for 5 occluded points. Figure 2.9 depicts the change in accuracy with increasing number of occluded points on CarFusion dataset. For the case of 4 invisible points configuration, our approach is nearly 25% higher compared to the baseline. To conclude we observe that the accuracy of KGNN on occluded points is higher than using the baseline method.

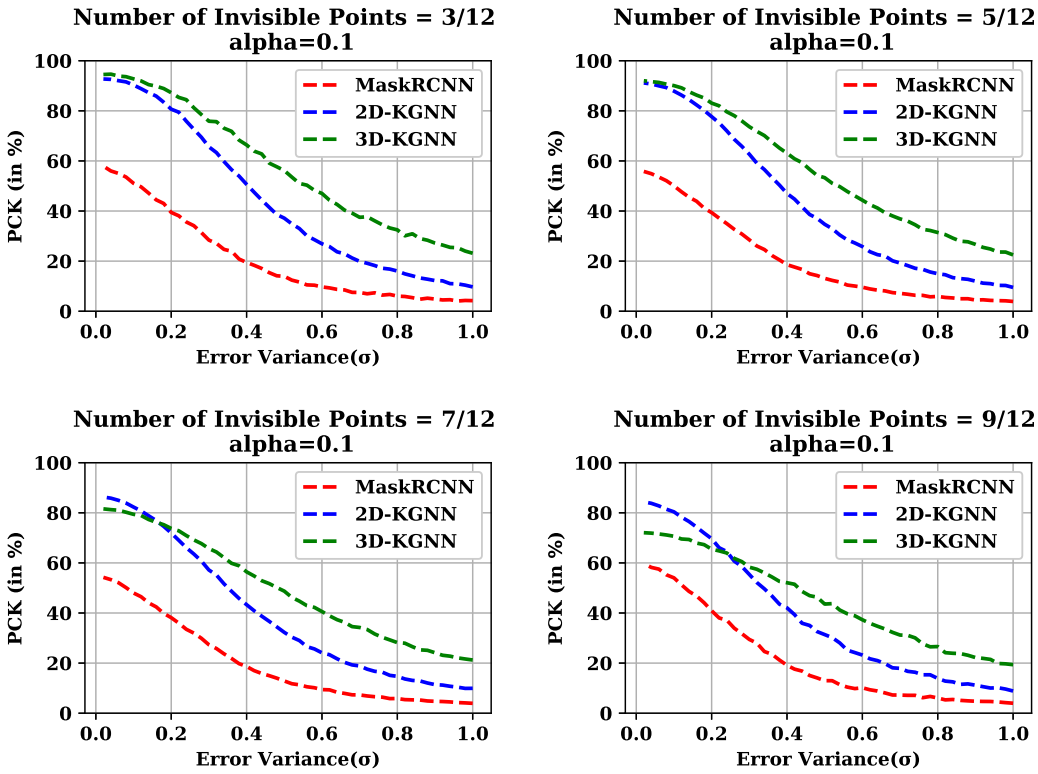


Figure 2.10: The plots depict the change in accuracy for the methods when Gaussian noise is added to the input keypoints. As expected, 3D-KGNN (green) performs much better in the presence of strong noise.

2.3.3 Qualitative Evaluation

In this section, we analyze the visual improvements of our method across different categories of occlusion. Figure 4.11 depicts the visual results of the algorithm in different occlusion situations. We demonstrate results on four occlusion types namely, self-occlusion, vehicle occluding car, other objects occluding car, and truncation where the car is partially visible. The first column depicts the output from the MaskRCNN keypoints. The color is coded blue because the output from heatmaps does not give statistics about the occlusion categories of the keypoints. The other column show ablation results on our approach. The results demonstrate that predicting occluded keypoints as a heatmap generate large errors in localization while learning a graph based latent space improves the location of the occluded keypoints with respect to the visible points. Specifically, in high occlusion scenarios, graph-based methods show large improvement visually compared to MaskRCNN. We further show the results of our method on multiple cars simultaneously in Figure 2.7. Our method performs accurate occluded keypoint localization on very challenging occluded cars.

2.4 Conclusion

We presented a novel graph based architecture to predict the 2D and 3D locations of occluded keypoints. Since supervision for 2D occluded keypoints is challenging, we computed the error using labeled visible keypoints from different views. We proposed a self-supervised network to lift the 3D structure of the keypoints from the 2D keypoints. We demonstrated our approach on synthetic CAD data as well as a large image set capturing vehicles at many busy city intersections and improve localization accuracy (about 10%) with respect to the baseline detection algorithm.

Limitations The algorithm need multi-view data as supervision for learning occluded keypoints is the main bottleneck of the supervision. In the network architecture, we disentangle the image feature from the graph network making us loose the gradient from the image to the occluded keypoints in future versions using methods that can transform the heatmap to key-point coordinates in a differentiable manner.

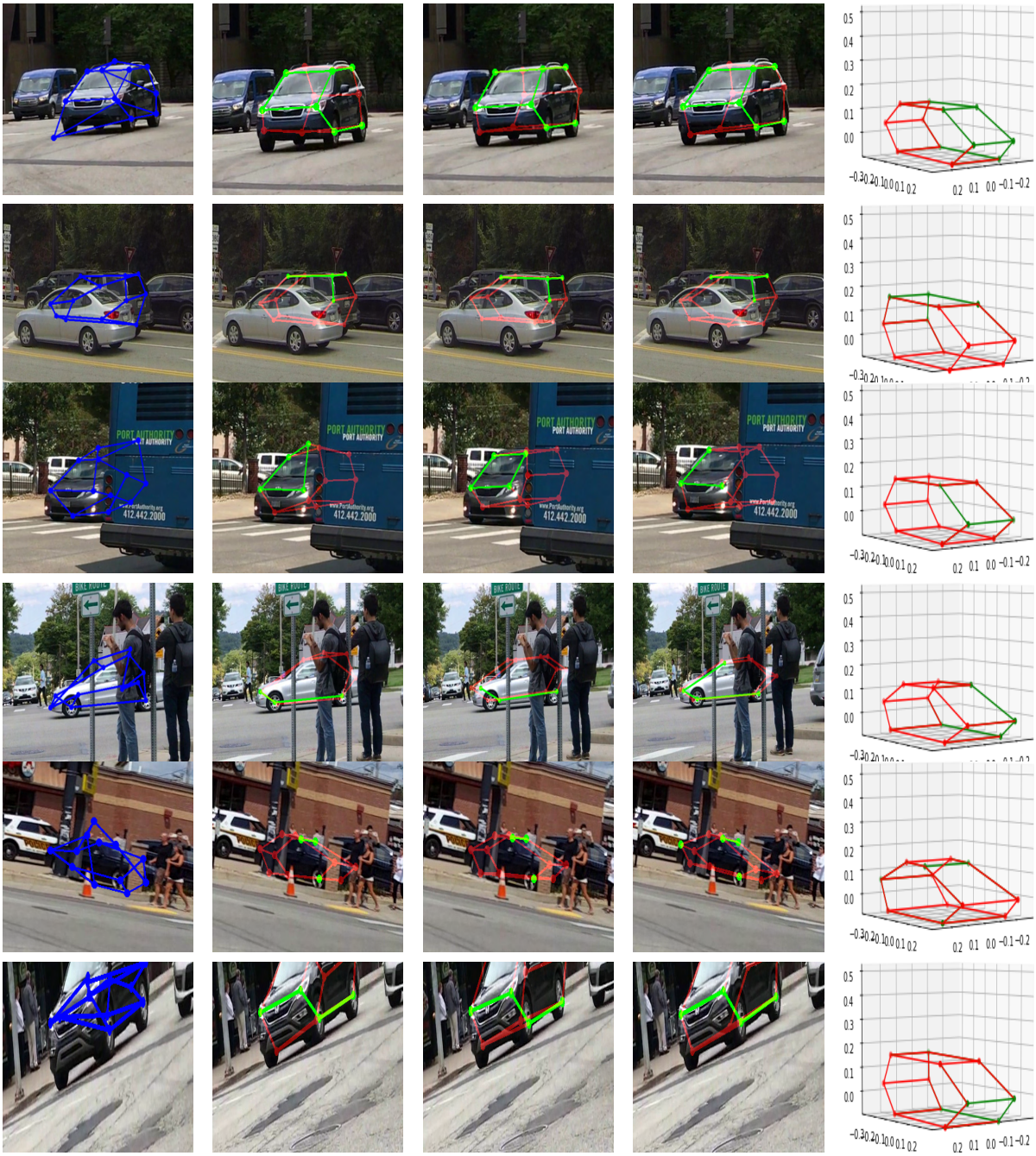


Figure 2.11: Qualitative evaluation of the 2D/3D keypoint localization for different occlusion categories of cars from the CarFusion dataset. The initial detector was trained using the MaskRCNN on the visible 2D keypoints. We use our self-supervised 2D-KGNN and 3D-GNN to localize keypoints from a single view. 2D reprojections of the 3D keypoints are shown in third column. The second and third columns show clear improvement in the localization of the occluded keypoints with respect to the baseline MaskRCNN. The canonical 3D views computed using 3D-KGNN are shown in the last column. The ground truth is obtained by applying trifocal tensor on the human labeled visible points to estimate the invisible points. Green represents visible edges and red represents occluded edges.

Chapter 3

Supervision For Temporal Occlusions

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 [66, 67] 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 [68] or Argoverse [69] 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 [70, 71, 72, 73, 74], they focus on short intervals such as over a few frames or 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 [15] 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. 6.1. We refer to this process as **longitudinal self-supervision**. Our main contributions are summarized below and the entire framework is shown in Fig. 6.2:

(a) Joint optimization for longitudinal reconstruction (Sec 3.2.2): 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

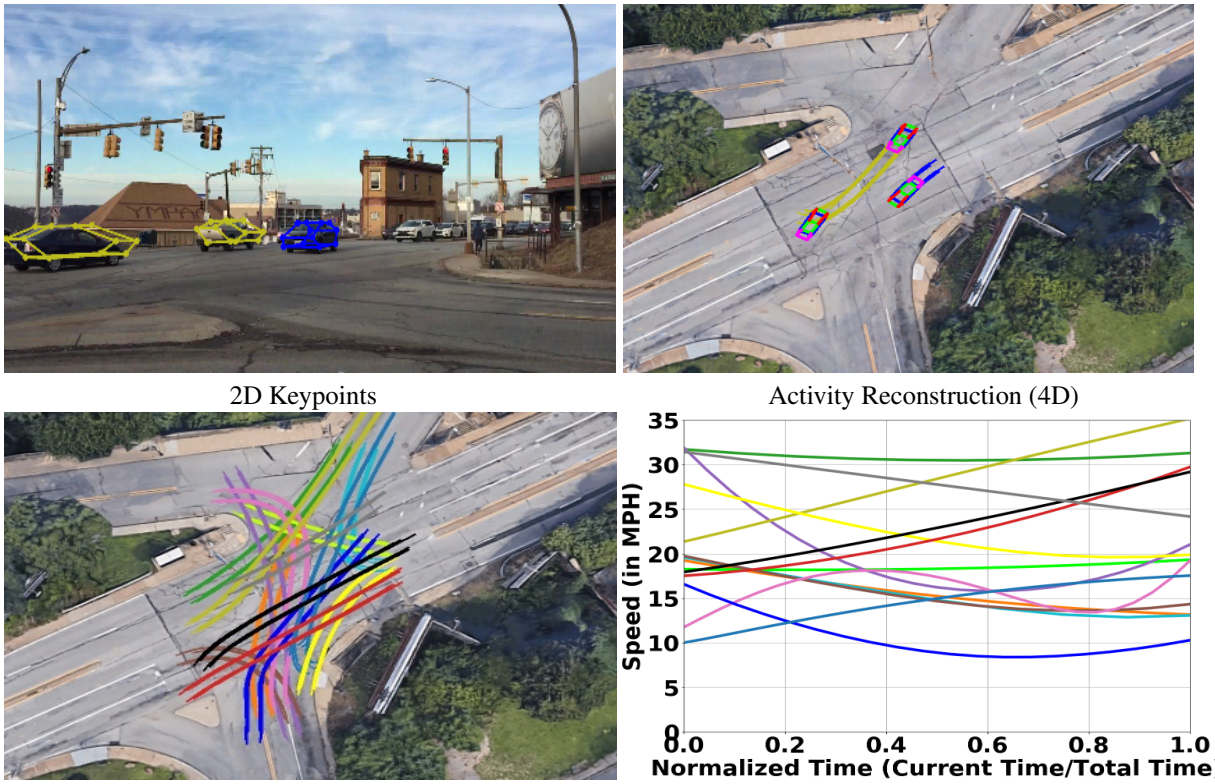


Figure 3.1: Long term repetitious vehicular activity is used as self-supervision to compute accurate 2D and 3D keypoints, trajectories and velocities from a single fixed camera. Reconstruction accuracy improves significantly over 20 minutes at this intersection as compared to methods that enforce consistency over short periods (a few frames to seconds).

art [3].

(b) Scene-specific repetitious activity clustering (Sec 3.2.3): 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 3.2.4): 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 [75]. 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**.

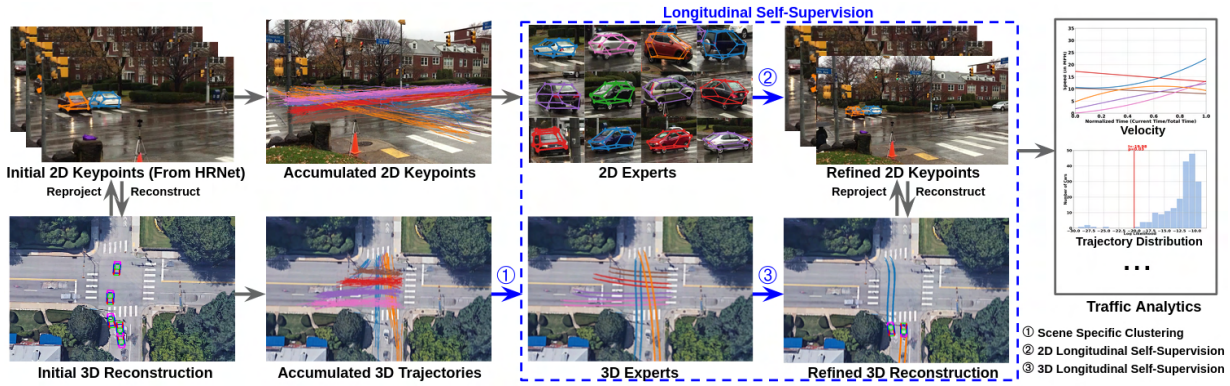


Figure 3.2: Framework for self-supervised 4D reconstruction of repetitious activity. Our method takes off-the-shelf 2D keypoint detections as input, reconstructs 3D keypoints with an active shape model, fits an analytic trajectory model to each vehicle's 3D poses along with frames, and accumulates them 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.

3.1 Related Work

Single View Reconstruction: Many methods utilize Lidar [**?** **?**], IMU [**?**], UAV [76] to acquire 3D information, or deploy deep networks to infer 3D geometric properties from RGB images, largely in a supervised manner [66, 77, 78, 79]. 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 [3, 67, 80, 81] build active shape models to optimize/retrieve 3D shapes from 2D images. Recent works [3, 80, 82] 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 [83, 84] and clustering [85, 86]. Specific to modeling repetitious activity, [6] 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 [87, 88, 89] and style consistency [90, 91]. In addition, many works [92, 93, 94] utilize alignment between frames to learn optical flow; [71, 95] 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 chapter 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.

3.2 Self-Supervised 4D Reconstruction

3.2.1 Background

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

Notation used in the chapter: We use three coordinate systems, i.e. camera, world and map coordinates as shown in Fig. 3.3. 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. 3.3). 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 [81] 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 [27]. 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 $\boldsymbol{\alpha}_n$ 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 $\boldsymbol{\alpha}_n$ 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

$$\hat{t}^{(w)}_{n,x}(t) = a_h t^h + ... + a_2 t^2 + a_1 t + a_0 \tag{3.1}$$

$$\hat{t}^{(w)}_{n,y}(t) = b_h t^h + ... + b_2 t^2 + b_1 t + b_0 \tag{3.2}$$

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

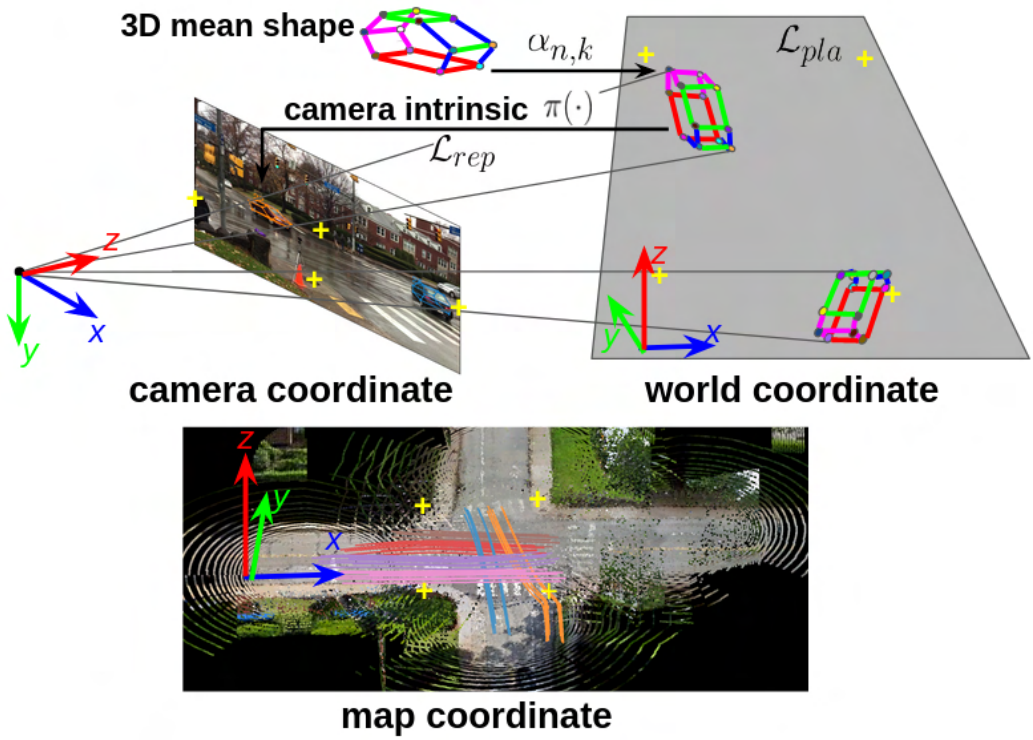


Figure 3.3: 3D reconstruction coordinate frames. Vehicle 3D keypoints are computed in camera coordinates. The world coordinate is defined with XY as the ground plane, in which we perform analytic model fitting and repetitious activity clustering. Map coordinates are defined based on Google maps, whose XY plane is also the ground. This is used to estimate real-world location and speed. Yellow cross landmarks transform world to map coordinates.

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 3.2.2) 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.

$$\mathcal{L}_{fit,n} = \sum_m \Bigg(\left\|\hat{\mathbf{t}}^{(w)}_{n,m} - \mathbf{t}^{(w)}_{n,m}\right\|^2 + \beta_1 \left\|\mathbf{u}(\nabla\hat{\mathbf{t}}^{(w)}_{n,m}) - \mathbf{u}(\mathbf{r}^{(w)}_{n,m})\right\|^2 + \beta_2 a_3^2 + \beta_3 b_3^2 \Bigg) \tag{3.3}$$

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

In this section, we explain our approach to utilize longitudinal consistency in repetitious vehicular activity for accurate 4D reconstruction. Fig. 6.2 shows the overall pipeline with the three stages described below.

3.2.2 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) [3, 81]. 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 [96] 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 [97]. 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 $\boldsymbol{\alpha}_n$ with object pose $(\mathbf{r}^{(c)}_{n,m}, \mathbf{t}^{(c)}_{n,m})$ as:

$$\mathbf{P}^{(c)}_{n,m} = \mathbf{R}^{(c)}_{n,m}(\bar{\mathbf{Q}} + \sum_{k=1}^{K} \alpha_{n,k}\mathbf{Q}_k) + \mathbf{t}^{(c)}_{n,m} \tag{3.4}$$

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

$$\mathcal{L}_{rep} = \sum_{n,m,j} \|\boldsymbol{\pi}(\mathbf{P}^{(c)}_{n,m,j}) - \mathbf{p}_{n,m,j}\|^2 \tag{3.5}$$

(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 $\boldsymbol{\eta}^{(c)}$.

$$\mathcal{L}_{pla} = \sum_{n,m} \frac{(\eta_1 P^{(c)}_{n,m,j_b,x} + \eta_2 P^{(c)}_{n,m,j_b,y} + \eta_3 P^{(c)}_{n,m,j_b,z} + \eta_4)^2}{\eta_1^2 + \eta_2^2 + \eta_3^2} \tag{3.6}$$

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

3.2.3 Scene-Specific Repetitious Activity Clustering

Capturing repetitious motion patterns over a long duration plays an important role in deciphering higher level semantics of the environment. We observe and demonstrate using experiments that such higher order semantics are much more distinguishable in 3D compared to 2D [4, 6]. Thus, we first fit a polynomial model to each object's 3D poses to suppress noise and reduce data dimension as described in Section 3.2.1. 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

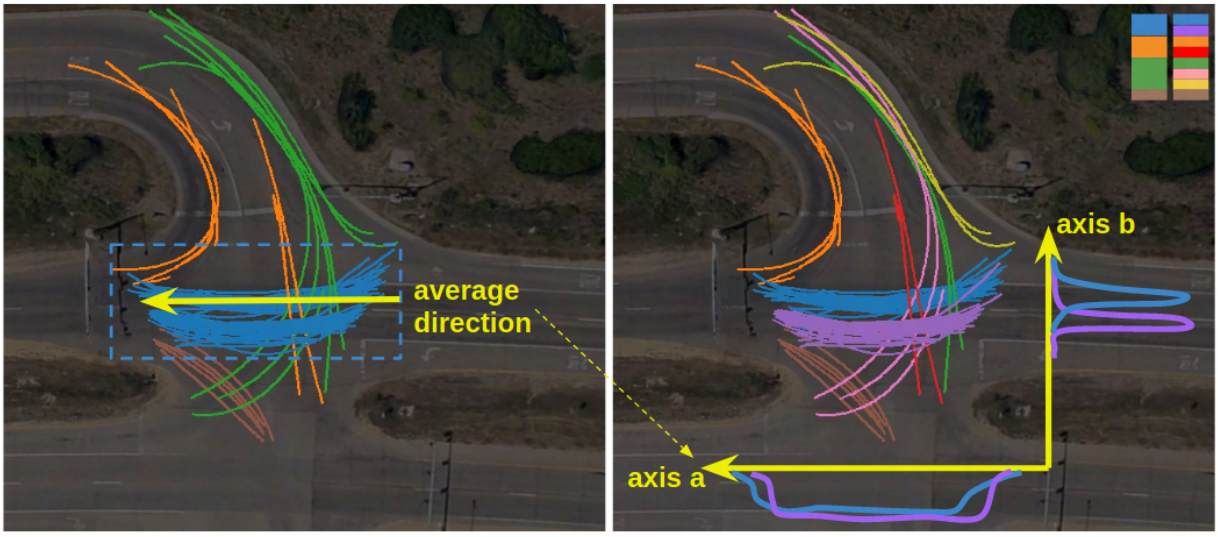


Figure 3.4: Demonstration of our hierarchical clustering in birds-eye view. **Left:** First stage clusters and the average direction of the blue cluster. **Right:** Second stage clustering. Trajectories are projected along their average direction, maximizing the spatial difference between near clusters. 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**.

