Scaling up Camera Calibration and Amodal 3D Object Reconstruction for Smart Cities

Khiem Vuong

CMU-RI-TR-23-38

August 15, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Prof. Srinivasa G. Narasimhan, *chair*
Prof. Shubham Tulsiani
Yufei Ye

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

Smart cities integrate thousands of outdoor cameras to enhance urban infrastructure, but their automated analysis potential remains untapped due to various challenges. Firstly, the lack of accurate camera calibration information, such as its intrinsics parameters and external orientation, restricts the measurement of real-world distances from the captured video. To address this issue, we propose a scalable framework leveraging publicly available street-level imagery and map data to automatically reconstruct a metric 3D model of the surrounding scene, allowing for accurate calibration of in-the-wild traffic cameras around the world.

Secondly, the presence of occlusions poses significant challenges in object understanding. For example, objects in the scene may be partially occluded by other static or dynamic objects, truncated by the camera's field of view, or be self-occluded, i.e., only one side of an object is visible from a specific view. We present a holistic approach to handle such occlusions for amodal 3D shape reconstruction. The approach starts by 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 from traffic cameras. This dataset provides strong 2D and 3D self-supervision to a network that jointly learns amodal 2D keypoints and segmentations, which are then optimized to reconstruct 3D shapes under constraints provided by occlusion categories. 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 in-vehicle cameras, thus enhancing the potential of smart cities to utilize outdoor cameras for effective urban planning.

Acknowledgments

I would like to begin by expressing my gratitude and appreciation to my advisor, Srinivasa Narasimhan, for his unwavering support over the past two years. Your commitment to high research standards and your ability to cut through the clutter and provide clear insights have been truly inspiring. Perhaps more than anything, thank you for always being available, and I couldn't possibly ask for more.

I also extend my appreciation to my committee members, Shubham Tulsiani and Yufei (Judy) Ye, for their valuable time and feedback on this thesis presentation.

To my wonderful labmates and friends at CMU, in no particular order - Guanzhou, Tianyuan, Bowei, Karnik, Sriram, Gaurav(s), Anurag, Dinesh, Robert, Adithya, Mark, Mani, Shumian, Tom, Swami - and many others who have collaborated with me on various projects, I am grateful for the energy, insights, and countless conversations we have shared. Your presence has been a constant source of inspiration.

To my family, thank you for your constant support and the selfless sacrifices you have made, even during times when I may not have reached out as often as I should. Last but certainly not least, I want to thank my girlfriend Thu for bringing me so much joy during both this endeavor and the exciting future that lies ahead.

Contents

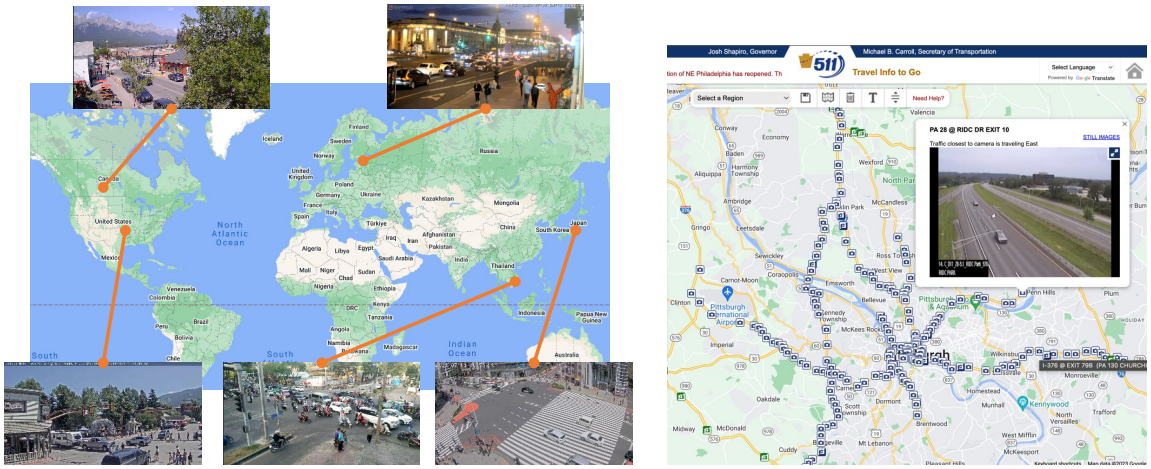
When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction



(a) Sample traffic video streams from YouTube

(b) Publicly available traffic cameras in Pittsburgh, PA (source: 511pa.com)

Figure 1.1: The ubiquity of publicly available CCTV cameras from all around the world, obtained either through a quick YouTube search (*left*) or provided by the government (*right*).