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. 3.4. We face two challenges here: (a) vehicles on different lanes can be close to each other (see blue and purple lines in Fig. 3.4) 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. 3.4), 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 $\Delta\mathbf{t}^w_{n,\min t_n,\max t_n} = \mathbf{t}^w_{n,\max t_n} - \mathbf{t}^w_{n,\min t_n}$, 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:

$$\hat{t}'_n(t) = \frac{\hat{t}^{(w)}_{n,x}(t)p_{i,x} + \hat{t}^{(w)}_{n,y}(t)p_{i,y}}{\|\mathbf{p}_i\|} \tag{3.7}$$

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. 3.4, axis **a** is the average direction. Blue and purple trajectories are projected along axis **a** to axis **b**. We notice the overlapping between the two lanes is mostly eliminated, so they become easily distinguishable. Our method is unsupervised and takes scene-specific information (say, the geometry of traffic lanes) into account to maximize the separation between similar clusters (lanes). For each fine-grained cluster, we then save the average of the parameters of all trajectories.

3.2.4 2D and 3D Longitudinal Self-Supervision

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

2D Longitudinal Self-Supervision: Learning-based detectors produce precise as well as erroneous keypoints. We would like to use the accurate detections to improve the badly localized keypoints. We distinguish the good ones from the erroneous by using a threshold δ_{rep} on the re-projection 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 $\mathbf{s}^{(b)}_{n,m}$ and translation $\mathbf{t}^{(b)}_{n,m}$ from the 2D expert bounding box $\hat{\mathbf{b}}_{n,m}$ to the instance bounding box $\mathbf{b}_{n,m} = \mathbf{s}^{(b)}_{n,m}\hat{\mathbf{b}}_{n,m} + \mathbf{t}^{(b)}_{n,m}$. Then the optimized transformations $\mathbf{s}^{(b)}_{n,m}$, $\mathbf{t}^{(b)}_{n,m}$ 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 $\mathbf{t}^{(c)}_{n,m}$, $\mathbf{r}^{(c)}_{n,m}$ in frame m and the 3D expert of one specific cluster, we find a point $\hat{\mathbf{t}}^{(c)}_{n,m}$ on the 3D expert minimizing its distance to the object position $\mathbf{t}^{(c)}_{n,m}$. 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: $d_{n,cham} = \sum_m \|\mathbf{t}^{(c)}_{n,m} - \hat{\mathbf{t}}^{(c)}_{n,m}\|$. 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 $\mathbf{t}^{(c)}_{n,m}$, $\mathbf{r}^{(c)}_{n,m}$ in frame m, we find its closest point $\hat{\mathbf{t}}^{(c)}_{n,m}$ on the 3D expert when calculating Chamfer distance. $\hat{\mathbf{r}}^{(c)}_{n,m}$ is the tangent direction of the 3D expert at $\hat{\mathbf{t}}^{(c)}_{n,m}$.

We propose the 3D longitudinal loss to learn correspondence between individual pose and 3D experts by minimizing

$$\mathcal{L}_{long} = \beta_4 \|\mathbf{t}_{n,m}^{(c)} - \hat{\mathbf{t}}_{n,m}^{(c)}\|^2 + \beta_5 \|\mathbf{r}_{n,m}^{(c)} - \hat{\mathbf{r}}_{n,m}^{(c)}\|^2 \tag{3.8}$$

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

3.3 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 [75] 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.

3.3.1 Datasets

TRAFFIC4D Dataset: This is a novel dataset proposed in the chapter 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 [75] 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.

3.3.2 Evaluation Metrics and Baseline Methods

CarFusion dataset [98] is used to pretrain our 2D keypoint detector [96]. 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.

We analyze the accuracy of our reconstruction by using metrics both in 2D and 3D. We use 3D-PCK (Percentage of Correct Keypoints) [62] between our 3D reconstructed keypoints and 3D ground truth keypoints for evaluating the reconstruction. We further evaluate the reconstruction

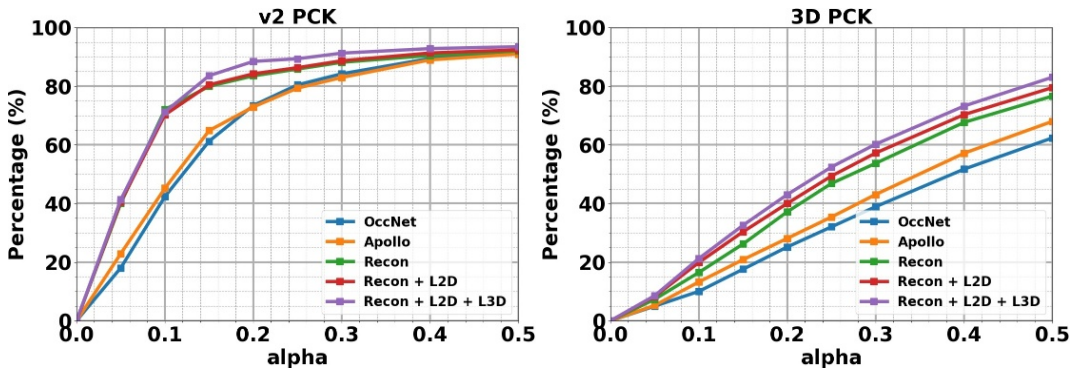


Figure 3.5: Accuracy of reconstruction with respect to varying window size (α) on TRAFFIC4D stereo pairs. **Left** and **right** are keypoints projected to the second view of stereo and reconstructed in 3D respectively. "Recon" indicates using our joint optimization for reconstruction. Note that longitudinal 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.

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 [3]. 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 [3], 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 [3] and Occnet [81]. 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 [6], MS [4], MBMS [5]. 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 3.2.2 to all the algorithms. We report the proportion of correctly clustered trajectories metric to evaluate our method as proposed in [6].

3.3.3 Accuracy Analysis

Reconstruction Analysis: Fig. 3.5 compares reconstruction on the stereo pairs of TRAFFIC4D. We observe higher PCK accuracy compared to [81] and [3] in 2D and 3D. Specifically, when no longitudinal self-supervision is used, our second view (v2) and 3D PCK are significantly higher than the others, indicating our reconstruction is more consistent in 3D. We emphasize that the global co-planar loss contributes to the improvement in reconstruction accuracy as it regularizes all the vehicles' poses in the video for better spatial consistency. Moreover, our method achieves better accuracy after 2D and 3D longitudinal self-supervision.

Figure 3.6: 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. 3.6 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 **??** shows improvement on A3DP for our method compared to baselines on S01 and S02 sets of AI City dataset. Similar to Fig. 3.5, adding 2D and 3D longitudinal self-supervision improves A3DP as well.

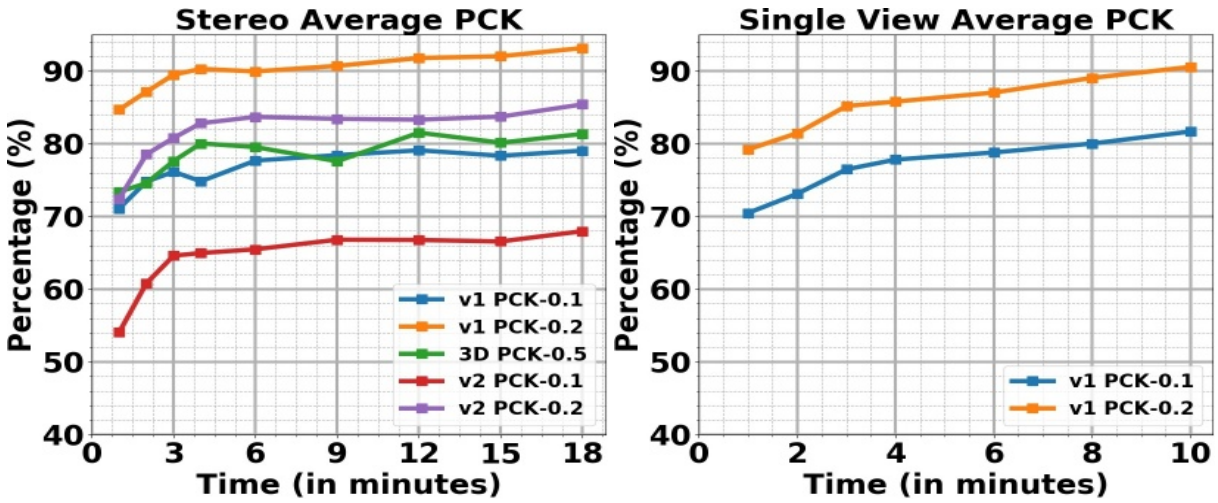


Figure 3.7: The plot depicts PCK-α accuracy improving over time by using longitudinal self-supervision. We observe 11% absolute and 16% relative improvement in average accuracy of 3D reconstruction and detections over stereo cameras (**left**) in TRAFFIC4D dataset with 18 minutes of continuous learning. 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.

			S01			S02		
Method	L2D	L3D	A3DP-Rel			A3DP-Rel		
			mean(in %)	c-l(in %)	c-s(in %)	mean(in %)	c-l(in %)	c-s(in %)
OccNet [81]			9.30	45.44	8.90	12.21	51.54	6.98
Apollo [3]			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 3.1: Comparing to state of the art trajectory reconstruction methods on AI City dataset using A3DP metric. "Mean", "c-l", and "c-s" denote mean, loose and strict criteria with different thresholds relative ("Rel") to depth [3]. Traffic4D shows an average improvement of 14.62%(in absolute terms) and 34.2% (in relative terms) compared to [3] on both sequences, without any manual supervision.

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. 3.7 left illustrates the effect on reconstruction accuracy for varying sub-sequence length on TRAFFIC4D dataset stereo cameras. We observe a clear increase in accuracy with an increase in sub-sequence length illustrating that longitudinal supervision enhances the reconstruction accuracy. The accuracy converges after a specific duration of time emphasizing that the activity clustering for the sequence has been learned. We observe similar improvements in PCK accuracy on single view cameras as shown on the right in Fig. 3.7.

Repetitious Activity Clustering Analysis: Table **??** 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

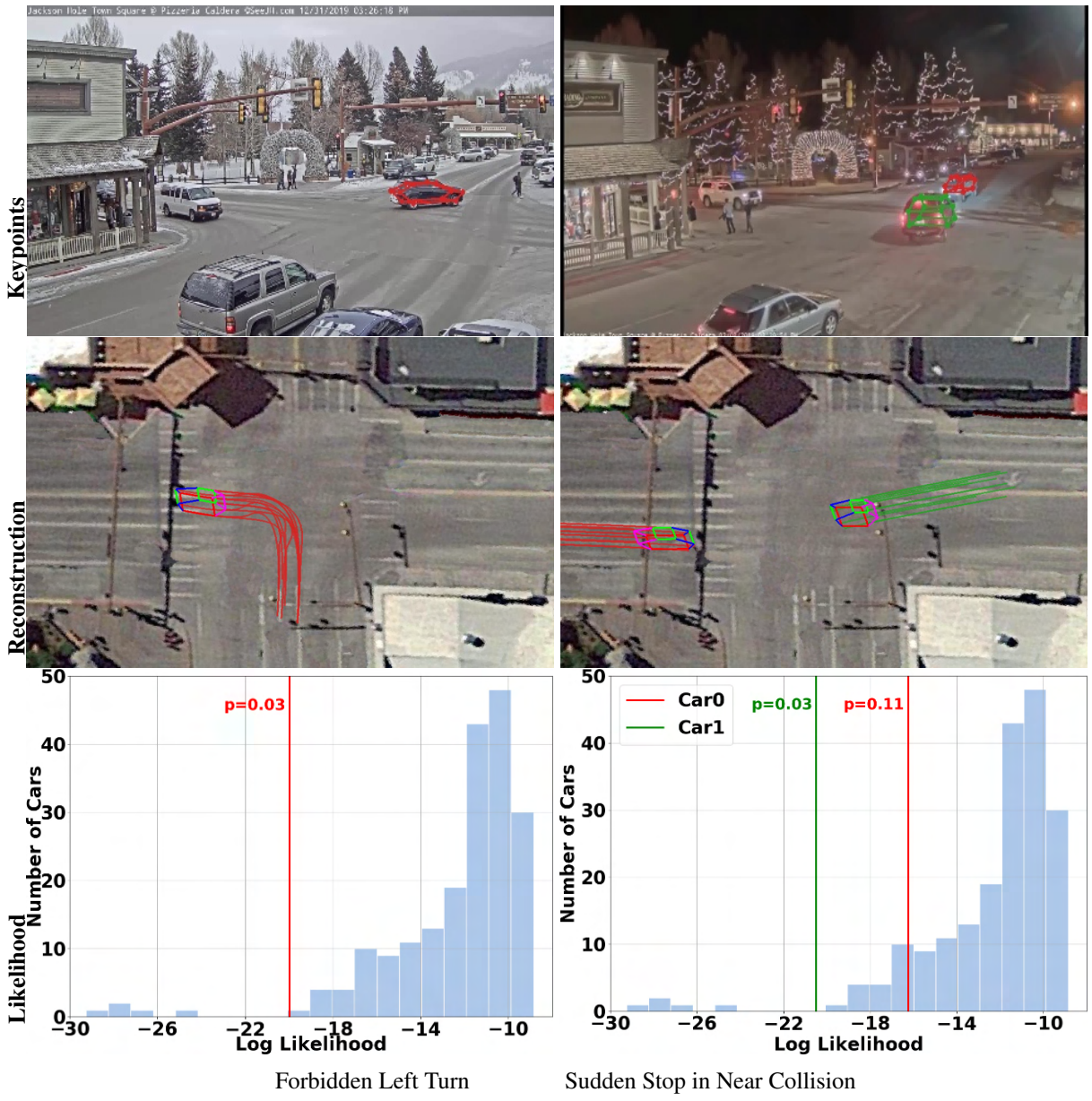


Figure 3.8: Automatic anomaly detection. The plot shows different anomalies like vehicles making forbidden left turn **(Left column)**, sudden stop in near collision **(Right column)** using our method. **Last row** shows the anomaly's log likelihood (red/green lines, p represents the probability) is much lower than the normal trajectories (blue bars) in the cluster.

they are distant or the camera looks straight forward, while 3D clustering eliminates the view angle and perspective effect by converting them to 3D.

3.3.4 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

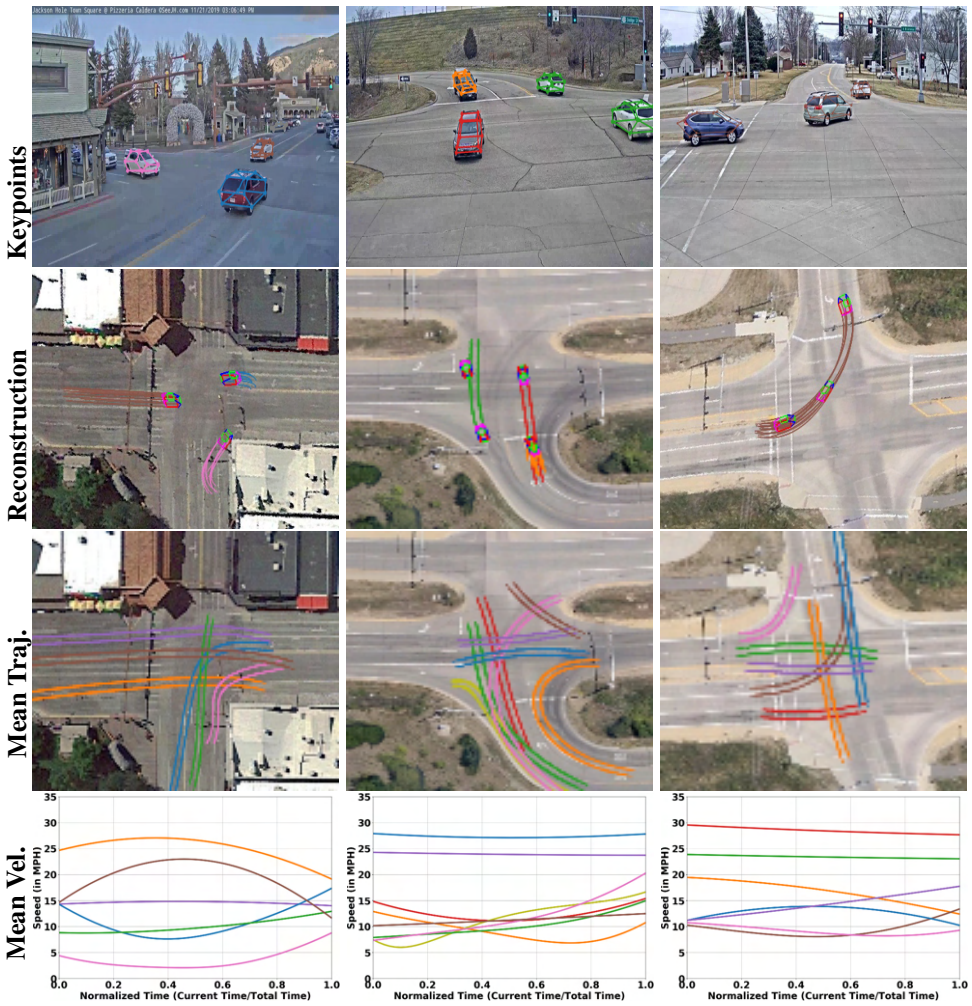


Figure 3.9: The keypoints **(first row)** and 3D reconstructions overlaid on Google map **(second row)** at different times, as well as 3D mean trajectories **(third row)** and velocities of the mean trajectories **(fourth row)** for three intersections. These mean trajectories represents typical vehicle motions and are used for 3D longitudinal self-supervision.

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. 3.9 shows the accurate reconstruction results of individual vehicles, 3D mean trajectories and speed profile after longitudinal self-supervision.

(2) Anomaly analysis: As an application of our model, vehicular anomalies can be identified. The log likelihood of a trajectory belonging to a specific cluster is obtained by sampling from the corresponding Gaussian component in the clustering model. The trajectory is considered as an anomaly if its likelihoods are lower than a threshold in all the clusters. Compared to previous anomaly detection methods purely in 2D, the 3D anomaly trajectory also reveals the anomaly vehicle's position and velocity in 3D real world. Fig. 3.8 shows the trajectories and likelihood of anomalies.

(3) Vehicle counting: The number of vehicles in each direction and lane is counted based on

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**

Table 3.2: Comparing the accuracy of TRAFFIC4D clustering algorithm with previous clustering methods MS [4], MBMS [5], AMKS [6]. 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).

cluster ids. The supplementary video and webpage show the results.

3.4 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.

Limitations: The algorithm only works with a single view camera data captured over significant duration of time and is ideal for stationary cameras in the wild. The method is heavily dependent on keypoint detector and needs more research to generalize to other representations like segmentation etc.

Chapter 4

Supervision For Occlusion by Others

While there has been strong progress in data-driven methods for object detection[47, 99, 100, 101], tracking[19, 20, 21, 22], segmentation[15, 102, 103, 104, 105] and reconstruction[10, 23, 24, 106] with limited occlusions, most methods under-perform in severely occluded scenarios. Severe occlusions are common in busy intersections and crowded places. Even in less dense scenes, pedestrians and vehicles often pass each other or pass behind other objects. As a result, objects are either not detected at all, or the 2D bounding boxes and segments are truncated and produce errors in downstream processes such as 3D reconstruction [10, 33, 107, 108, 109, 110].

Much of this state of affairs can be attributed to the fact that occlusions are treated as noise that must be overcome by robust measures [25, 26, 27, 28, 111, 112]. There are several challenges that make this strategy hard to succeed. First, it is much harder to label object bounding boxes or segments that are occluded, even for humans [7, 8, 113]. Thus, even large datasets like COCO[11] and ImageNet[12] have relatively few objects labeled that are severely occluded [7, 8]. This creates a strong bias against learning robustness to occlusions [114, 115, 116]. Further, the evaluation metrics are often reported on the entire datasets [11, 13, 14] that could hide problems in occluded scenarios.

As a result, there is growing recognition that occlusions must be explicitly modeled and learned [105, 105, 117, 118, 119**?**]. This has led to new efforts in labeling occlusions explicitly in multiple datasets [7, 8, 120]. Using such supervision, amodal, or holistic, representations (e.g. segmentations and bounding boxes) of objects are learned from partially occluded observations [121, 122, 123]. While producing significantly better results than before, these commendable efforts are still plagued by the same challenges - difficulty for humans to label occlusions in real scenes and the limited dataset size. To supplement such limited data, focus has turned toward synthesizing objects in occluded scenarios using synthetic inpainting[105, 121**?**] using computer graphics [124, 125, 126]. CG can generate a large amount of data for supervision (given today's cloud computing resources) but even the best renderers [127, 128, 129] leave a notable domain gap to the real data, which needs to be bridged [130, 131].

In this work, we present the best of both the real and synthetic worlds for automatic occlusion supervision using a large source of hitherto unexploited data: time-lapse imagery from stationary cameras observing street intersections over weeks, months, and even years[1]. We exploit this data

[1]In the past decades, much analysis on time-lapse data was conducted for illumination and weather understanding [132][133], object insertion and rendering, from thousands of webcams all over the world citeja-

Figure 4.1: We visualize the prediction of amodal representation of vehicles and people under severe occlusions trained using our longitudinal self-supervision framework. The method shows significant improvement in amodal detection and segmentation with images captured from different cameras.

in a novel way to first mine a large dataset of real *unoccluded* objects over time and then use them to synthesize a large number of occlusion scenarios. We develop a new method to classify unoccluded objects based on the idea that when objects on the same ground plane occlude one another, their bounding boxes overlap in a particular common configuration. Once unoccluded objects are discovered, they are composited in layers back into the same scene. These compositions have artifacts that perhaps do not make them too useful for visualization. But they are close enough to real data to reduce the domain gap for a deep network that explicitly predicts the object, its occluder, and the occluded.

Being patient pays off here. Over time, our method discovers tens of thousands of unoccluded objects at diverse positions, orientations, and appearances due to lighting and weather conditions, even in busy scenes. We speed up this discovery by combining sparse time sampling of the data with burst local tracking. This step reduces the required observation period from many months to several days (images captured every few mins.). The data enables us to analyze the performance of our approach over different durations and confidences of self-supervision. Specifically, we relate the confidence in *unoccluded* object prediction to the rate and accuracy of training *occluded* objects. In the beginning, including lower confidence predictions increases more supervision to speed up training, but is quickly passed by training only on high confidence supervisions.

We introduce a new dataset, Watch and Learn Time-lapse (WALT), consisting of 12 (4K or 1080p) cameras capturing urban environments over a year. The cameras view a diverse set of scenes from traffic intersections to boardwalks. The performances of pedestrian and vehicle detection and segmentation improve significantly on all cameras. Like in [111, 113, 134], we report performances at different levels of occlusion and show that the performance drops more

cobs07amos,baatz2012large, Li-2021-127410.

slowly as occlusion increases, compared to methods that do not use longitudinal self-supervision. Because of this, we achieve strong results in detecting and tracking objects as they pass each other - a common failure mode of existing approaches. The methods we present are simple but provide an effective baseline to inspire future work on exploiting longitudinal supervision for computer vision under strong occlusions.

4.1 Watch and Learn Amodal Representation

We address the problem of layer representation of objects in a scene under severe occlusions. We propose a continuous learning framework to resolve occlusion ambiguities from images. Initially, given a time-lapse stream of data from a stationary camera, we detect and mine all the unoccluded objects over a long duration of time. These unoccluded objects collected over time automatically act as supervision that we term *longitudinal self-supervision*. We follow a clip art-based integration method to place these unoccluded objects within the scene at the same detected location but overlapping with another unoccluded object from the database. This generates many realistic occlusion configurations for training a network to disentangle holistic object segmentation from a cluttered scene. We further show how to speed up the training for learning amodal representations by tracking around unoccluded detections.

4.1.1 Unoccluded Object Mining

We exploit the time-lapse data in a novel way to mine a large dataset of real unoccluded objects over time. We develop a new method to classify unoccluded objects based on the idea that when objects on the same ground plane occlude one another, their bounding boxes overlap in a particular common configuration.

Preprocessing Videos: On the time lapse feed from a camera, we run instance segmentation[101] on each frame. We use Intersection-Over-Union based tracker[135] to track the detected bounding box and segmentation. We represent the detections as $D^{t_0 \ldots t_N}_{m=0,\ldots,M}$, where t_N represents time, while N represents the number of images and m corresponds to the index of the object from a total of M detections.

Occlusion Classification: We locate and segment unoccluded objects in the scene from time lapse video sequences. The unoccluded objects are detected by exploiting overlap between objects detected in an image as shown in Fig 4.2. For every detection D_i at time instance t_j, we compute the occlusion indicator $O(D_i^{t_j})$ using

$$O(D_i^{t_j}) = \begin{cases} 0, & \text{if } D_i^{t_j} \cap D^{t_j} = 0 \text{ or } B(D_i^{t_j}) \cap D^{t_j} < \delta \\ 1, & \text{otherwise.} \end{cases} \tag{4.1}$$

We use two hypotheses to classify the detected objects as occluded or fully visible. The first constraint is that the bounding box should not intersect any other detected objects D^{t_j} from the same time instance. Secondly, for every overlapping bounding box, we disentangle the occluded object and the occluder assuming planar constraints. When both objects are on the same plane, we observe that the bottom of the occluded bounding box always intersects with another bounding

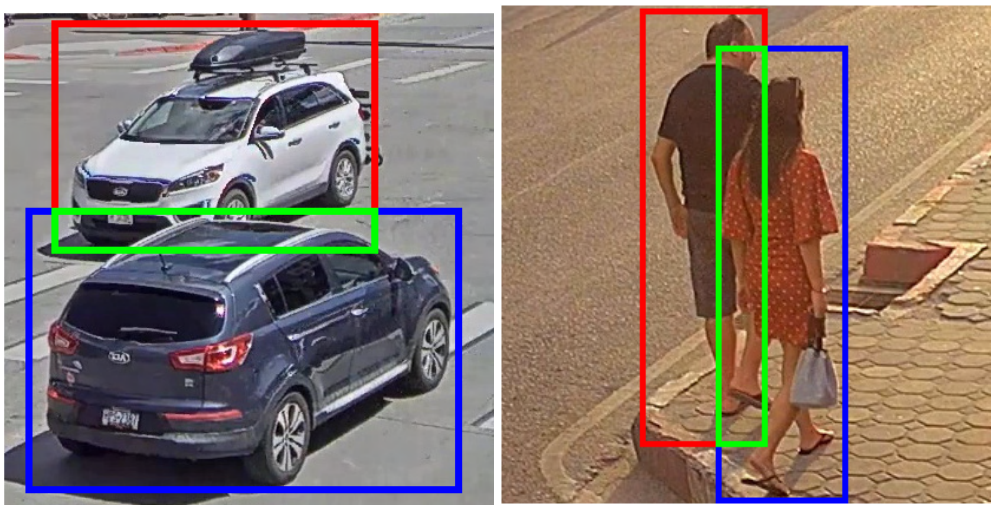


Figure 4.2: Illustrating the region used to classify unoccluded (Blue) and occluded objects (Red) using planar based IOU (Green) for different categories of objects like vehicles and people.

Figure 4.3: We illustrate generated training images(top) from Clip Art WALT dataset. The synthesized Ground-Truth amodal segmentation map(bottom) captures multiple layers(darker represents higher order of occlusion) of occlusions for training. The Clip Art images have realistic occlusions because the inpainting is performed by superimposing the object at the same location as it was observed but from varying time instances.

box from the scene. We exploit this observation and find the intersection of the occluding bounding box with the bottom of the occluded bounding box $B(D_i^{t_j})$. If the intersection is larger than a threshold δ, we classify the object as occluded. This classification is computed iteratively over all the detections $D_{m=0,\ldots,M}^{t_0 \ldots t_N}$ and unoccluded object detections and segmentations are extracted.

4.1.2 Clip-Art based Self-Supervision

Once unoccluded objects are discovered, they are composited in layers back into the same scene as shown in Fig 5.5. These are close enough to real data to reduce the domain gap for a deep network that explicitly predicts the object, its occluder, and the occluded.

Background Computation: Given a sequence of images from a stationary camera, we compute the median image by finding the median RGB value per pixel from a collection of images. Since the camera is captured throughout the day and in different weather computing a single median image is unrealistic. To create realistic background images, we generate median images for

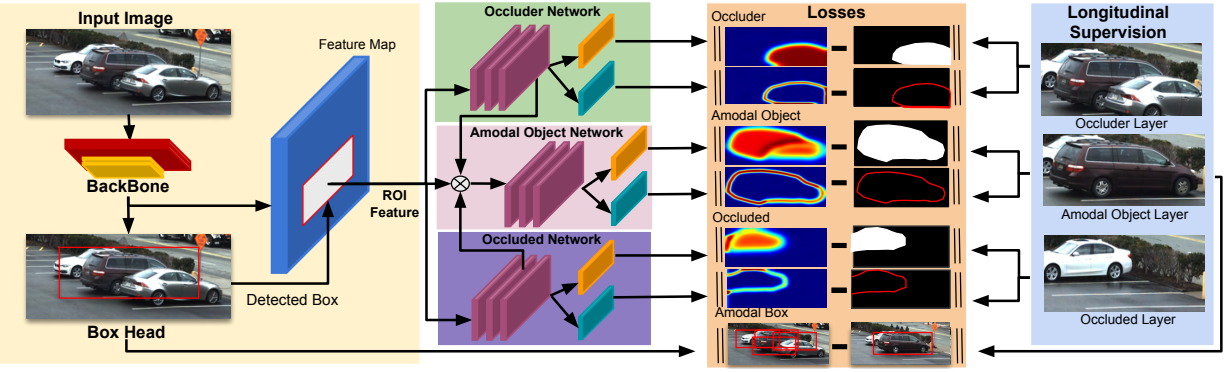


Figure 4.4: The composite images are passed through our Network to train for amodal representations of the scene. The feature map from the backbone is passed through the box head to produce the amodal bounding box. This bounding box is combined with the feature map from the backbone to produce an ROI feature. The ROI feature is used to train for amodal segmentation. The key to predicting holistic object representation is to understand the occluder and the occluded objects in the amodal bounding box. The features from occluder and occluded are concatenated with the ROI feature to produce accurate amodal segmentation. We supervise this network with a segmentation map generated using Clip-Art based Self-Supervision.

varying imaging conditions like time of the day or different weather i.e. sunny, rainy, etc. This is computed by sampling the images under different conditions. We also compute the spatial distribution of the object occurrence for each median image to simulate the occlusion patterns similar to the real-world images.

Generating Layered Representation: We randomly select a background image and its object occurrence data distribution. We sample P unoccluded objects from the data distribution $D^{t_0,\ldots t_N}_{m=0,\ldots,M}$ where $O(D^{t_j}_i) == 0; i \in P$. These sampled objects and their segmentation masks are segregated into different layers for generating varied occlusions of the scene. We iterate through each layer and composite the objects onto the background image using the segmentation masks. Since they are composited layer-wise onto the image, an amodal segmentation map is automatically generated using the segmentation mask for all the objects in the scene. Since we use longitudinal information (images over a long period of time) to generate these objects the network learns from large variations of objects as well as different occlusion configurations. The composited image and the amodal segmentation map are passed to the network for training the Amodal Representation.

4.1.3 Watch and Learn Time-lapse Network

We learn the amodal representation of the scene by training a network using the composite image and its amodal segmentation map as shown in Fig 5.3. The input image is passed through a backbone network [101] to produce feature maps. The feature map produced from the backbone is passed through the box head [136] to produce an amodal bounding box. The amodal bounding box is combined with the feature map to produce the amodal segmentation by learning Object-Occluder-Occluded interaction.

Amodal Bounding Box: The feature map from the backbone is passed through the box head to compute the amodal bounding box hypothesis. We train this box head using FCOS [136] based

losses as:

$$L_{AmodalBox} = L_{Regression} + L_{Centerness} + L_{Class} \tag{4.2}$$

The ground truth bounding box is computed using the amodal segmentation map obtained by compositing. Bounding box hypotheses are combined with the backbone feature map to learn the amodal segmentation network.

Object-Occluder-Occluded Interaction: We learn the interaction between the object and other layers present in the bounding box. Every amodal bounding box has three components .i.e. the object we want to detect (amodal object(AO)), object occluding the amodal object (occluder(OR)), objects occluded other than background(occluded(OD)). To learn a holistic representation of the object, the interaction of the object with both the occluder and occluded must be exploited by the learning framework. To train for such interactions we propose using different modules for each of the categories. The occluder network takes as input the ROI features and predicts the occluder layer in the amodal bounding box. The occluded network predicts the occluded layer of the amodal bounding box from the ROI features. The object network predicts the amodal object segmentation by robustness to the occluder and the occluded. We combine the occluder and occluded features with the object features to make the network robust to different occlusions. We use both the boundary and segmentation mask to learn the amodal segmentation. We train the boundary for each component using the loss function L^B:

$$L_M^B = L_{BCE}(W_B F_M^B, GT_M^B) \tag{4.3}$$

We train the segmentation for each component using the loss function L^S:

$$L_M^S = L_{BCE}(W_S F_M^S, GT_M^S) \tag{4.4}$$

Here, $M \in [AO, OR, OD]$ denotes different network components, and L_{BCE} denotes binary cross-entropy loss between the Ground-Truth GT and the predicted heatmap. W_S and W_B denote the weights trained for segmentation and boundary respectively. F_M^S and F_M^S are the computed feature map for segmentation and boundary respectively for each M. To make the amodal segmentation robust, we combine the occluder F_{OC}, occluded F_{OD} and input feature maps to produce the amodal object feature map F_{AO}.

End-to-End Parameter Learning: The whole amodal representation framework can be trained in an end-to-end manner defined by a multi-task loss function L as,

$$L = \lambda_b L_{AmodalBox} + L_{AO} + L_{OR} + L_{OD} \tag{4.5}$$

$$L_{Object} = \lambda_{AO}^S L_{AO}^S + \lambda_{AO}^B L_{AO}^B \tag{4.6}$$

where, L_{AO},L_{OR},L_{OD} are losses for Amodal object, Occluder and Occluded networks, respectively. As shown in Eq(4.6), for each layer the loss is a summation of the boundary loss and the segmentation loss. Similar to Eq(4.6), we compute the boundary and segmentation loss for both the occluder and occluded layers. Finally, the network is trained with an end-to-end framework optimizing all the losses.

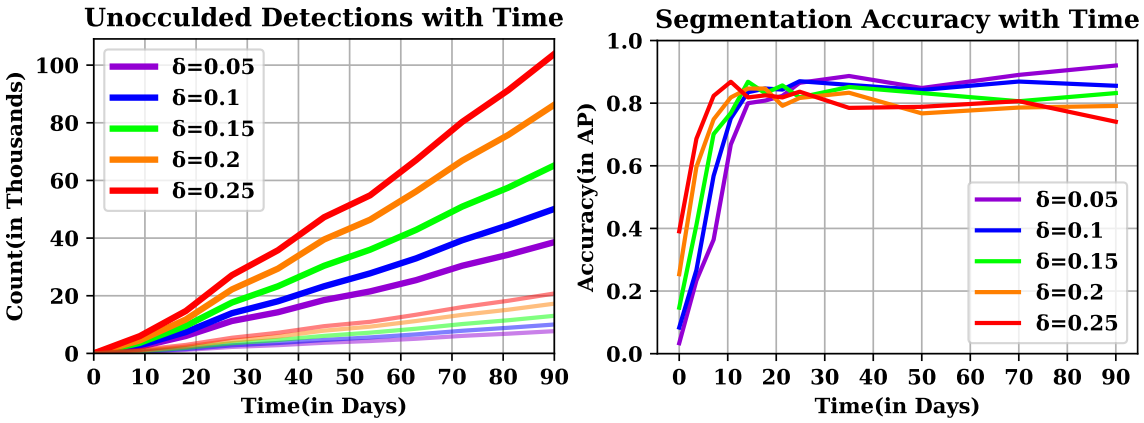


Figure 4.5: We compare the number of detected unoccluded objects (bold) using our unoccluded tracking framework compared to uniform sampling (transparent) on the left image. Using the new module, achieving high accuracy faster(within 15 days) compared to uniform sampling for nearly all thresholds of γ (right).

4.1.4 Speeding Up Amodal Learning

The accuracy of the amodal representation is affected by the quality and quantity of the unoccluded objects. We speed up the discovery of unoccluded objects by combining sparse time sampling of the data with burst local tracking. This step reduces the required observation period from many months to several days (images captured every few mins.). We discover nearly 3 times more unoccluded objects with different thresholds of detection using this strategy, as shown by the thin transparent lines on the left of Fig 4.5. These additional mined unoccluded objects speed up the training by more than 5 times and plateau in just 14 days of observation as shown in Fig 4.5 for different thresholds δ. Another important insight is that the network learns faster with higher δ but loses accuracy as the mined unoccluded objects are erroneous. On the other hand, lower δ shows that the network takes longer to learn but gains accuracy with the addition of more samples. We reduce δ linearly with the number of days captured for faster training.

4.2 Dataset and Metrics

We introduce a new dataset, called WALT, of 12 (4K and 1080p) cameras capturing data over a year in short bursts. Further, we propose a novel evaluation method using stationary objects to improve on the shortcomings of human-annotated or synthetic datasets on real images.

Watch And Learn Time-lapse (WALT) Dataset: The dataset consists of 6 4K resolution cameras setup by us and 6 1080p YouTube public live streams. The cameras overlook public urban settings analyzing the flow of traffic and people with severe occlusions, as shown in Fig 5.4. We used 4 cameras from our setup and 6 cameras from YouTube for training. Data captured from 2 cameras are used for testing. The data is captured for 3-second bursts at 30 FPS every few minutes. Only the images with notable changes from the previous image are stored. This results in storing approximately 5000 images per day for a year. We will be releasing months of data captured from cameras set up by us and publish a live stream video of the cameras on YouTube for research purposes. The code to automatically capture and process data from YouTube live

Figure 4.6: Sample visualizations from the WALT(Right) and Rendered WALT(Left) dataset. The dataset contains diverse objects with severe occlusions captured over years. The results show significant performance in amodal representation learning on such large scale real data for the first time.

streams will be released.

Potential Societal Impact: We do not perform any human subject studies from these cameras. To discourage any human subject study and preserve the privacy of the object captured in the images, we blur the faces and license plates in all the images to be released. The data is captured in short bursts around random time instances to discourage identification of movement patterns of particular persons or vehicles. This study is designated as non-human subjects research by our Institutional Review Board (IRB).

Rendered WALT Dataset(RWALT): We replicate the WALT Dataset using computer graphics rendering[129]. We use a parking lot 3D model and simulate object trajectories similar to the real-world parking lot. We render 1000 time-lapse images of the scene from multiple viewpoints. The cameras for rendering are placed on the dashboard of the vehicles or on infrastructure around the parking lot. Sample rendered images from the dataset are shown in Fig 5.4. We use rendering from 100 cameras for training and 20 cameras for testing. We use the dataset to compute the ablation study of the network using Ground-Truth from rendering.

Metrics: We use average precision (AP) for evaluating bounding box and segmentation accuracy throughout our experiments unless specified otherwise. We evaluate our method on three different categories of data generated from the WALT Dataset: the Rendered WALT Dataset (RWALT), Clip Art WALT Dataset (CWALT), and Stationary Objects WALT Dataset (SWALT). For the Rendered WALT Dataset, the amodal representation is computed on the synthetic image and compared to the Ground-Truth silhouette produced from rendering. For Clip Art WALT Dataset, we compute the unoccluded objects for 90 days on the test and train cameras of the

Dataset	Amodal Object(AO)			Occluder(+OR)			Occluded(+OD)		
	B	M	BM	B	M	BM	B	M	BM
RWALT	55.3	60.5	61.4	64.2	65.5	66.3	66.2	67.9	**68.1**
CWALT	62.3	65.5	66.1	70.2	71.2	73.2	73.9	74.2	**75.3**

Table 4.1: Ablation analysis of the proposed learning architecture on Rendered and CWALT Dataset. Note that each component .i.e Occluder (+OR) and Occluded (+OD) network improves the accuracy of segmentation. Training with Boundary(B) and Segmentation Mask(M) consistently outperforms models trained only with Boundary or Segmentation Mask.

WALT Dataset and synthesize 10000 composite images per camera using the method from Sec 4.1.2. We pass the layered image through the network and compare the results with generated Ground-Truth for test images.

Stationary Object-Based Evaluation (SWALT): Since human annotators can only hallucinate the object extent in the occluded region, their labeling is not reliable. To circumvent this problem, we propose using consistency in stationary object segmentation and detection under occlusions as a metric to quantify the accuracy of the algorithm. From the test set of WALT, we mine unoccluded stationary objects by clustering objects detected at the same location. We use unoccluded bounding box and segmentation of the stationary object as ground truth to compare predictions when the object is occluded by another object at a different time instance. The mean Intersection-over-union (IOU) between the Ground Truth and prediction is computed for the stationary object when it is occluded by greater than a threshold of γ. γ is computed as the overlap between the Ground-Truth bounding box and the bounding box of other objects in the scene. Using this strategy, we extracted 536 stationary objects observed over 60k frames for evaluation.

4.3 Evaluations and Ablation Analysis

The performances of pedestrian and vehicle detection and segmentation improve significantly in all of the cameras. we report performances at different levels of occlusion and show that the performance drops more slowly as occlusion increases, compared to methods that do not use Clip-Art Based self-supervision.

Notations: Modal represents a model trained using visible segmentations or bounding boxes, while Amodal uses our amodal supervision. In Amodal methods, just using the Amodal object network is represented as AO, while adding just occluder network as +OR. +OD is given as a combination of final layers from both occluder and occluded networks. B and M represent boundary and segmentation Mask respectively, while BM represents training jointly.

Occluder and Occluded Networks Analysis: We observe that adding features from the occluder and occluded networks to the amodal object prediction network increases the accuracy of amodal segmentation for the Rendered WALT Dataset and the Clip Art WALT Dataset as shown in Fig 4.1. We observe robust segmentation accuracy with an increase in occlusion percentage when using the occluder and occluded networks in Fig 4.7 for both vehicles and people.

Boundary and Mask Prediction Analysis: Segmentation based methods are observed to be better than boundary based methods. We observe that combining the object boundary with seg-

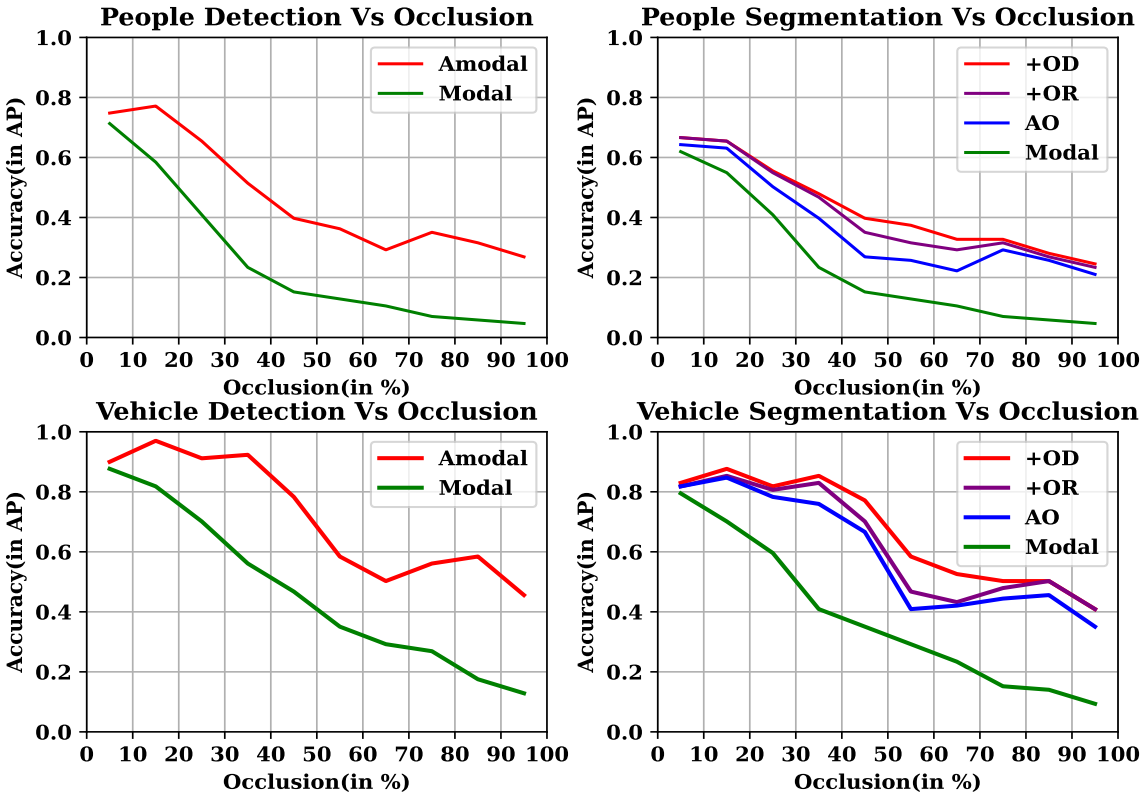


Figure 4.7: Comparative analysis of Segmentation and Detection accuracy of people and vehicles. Clearly Amodal(Holistic Representation) based methods outperform Modal(only visible representation) based methods in detection and segmentation. Addition of each Network(AO, +OD, +OR) to amodal training improves accuracy of segmentation for severely occluded scenarios. At 50 % occlusion we observe nearly 90 % and 60 % improvement in detection accuracy compared to modal based for people and vehicle respectively. Similarly, at 50 % occlusion we observe 20 % and 12 % improvement in segmentation accuracy compared to Occluder(+OR) for people and vehicles respectively.

mentation mask consistently improves accuracy on both the Datasets as shown in Tab 4.1.

Robustness to Occlusions: We evaluate the accuracy of our algorithm with different percentages of occlusions using CWALT Dataset. We use the Ground-Truth segmentation masks from the dataset to group objects based on the percentage of occlusion. Fig 4.7 shows the accuracy of detection and segmentation on the Clip Art WALT Dataset with different occlusion percentages. Clearly, we observe that the proposed method is very robust to occlusion compared to other methods for both people and vehicles.

Occlusions Over Time: We analyze the accuracy of amodal representation with respect to training data from different lengths of the Clip Art WILD dataset, in Fig 4.8. The N-th day plot corresponds to a model trained with N days of unoccluded object detection. We observe from the heatmap that the accuracy increases with time as more unoccluded objects are used to train but decrease with occlusion percentage. We further observe that accuracy improves over time for more severe occlusions, emphasizing that longitudinal learning is important to handle severe occlusions.