The decreasing costs of sensing devices have led to the widespread adoption of large-scale public camera networks in smart cities (see examples in Figure 1.1). As of 2016, there are over 350 million CCTV cameras installed worldwide [65]. These cameras are deployed by authorities to monitor traffic, aid in urban planning, and ensure city safety. Additionally, citizens install cameras for various purposes, such as monitoring

private properties and preventing theft. The ubiquity of CCTV cameras presents numerous opportunities for cities to gain real-time insights into road complexities. Cameras within these networks can track the interactions, trajectories, speeds, and densities of road users, providing valuable data that can inform decision-making for strategic urban planning. By leveraging this data, cities can optimize transportation systems, improve traffic flow, and reduce congestion, ultimately leading to safer, more efficient, and sustainable cities. However, effectively harnessing the potential of publicly available CCTV cameras remains a challenging task. In this thesis, we propose a comprehensive framework to tackle two important technical challenges. Our framework addresses two closely interconnected problems: 1) *Automatic scene reconstruction and camera calibration*, and 2) *Amodal object understanding under occlusion*. By developing novel and scalable solutions to these problems, we aim to unlock the full potential of public traffic camera networks and enable the development of useful applications for smart cities.

1.1 Automatic scene reconstruction and camera calibration

With the recent advancements in computer vision techniques, traffic cameras have gained numerous applications, including vehicle speed measurement, automated traffic analytics, and detecting near-misses or near-accidents for urban planning improvement [6], just to name a few. Additionally, an important use case is understanding human-vehicle interaction behavior for accident prediction and prevention, as well as achieving multi-camera fusion by aligning different cameras' views to a common frame of reference [50]. To enable the development of such applications using publicly available video feeds, camera calibration is a crucial requirement. Along with determining intrinsic parameters like focal length and distortion coefficients, camera calibration also involves estimating extrinsic parameters, which refer to the orientation and position of the camera in real-world coordinates. Moreover, for accurate downstream applications, knowledge about scene's geometry like ground plane parameters and traversible lanes is often necessary, necessitating some form of metric reconstruction of the scene. However, calibrating each individual camera becomes a

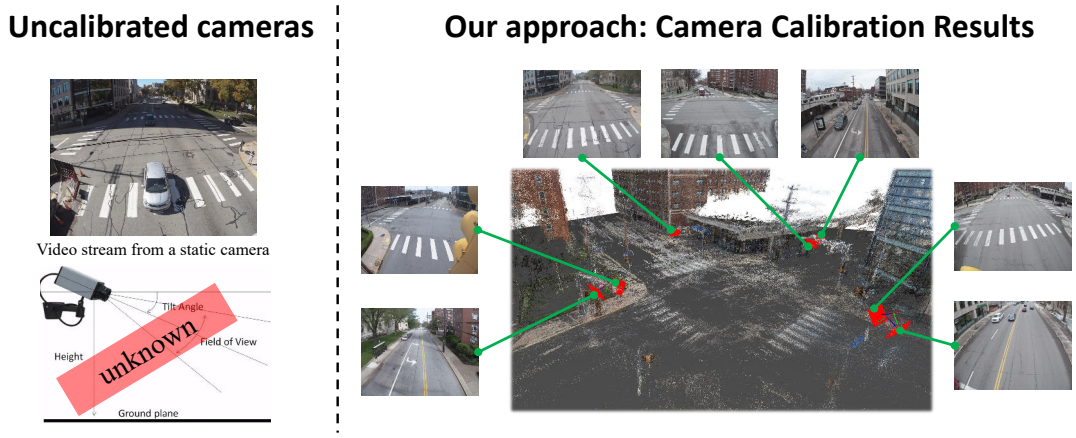


Figure 1.2: **Left**: In an in-the-wild scenario, a static camera observes a traffic scene where crucial information such as its height to the ground plane, mounting orientation, and field of view are unknown. **Right**: Our approach demonstrates the accurate localization of 7 cameras at an intersection in Pittsburgh, PA, showcasing the effectiveness of our methodology.