Comparison to Human Annotated Datasets: We reiterate that human annotations, especially for strong occlusions, are imprecise to learn amodal representations. Compared to human anno-

	KINS	COCOA	SWALT	
			γ = 0.01	γ = 0.5
ASN	24.9	29.6	79.4	76.91
BCN	27.3	32.7	82.79	77.44
Ours	**27.9**	**33.1**	**83.6**	**78.2**

(a) Trained on KNIS[7]+COCOA[8]

	CWALT	SWALT	
		γ=0.01	γ=0.5
ASN	66.1	83.1	81.9
BCN	73.2	89.9	88.3
Ours	**75.3**	**92.19**	**91.7**

(b) Trained on CWALT

Table 4.2: Amodal Segmentation comparisons trained on Human annotated datasets (a) and Clip-Art WALT Dataset (CWALT) (b) with respect to three different network architectures ASN[7], BCNet[8] and Ours. Tab. 4.2a shows that Human annotated dataset training only achieves around 78% accuracy on SWALT. On the other hand, Tab.4.2b reports 91.7% accuracy on SWALT showing the advantage of training on CWALT. In fact, all methods show improvement on SWALT by training on CWALT. γ represents the percentage of occlusion for each object in SWALT but needs further study to report for human-annotated datasets.

tated datasets .i.e. KINS or COCOA, our SWALT based evaluation methodology produces more accurate ground truth. Further, SWALT methodology generates much larger test sets compared to any existing human annotated datasets (60K images from WALT dataset compared to 6157 images in KINS dataset) and is expected to grow significantly as data is captured from more cameras in the following years. Scaling human annotations on such expanding datasets is costly and infeasible and our self-supervision based methodology automatically generates accurate and large training and testing datasets for amodal evaluation. Nonetheless, we report accuracy of our method when trained on Human annotated datasets and tested on KINS, COCOA and SWALT in Tab 4.2a. Our method slightly outperforms previous methods here.

Comparisons to other Networks: We analyze the advantage of training/testing different methods on our data (CWALT/SWALT). The test scores show improvement in amodal accuracy as compared to other methods. In fact, all methods improve by training on CWALT and testing on SWALT as shown in Tab 4.2b. We show a qualitative comparison of these methods on multiple real-world images with severe occlusions in Fig 4.11.

Robust Tracking Using Amodal Representations: We demonstrate that learning robust amodal representation automatically improves tracking of severely occluded objects, as shown in Fig 4.10 for people and Fig 4.9 for vehicles. Specifically, observe that the objects are well-segmented and consistent across frames with various levels of occlusions. See supplementary material for more results and videos.

4.4 Conclusion and Limitations

Limitations: Generalization of the amodal segmentation on new cameras that view significantly different scenes needs to be analyzed. Speeding up learning rate even further needs to be inves-

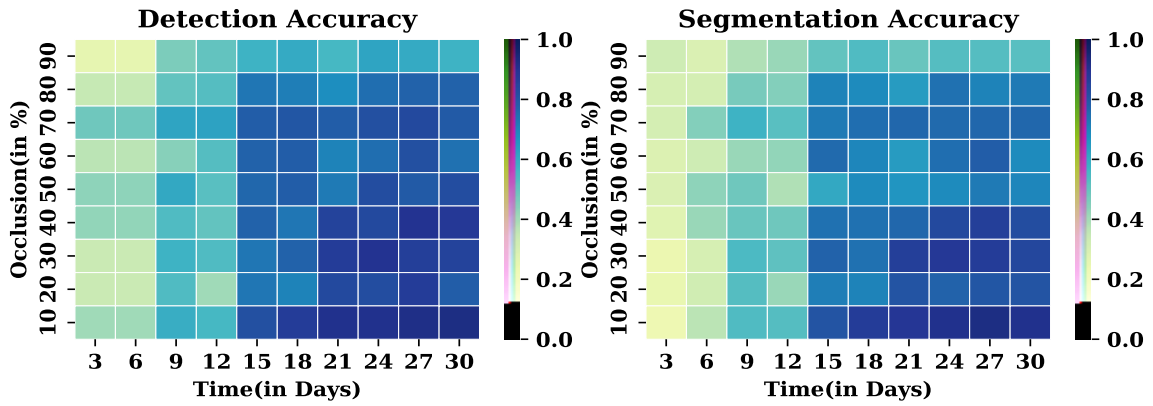


Figure 4.8: Heatmap of accuracy with different occlusion levels over time on the CWALT Dataset. Observe that the accuracy improves drastically with time for severe occlusions(.i.e >50%) emphasising that our framework learns robust amodal segmentation.

Figure 4.9: Accurate amodal segmentation of vehicles during occlusion while passing each other(Top) or when a vehicle is parking. Our method is able to provide consistent segmentation and detection of all the vehicles in severe occlusions and motions. This can lead to a drastic improvement in tracking objects with occlusions.

tigated for broader application of our approach.

Conclusion: This work demonstrates that real longitudinal data can be used effectively to self-supervise amodal learning. The key insight is that it is easier to discover unoccluded objects accurately and quickly (over several days) and use them to learn amodal segmentations from any stationary camera observing a scene over time. The confidence of this discovery can be used as a quasi-learning rate to speed up amodal training of occluded objects. We introduce a new dataset, called WALT, of 12 (4K and 1080p) cameras capturing data over a year in short bursts every 5 minutes or so. The data will be released with faces and license plates anonymized to help preserve privacy. The results show significant performance in amodal representation learning on large scale real data for the first time. In the future, we will extend our approach to learn from cameras placed on vehicles for self-driving applications.

Figure 4.10: Accurate prediction of amodal segmentation of people when a person passes by another(top) or when they walk occluding throughout the video(bottom). Such representation directly extrapolates to improved tracking of people in generic videos.

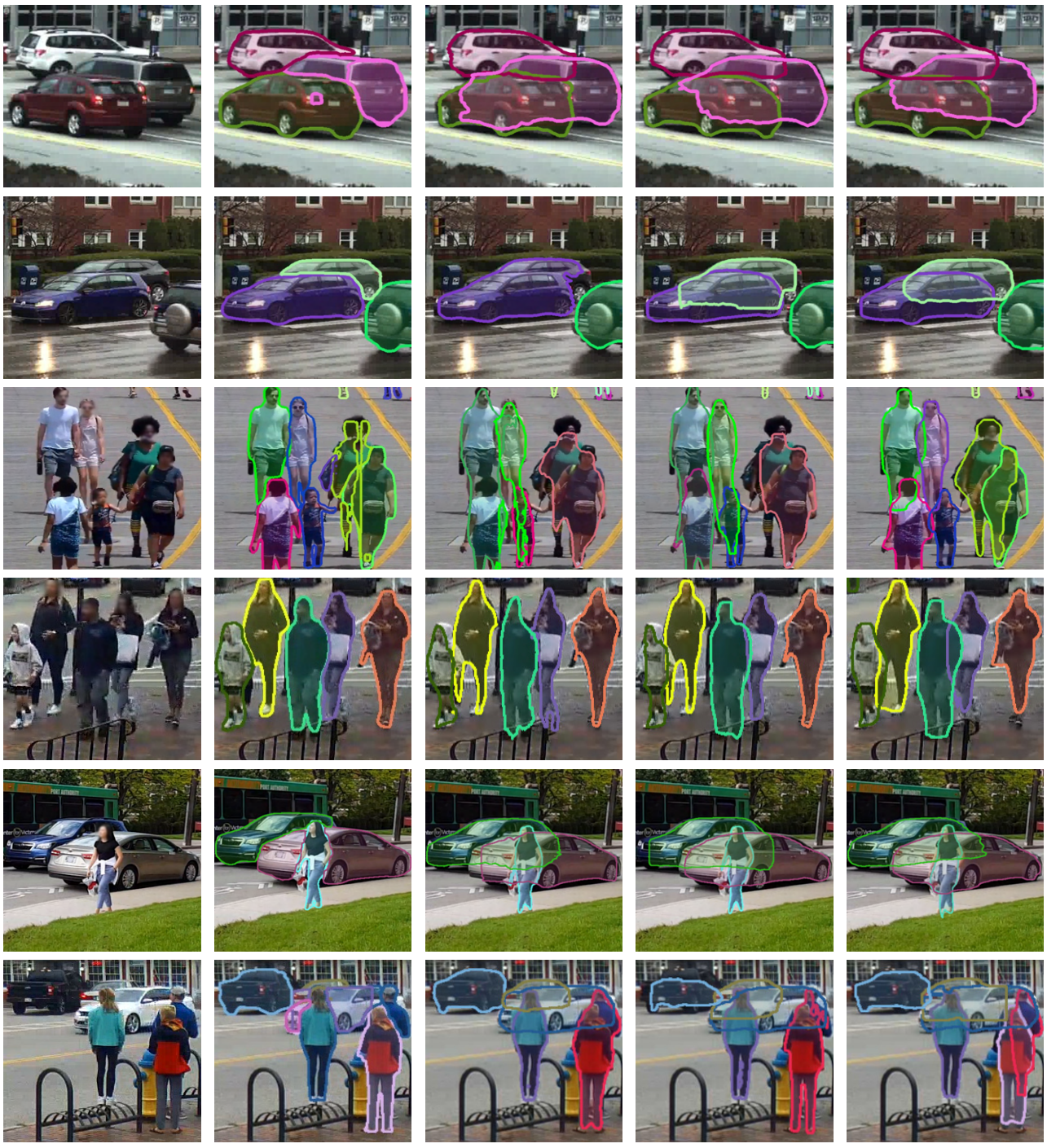


Figure 4.11: Quantitative results comparing our method to the state-of-the-art images captured from different datasets. The first two rows show vehicles occluding vehicles scenarios while the next two show people occluding people. Finally, we also show examples of people and vehicles occluding each other in the bottom two rows. Observe that our method consistently outperforms other baselines in predicting the amodal segmentation due to longitudinal self-supervision formulation. We perform accurate segmentation in difficult occlusions scenarios like objects having similar colors (Second Row) or large occlusions(Third Row, Sixth Row) or multiple layers of occlusions(First Row, Fifth Row). Our method even works with low-resolution images(Fourth Row) and inter-object interactions(Fifth Row, Sixth Row).

Chapter 5

Exploiting Occlusion Categories

Occlusions are everywhere and handling them is crucial for scene understanding [28, 117, 119, 137, 138, 139, 140]. Treating occlusions as outliers [25, 26, 28, 112, 141, 142] does not often work reliably as there may be too many in a scene. Explicitly modeling occlusions is challenging because of the range of occlusion types in the scene [119, 142]: an object may be partially occluded by other objects, truncated by the camera's field-of-view and even if there is only one object, the viewable side of the object (front) occludes the non-viewable side of the object (back) (see examples in Figure 5.1). Learning occlusions requires a large annotated, realistic dataset. Unfortunately, such datasets are lacking because labeling hidden parts of objects is a difficult task for people to consistently accomplish [7, 8, 113]. Overcoming these challenges is important to advance many smart cities applications, where the number of cameras on vehicles and city infrastructure is rapidly increasing [143, 144, 145].

To address these challenges, there have been several recent advances in amodal scene understanding by modeling occlusions. There have been small datasets where humans have annotated occlusions to the best of their abilities and methods have been developed with such supervision [7, 8, 118, 120], even if the labels are inaccurate or insufficient. Occlusion-Net [113, 146] provides an accurate method to supervise self-occluded keypoints using multiple views. To expand the supervision, several methods synthesize occlusions to varying degrees of realism. But pure CG renderings [105, 115, 121, 124, 125, 126, 142] suffer from a wide domain-gap [28, 131]. To address this domain gap, methods such as WALT [9] introduces a hybrid approach to composite real image segments of self-occluded objects captured from time-lapse data to create a 2D clip-art dataset of a large number of occlusion configurations. They used this clip-art dataset to train a network and showed significant performance improvement in 2D amodal segmentations. While these approaches have advanced the state of the art, they focused on only one type of occlusion (self-occlusion or occlusion-by-others) and there is still a strong need for a holistic approach for 3D amodal reconstruction under all types of occlusions.

In this work, we present an approach to address occluded by other objects, self-occlusion, and occluded by truncation to produce 3D amodal reconstruction for the first time. We start by making an observation that human supervision is highly accurate in categorizing (not localizing) occlusions in images and an accurate category classifier can be learned. We show that this categorization (occluded by other objects, self-occlusion, or occluded by truncation) can be exploited to develop both 2D and 3D amodal reconstructions of all objects. Key to achieving this is our use of mixed-representation for objects - keypoints, segmentations and statistical shape models

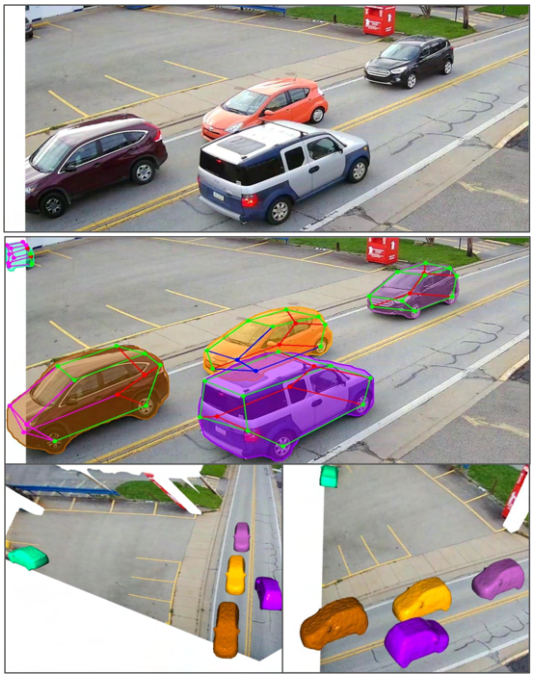


Figure 5.1: Top: Example scene with objects (vehicles) exhibiting different types of complex occlusion. **Middle**: Our method is able to recover amodal 2D segmentation and keypoints. Different types of occlusion are shown by different colored wireframe segments. Every object in this scene has visible regions (green) and one or more type of occlusion like Self-Occlusion (red), Truncation (magenta), and Occlusion-by-Others (blue). **Bottom**: Our method is able to reconstruct amodal 3D shapes and poses of the objects by exploiting these occlusion categories in densely populated scenes.

- since no single representation is satisfactory for all occlusion configurations. Segmentations provide natural layered representations in 2D, keypoints are useful to represent self-occluded regions and statistical shape models regularize 2D-to-3D optimization.

We have also develop an approach to automatically generate realistic supervision data for training from time-lapse imagery. But instead of doing this in 2D [9, 105, 147], our method generates this data in 3D by first mining only self-occluded objects, reconstructing them using object motion and then compositing them in 3D. This hybrid 3D composited data is then used to train both layered amodal keypoints and segmentations. Finally, the amodal 2D representations are lifted to 3D using shape basis optimization. Binary visibility of keypoints is used to supervise the entire pipeline.

We demonstrate our holistic approach to amodal 3D reconstruction using several datasets, including WALT [9], Carfusion [146], and others (PASCAL3D+ [148], KITTI3D [142], ApolloCar3D [3]). The occlusion categorization method outperforms previous heuristics, making sure that the hybrid training data we generate is physically accurate. Quantitatively, our approach significantly improves upon previous the state-of-the-art by 12% for self-occlusion handling and layered occlusion 3D recovery. We demonstrate successful 3D reconstruction at busy urban scenes captured from a variety of viewpoints and distances including traffic-cams, hand-held cameras and under different lighting conditions including night. While we have focused mainly on vehicles in this work, it can be naturally extended to other classes of objects like people using the readily available shape models [149].

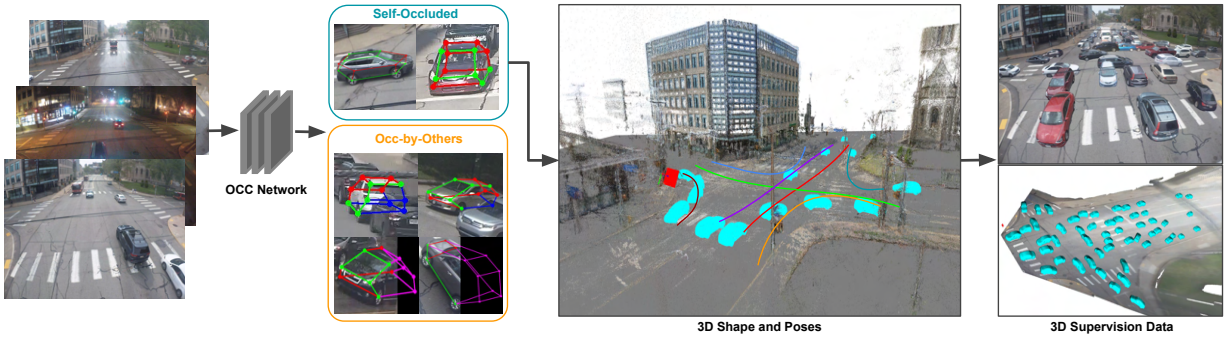


Figure 5.2: We illustrate the framework for 3D Supervision Generation using Clip-Art. The key idea is that we use the Occlusion Category Classification (OCC) network on a stream of data to mine for self-occluded objects. We then perform 3D spatio-temporal reconstruction of these mined self-occluded objects following [1] to get 3D shape and poses (showed on top of the actual 3D background scene reconstruction). These self-occluded objects are placed back in the same location they were detected to generate various occlusion configurations as 3D ground-truth supervision data to train for Amodal 2D/3D Representations. Note that per-keypoint occlusion category information are also later used in the occlusion consistency loss as an additional supervision signal.

5.1 Amodal 3D Reconstruction

We tackle the problem of 3D recovery of objects under severe occlusions by learning and exploiting different occlusion categories. We blend occlusion understanding into a 3D reconstruction framework for improving accuracy with no additional supervision. We start by classifying the input object into three different occlusion categories, i.e. self-occluded, occluded-by-others, and truncation.

5.1.1 Occlusion Category Classification

For every semantic keypoint on a single input object, we define its visibility status which can be classified into four different categories: visible, self-occluded, occluded-by-others, and occluded-by-truncation and its 2D location. The visibility category is crucial to facilitate better occlusion understanding since an object can simultaneously fall into different occlusion configurations, e.g., some parts of a vehicle are self-occluded and other parts can be occluded by other objects. We formulate the visibility prediction problem as a classification task that associates each detected keypoint with one of the four labels mentioned above. We will refer to this as the **OCC** (Occlusion Category Classification) module. We learn this module in a supervised manner using our new dataset which will be described in Section 5.2. This keypoint occlusion understanding is then used to generate ground-truth 3D occlusion-aware data for training.

5.1.2 Generating Occlusion-Aware Supervision

Using the OCC module, we are able to categorize types of occlusion for every semantic keypoint. This information is sufficient to generate a large amount of occlusion-aware data for training. In this section, we will look at how to exploit the occlusion understanding to generate supervision signal for different occlusion categories.

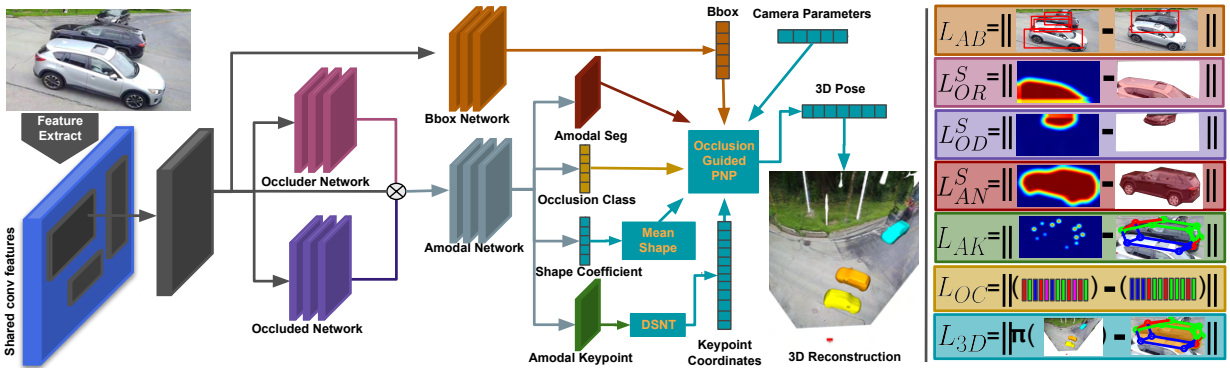


Figure 5.3: Given the Amodal Clip-Art Image and the corresponding 2D/3D representations of the objects from the occlusion-aware supervision, we illustrate the network used to train to predict 3D pose and shape of the object. The input image is passed through a backbone to extract ROI features. These features are passed through an occluder and occluded networks which help disentangle objects occluded-by-others. The features from these networks are concatenated and passed through an amodal network. The network learns to predict the amodal segmentation, keypoint locations, shape bases, and occlusion types. Finally, these representations are combined with the camera parameters and passed through a Occlusion-Guided Differentiable PNP to produce the amodal 3D pose. All the network losses are jointly optimized to produce 3D reconstruction.

Geometric Supervision for Self-Occlusion On an instance-level, each object can be categorized as Self-Occluded or Occluded-by-Others (see Fig. 5.2). Evidently, many downstream vision tasks such as segmentation, tracking, reconstruction, etc. work well for Self-Occluded objects. Using the occlusion category information from the OCC module, we classify each object instance into the Self-Occluded or Occluded-by-Others category. We then mine all the objects belonging to the Self-Occluded class and reconstruct them in a large joint optimization using planar constraints combined with the 3D static background reconstruction following [1]. Given the 2D keypoint locations of the object, we initialize the 3D poses using EPnP[61, 150] by using only visible keypoints. Further, during the joint optimization for the object poses and shape coefficients, only the visible keypoints predicted from the OCC network are used. This occlusion-aware reconstruction pipeline produces accurate 3D poses and shape coefficients. The mined Self-Occluded object segmentation, keypoint predictions and 3D poses are used in a clip-art based framework to generate 3D supervision data as illustrated in Fig. 5.2.

Clip-Art Supervision for Occluded-by-Others: Getting supervision for occluded-by-others is a challenging problem. Therefore, by using clip-art based image augmentation, we can automatically generate a large number of supervision signals in severe occlusions. By using the 3D poses of the objects, we generate a 3D-aware scene graph with non-intersecting 3D bounding boxes. These objects are placed back into the background image from the farthest to closest objects producing a realistic looking generated image as shown in Fig. 5.5. The generated image additionally can produce ground-truth data such as amodal segmentation, amodal 2D/3D bounding box, 3D poses and depth as shown in Fig. 5.5 for occluded-by-others category. This kind of supervision signal will play a major role for in deciphering different layers of occlusions for downstream tasks such as tracking and reconstruction.

5.1.3 Occlusion-Aware 3D Reconstruction

We have generated a large clip-art image dataset and corresponding amodal 2D/3D ground-truth representations with occlusion understanding. Using these supervision signals, we will recover the 3D pose of the object by disentangling each layer of occlusion in a network as shown in Fig. 5.3. We first run a feature extractor network on the input image and ROI features are passed through a Bbox Network to compute the amodal bounding box. We compute the loss between the predicted bounding box and the Ground-Truth Amodal Bounding box similar to [101] and is given as L_{AB}. The rest of the ROI features are passed through the 3D prediction network. We can simplify the network into two broad segments based on occlusion categories. Firstly, we learn to disentangle multiple occlusion layers in the scene.

Learning Occluded-by-Other layers: For computing the amodal features of an object initially, it is quintessential to learn different occlusion layers in the amodal bounding box. To achieve this goal, we learn the occluder-occluded-object interaction which helps us distinguish each object interacting with the bounding box to disentangle multiple objects in a scene. We train the segmentation loss for each of these components using the binary cross-entropy loss function L:

$$L_M^T = -W_T[G_M^T log(F_M^T) + (1 - G_M^T) log(1 - F_M^T)] \tag{5.1}$$

Here, $M \in [AN, OR, OD]$ denotes amodal network, occluder and the occluded network, while $T \in S, K$ denotes the type of representation, i.e. Segmentation and Keypoint respectively. We compute the binary cross-entropy loss between the Ground-Truth G and the predicted map F with the weights given by W. Note that the features from both the occluded and occluder layer are concatenated with the input ROI feature to produce an amodal feature vector, which helps improve the amodal network to learn to distinguish different objects in a bounding box. This combined amodal representation feature is used to compute the segmentation mask, keypoint locations, shape coefficients, and the occlusion category of each object. For the amodal segmentation computation, the output of amodal network is passed through multiple convolutions to produce a heatmap for segmentation and the loss is computed as:

$$L_{AS} = L_{OD}^S + L_{OR}^S + L_{AN}^S \tag{5.2}$$

Here L_{OD}^S, L_{OR}^S, L_{AN}^S represent binary cross-entropy loss for occluded, occluder, and amodal segmentation maps, respectively.

Similarly, the amodal features are passed through keypoint regression network to produce amodal keypoints, and the loss is given as:

$$L_{AK} = \sum_{k \in K} L_{OD}^k + L_{OR}^k + L_{AN}^k \tag{5.3}$$

Here L_{OD}^k, L_{OR}^k, L_{AN}^k represent binary cross-entropy loss for occluded, occluder, and amodal keypoints where the loss is summed over each $k \in K$ keypoints of the object. We also compute the per-keypoint occlusion category to understand the type of occlusion from the amodal network. The loss is given as:

$$L_{OC} = -\sum_{k \in K} \sum_{c \in M} y_c^k log(p_c^k) \tag{5.4}$$

where y_c^k is the binary indicator if keypoint k belongs to class c given by the OCC network while p_c^k is the predicted probability observation of class c for keypoint k from the network. This helps us distinguish multiple objects and their visibility accurately in predicted amodal bounding box. Generally, although we can compute the loss on both categories, segmentation representation is more beneficial compared to keypoint in understanding such layers. Note that the supervision is produced from clip-art based augmentation method as shown in Fig. 5.2.

Learning Self-Occlusion Using 3D Supervision: Once the occluder and occluded layers are disentangled from the input ROI feaure, only objects with self-occlusions remain to be learned. We use the geometrically consistent reconstruction from longitudinal data for supervising the self-occluded portion of the object. We pass the amodal representations through an Occlusion-Guided-Differentiable-PnP (OGD-PNP) to produce the 3D pose and shape parameters used for amodal 3D recovery. OGD-PNP is similar to [151, 152] but has additional occlusion supervision for improved pose estimation. The input to this module is the keypoints and segmentation mask transformed to the original image coordinate frame, the mean shape of the object, mean shape coefficients, camera parameters, and occlusion category class. We compute the loss for OGD-PNP as:

$$L_{3D} = \frac{1}{2}\sum_{i=1}^{N} ||w_i \circ (\pi(RX_i + t) - x_i)||^2 + \sum_{k \in K}\sum_{c \in M} y_c^k log(r_c^k) \tag{5.5}$$

The first term is the reprojection loss between the reconstructed shape and the predicted shape. Here w_i represents the weights of the reprojection loss, $\circ$ represents element-wise mutiplication, R and t represent the 3D poses of the object. N represents all the points in the mean shape. X_i and x_i represent the 3D mean shape and 2D predicted points. The second term is the occlusion consistency term which enforces that the occlusion configuration of the predicted 3D object should be as similar as possible to the predicted occlusion type. Specifically, r_c^k is computed by ray-tracing the reconstructed 3D keypoint and optimizing for the visibility constraint. We can learn these losses for objects occluded-by-truncation by minimizing the 3D reprojection loss on visible keypoints.

We compute the corresponding X_i for keypoints and masks using the same shape coefficients on the mean shape from [113] for keypoints and [153] for masks. For the 2D location x_i of the keypoints, we pass the predicted keypoints through a differentiable argmax module (DSNT [154]) to convert from ROI feature space to coordinate space. These ROI coordinates are transformed with respect to the bounding box to produce the keypoint locations in the original image space. Similarly, we transform the amodal segmentation mask to the image coordinate frame as well. Finally, the reprojection loss and occlusion consistency loss are optimized to produce amodal 3D pose and shape of the object.

End-to-End Optimization: The final step is to optimize for the 3D poses from the input clip-art image with 2D/3D supervision signals. The final loss term is given as the sum of the losses for the amodal bounding box, segmentation heatmap, keypoints, and OGD-PNP:

$$L = L_{AB} + L_{AS} + L_{AK} + L_{3D} \tag{5.6}$$

For a object, we learn amodal bounding box, segmentation, keypoint locations, occlusion category, 3D shape and pose in an end-to-end differentiable joint optimization.

Figure 5.4: Sample images from our new Occlusion Category Classification (OCC) Dataset. Our dataset contains a wide range of appearance variations: nighttime driving, traffic cams, etc.

5.2 Dataset and Implementation Details

There are multiple vehicle keypoints datasets [3, 142, 146, 148] but none of them provides detailed occlusion category information. Moreover, they also lack the appearance diversity to perform well on in-the-wild evaluation data. To tackle this problem, we propose a new dataset called *Occlusion Category Classification (OCC)* Dataset.

Occlusion Category Classification (OCC) Dataset: Our new dataset consists of images collected from many freely available in-the-wild sources, including in-vehicle, handheld, and public traffic cameras. The dataset also captures a large number of appearance variations including day/night and different countries, weather conditions, and seasons. It contains of 7,018 images with 42,547 car instances with 90/10% training and testing split. We manually annotated 12 semantic keypoints for each vehicle as well as the corresponding occlusion category. Among these, 5,384 instances are marked as Occluded-by-Others and 1,467 instances as Occluded-by-Truncation. The dataset is used for pre-training and evaluation purposes and will be released for the research community. Examples from our OCC dataset are shown in Fig. 5.4.

WALT Dataset [9]: This dataset contains images from 12 cameras overlooking urban scenes captured over multiple years. The images are either 4K or HD and are captured at 60fps in short bursts. We used 30 days of data from 10 cameras amounting to approximately 3.3 million car instances for our experiments. We use the WALT raw dataset to generate the 3D Clip-Art supervision dataset.

3D Clip-Art Supervision: From the WALT dataset, we mine for objects containing Self-Occlusion

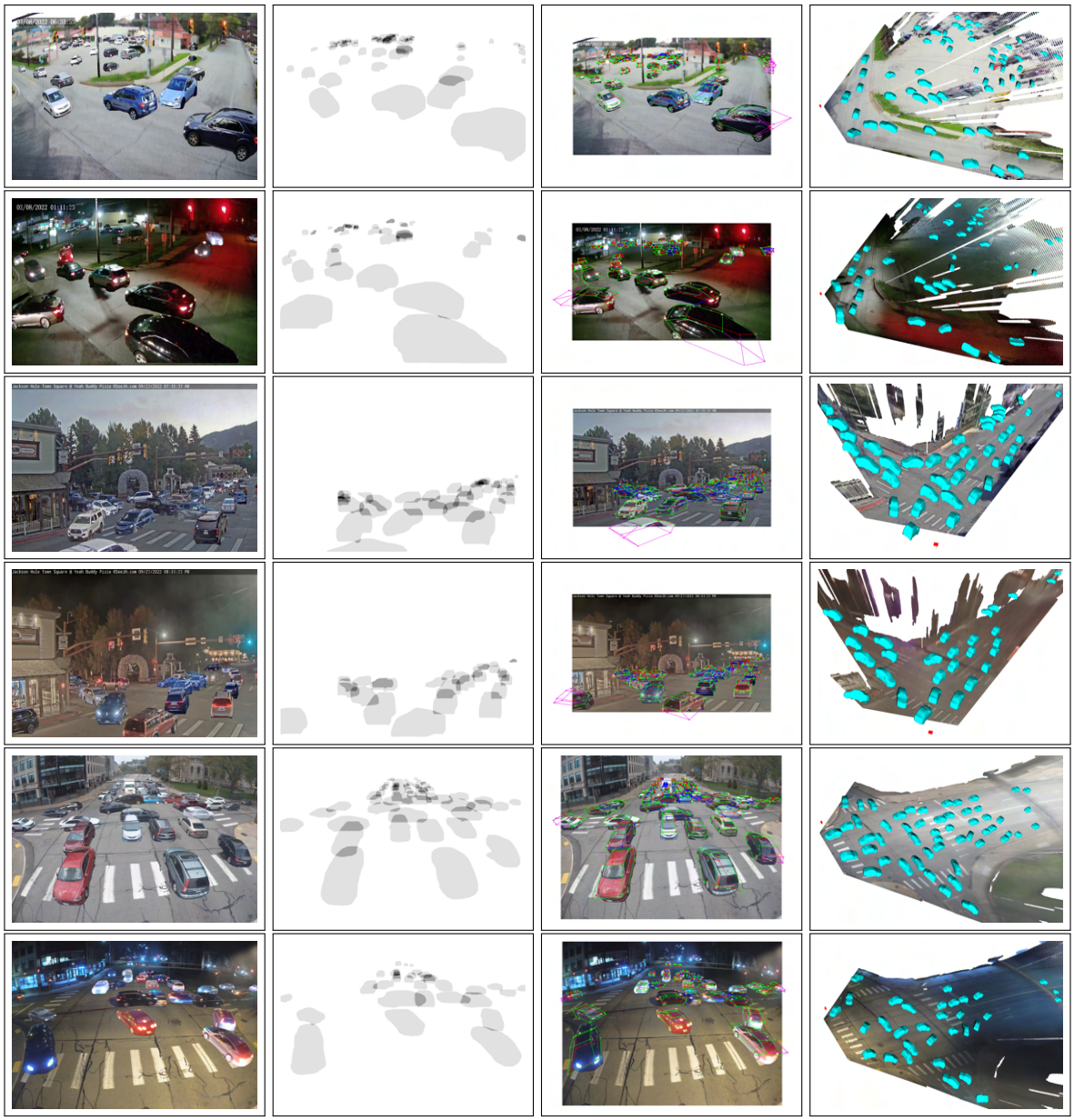


Figure 5.5: We show samples of the generated 3D Clip-Art dataset on images captured from WALT dataset. We show the 3D Clip-Art generated realistic image**(column 1)** and their respective amodal segmentation mask **(column 2)** and keypoint locations **(column 3)**. In **column 4**, we show the reconstructed 3D poses of vehicles using the 3D Clip-Art generation pipeline. Observe that the method can generate results across multiple cameras with varied weather and lighting conditions with realistic occlusion configurations. This acts as a very strong supervision signal to learn 3D amodal network.

resulting in 2.1 million objects. We paste them back into the scene with different backgrounds generating 10000 training and 500 testing images per camera. The resulting Clip-Art dataset covers all occlusion categories in different lighting and weather conditions. Sample 3D Clip-Art dataset are shown in Fig 5.5.

Camera Parameter Estimation: We leveraged Google Street View (GSV) [155] to perform camera calibration automatically on all the cameras. We sample multiple panoramas around

Method	PCK@0.1	Visibility Class.	Occ. Type Classification		
			Self-Occ	Occ-Oth	Occ-Trunc
MaskRCNN [15]	62.45	74.27*	✗	✗	✗
Occ-Net [150]	66.41	80.15	✗	✗	✗
Ours	**80.12**	**86.18**	**80.80**	**61.74**	**63.01**

Table 5.1: We show the keypoint prediction, visibility classification, and occlusion type classification accuracy on our OCC dataset. (✗: not available, *: using best confidence score threshold)

Metric	$\delta = 0.01$	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.5$	**OCC (ours)**
Recall	0.60	0.42	0.17	0.01	**0.81**
Precision	0.32	0.41	0.52	0.57	**0.70**

Table 5.2: Accuracy of our OCC module compared with heuristics baseline using bbox IOU threshold δ [9] in detecting Occluded-by-Others objects.

the desired camera's location and use COLMAP [156] to reconstruct the 3D scene geometry. We then establish 2D-3D correspondences between the camera's image and GSV reconstruction using [157, 158] and jointly optimize for the intrinsic and extrinsic parameters in a bundle adjustment step.

Metrics: We follow the Mean Average Precision (IoU=0.5)[11] for bounding box detection, object segmentation, and 3D pose estimation. In the case of 3D pose estimation, we compare the predicted 3D bounding box with respect to the ground-truth bounding box from the 3D Clip-Art generated 3D poses. For the case of keypoints, we use the Percentage of Correct Keypoints (PCK) metric where a keypoint is considered correct if it lies within the radius α of the ground-truth keypoint (normalized by the maximum of length and width of the bounding box and $0 < \alpha < 1$).

Baselines: We use MaskRCNN[15], Occ-Net[113] and 3DRCNN[159] with SWIN[101] backbone trained on multiple vehicle datasets[3, 142, 146, 148]. We use WALTNet and WALTNet-KPS [9] on the WALT dataset to evaluate for amodal representations.

5.3 Ablation Analysis and Results:

Keypoint and Occlusion Category Accuracy: Using Occ-Net [113], we further finetune on our OCC dataset and evaluate the accuracy of 2D keypoint localization and per-keypoint occlusion classification on OCC testing data. From Table 5.1, we can draw two conclusions: 1) There exists a big domain gap between previous datasets and ours (OCC) leading to an improvement of 20% in keypoint localization accuracy and 8% in visibility classification when finetuned on our new dataset, and 2) Per-keypoint occlusion type annotations provided by our dataset enables us to effectively learn a category classifier and subsequently improve the performance of downstream 3D reconstruction tasks.

Occluded-by-Others Detection To detect Occluded-by-Others objects, WALT [9] used a simple heuristics where an object is classified as Occluded-by-Others if its bounding box Intersection-

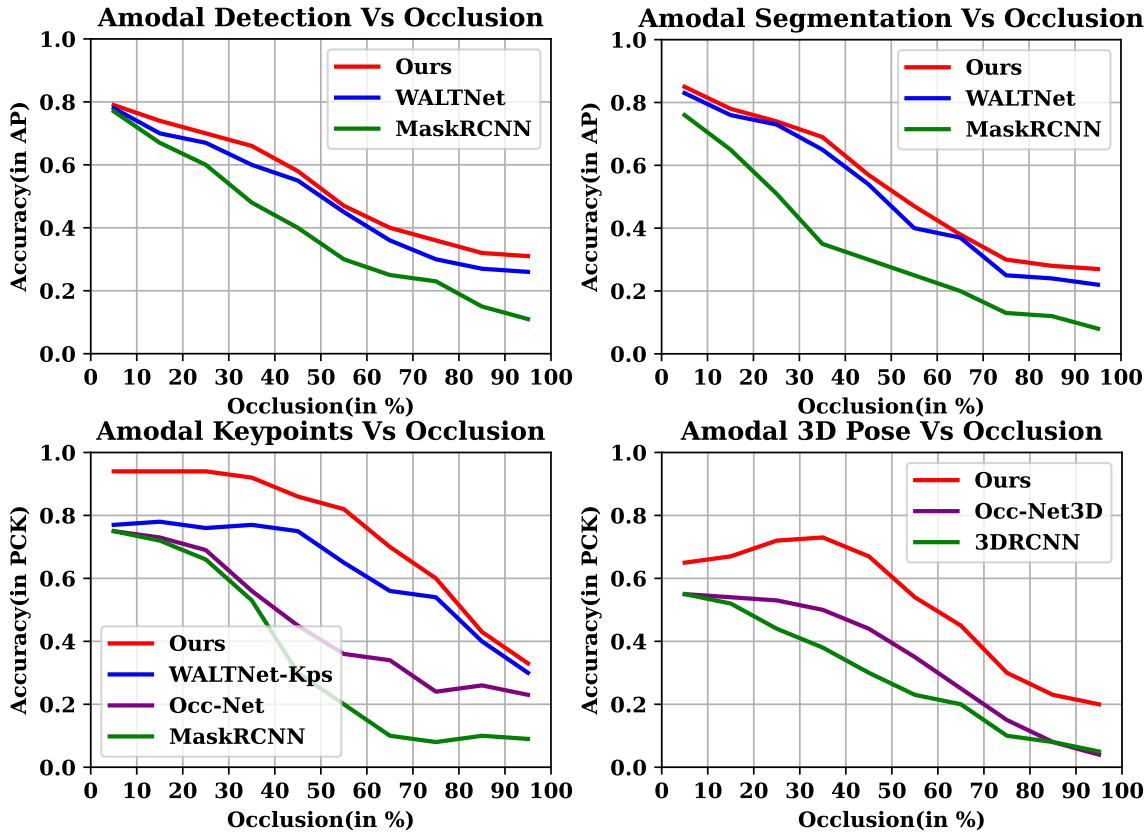


Figure 5.6: We show the accuracy of our method with respect to an increasing percentage of occlusion on multiple tasks like amodal detection, segmentation, keypoint, and 3D pose estimation. Observe that our method consistently performs better than other baselines showing robustness to increasing occlusion percentage.

over-Union (IOU) with other objects is greater than δ. In Table 5.2, we compare this heuristics baseline (using different thresholds of δ) with our OCC network in detecting Occluded-by-Others objects. We show that our OCC module is significantly more effective compared to the naive heuristics, allowing us to effectively filter out unwanted Occluded-by-Others objects in the training dataset, thus simultaneously reduce training time and improving training data's purity.

Dissecting the Network We analyze the advantages and disadvantages of different network choices in Table 5.3. Observing that with the addition of Occluder and Occluded networks, the accuracy of segmentation improves drastically but the 3D network does not show substantial improvement in segmentation. Keypoint detection improves marginally with the addition of Occluder and Occluded network but improves substantially using the 3D loss. Each of these elements helps improve the accuracy of the 3D pose showcasing that both the representations of mask and keypoint are helpful as well as the network choices help improve accuracy by nearly 8%.

Robust 3D Recovery with Occlusions: Our method is robust in detection, segmentation, keypoint estimation, and 3D pose estimation with increasing occlusion compared to to previous proposed methods as can be seen from Figure 5.6. We observe specifically that the 3D recovery consistently outperforms other baselines both in the case of self-occlusion and occlusion-by-others.

	Amodal Network (AN)			AN+OR+OD			AN+OR+OD+3D		
Accuracy	Kps	Segm	Both	Kps	Segm	Both	Kps	Segm	Both
Segm (AP)	✗	72.3	72.5	✗	76.3	**76.9**	✗	76.4	76.5
Kps (PCK)	73.5	✗	73.8	74.3	✗	81.2	85.1	✗	**85.3**
3D Pose (AP)	55.4	42.3	56.5	58.5	46.9	58.3	62.3	50.3	**63.4**

Table 5.3: Accuracy analysis of each network component with different representations, i.e. keypoints and segmentation. Observe that with the addition of each constraint, the accuracy of 3D pose estimation improves, showcasing that the additional supervision data is helpful in improving 3D recovery.

Segmentation vs. Keypoints for Amodal 3D: We show analysis of using different representations, i.e., segmentation and keypoints for 3D recovery in Table 5.3. We observe that segmentation helps improve the accuracy in occlusion-by-other cases while keypoints and mean shape help in self-occlusion. Therefore, we exploit both of them to produce accurate 3D Amodal Reconstruction.

Comparison to Baselines: We show analysis of our method compared to other baselines in Figure 5.6. We observe marginal improvement over WALTNet for segmentation and bounding box detection due to marginal change in the Clip-Art generation methodology. However, we do observe a substantial improvement in accuracy for 3D Detection (12%) and keypoint estimation (8%) in severe occlusions compared to Occ-Net and 3DRCNN. This can be attributed to the novel 3D clip-art based supervision for both the self-occlusion and occlusion-by-others cases.

5.4 Conclusion and Limitations

Conclusions: We demonstrated our holistic approach to amodal 3D reconstruction on several datasets. The occlusion categorization method outperforms previous heuristics making sure that the hybrid training data we generate is physically accurate. We demonstrated successful 3D reconstruction at busy urban scenes captured from a variety of view points and distances including traffic-cams, hand-held cameras and under different lighting conditions including night. Our framework can be used in a variety of smart city applications. For example, reliable amodal 3D reconstruction of vehicles will permit vehicle-based analytics (e.g., gross counts, speed estimation, etc.) that may supplement best practices used by urban planners.

Limitations: The method applies to one camera at a time and more research is needed for generalization. The method assumes that a mean shape model for the object is available. **Societal Impact:** Our framework could strongly benefit smart city applications. We do not perform any human subjects research or compute identifying information from the data as required by our Institutional Review Board (IRB).

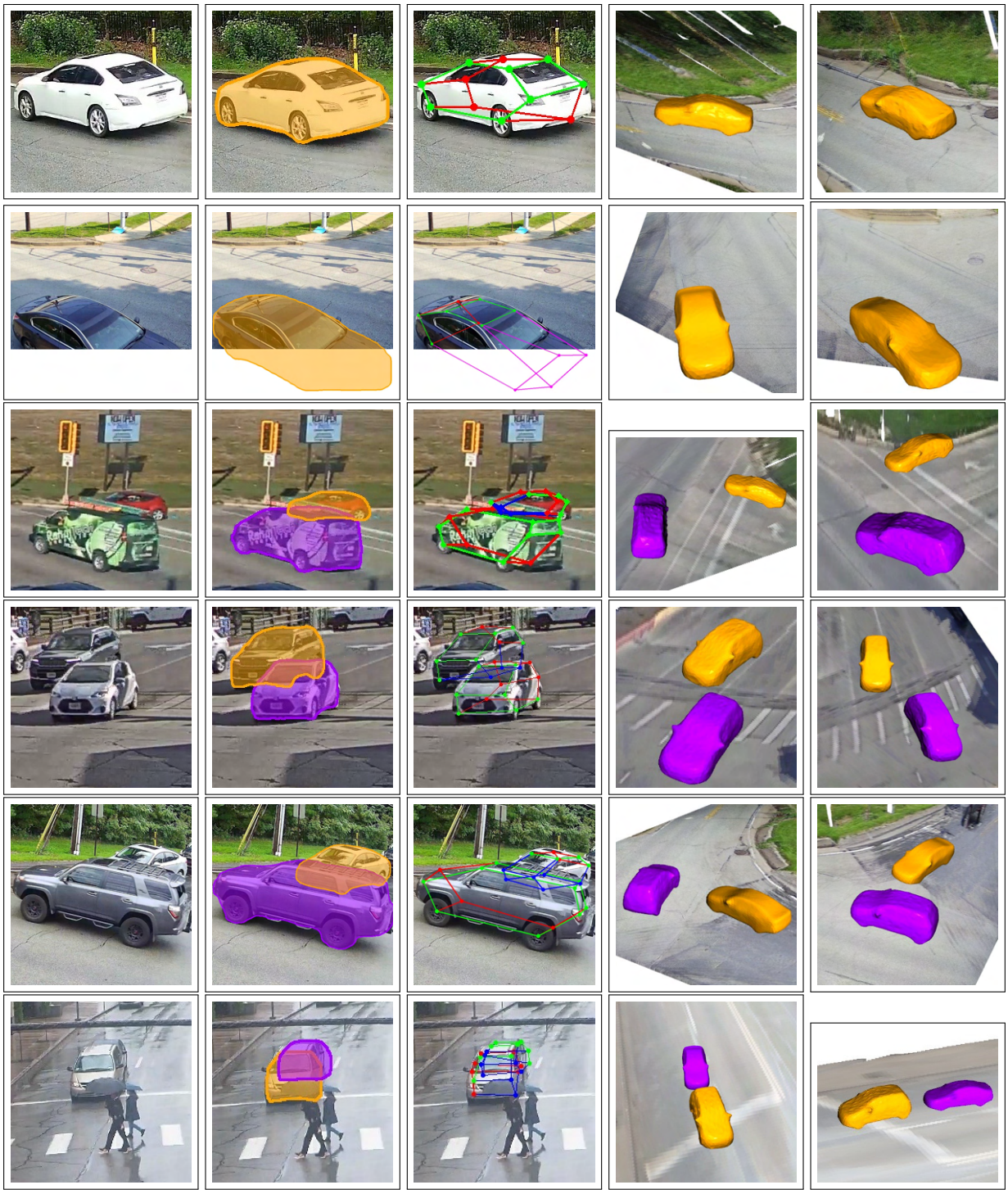


Figure 5.7: We show qualitative results of our method on multiple sequences of the WALT dataset. The input image **(col 1)** to the pipeline produces amodal segmentation mask **(col 2)** and keypoint locations **(col 3)**. Our method spits out 3D poses of the objects using an end-to-end a differentiable optimization to produce the 3D poses of the objects. We show the reconstructed 3D poses of the objects from two views **(col 4 and col 5)**.We observe accurate reconstruction of vehicles in wide-ranging poses and different occlusion configurations.