challenging task when information regarding its intrinsic properties and mounting specifications is not readily available in most cases (see Fig. 1.2). Despite the existence of extensive literature on traffic camera calibration, existing approaches suffer from various limitations. Some methods are impractical for traffic cameras, such as those relying on checkerboard-based calibration [80]. Other techniques require manual inputs, such as identifying landmarks with known dimensions like road markings which can be time-consuming and subject to human error [7, 72]. Moreover, certain approaches rely on estimating and/or assuming specific priors, such as vanishing points [16, 35, 68], average vehicle size [12], or camera height [72], which can introduce potential inaccuracies and limit generalizability.

In this thesis, we introduce a procedure for acquiring accurate metric 3D scene reconstruction and calibration of stationary cameras in real-world street intersections in an automated manner (see Fig. 1.2). To achieve this, we leverage the vast amount of high-quality, geo-referenced, and calibrated images available in Google Street View (GSV). By utilizing GSV, we construct a metric-scale 3D scene reconstruction at the desired camera location. Next, we employ state-of-the-art (SOTA) camera localization techniques, leveraging recent advances in learned feature matching, such as SuperPoint [13] and SuperGlue [61], to establish robust 2D-3D correspondences. This enables us

to infer the traffic camera's intrinsic and extrinsic parameters accurately. Through extensive quantitative and qualitative experiments, we demonstrate the significant improvements of our method over existing SOTA methods in both intrinsic and extrinsic calibration. Notably, our approach is efficient and capable of reconstructing and calibrating more than 100 stationary cameras in various real-world traffic scenes across the globe, showcasing the scalability and versatility of our method.

1.2 Amodal object understanding

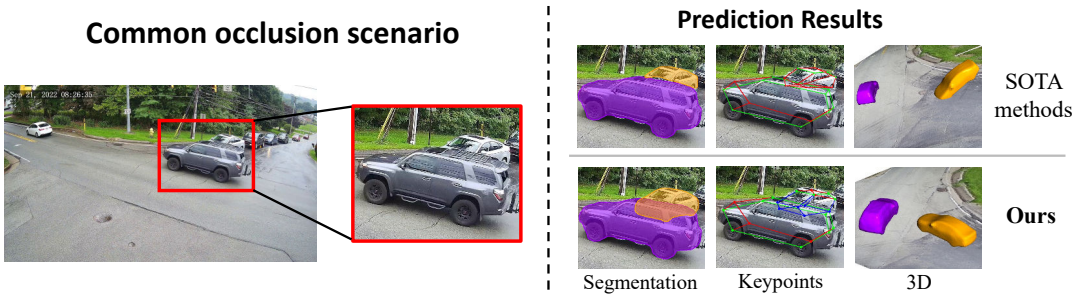


Figure 1.3: **Left**: In a common scenario with occlusion, the front car occludes a significant portion of the car behind it. **Right**: Existing SOTA methods struggle to handle occlusion, as seen in the case of the occluded (yellow) car. In contrast, our approach predicts accurate 2D/3D amodal representations of objects, even in the presence of occlusion.

In virtually every scene, occlusions are present, as depicted in Fig. 1.3. Explicitly modeling occlusions is challenging because of the range of occlusion types in the scene [41, 78]: an object may be partially occluded by other objects, truncated by the camera's field-of-view, and even when there is only a single object in the scene, self-occlusions occur because a camera can only capture one side of the object (either left or right, front or back). Despite significant advancements in data-driven methods for downstream scene understanding tasks such as object detection, tracking, segmentation, reconstruction, the performance of these approaches often falls short in scenarios with severe occlusions [58] (see Fig. 1.3).

The limited performance in occluded scenarios can indeed be attributed to the absence of specialized treatments for occlusions. Treating occlusions as outliers or

noise in the data has been attempted in various studies [21, 22, 29, 41, 64, 73], but this approach often falls short in providing reliable results, particularly when the number of occlusions in a scene is substantial (e.g., in an urban area). Learning occlusions requires a large annotated, realistic dataset. Unfortunately, labeling hidden parts of objects consistently proves to be difficult for human annotators, leading to a scarcity of realistic datasets with comprehensive occlusion annotations [14, 54, 56, 83]. As a result, there exists a significant bias against learning robustness to occlusions.

Our work addresses the challenge of amodal object understanding by proposing an automated way of generating occlusion supervision. Specifically, we exploit the time-lapse imagery captured by stationary traffic cameras observing street intersections over extended periods, ranging from weeks to months and even years, to synthesize diverse occlusion scenarios. Specifically, from time-lapse video streams, we mine for unoccluded objects using an accurate classifier. Then, using the mined unoccluded objects and leveraging object motion and 3D static scene constraints such as camera intrinsics and ground-plane equation, we reconstruct them in a physically accurate 3D manner. These objects are then composited in both 2D and 3D to generate realistic occlusion configurations. The combination of real data and synthetic occlusions results in a hybrid dataset that captures the richness of real-world objects and the variety of occlusion patterns. To train our model, we utilize this hybrid 3D composited data as amodal supervision. We train layered amodal keypoints and segmentations, and the amodal 2D representations are lifted to 3D using shape basis optimization. Throughout the pipeline, the occlusion category is used for supervision, ensuring accurate amodal 3D reconstructions.

Chapter 2

Background

2.1 3D Scene Reconstruction and Camera Calibration

Reconstructing the complete 4D vehicular activity, which encompasses 3D space and time, from a single stationary camera operating in real-world conditions presents a formidable challenge. This problem is inherently complex and necessitates precise camera calibration. The calibration process involves two key aspects: 1) intrinsics calibration, which takes into account perspective projection and potentially corrects for radial and tangential distortion, and 2) extrinsics calibration, which handles varying camera rotations and addresses the unknown distance between the camera and the ground plane of the road. Therefore, to achieve accurate reconstruction, it is crucial to determine both intrinsic and extrinsic camera parameters, as well as establish the scene geometry and/or the camera's distance from the road plane.

Existing approaches: When it comes to camera calibration in general, various approaches exist. The widely used method by Zhang et al. [80] employs a calibration checkerboard to obtain intrinsic and extrinsic camera parameters. However, this traditional checkerboard-based method becomes impractical for traffic cameras in inaccessible locations, particularly in-the-wild scenarios. In the context of traffic scene analysis, alternative methods have been proposed. Some approaches [7, 24, 27] rely on detecting vanishing points at road marking intersections, utilizing vehicle motion to

calibrate the camera [12, 16, 17, 62], or involving manual measurements of dimensions on the road plane [15, 39, 46, 47, 48, 51, 67]. Various techniques have also been proposed for estimating the scene scale. For example, [16] employed a 3D bounding box around vehicles and their average dimensions to compute the scale, and [68] suggested using the alignment of a 3D model and a bounding box for scale inference. However, it is important to note that these methods have limitations in terms of scalability and accuracy. Manual methods require laborious landmark and dimension setting, while automatic methods still exhibit high errors and high sensitivity to the quality of estimated geometric cues such as vanishing points, rendering them unsuitable for achieving precise 3D reconstruction. A comprehensive survey on *Monocular Visual Traffic Surveillance* has been conducted by Zhang et al. [79].

Our approach: To achieve accurate metric 3D scene reconstruction and automatic camera calibration, we propose leveraging the extensive collection of geo-registered panoramic imagery from Google Street View [23] (GSV). This approach offers a cost-effective and automatable solution for reconstructing urban areas in 3D. We employ structure-from-motion (SfM) [63] to reconstruct the scene's metric geometry using multiple geo-referenced panoramic images from GSV, sampled around a specific physical location. To localize a query image from a traffic camera stream within the 3D reconstruction obtained from GSV images, we leverage robust 2D-3D correspondences generated by a learned feature matching method called SuperGlue [61]. SuperGlue, in combination with SuperPoint [13] features, produces a large number of accurate matches between the query image and the GSV images. Despite the challenging differences in appearance, these matches enable us to robustly recover both the camera's intrinsic parameters and its 6 degrees of freedom (6DoF) extrinsic parameters.

2.2 Amodal Object Understanding

Occlusion Reasoning: Understanding and reasoning occlusions has been extensively studied for decades [21, 22, 64]. Bad predictions due to occlusions are dealt with as noise/outliers in robust estimators. On the other hand, occlusions are explicitly treated as missing parts in model fitting methods [70, 81]. But severe occlusions, such as when a large part of an object is blocked, can result in poor model fitting [25, 84]. Furthermore, 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. While these approaches have advanced the state-of-the-art, they focused on only one type of occlusion (either 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.