Chapter 6

End-to-End Occlusion Learning

This chapter addresses the problem of tracking and reconstructing in 3D articulated poses of multiple individuals seen in an arbitrary number of camera feeds. This task requires identifying the number of people in the scene, reconstructing their 3D body joints into consistent skeletons, and associating 3D body joints over time. We do not make any assumption on the number of available camera views and focus on real-world scenarios that often include multiple close-by interacting individuals, fast motions, self- and person-person occlusions. A key challenge in such scenarios is that people might strongly overlap and expose only a subset of body joints due to occlusions or truncations by image boundaries (Fig. 6.1), which makes it harder to reliably reconstruct and track articulated 3D human poses. Most multi-view strategies rely on multi-stage inference [10, 33, 160, 161, 162, 163, 164, 165] to first estimate 2D poses in each frame, cluster same person poses across views, reconstruct 3D poses from clusters based on triangulation, and finally link 3D poses over time [160, 163]. Solving each step in isolation is sub-optimal and prone to errors that cannot be recovered in later stages. This is even more true for monocular methods [107, 150, 166, 167, 168] where solving each step in isolation often represents an ill-posed problem.

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

Our main contribution is a novel spatio-temporal formulation that allows simultaneous 3D body joint reconstruction and tracking of multiple individuals. In contrast to the multi-person 3D pose estimation approach of [169] who similarly aggregate per frame information in 3D voxel space, we address a more challenging problem of multi-person 3D pose tracking and propose end-to-end person-specific representation learning. TesseTrack does not make assumptions on the available number of camera views and performs reasonably well even in the purely monoc-

Figure 6.1: We illustrate the output of Tessetrack on the Tagging sequence. The top two row potray the projections of keypoints on two views, while the bottom row shows the 3D pose tracking. Observe smooth tracking of people in the wild with moving cameras for long duration of time.

ular setting. Remarkably, using only a single view allows achieving similar MPJPE 3D joint localization error compared to the five-view setting of [169], while using the same five-view setting results in $2.4\times$ reduction in MPJPE error (Sec. 6.3). In contrast to the multi-person 2D pose tracking method of [141] who rely on short-term spatio-temporal representation learning, our approach operates on the aggregated spatio-temporal voxel volume and provides a richer hypothesis comprising of tracked 3D skeletons.

Our second contribution is a novel learnable tracking formulation that allows extending person-specific spatio-temporal representation learning to arbitrary-long sequences. In contrast to [141] who use a heuristic pairwise tracking score based on pose distance and perform matching using the Hungarian method, we rely on an attention aggregation layer and a differentiable representation matching layer based on the Sinkhorn algorithm. Importantly, we match person-specific representations instead of the determined body pose tracklets, which allows to learn more expressive representations. In Sec. 6.3 we demonstrate that the proposed learnable tracking formulation not only improves tracking accuracy but also improves joint localization.

Our third contribution is a novel framework for the evaluation of multi-person articulated 3D pose tracking. Experimental evaluation on the Panoptic dataset [164] shows that TesseTrack achieves significant improvements in per-joint tracking accuracy compared to strong baselines.

Finally, our fourth contribution is an in-depth ablation study of the proposed approach and thorough comparisons to current methods on several standard benchmarks. In Sec. 6.3 we demonstrate that proposed design choices result in significant accuracy gains, thereby establishing a new state of the art on multiple datasets.

6.1 Related Work

Person 3D Pose Estimation methods can be sub-divided into multi-view and monocular approaches. Multi-view approaches often rely on triangulation [57] of per view 2D poses to determine a 3D pose [160, 161, 164]. To improve robustness to 2D pose estimation errors, [170, 171] jointly reason over 2D poses seen from multiple viewpoints. Recent monocular approaches typically lean on powerful neural networks to mitigate the ambiguity of recovering 3D from 2D joint locations [10, 107, 108, 109, 110, 171, 172, 173]. [107, 172] directly regress 3D poses from 2D joint locations using deep networks. While being quite simple, they suffer from inaccuracies of 2D joint localization and the fact that appearance is not used during 3D pose prediction. [10, 108, 173, 174] intend to overcome these limitations by predicting a 3D volumetric representations from images: [174] augments 2D detection heatmaps with latent 3D pose features to predict 3D pose, [10] projects 2D feature maps to 3D volume and processes the volume to predict 3D joint locations. Similarly to [10, 108, 173, 174], we cast per-frame deep learning features from single or multiple views into a common discretized space. However, we address a more challenging problem of multi-person 3D pose tracking and process 4D spatio-temporal volumes to compute person-specific representations that allow to predict spatially and temporally consistent skeletons of multiple people. Our method is also related to [109, 110] who perform spatio-temporal representation learning optimized specifically for monocular case by introducing occlusion-aware training and spatio-temporal pose discriminator [109]. In contrast, our approach was not yet tuned to a monocular case and thus is expected to improve when using similar strategies.

Multi-person 3D Pose Estimation methods typically split the problem into 2D joint grouping in single frames and 3D pose reconstruction. 2D grouping is done using bottom-up [175, 176, 177, 178] or top-down [40, 179] strategies. In multi-view scenarios, recent approaches typically rely on triangulation of 2D poses of the same individual to reconstruct 3D poses [161, 165], while earlier methods extend pictorial structures model to deal with multiple views [162, 163, 180]. Independently solving 2D pose estimation, multi-view matching and triangulation are prone to errors. [169] project per view 2D joint heatmaps into a voxelized 3D space and directly detect people and predict their 3D poses in this space. Monocular approaches [181, 182] encode 2D and 3D pose features and jointly decode 3D poses of all individuals in the scene. Encoding the pose for all joints/limbs of the full-body, regardless of available image evidence, leads to potential encoding conflicts when similar body parts of different subjects overlap. Similar to [169] we cast per-frame feature maps into a voxelized 3D space and follow a top-down approach which starts with detecting people in this space. However, we address a more challenging problem of multi-person 3D pose tracking, which requires reasoning in spatio-temporal volumes extracted around person detections and merging extracted person-specific representations to reliably reconstruct and track 3D skeletons in arbitrarily long sequences. In contrast to [169] and similarly to [181,

182] our approach can operate in a purely monocular setting. However, unlike [181, 182] our approach does not suffer from encoding conflicts, since we cast feature maps into a common voxelized 3D space.

Multi-person 3D Pose Tracking was only addressed by few approaches [160, 166, 183, 184]. The multi-view approach of [160] follows a multi-stage inference where 2D poses are first predicted per frame, same person 2D poses are triangulated across views to recover 3D poses which are finally linked over time. In contrast, our formulation operates in a common spatio-temporal volume, is end-to-end learnable, and is not restricted to the multi-view setting only. An earlier monocular approach [166] relies on 2D tracking-by-detection and 2D-to-3D lifting to track 3D poses of walking pedestrians with a little degree of articulation. In contrast, we do make no assumptions about the type of body motions or people activities and address a harder problem of multi-person *articulated* 3D pose tracking. [183] compute per frame 2D and 3D pose and shape hypothesis and perform joint space-time optimization under scene constraints to reconstruct and track 3D poses. [184] encodes per frame 2D and 3D pose features and identities for all visible body joints of all people and employs a fully-connected deep network to decode features into complete 3D poses, followed by a spatio-temporal skeletal model fitting. In contrast, to [183, 184] who resort to a piece-wise trainable strategy, our approach is end-to-end trainable and thus can propagate people detection, tracking, and pose estimation errors back to input image pixels. Furthermore, our formulation seamlessly incorporates additional views, if available, to boost accuracy. We envision though that similar spatio-temporal model fitting strategies as in [183, 184] can be used to refine the output of our method.

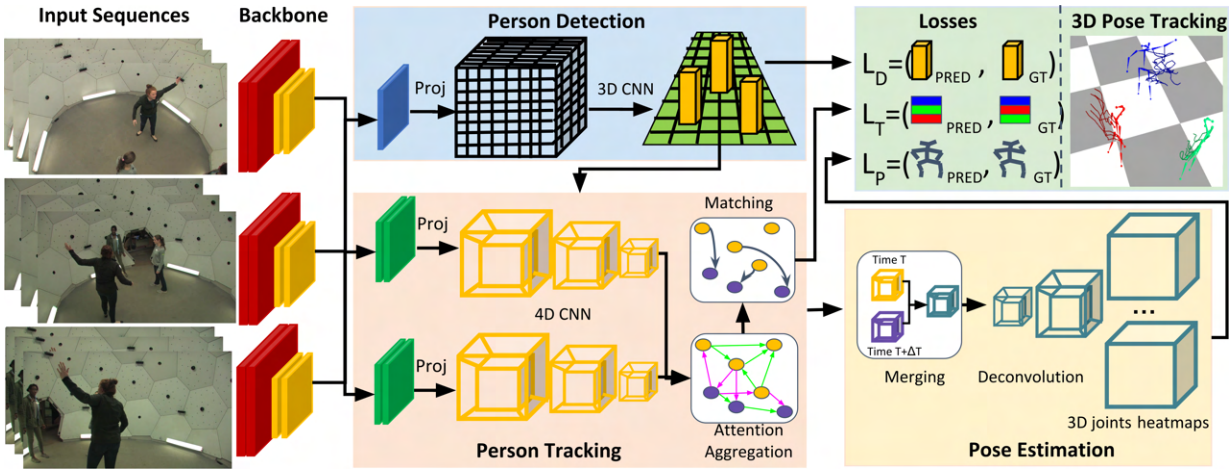


Figure 6.2: The complete pipeline of tessetrack has been illustrated. Initially, the video feed from multiple cameras is passed through shared HRNet to compute the features required for detection and 3D pose tracking. The final layer of the HRNet is passed through a 3D convolution to regress to the center of the human 3D bounding boxes. Each of the hypotheses is combined with the HRNet final layer to create a spatio-temporal Tube called tesseract. We use a learnable 3D tracking framework for a person association over time using spatio-temporal person descriptors. Finally, the associated descriptors are passed through deconvolution layers to infer the 3D pose. Note that the framework is end-to-end trainable except for the NMS layer in the detection network.

6.2 TesseTrack: Multi-Person 3D Pose Tracking

To learn person tracking and pose estimation in 3D we build multiple differentiable layers with intermediate supervisions. Our network is made up of three main blocks, each one with an associated loss. The first block is a person detection network in 3D voxel space (6.2.1). Given person detections, a 4D CNN extracts a spatio-temporal representation of each detected person over a short period of time. In order to track people, we then solve an assignment problem between the set of descriptors for two frames t and $t + \Delta t$ (6.2.2). All matched descriptors which overlap are then merged into a single descriptor which is finally deconvolved into a 3D pose for the person tracked at central frame (6.2.3).

6.2.1 Person Detection Network

Our approach starts with a multi-view person detection network (PDN) trained to detect people in 3D at a specific time instance. We use HRNet [179] as our backbone for extracting image-based features at each frame. We use the pre-final layer of the network and pass it through a single convolution layer to convert it into a feature map of size R. The feature maps coming from all the camera views are then aggregated into a 3D voxelized volume by an inverse image projection method, similarly to [10], with the critical difference that we don't fuse the 2D joint heatmaps in 3D but the richer feature vectors picked from the pre-final layer of HRNet. The voxel grid is initialized to encompass the whole space observed by the cameras. Using the camera calibration data, each voxel center is projected into the camera views. We aggregate all the feature vectors picked in image space by concatenating them and passing through a shallow network with a softmax layer. This produces a unique feature vector of size R. We thus end up with a data structure of size $R \times W \times H \times D$ dimensions, where W, H, D are the dimensions of the voxel grid and R is the dimension of the feature maps. We then apply 3D Convolutions to this volume to generate detection proposals. For each person, we train the network to detect its "center", which is defined as the midpoint between neck and center of the hips. The loss at each time t is expressed directly as a distance between the expected heatmap and the output heatmap, similarly to the CenterNet approach [185], except that our framework is in 3D instead of 2D:

$$L_D^t = \sum_{w=1}^{W} \sum_{h=1}^{H} \sum_{d=1}^{D} ||V_{Pred}^{w,h,d} - V_{GT}^{w,h,d}|| \tag{6.1}$$

We apply non-maximum suppression (NMS) on the 3D heatmaps and only retain the detections with large score.

6.2.2 Spatio-Temporal Descriptors and Tracking

For each detected person we create a spatio-temporal volume of fixed dimension centered on the person and use a 4D CNN to produce a short time description of the person around the detection frame. We call this spatio-temporal volume a *tesseract* as it is a 4D volume of size R×T×X×Y×Z, where T represents temporal window size and X,Y,Z are the dimensions of the cuboid centered on the detected person. The goal of extending the volume in time around

the detection frame is twofold. First, using a temporal context allows to better estimate the joint positions in the central frame, and especially to extrapolate/interpolate occluded joints or to handle pose or appearance ambiguities in a single frame. Second, extending a person's description in time generates a descriptor which overlaps with adjacent frames, hence producing descriptors that can be matched by similarity for tracking purposes.

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

Attention Aggregation. Before temporal matching, as illustrated in Fig 6.3, we pass the features into a Graph Neural Network to integrate contextual cues and improve the features distinctiveness. We use two types of undirected edges: self edges, connecting features belonging to the same time instance and cross edges, connecting features from adjacent time instances. We use a learnable message passing formulation to propagate the information in the graph. The resulting multiplex network starts with a high-dimensional state for each node and computes at each layer an updated representation by simultaneously aggregating messages across all incident edges for all nodes.

Let ${}^{(l)}\mathrm{x}_i^t$ be the intermediate representation for element i at time instance t at layer l. The message $m_{\epsilon\to i}$ is the result of the aggregation from all features of persons $j : (i,j) \in \epsilon$, where $\epsilon \in \epsilon_{self}, \epsilon_{cross}$. Following [186, 187, 188] we pass the input through multiple message passing updates to get a final matching descriptors given as linear projections. They are given as $f_i^t = W.^{(L)}\mathrm{x}_i^t + b.$ for features at time t and $f_i^{(t+\Delta t)} = W.^{(L)}\mathrm{x}_i^{t+\Delta t} + b.$ at time $t+\Delta t$, where W are the weights learned for the GNN.

Temporal Matching Layer. The final features of the attention module are passed through a trained matching layer, which produces an assignment matrix. For a given time instance t, we consider the features of N and M persons at time t and $t+\Delta t$ respectively. As in the standard bipartite graph matching formulation, an optimal assignment P is a permutation matrix which maximizes the total score $\sum_{i,j} S_{i,j}P_{i,j}$ where $S \in R^{M\times N}$ is a score matrix. We compute the similarity $S_{i,j}$ between the descriptor i at time t and the descriptor j at time $t+\Delta t$ using the inner product between descriptors $S_{i,j} =< f_i^t, f_j^{(t+\Delta t)} >$. As opposed to learned visual descriptors, the matching descriptors are not normalized, and their magnitude can change as per the feature during training to reflect the prediction confidence.

To let the network suppress some predicted persons (false detections) and to handle changes in the number of persons in the scene, we augment each set with a dustbin so that matching is always computed on a fixed length feature vectors. This leads to optimal assignments for each available detection and the rest unassigned dustbins always correspond one-to-one with the next time instance. Following recent end-to-end learning approaches which include an optimal assignment step, such as [186, 189], we use the Softassign algorithm [190] to solve the assignment problem by a differentiable operator. The Softassign algorithm is based on Sinkhorn iterative matrix balancing, which projects an initial score matrix into a doubly stochastic matrix by itera-

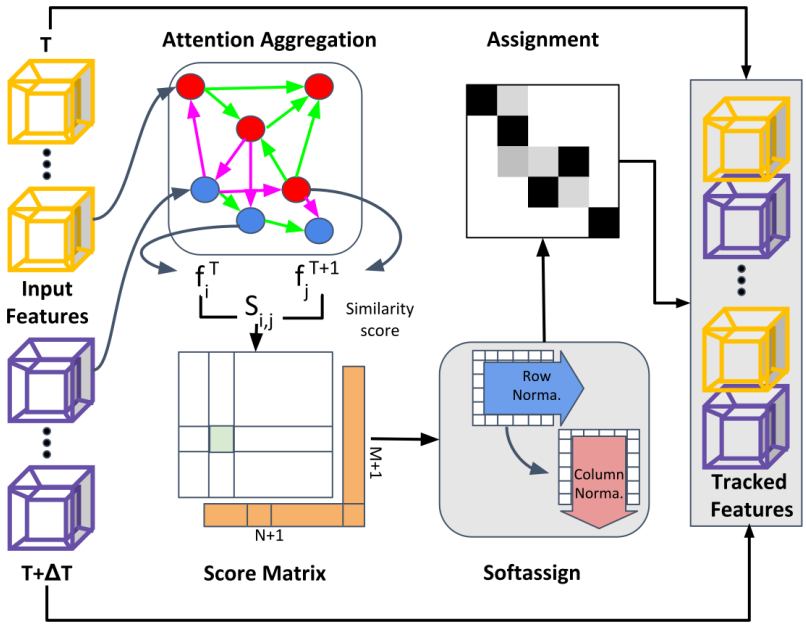


Figure 6.3: The learnable tracking framework. The input is the tesseract features for multiple detected humans at two different time instances. The output is an assignment matrix providing the correspondence between the detected persons at different times.

tively normalizing the matrix along rows and columns. When applied to the matrix $exp(S^{-}/\tau)$, it has been shown that Sinkhorn balancing corresponds to solving an entropy regularized problem which converges to the optimal assignment solution as τ goes to 0 [189]. The Softassign algorithm can be efficiently implemented on GPU by unrolling a fixed number of Sinkhorn iterations. After $T = 100$ iterations, we get a final score matrix P and the association for the detection i at time t is then extracted as $\arg\max_j P_{i,j}$.

Since all of the above layers are differentiable, we can train the tracking module in a supervised manner with respect to the ground truth. Given ground truth associations G between time t and $t + \Delta t$, the objective function to be minimized is the log likelihood of the assignment P:

$$L_T^t = -\sum_{(i,j)\in G} \log P_{i,j} \tag{6.2}$$

6.2.3 3D Pose Estimation

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

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

Tesseract deconvolution. The merged tesseract is finally passed through multiple 4D deconvolution layers to produce 3D heatmaps of person's joints at time t. If T^q_{Pred} denotes the 3D heatmap obtained for the joint q, the predicted joint position k^q_{Pred} is obtained by a soft-argmax operator, i.e. by a heatmap scores-weighted average of the voxel centers.

Similar to [10], we then combine two loss functions for the pose estimation task: a L1 distance computed on the keypoints positions and a loss on the response of the heatmap at the ground truth joint position:

$$L^{t,d}_P = \sum_{q=1}^{Q} \left[||k^q_{Pred} - k^q_{GT}||_1 - \beta . \log(T^q_{Pred}(k^q_{GT})) \right], \tag{6.3}$$

where Q is the number of joints. In the end, we train our network end-to-end to minimize the sum of the three losses defined above over time, the person detection loss L^t_D, the tracking loss L^t_T and the pose estimation loss $L^{t,p}_P$:

$$L = \sum_{t \in D} \left[L^t_D + \alpha L^t_T + \gamma \sum_{p \in TP(t)} L^{t,p}_P \right], \tag{6.4}$$

where D is the total duration of the sequence and $TP(t)$ represent the true positive detections at time t. The gradient is propagated back to the initial images, including through the HRNet backbone which is shared by the detection module and the tracking + pose estimation modules.

6.3 Experiments

6.3.1 Datasets and Metrics

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

Human3.6M [191] was captured from 4 cameras with a single human performing multiple actions. The dataset contains 8 actors performing 16 actions captured in controlled indoor settings. Motion capture was used to create ground truth 3D poses. We use 6 sequences to train and 2 sequences (S09, S11) to test our algorithm.

TUM Shelf [180] was captured indoors using 5 stationary cameras, with 4 people disassembling a shelf. The dataset provides sparse 3D pose annotations. Severe occlusions and random motion of the persons are the key challenges.

TUM Campus [180] was captured outdoors using 3 stationary cameras, with 3 people interacting on campus grounds. Similar to *Shelf*, it provides sparse 3D pose annotations. The dataset is challenging for 3D pose estimation due to a small number of cameras and wide baseline views.

CMU Panoptic [164] was built to understand human interactions in 3D. It contains 60 hours of data with 3D poses and tracking information captured by 500 cameras. We follow [169] and sample the same 5 cameras for evaluation, and use the same sequences for training. We split the training and testing sequences following [192].

Tagging [193] was captured in unconstrained environments where people are interacting in a social setting. There are no constraints on the motion of the cameras or the number of persons

during the capture. This "in the wild" setting makes this dataset particularly interesting for 3D pose tracking. However, since no GT pose annotations are available, we only use this dataset for qualitative evaluation.

Evaluation details. Mean Per Joint Position Error (MPJPE) [194] evaluates 3D joint localization accuracy in mm and represents L2 distance between the GT and predicted joint locations. Percentage of Correct Keypoints (3D-PCK) [161] provides a more global view on the accuracy of 3D pose estimation and is computed similarly to its 2D PCK counterpart [195]. On Human3.6M we follow [10] and provide all comparisons using root-centered MPJPE metric. On Panoptic dataset, we follow [169] and provide all comparisons using non-root-centered MPJPE.

Implementation Details. We train TesseTrack on 8 $V100$ GPUs with 32 GB memory each. As model does not fit into a single GPU, we share the *tesseract* convolutions and the backbone across 2 GPUs. Each GPU has propagation weights of a single time instance. The tracking and the deconvolution modules are shared among both GPUs. During testing, the model can be computed on a single GPU using sequential processing. A learning rate of 0.01 is used for all the modules. The Temporal Window (T) and the step size (Δt) used across the experiments is 5 unless specified. The module was trained with $Q = 19$ keypoints with the voxel volumes size 64. For all indoor experiments (*Panoptic*, *Human3.6M* and *Shelf*) we use a voxel volume of 12m and for outdoor experiments (*Campus*, *Tagging*) the size is 50m. For the tesseract a fixed volume size of 2.5m is used across all datasets. We use panoptic [164] keypoint format in all the experiments except for *Human3.6M* evaluation. As *Shelf*, *Campus* and *Tagging* datasets have no training GT annotations we use multi-view triangulation to obtain auto-annotated 3D labels to finetune PDN module only. We use HRNet [179] for feature extraction with $R = 32$ and $\alpha = 1$, $\beta = \gamma = 0.01$ in all experiments.

TesseTrack variants. We consider possible design choices for TesseTrack components: *F* - casting backbone's pre-final layer features into the voxelized space, *H* - using 2D joint detection heatmaps instead [169]; *T* - prediction using *tesseract* spatio-temporal module, *I* - instantaneous prediction per time instance instead; *D* - tracking using learned matcher, *G* - using heuristic matching using the Hungarian algorithm instead [141]; *L* - learned descriptor merging, *A* - simple heatmaps averaging instead [141]. This results into six TesseTrack variants: *HI*, *FI*, *FT*, *FTGA*, *FTGL*, *FTDL*. We also consider a simple tracking baseline that performs instantaneous prediction followed by the Hungarian matching of poses across time, which we denote as *FIG*.

6.4 Multi-Person 3D Pose Estimation

In this section, we evaluate TesseTrack on the task of multi-person 3D pose estimation. First, we demonstrate the improvements due to various design choices and show the robustness of TesseTrack to the number of available camera views on the *Panoptic* dataset. Then, we compare to the state of the art on *Panoptic*, *Shelf* and *Campus* datasets.

Ablation analysis on Panoptic dataset. MPJPE metric is used for comparison. Results are shown in Tab. 6.1.

Impact of Temporal Volumes. Tessetrack can operate without temporal information, which leads to -5.8 mm MPJPE loss on Panoptic dataset (*FI* vs. *FT* in Tab.6.6).