2D Amodal Representation: Although the effects of occlusion on visual reasoning has been widely studied, estimating the amodal representation (i.e. both the occluded and visible regions) has only been recently explored. Initial attempts [20, 26, 54, 83] use a supervised learning paradigm using small datasets [54, 83] where humans have annotated occlusions to the best of their abilities. Some methods [55, 56, 66] have explored using multiple views to provide accurate supervision for occluded parts but are not scalable due to capture limitations. To expand supervision, several methods synthesize occlusions to varying degrees of realism. But pure CG renderings [2, 18, 19, 30, 33, 41, 77] suffer from a wide domain-gap [36, 64]. To address this domain gap, methods like WALT [58] introduce a hybrid approach to composite real image segments of unoccluded objects captured from time-lapse data to create a 2D clip-art dataset of a large number of occlusion configurations. Most of these methods only classify occlusion types into visible and occluded (2-classes), while our proposed method is able to extend this object categorization to different occlusion categories which we show to be more effective for amodal 2D and 3D reconstruction.

3D Amodal Reconstruction: Amodal 3D reconstruction is still in the nascent stages of research. Most of the algorithms developed have been for self-occluded objects with shape completion from partial observations [11, 52, 82]. On the other hand, shape models fitting for objects only with the visible regions either from images [25, 31, 38, 59] or depth sensors [1, 53, 74] have been explored. Compared to these methods, our pipeline and network focus on learning both 2D and 3D amodal representations under a variety of occlusion configurations from just streams of images.

Chapter 3

Scene Reconstruction and Camera Calibration

Our objective is to construct a metric 3D reconstruction of the scene around a desired traffic camera's GPS location, such as an intersection. Subsequently, we aim to localize the traffic camera within the obtained reconstruction. The overall framework of our approach is illustrated in Figure 3.1.

3.1 3D Scene Reconstruction

To perform the reconstruction, we leverage Google Street View (GSV) [23] to build the scene's geometry around a specific GPS location. GSV is a street-level imagery database and a rich source of millions of panorama images with wide coverage all over the world. Every panorama image is geo-tagged with accurate GPS coordinates, capturing 360° horizontal and 180° vertical field-of-view (FoV) with high resolution (see Fig. 3.2). This panorama is also known as an equirectangular image, which can be thought of as a sphere mesh unwraped on a flat rectangular plane surface.

In particular, we first sample N panoramas (equirectangular) frames $\mathcal{E} = \{\mathcal{E}_i | i = 1 \dots N\}$ around the desired camera's location inside a radius of 40 meters from the GPS location. As most components of a *structure-from-motion* (SfM) pipline [63] are only well-designed for rectilinear perspective images, we extract ideal, pinhole camera-style perspective projections from an equirectangular image before performing

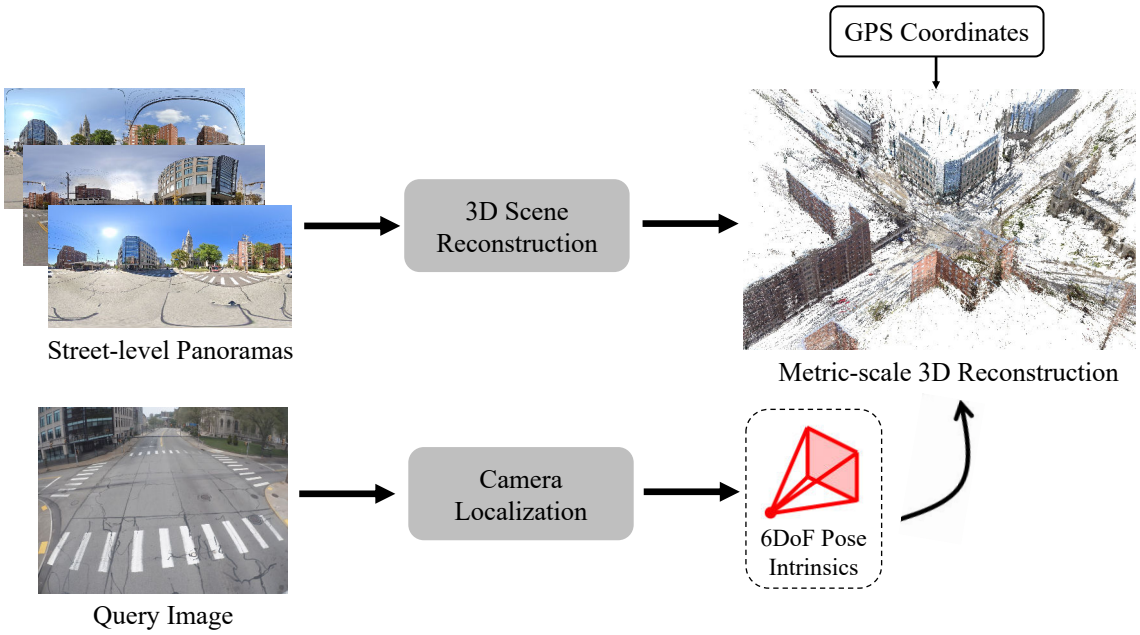


Figure 3.1: Overview of the scene reconstruction and camera calibration pipeline. **Top**: Using street-level panoramas (equirectangular images) and corresponding GPS coordinates from Google Street View (GSV), we perform 3D scene reconstruction to obtain a metric-scale 3D representation of the scene. **Bottom**: Given a query image from a traffic camera stream, we perform camera localization to accurately determine both the intrinsic parameters and the 6 degrees of freedom (6DoF) pose of the camera with respect to the scene.

3D reconstruction (more details in [4]). Specifically, from each equirectangular image $\mathcal{E}_i$, we extract T perspective images $\mathcal{I} = \{\mathcal{I}_{ij} | i = 1 \dots N, j = 1 \dots T\}$ which are uniformly sampled along the yaw direction with specified size and FoV, covering 360° horizontal FoV. Denoting $\mathbf{\Pi}$ as the projection function from equirectangular to perspective image, we can define each perspective image $\mathcal{I}_{ij}$ as:

$$\mathcal{I}_{ij} = \mathbf{\Pi}(\mathcal{E}_i, \texttt{pitch=0}, \texttt{yaw=}\frac{2\pi * j}{T}, \texttt{(height,width)=(H,W)}, \texttt{fov=FOV})$$

In practice, we found the set of hyperparameters $\{\texttt{T=12}, \texttt{H=1080}, \texttt{W=1920}, \texttt{FOV=90°}\}$ to be producing high-quality perspective images with sufficient overlap and minimal perspective distortions.

In the following paragraphs, we described the specific details from each step

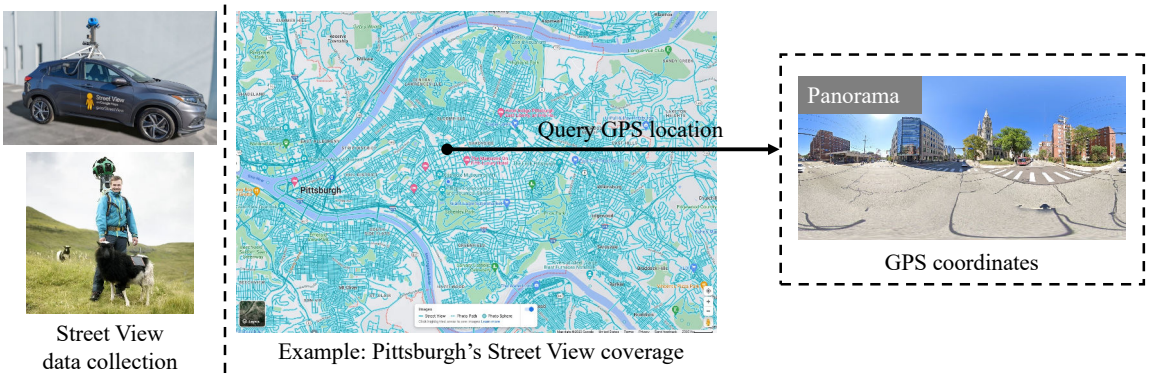


Figure 3.2: The coverage of Google Street View is highly extensive, with data being captured over the course of the last decade. At any given location, we can query the closest panorama (equirectangular image) that covers a 360°horizontal and 180°vertical field-of-view, together with its GPS coordinates.

of using structure-from-motion COLMAP [63] to estimate for each frame $\mathcal{I}_{ij}$ the intrinsic K_{ij} and extrinsics camera parameters $(R_{i,j}, t_{i,j})$. An overview of the pipeline is illustrated in Figure 3.3.