Model	HI	FI	FT	FTGA	FTGL	FTDL
MPJPE (mm)	16.3	13.8	8.0	8.1	7.5	**7.3**

Table 6.1: Ablation study of 3D pose reconstruction on the Panoptic dataset using non-root-centered MPJPE. We observe a clear increase in reconstruction accuracy with each additional improvement added to the model. Using the final layer of the backbone with a spatio-temporal descriptor-based network and learned matching and merging (FTDL) provides the best results in 3D reconstruction.

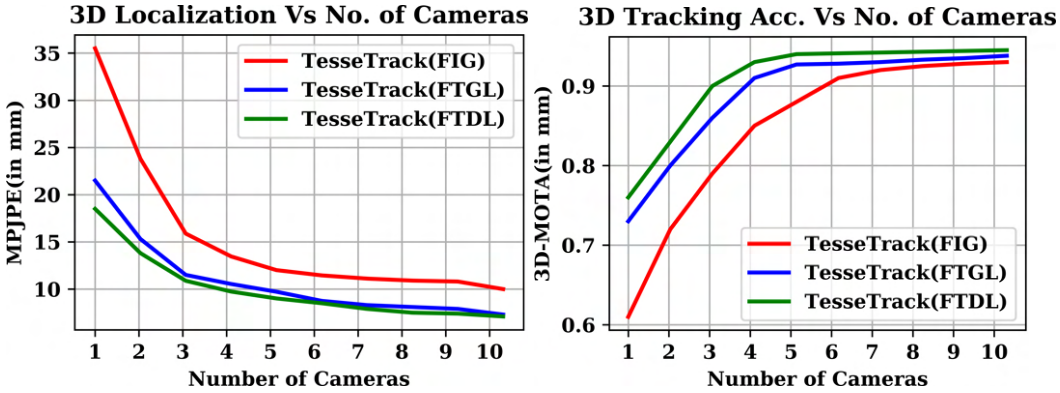


Figure 6.4: Impact of number of cameras on body joint localization error (MPJPE) (left) and pose tracking accuracy (3D MOTA) (right). Tessetrack (FTDL) shows the greatest advantage with lower number of cameras.

Robustness to number of cameras. We evaluate the robustness of the best found *FTDL* architecture to the number of available camera views. To that end, we vary the number of cameras available at each time instance from one (monocular) to ten. Results are shown in Fig. 6.4 (left). First, we observe that *FTDL* can achieve a reasonable accuracy of 18.9mm in the pure monocular scenario, although it was not specifically tuned for this setting. Intuitively, increasing the number of camera views results in a clear improvement in joint localization accuracy. Compared to *FTGL* we observe noticeable improvements for fewer cameras, which underlines the advantages of differentiable matching. Compared to *FIG*, both *FTGL* and *FTDL* achieve dramatic improvements in localization accuracy, which demonstrates the importance of incorporating temporal information.

FI vs. HI. We observe an improvement in reconstruction accuracy when using backbone features. This is because 2D heatmaps learned from 2D pose supervision might be missing out on crucial information required for accurate 3D joint reconstruction.

FT vs. FI. Most of the state-of-the-art methods use instantaneous 3D pose estimation and might struggle due to a lack of consistency of keypoints over time. TesseTrack enforces smoothness of the keypoints showing a clear improvement in 3D pose reconstruction.

FTGL vs. FTGA. Corresponding the human poses across time instances and merging them is generally a neglected problem. Most of the methods just average joint locations from different time instance inferences. We observe that relying on a learned merging framework at the descriptor level improves accuracy.

FTDL vs. FTGL. Differentiable matching module learns person-specific representations that are essential for reliable tracking. As expected, it improves over heuristic matching based on the

	Multi-View (5 views)		Monocular	
Method	Tu et al. [169]	TesseTrack	Tu et al. [169]	TesseTrack
MPJPE (mm)	17.7	**7.3**	51.1	**18.9**

Table 6.2: Comparison to the state of the art on the Panoptic dataset in multi-view and monocular settings. We show substantial improvement in reconstruction compared to the baseline method due to temporal consistency and end-to-end learnable framework.

Method	Actor-1	Actor-2	Actor-3	Total
Belagiannis et. [162]	93.5	75.7	84.4	84.5
Ershadi et. [165]	94.2	92.9	84.6	90.6
Dong et. [161]	97.6	93.3	98.0	96.3
Tu et al. [169]	97.6	93.8	98.8	96.7
TesseTrack	**97.9**	**95.2**	**99.1**	**97.4**

Table 6.3: Evaluation of 3D-PCK accuracy on the Campus dataset. TesseTrack ourperforms baselines due to the temporal consistency constraints.

Method	Actor-1	Actor-2	Actor-3	Total
Belagiannis et. [162]	75.3	69.7	87.6	77.5
Ershadi et. [165]	93.3	75.9	94.8	88.0
Dong et. [161]	98.8	94.1	97.8	96.9
Tu et al. [169]	**99.3**	94.1	97.6	97.0
TesseTrack	99.1	**96.3**	**98.3**	**98.2**

Table 6.4: Evaluation of 3D-PCK accuracy on the Shelf dataset. TesseTrack ourperforms baselines even in severe occlusions of the Shelf dataset.

Hungarian algorithm.

Comparison to the State of the Art on Panoptic dataset. We compare *FTDL* to the state-of-the-art approach of [169] in Tab. 6.2. TesseTrack achieves $2.4\times$ reduction in MPJPE in multi-view setting, and $2.7\times$ reduction in monocular scenario, which clearly shows the advantages of the proposed spatio-temporal formulation over [169].

Comparison to the State of the Art on TUM datasets. We use 3D-PCK metric and compare on TUM Campus in Tab. 6.3 and on TUM Shelf in Tab. 6.4. *FTDL* achieves significant improvements over the state of the art on both datasets.

6.4.1 Multi-Person Articulated 3D Pose Tracking

Most recent works on multi-person articulated 3D pose tracking [160, 183, 184] focus on evaluation of 3D pose reconstruction accuracy using MPJPE [194] or 3D-PCK [200]. However, this is not clear how existing methods advance actual body joint tracking accuracy in multi-person scenarios. We thus intend to fill in this gap and propose a set of novel evaluation metrics for

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

Table 6.5: 3D MOTA evaluations on the Panoptic dataset. Using an end-to-end learnable framework (*FTDL*) systematically improves the accuracy of 3D pose tracking across all keypoints.

Protocol #1	Dir	Disc	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD	Smok	Wait	Walk	WalkD	WalkT	Total
Monocular methods, (MPJPE, mm)																
Martinez et al. [196]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Iskakov et al. (monocular) [10]	41.9	49.2	46.9	47.6	50.7	57.9	41.2	50.9	57.3	74.9	48.6	44.3	41.3	52.8	42.7	49.9
Pavllo et al. [197]	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
Cheng et al. [110]	38.3	41.3	46.1	40.1	41.6	51.9	41.8	40.9	51.5	58.4	42.2	44.6	**41.7**	33.7	30.1	42.9
Cheng et al. [109]	**36.2**	**38.1**	**42.7**	**35.9**	**38.2**	**45.7**	**36.8**	42.0	45.9	**51.3**	**41.8**	**41.5**	43.8	**33.1**	**28.6**	**40.1**
TesseTrack	38.4	46.2	44.3	43.2	44.8	48.3	52.9	**36.7**	**45.3**	54.5	63.4	44.4	41.9	46.2	39.9	44.6
Multi-view methods, (MPJPE, mm)																
Martinez et al. (multi-view) [196]	46.5	48.6	54.0	51.5	67.5	70.7	48.5	49.1	69.8	79.4	57.8	53.1	56.7	42.2	45.4	57.0
Pavlakos et al. [198]	41.2	49.2	42.8	43.4	55.6	46.9	40.3	63.7	97.6	119.0	52.1	42.7	51.9	41.8	39.4	56.9
Kadkhodamohammadi & Padoy [199]	39.4	46.9	41.0	42.7	53.6	54.8	41.4	50.0	59.9	78.8	49.8	46.2	51.1	40.5	41.0	49.1
Iskakov et al. [10]	19.9	20.0	18.9	18.5	20.5	19.4	18.4	22.1	22.5	28.7	21.2	20.8	19.7	22.1	20.2	20.8
TesseTrack (FI)	18.0	19.8	19.9	19.0	20.1	17.6	21.1	23.7	26.8	20.6	20.0	19.5	19.2	21.7	18.6	20.4
TesseTrack	**17.5**	**19.6**	**17.2**	**18.3**	**18.2**	**17.7**	**18.0**	**18.0**	**20.5**	**20.3**	**19.4**	**17.2**	**18.9**	**19.0**	**17.8**	**18.7**

Table 6.6: 3D pose reconstruction accuracy of different methods on the Human3.6M dataset using root-centered MPJPE metric and *Protocol #1* from [10].

multi-person articulated 3D pose tracking. To that end, we build on the popular Multiple Object Tracking (MOT) [201] and articulated 2D pose tracking metrics [202] and extend them to the 3D pose use case. The proposed metrics require predicted 3D body poses with track IDs. First, for each pair of (predicted pose, GT pose) 3D-PCK is computed. Predicted and GT poses are matched to each other by a global matching procedure that maximizes per pose 3D-PCK. Finally, Multiple Object Tracker Accuracy (MOTA), Multiple Object Tracker Precision (MOTP), Precision, and Recall metrics are computed.

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

Impact of temporal representations on tracking. Results are shown in Tab. 6.5. Using temporal person descriptors (*FTDL* and *FTGL*) significantly improves tracking accuracy compared to instantaneous person descriptor (*FIG*). Using a end-to-end learnable tracking framework (*FTDL*) instead of a Hungarian matching algorithm (*FTGL*) further improves tracking accuracy. This can be attributed to the fact that the learnable descriptors matching can distinguish interacting people much better than graph-based tracking methods.

Robustness to number of cameras. We analyze the accuracy of 3D pose tracking with respect to a varying number of cameras. Results are shown in Fig. 6.4 (right). While an increasing number of cameras allows improving the accuracy of all variants, we observe that relying on spatio-temporal representation learning results in significant tracking accuracy improvements specifically in the few cameras mode (*FTDL* and *FTGL* vs. *FIG*). Furthermore, using a learnable tracklet matcher (*FTDL*) results in consistent increase in tracking accuracy over a wide range of number camera views. Both observations underline the advantages of the proposed formulation when only a few cameras are available. Finally, in the pure monocular setting, *FTDL* achieves a reasonable 76% 3D MOTA accuracy, despite not being specifically tuned in this setting. We

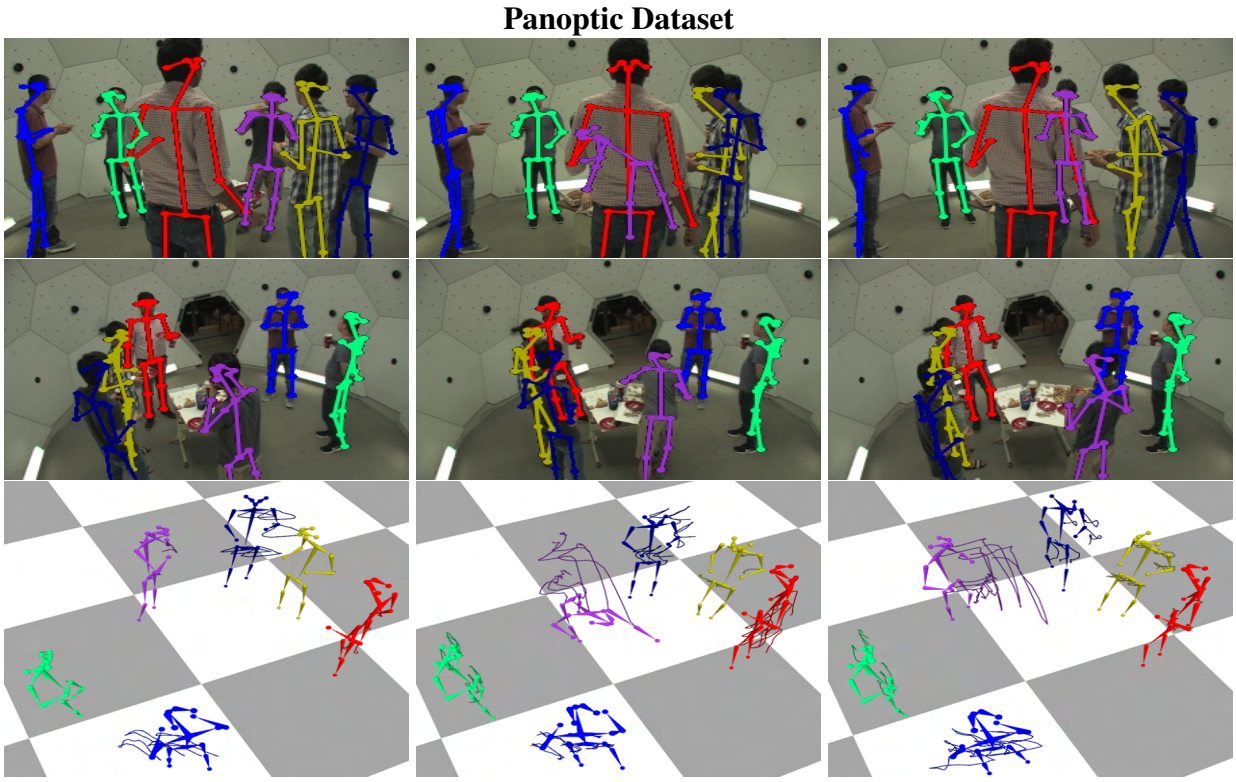


Figure 6.5: Qualitative results on Panoptic datasets. TesseTrack can track people in the wild as well as when interacting in close proximity.

envision that incorporating scene constraints and performing spatio-temporal articulated model fitting [183, 184] should significantly boost the accuracy of TesseTrack in monocular setting.

6.4.2 Single Person 3D Pose Estimation

We compare to the state-of-the-art methods on Human 3.6M using the MPJPE metric under *Protocol #1*.

Multi-View scenario. Comparison to multi-view approaches is shown in Tab. 6.6 (bottom). TesseTrack clearly improves over the state of the art, which underlines the advantages of the proposed spatio-temporal formulation. Specifically, using temporal consistency improves the joint localization accuracy for ambiguous poses like sitting down and walking a dog. We conclude that temporal constraints boost reconstruction accuracy in challenging actions.

Monocular scenario. Comparison to monocular methods is shown in Tab. 6.6(top). Despite not being specifically tuned for the monocular scenario, TesseTrack without bells and whistles outperforms most of the monocular approaches [109, 110]. Both [109, 110] also rely on spatio-temporal representation learning, but introduce occlusion-aware training which proved to be very useful specifically in monocular case, while [109] further reduce the error by adding a spatio-temporal discriminator to verify pose plausibility. Both improvements are orthogonal to our approach and thus can be incorporated to improve monocular case.

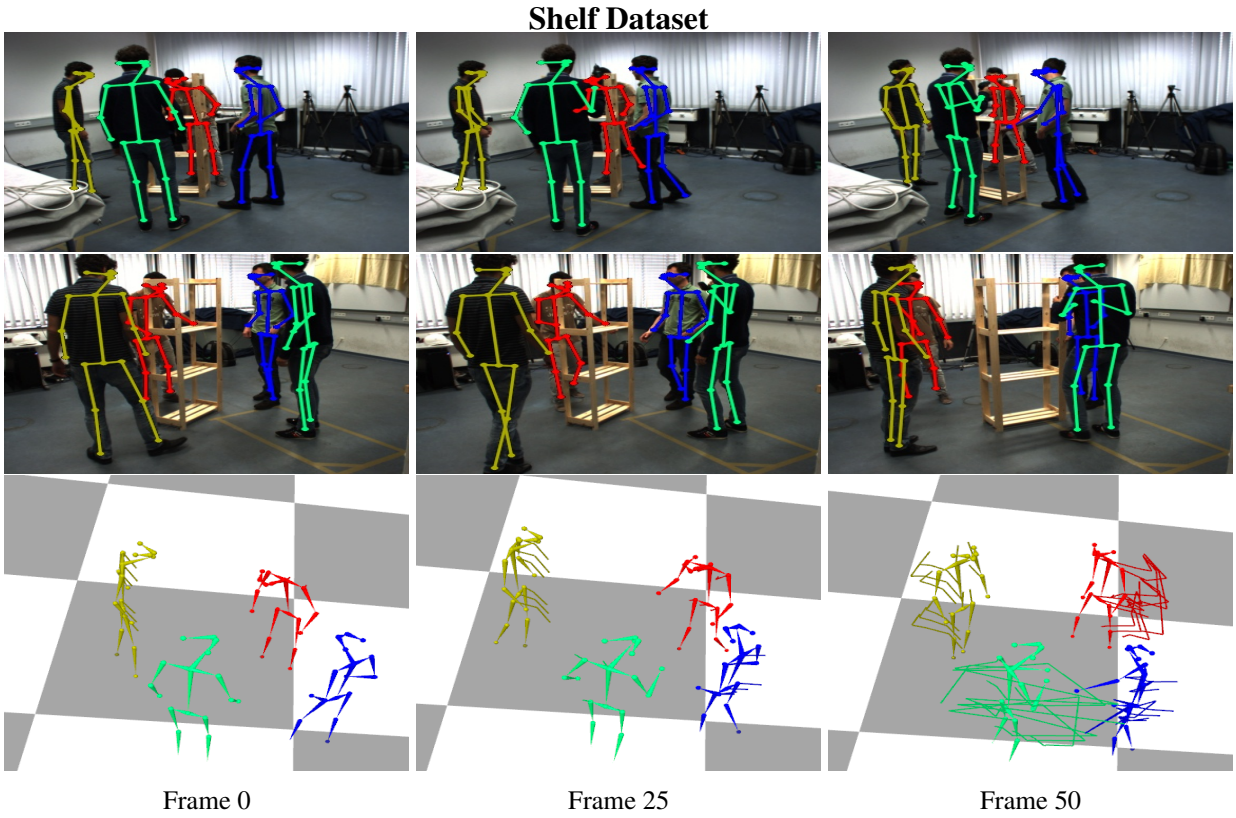


Figure 6.6: Qualitative results on Shelf datasets. TesseTrack can track people in the wild as well as when interacting in close proximity.

6.5 Conclusion

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

Limitations The algorithm is heavily dependent on extensive compute using 4D convolutions and the temporal data uses up a lot of GPU memory and needs to be optimized to work with lower memory footprint.

Chapter 7

Conclusion and Future Work

7.1 Analysis of Pros and Cons of Each Chapter

This thesis has shown different supervision signals for learning occlusions in different representations. We will briefly discuss the advantage and disadvantages of using different supervision signals and the situation to use them.

If you want to train for self-occlusions for data in the wild generally multi-view data is essential for gaining the supervision signal for regions occluded by the object. In the Occlusion-Net framework we explore such constraints to learn for self-occlusions. here multi-view data can be produced either from multi-camera setup or from video data.

Similarly when the data is only an image and we want to infer the complete region of objects i.e. amodal representation of objects using longitutdinal data as supervision can easily disentangle such occlusions. We found that data augmentation by copy and paste enhances the accuracy of the amodal segmentation from single view video as shown from the WALT method.

By combining the above two methodology we will be able to get a complete representation of the objects in severe occlusions and if an approxiamte shape of the object is provided or learned that can be used to infer the 3D space of the object automatically. Here the 3d location is generally only inferred from the visible regions of the object but using the amodal shapes we will be able to infer the shape of the object more accurately as shown in Chap 5.

Finally if we are given the multi-view video data we can automatically learn the 3D pose and track them using an unified learning framework. here the occlusions are automatically learned in an end-to-end fashion because of the spatio-temporal constraints in 3d. We explored this direct using the chap 6.

7.2 Joint Multi-View and Longitudinal Constraints

We have explored two different methodology for learning occlusions. The first was using multi-camera based multi-views constraints to learn occlusions. This has a severe bottle neck as capturing such datasets in real world is challenging and the computation time to do reconstruction is a very costly step. On the contrary the longitudinal data is very easy to capture but faces issue in generalizing as the number of views are minimal and cannot be generalized easily. We plan on combining the best of both world by using a multi-view longitudinal dataset captured over years

of time to do accurate reconstruction and supervise occlusions to automatically improve single view detectors. For the future work, these algorithms open door for different future directions and interesting applications. For example the Carfusion dataset can be used to learn dynamic novel view synthesis in the real world data using spatio-temoral frameworks which can be used by different methods.

7.3 Occlusions for in-the-Wild Object Categories

We live in a dynamic and open world – over short time periods, objects move and vary their shapes under the constraints of physics; over the long term, novel objects are created and new scenes are formed. Three-dimensional perception in such dynamic and open environments has been a longstanding problem in computer vision and machine learning, with tremendous impact in real life. With the current proposal, we plan to solve the problem of **generic** object 3D reconstruction from **unlabeled** videos, to create a system for capturing the dynamic 3D world. We can explore an new regime: *Can one reconstruct dynamic 3D structures from unlabeled videos without relying on strong shape or semantic priors?*

We can address the following tasks to tackle the problem of open world dynamic 3D reconstruction from **unlabeled videos**: (1) How does one learn deformable 3D shape templates from videos without relying on strong shape priors? (2) How does one segment never-before-seen objects from a video? (3) Can we use longitudinal self-supervision to improve reconstruction?

Learning 3D Shape Templates from Videos: To extract 3D shapes of objects, prior works either rely on 3D data – building or learning 3D shape models from RGBD scans, or learn category-specific 3D models from image collections with 2D annotations. However, depth data are generally difficult to acquire and scale-up due to specialized sensor availability. Although image collections of the specific object categories are relatively easy to obtain, a single image of an object does not provide enough constraints to reconstruct the full 3D shape at test time. Instead of inferring 3D shape from category-specific image collections, the proposed work builds a library of shape models from longitudinal observation of a single video of an object, or multiple videos of similar objects. one key hypothesis is that recent progress in differentiable rendering and optical flow allows one to recast the problem as analysis-by-synthesis task, solving the inverse graphics problem of recovering the 3D shape and trajectory, camera parameters as well as space-time deformations of an object that fit observed flow measurements.

Open-world Object Discovery from Videos: While one can build accurate detectors for many categories of objects, class-specific detectors rely heavily on appearance cues and categories present in a training set. Consider a trash can that falls on the street; current *closed-world* detectors will not likely be able to model all types of moving debris. This poses severe implications for robustness of object segmentation in the open-world [203]. The proposed work follows classic work on motion-based perceptual grouping from an observed motion field [204, 205, 206], and extend these work that segments rigid bodies using geometric consistency in two-frames [207] to videos. We can exploit the reconstructions from the previous stage for real world objects on the continuously captured data at city scale. We can plan to further show that the data captured for long duration (longitudinal) can be used as self-supervision in improving the accuracy of reconstruction for dynamic objects.

7.4 Occlusion Uncertainty Reduction

Uncertainty in prediction of occluded object boundaries even when the object is stationary represents that the network has high variance in the values being predicted. We can attempt to reduce the uncertainty of occluded object predictions in the network and build a self-supervised framework to reduce such uncertainty.

Funding Acknowledgements

This work was funded in parts by Heinz Endowments, US DOT RITA (University Transportation Center and Mobility 21 Center), NSF #CNS-1446601, DARPA REVEAL Phase 2 contract HR0011-16-C-0025, ARL Grant W911QX20F016, NSF CNS-2038612, and DOT RITA Mobility-21 Grant 69A3551747111, a Qualcomm Innovation Fellowship, NSF Grants IIS-1900821 and CNS-2038612, DOT RITA Mobility-21 Grant 69A3551747111, a PhD fellowship from Amazon Go.

Bibliography

[1] F. Li, N. D. Reddy, X. Chen, and S. G. Narasimhan, "Traffic4d: Single view longitudinal 4d reconstruction of repetitious activity using self-supervised experts," in *IEEE Intelligent Vehicles Symposium*, 2021. (document), 5.2, 5.1.2

[2] C. Li, M. Z. Zia, Q. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with intermediate concepts," *CoRR*, vol. abs/1801.03399, 2018. [Online]. Available: http://arxiv.org/abs/1801.03399 (document), 2.1, **??**, 2.1

[3] 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. (document), 3, 3.1, 3.2.2, 3.3.2, 3.3.3, **??**, 3.1, 5, 5.2, 5.2

[4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *TPAMI*, vol. 24, no. 5, pp. 603–619, 2002. (document), 3.2.3, 3.3.2, 3.2

[5] W. Wang and M. A. Carreira-Perpinán, "Manifold blurring mean shift algorithms for manifold denoising," in *CVPR*, 2010. (document), 3.3.2, 3.2

[6] H. Xu, Y. Zhou, W. Lin, and H. Zha, "Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage," in *ICCV*, 2015. (document), 3.1, 3.2.3, 3.3.2, 3.2

[7] L. Qi, L. Jiang, S. Liu, X. Shen, and J. Jia, "Amodal instance segmentation with kins dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3014–3023. (document), 4, 4, 4.2a, 4.2, **??**, 5

[8] Y. Zhu, Y. Tian, D. Metaxas, and P. Dollár, "Semantic amodal segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1464–1472. (document), 4, 4, 4.2a, 4.2, 5

[9] N. D. Reddy, R. Tamburo, and S. G. Narasimhan, "Walt: Watch and learn 2d amodal representation from time-lapse imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 9356–9366. (document), 5, 5.2, 5.2, 5.2, 5.3

[10] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7718–7727. (document), 4, 6, 6.1, 6.2.1, 6.2.3, 6.3.1, **??**, **??**, 6.6

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755. 1, 4, 5.2

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convo-

lutional neural networks," in *NIPS*. 1, 4

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 4

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 4

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017. 1.2, 2, 2, 2.2.1, 2.2.3, 2.3, 3, 4, **??**, 5.2

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *TPAMI*, 2010. 2

[17] 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. 2

[18] 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. 2

[19] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *CVPR*, 2008. 2, 4

[20] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *ICCV*, 2015. 2, 4

[21] 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*, 2015, pp. 4705–4713. 2, 4

[22] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 484–501, 2017. 2, 4

[23] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Category-specific object reconstruction from a single image," in *CVPR*, 2015. 2, 4

[24] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," *CoRR*, vol. abs/1803.07549, 2018. [Online]. Available: http://arxiv.org/abs/1803.07549 2, 4

[25] R. Fransens, C. Strecha, and L. Van Gool, "A mean field em-algorithm for coherent occlusion handling in map-estimation prob," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 300–307. 2, 4, 5

[26] T. Gao, B. Packer, and D. Koller, "A segmentation-aware object detection model with occlusion handling," in *CVPR 2011*. IEEE, 2011, pp. 1361–1368. 2, 4, 5

[27] C. Li, M. Z. Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep su-

pervision with shape concepts for occlusion-aware 3d object parsing," *arXiv preprint arXiv:1612.02699*, 2016. 2, 2.1, 2.2.2, 2.3.1, 3.2.1, 4

[28] S. Schulter, M. Zhai, N. Jacobs, and M. Chandraker, "Learning to look around objects for top-view representations of outdoor scenes," in *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 4, 5

[29] X. Zhou, S. Leonardos, X. Hu, and K. Daniilidis, "3d shape estimation from 2d landmarks: A convex relaxation approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4447–4455. 2

[30] A. Vedaldi and A. Zisserman, "Structured output regression for detection with partial truncation," in *Advances in neural information processing systems*, 2009, pp. 1928–1936. 2

[31] M. Z. Zia, M. Stark, and K. Schindler, "Towards scene understanding with detailed 3d object representations," *IJCV*, 2015. 2, 2, 2.1, **??**

[32] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," *CoRR*, vol. abs/1704.06254, 2017. [Online]. Available: http://arxiv.org/abs/1704.06254 2

[33] M. V. N Dinesh Reddy and S. G. Narasimhan, "Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicle," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*. IEEE, June 2018. 2, 2.1, 2.3.1, 2.3.1, 4, 6

[34] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1145–1153. 2, 2.1

[35] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Computer Vision and Pattern Regognition (CVPR)*, 2017. 2

[36] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 483–499. 2.1, 2.2.1

[37] 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*, 2016, pp. 4724–4732. 2.1

[38] 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. 2.1

[39] 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*. Springer, 2014, pp. 466–480. 2.1

[40] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," *arXiv preprint arXiv:1804.06208*, 2018. 2.1, 6.1

[41] P. Moreno, C. K. Williams, C. Nash, and P. Kohli, "Overcoming occlusion with inverse graphics," in *European Conference on Computer Vision*. Springer, 2016, pp. 170–185.

2.1

[42] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *CVIU*, 1995. 2.1

[43] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler, "Detailed 3d representations for object recognition and modeling," *TPAMI*, 2013. 2.1

[44] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman, "Single image 3d interpreter network," in *European Conference on Computer Vision*. Springer, 2016, pp. 365–382. 2.1

[45] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on computers*, vol. 100, no. 1, pp. 67–92, 1973. 2.1

[46] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. R. Soc. Lond. B*, vol. 200, no. 1140, pp. 269–294, 1978. 2.1

[47] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005. 2.1, 4

[48] L. Sigal, M. Isard, H. Haussecker, and M. J. Black, "Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation," *International journal of computer vision*, vol. 98, no. 1, pp. 15–48, 2012. 2.1

[49] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2224–2232. [Online]. Available: http://papers.nips.cc/paper/5954-convolutional-networks-on-graphs-for-learning-molecular-fingerprints.pdf 2.1

[50] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016. 2.1

[51] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *CoRR*, vol. abs/1312.6203, 2013. [Online]. Available: http://arxiv.org/abs/1312.6203 2.1

[52] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *CoRR*, vol. abs/1506.05163, 2015. [Online]. Available: http://arxiv.org/abs/1506.05163 2.1

[53] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *CoRR*, vol. abs/1606.09375, 2016. [Online]. Available: http://arxiv.org/abs/1606.09375 2.1

[54] L. Yi, H. Su, X. Guo, and L. J. Guibas, "Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation." in *CVPR*, 2017, pp. 6584–6592. 2.1

[55] O. Litany, T. Remez, E. Rodola, A. Bronstein, and M. Bronstein, "Deep functional maps: Structured prediction for dense shape correspondence," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 5660–5668. 2.1

[56] H. Maron, M. Galun, N. Aigerman, M. Trope, N. Dym, E. Yumer, V. G. Kim, and Y. Lipman, "Convolutional neural networks on surfaces via seamless toric covers," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 71, 2017. 2.1

[57] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2.2.1, 2.2.2, 6.1

[58] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015. 2.2.2

[59] A. Kundu, Y. Li, and J. M. Rehg, "3d-rcnn: Instance-level 3d object reconstruction via render-and-compare," in *CVPR*, 2018. 2.2.2

[60] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015. 2.3.1

[61] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009. 2.3.1, 5.1.2

[62] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1385–1392. 2.3.2, 3.3.2

[63] A. Kanazawa, D. W. Jacobs, and M. Chandraker, "Warpnet: Weakly supervised matching for single-view reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3253–3261. **??**

[64] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," in *Robotics: Science and Systems*, 2016. 2.3.2

[65] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2014, pp. 75–82. 2.3.2

[66] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *CVPR*, 2017. 3, 3.1

[67] 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, 3.1

[68] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012. 3

[69] 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. 3

[70] 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. 3

[71] 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. 3, 3.1

[72] S. Tulsiani, A.Efros, and J. Malik, "Multi-view consistency as supervisory signal for learning shape and pose prediction," in *CVPR*, 2018. 3

[73] J. Gwak, C.Choy, M.Chandraker, Garg, and Savarese, "Weakly supervised 3d reconstruction with adversarial constraint," in *3DV*, 2017. 3

[74] N. D. Reddy, P. Singhal, V. Chari, and K. M. Krishna, "Dynamic body vslam with semantic constraints," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1897–1904. 3

[75] 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. 3, 3.3, 3.3.1

[76] C. J. de Frias, A. Al-Kaff, F. M. Moreno, A. Madridano, and J. M. Armingol, "Intelligent cooperative system for traffic monitoring in smart cities," in *IVS*, 2020. 3.1

[77] 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. 3.1

[78] 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. 3.1

[79] P. Li, H. Zhao, and F. Cao, "RTM3D: real-time monocular 3d detection from object keypoints for autonomous driving," in *ECCV*, 2020. 3.1

[80] 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. 3.1

[81] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Occlusion-net: 2d/3d occluded keypoint localization using graph networks," in *CVPR*, 2019. 3.1, 3.2.1, 3.2.2, 3.3.2, 3.3.3, **??**

[82] Y. Chen, L. Tai, K. Sun, and M. Li, "Monopair: Monocular 3d object detection using pairwise spatial relationships," in *CVPR*, 2020. 3.1

[83] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Trajectory space: A dual representation for nonrigid structure from motion," *TPAMI*, 2011. 3.1

[84] T. Zhang, H. Lu, and S. Z. Li, "Learning semantic scene models by object classification and trajectory clustering," in *CVPR*, 2009. 3.1

[85] 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. 3.1

[86] 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. 3.1

[87] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised learning of probably symmetric deformable 3d objects from images in the wild," in *CVPR*, 2020. 3.1

[88] R. Yeh, Y.-T. Hu, and A. Schwing, "Chirality nets for human pose regression," in *NeurIPS*, 2019, pp. 8163–8173. 3.1

[89] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *ECCV*, 2018. 3.1

[90] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *CVPR*, 2018. 3.1

[91] J.-Y. Zhu, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *CVPR*, 2017. 3.1

[92] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *CVPR*, 2019, pp. 2566–2576. 3.1

[93] 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. 3.1

[94] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *CVPR*, 2018. 3.1

[95] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele, "Kinematic 3d object detection in monocular video," in *ECCV*, 2020. 3.1

[96] 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. 3.2.2, 3.3.2

[97] 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. 3.2.2

[98] 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. 3.3.2

[99] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893. 4

[100] 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*, 2014, pp. 580–587. 4

[101] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 012–10 022. 4, 4.1.1, 4.1.3, **??**, 5.1.3, 5.2

[102] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818. 4

[103] M. Ren and R. S. Zemel, "End-to-end instance segmentation with recurrent attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6656–6664. 4

[104] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[105] L. Ke, Y.-W. Tai, and C.-K. Tang, "Deep occlusion-aware instance segmentation with overlapping bilayers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 4019–4028. 4, 4, **??**, 5

[106] N. Silberman, L. Shapira, R. Gal, and P. Kohli, "A contour completion model for augmenting surface reconstructions," in *European Conference on Computer Vision*. Springer, 2014, pp. 488–503. 4

[107] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, 2017. 4, 6, 6.1

[108] G. Pavlakos, X. Zhou, and K. Daniilidis, "Ordinal depth supervision for 3d human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4, 6.1

[109] Y. Cheng, B. Yang, B. Wang, and R. T. Tan, "3d human pose estimation using spatio-temporal networks with explicit occlusion training," *arXiv preprint arXiv:2004.11822*, 2020. 4, 6.1, **??**, 6.4.2

[110] Y. Cheng, B. Yang, B. Wang, Y. Wending, and R. Tan, "Occlusion-aware networks for 3d human pose estimation in video," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, pp. 723–732. 4, 6.1, **??**, 6.4.2

[111] A. Wang, Y. Sun, A. Kortylewski, and A. L. Yuille, "Robust object detection under occlusion with context-aware compositionalnets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 645–12 654. 4, 4

[112] E. Hsiao and M. Hebert, "Occlusion reasoning for object detectionunder arbitrary viewpoint," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 9, pp. 1803–1815, 2014. 4, 5

[113] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Occlusion-net: 2d/3d occluded keypoint localization using graph networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4, 4, 5, 5.1.3, 5.2, 5.3

[114] B. Pepikj, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3286–3293. 4

[115] K. Ehsani, R. Mottaghi, and A. Farhadi, "Segan: Segmenting and generating the invisible," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6144–6153. 4, 5

[116] J. Tighe, M. Niethammer, and S. Lazebnik, "Scene parsing with object instances and occlusion ordering," in *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition, 2014, pp. 3748–3755. 4

[117] G. Ghiasi, Y. Yang, D. Ramanan, and C. C. Fowlkes, "Parsing occluded people," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2401–2408. 4, 5

[118] P. Follmann, R. König, P. Härtinger, M. Klostermann, and T. Böttger, "Learning to see the invisible: End-to-end trainable amodal instance segmentation," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1328–1336. 4, 5

[119] X. Zhan, X. Pan, B. Dai, Z. Liu, D. Lin, and C. C. Loy, "Self-supervised scene deocclusion," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, June 2020. 4, 5

[120] R. Guo and D. Hoiem, "Beyond the line of sight: labeling the underlying surfaces," in *European Conference on Computer Vision*. Springer, 2012, pp. 761–774. 4, 5

[121] X. Yuan, A. Kortylewski, Y. Sun, and A. Yuille, "Robust instance segmentation through reasoning about multi-object occlusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 11 141–11 150. 4, 5

[122] Y. Sun, A. Kortylewski, and A. Yuille, "Weakly-supervised amodal instance segmentation with compositional priors," *arXiv preprint arXiv:2010.13175*, 2020. 4

[123] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Amodal completion and size constancy in natural scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 127–135. 4

[124] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 430–446. 4, 5

[125] Y.-T. Hu, H.-S. Chen, K. Hui, J.-B. Huang, and A. G. Schwing, "SAIL-VOS: Semantic Amodal Instance Level Video Object Segmentation – A Synthetic Dataset and Baselines," in *Proc. CVPR*, 2019. 4, 5

[126] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets deep learning for car instance segmentation in urban scenes," in *British machine vision conference*, vol. 1, 2017, p. 2. 4, 5

[127] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–17, 2019. 4

[128] Epic Games, "Unreal engine." [Online]. Available: https://www.unrealengine.com 4

[129] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org 4, 4.2

[130] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*. Springer, 2016, pp. 102–118. 4

[131] P. Krähenbühl, "Free supervision from video games," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2955–2964. 4, 5

[132] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Webcam clip art: Appearance and illuminant transfer from time-lapse sequences," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, pp. 1–10, 2009. 1

[133] S. G. Narasimhan, C. Wang, and S. K. Nayar, "All the images of an outdoor scene," in *European conference on computer vision*. Springer, 2002, pp. 148–162. 1

[134] A. Kortylewski, J. He, Q. Liu, and A. L. Yuille, "Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8940–8949. 4

[135] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468. 4.1.1

[136] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. Int. Conf. Computer Vision (ICCV)*, 2019. 4.1.3

[137] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *TPAMI*, 2014. 5

[138] Z. Wang, B. Liu, S. Schulter, and M. Chandraker, "A parametric top-view representation of complex road scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 325–10 333. 5

[139] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "Hd maps: Fine-grained road segmentation by parsing ground and aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3611–3619. 5

[140] S. Sengupta, P. Sturgess, L. Ladickỳ, and P. H. Torr, "Automatic dense visual semantic mapping from street-level imagery," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 857–862. 5

[141] M. Wang, J. Tighe, and D. Modolo, "Combining detection and tracking for human pose estimation in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5, 6, 6.3.1

[142] C. Li, M. Zeeshan Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with shape concepts for occlusion-aware 3d object parsing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5465–5474. 5, 5.2, 5.2

[143] H. Glasl, D. Schreiber, N. Viertl, S. Veigl, and G. Fernandez, "Video based traffic congestion prediction on an embedded system," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2008, pp. 950–955. 5

[144] M. Naphade, S. Wang, D. C. Anastasiu, Z. Tang, M. Chang, Y. Yao, L. Zheng, M. S. Rahman, A. Venkatachalapathy, A. Sharma, Q. Feng, V. Ablavsky, S. Sclaroff, P. Chakraborty, A. Li, S. Li, and R. Chellappa, "The 6th ai city challenge," in *2022 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE Computer Society, June 2022, pp. 3346–3355. 5

[145] A. Aslam, "Detecting objects in less response time for processing multimedia events in smart cities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2044–2054. 5

[146] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5, 5.2, 5.2

[147] X. Zhan, X. Pan, B. Dai, Z. Liu, D. Lin, and C. C. Loy, "Self-supervised scene de-occlusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5

[148] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision*, March 2014, pp. 75–82. 5, 5.2, 5.2

[149] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015. 5

[150] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Occlusion-net: 2d/3d occluded keypoint localization using graph networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7326–7335. 5.1.2, **??**, 6

[151] H. Chen, P. Wang, F. Wang, W. Tian, L. Xiong, and H. Li, "Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5.1.3

[152] B. Chen, A. Parra, J. Cao, N. Li, and T.-J. Chin, "End-to-end learnable geometric vision by backpropagating pnp optimization," in *CVPR*, 2020. 5.1.3

[153] L. Ke, S. Li, Y. Sun, Y.-W. Tai, and C.-K. Tang, "Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision," in *European Conference on Computer Vision*. Springer, 2020, pp. 515–532. 5.1.3

[154] A. Nibali, Z. He, S. Morgan, and L. Prendergast, "Numerical coordinate regression with convolutional neural networks," *arXiv preprint arXiv:1801.07372*, 2018. 5.1.3

[155] Google, "Google Street View," https://www.google.com/streetview/. 5.2

[156] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5.2

[157] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019. 5.2

[158] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236. 5.2

[159] A. Kundu, Y. Li, and J. M. Rehg, "3d-rcnn: Instance-level 3d object reconstruction via render-and-compare," in *CVPR*, 2018. 5.2

[160] L. Bridgeman, M. Volino, J.-Y. Guillemaut, and A. Hilton, "Multi-person 3d pose estimation and tracking in sports," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 6, 6.1, 6.4.1

[161] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou, "Fast and robust multi-person 3d pose estimation from multiple views," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7792–7801. 6, 6.1, 6.3.1, **??**, **??**

[162] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3d pictorial structures revisited: Multiple human pose estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 1929–1942, 2015. 6, 6.1, **??**, **??**

[163] V. Belagiannis, X. Wang, B. Schiele, P. Fua, S. Ilic, and N. Navab, "Multiple human pose estimation with temporally consistent 3d pictorial structures," in *ECCVw*, 2014. 6, 6.1

[164] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social motion capture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3334–3342. 6, 6, 6.1, 6.3.1

[165] S. Ershadi-Nasab, E. Noury, S. Kasaei, and E. Sanaei, "Multiple human 3d pose estimation from multiview images," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15 573–15 601, 2018. 6, 6.1, **??**, **??**

[166] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. 6, 6.1

[167] G. Moon, J. Chang, and K. M. Lee, "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image," in *The IEEE Conference on International Conference on Computer Vision (ICCV)*, 2019. 6

[168] S. Li, L. Ke, K. Pratama, Y.-W. Tai, C.-K. Tang, and K.-T. Cheng, "Cascaded deep monocular 3d human pose estimation with evolutionary training data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 6

[169] H. Tu, C. Wang, and W. Zeng, "Voxelpose: Towards multi-camera 3d human pose estimation in wild environment." ECCV, 2020. 6, 6.1, 6.3.1, **??**, **??**, **??**, **??**, 6.4

[170] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele, "Multi-view pictorial structures for 3d human pose estimation," in *British Machine Vision Conference, BMVC*, 2013. 6.1

[171] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng, "Cross view fusion for 3d human pose estimation," in *International Conference on Computer Vision (ICCV)*, 2019. 6.1

[172] F. Moreno-Noguer, "3d human pose estimation from a single image via distance matrix regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6.1

[173] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3d human pose estimation in the wild: A weakly-supervised approach," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 6.1

[174] I. Habibie, W. Xu, D. Mehta, G. Pons-Moll, and C. Theobalt, "In the wild human pose estimation using explicit 2d features and intermediate 3d representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6.1

[175] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 6.1

[176] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6.1

[177] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6.1

[178] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *Advances in Neural Information Processing Systems*, 2017. 6.1

[179] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *CVPR*, 2019. 6.1, 6.2.1, 6.3.1

[180] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3d pictorial structures for multiple human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 6.1, 6.3.1

[181] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," in *3D Vision (3DV), 2018 Sixth International Conference on*. IEEE, sep 2018. [Online]. Available: http://gvv.mpi-inf.mpg.de/projects/SingleShotMultiPerson 6.1

[182] A. Zanfir, E. Marinoiu, M. Zanfir, A.-I. Popa, and C. Sminchisescu, "Deep network for the integrated 3d sensing of multiple people in natural images," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018, pp. 8410–8419. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/6a6610feab86a1f294dbbf5855c74af9-Paper.pdf 6.1

[183] A. Zanfir, E. Marinoiu, and C. Sminchisescu, "Monocular 3d pose and shape estimation of multiple people in natural scenes - the importance of multiple scene constraints," 2018. 6.1, 6.4.1

[184] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "XNect: Real-time multi-person 3D motion capture with a single RGB camera," vol. 39, no. 4, 2020. [Online]. Available: http://gvv.mpi-inf.mpg.de/projects/XNect/ 6.1, 6.4.1

[185] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE International Conference on Computer*

Vision, 2019, pp. 6569–6578. 6.2.1

[186] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4938–4947. 6.2.2

[187] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018. 6.2.2

[188] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. 6.2.2

[189] G. Mena, D. Belanger, S. Linderman, and J. Snoek, "Learning latent permutations with gumbel-sinkhorn networks," *arXiv preprint arXiv:1802.08665*, 2018. 6.2.2

[190] S. Gold, A. Rangarajan *et al.*, "Softmax to softassign: Neural network algorithms for combinatorial optimization," *Journal of Artificial Neural Networks*, vol. 2, no. 4, pp. 381–399, 1996. 6.2.2

[191] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013. 6.3.1

[192] H. Joo, T. Simon, M. Cikara, and Y. Sheikh, "Towards social artificial intelligence: Nonverbal social signal prediction in a triadic interaction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 873–10 883. 6.3.1

[193] M. Vo, E. Yumer, K. Sunkavalli, S. Hadap, Y. Sheikh, and S. Narasimhan, "Automatic adaptation of person association for multiview tracking in group activities," *TPAMI*, 2020. 6.3.1

[194] L. Sigal, A. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87, no. 1, pp. 4–27, Mar. 2010. 6.3.1, 6.4.1

[195] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 6.3.1

[196] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2640–2649. **??**, **??**

[197] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7753–7762. **??**

[198] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Harvesting multiple views for marker-less 3d human pose annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6988–6997. **??**

[199] A. Kadkhodamohammadi and N. Padoy, "A generalizable approach for multi-view 3d human pose regression," *Machine Vision and Applications*, vol. 32, no. 1, pp. 1–14, 2020. **??**

[200] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. [Online]. Available: http://gvv.mpi-inf.mpg.de/3dhp_dataset 6.4.1

[201] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831 [cs]*, 2016. [Online]. Available: http://arxiv.org/abs/1603.00831 6.4.1

[202] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 6.4.1

[203] A. Bendale and T. Boult, "Towards open world recognition," in *CVPR*, 2015. 7.3

[204] M. Irani and P. Anandan, "A unified approach to moving object detection in 2d and 3d scenes," *PAMI*, 1998. 7.3

[205] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *CVPR*, 2007. 7.3

[206] X. Xu, L. F. Cheong, and Z. Li, "3d rigid motion segmentation with mixed and unknown number of models," *PAMI*, 2019. 7.3

[207] G. Yang and D. Ramanan, "Learning to segment rigid motions from two frames," *arXiv preprint arXiv:2101.03694*, 2021. 7.3