Pre-processing. Because dynamic objects often cause errors in the reconstruction, we apply a segmantic segmentation method [10] to segment out potential dynamic objects such as vehicles and people in every frame and suppress feature extraction in these areas. For each perspective image, as we know its exact focal length with no distortions (assuming it is extracted from a correct equirectangular), the intrinsic camera parameters K_{ij} is known. Therefore, we use `SIMPLE_PINHOLE` camera model and fixed the shared camera intrinsics for all the frames.

Feature matching. Although SIFT [45] features works well for the reconstruction process given the dense samples of GSV images, we instead use a learned feature extractor SuperPoint [13] as it will allows us to take advantage of learned feature matching in the later localization step. To find correspondences between the feature points in different images, instead of using exhausive matching where every image is matched against every other image, we use a modified version of vocabulary tree matching where every image is matched against its visual nearest neighbors using a vocabulary tree. To build the vocabulary tree, we first compute the descriptor centroids using `KMeans++` [5], then `KDTree` is used to build the vocabulary tree using VLAD [3] descriptors. This vocabulary tree can be thought of as a visual database

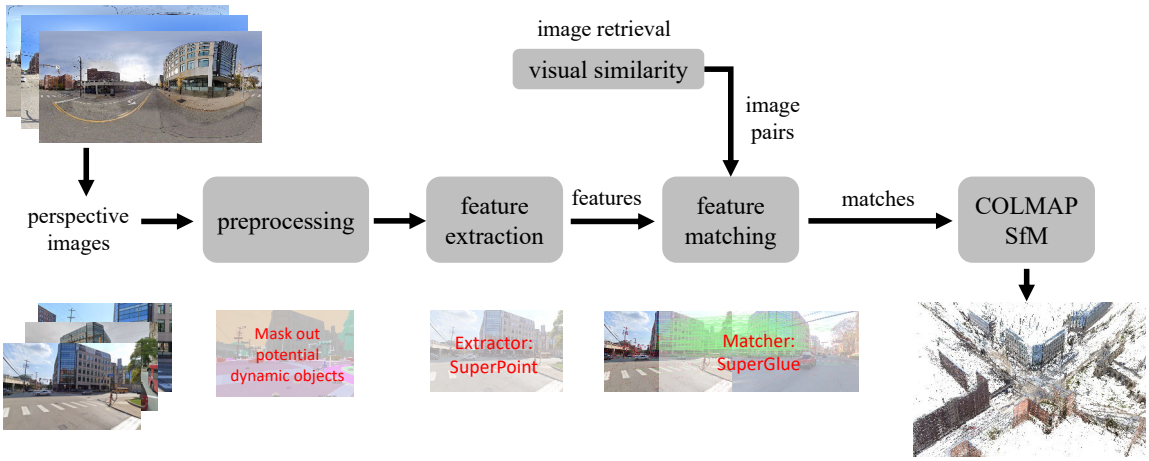


Figure 3.3: Overview of the 3D scene reconstruction pipeline. Using multiple retrieved panorama images from GSV, our objective is to create a metric-scale 3D reconstruction of the scene.

which we will use to retrieve the database images that are the most similar (in terms of visual appearance) to the query image.

Enforcing panoramic constraints for bundle adjustment. In a standard structure-from-motion (SfM) workflow, the input image collection is assumed to be unordered, and each image is treated independently. However, in our case where we extract perspective images by sampling from panoramas, we can leverage the known transformations or relative poses between frames that are sampled from the same panorama. To achieve this, in addition to the typical bundle adjustment (BA) optimization that minimizes the reprojection error to refine the reconstructed 3D points, intrinsics, and camera poses, we incorporate the enforcement of known relative poses between frames from the same panorama. In our case, two perspective images from the same panorama are related by a pure rotation around its z-axis since we initially sample and extract perspective images along the yaw direction. For each image $\mathcal{I}_{ij}$, represented by its extrinsic camera parameters $(R_{i,j}, t_{i,j})$, we introduce a loss term $\mathcal{L} = \mathcal{L}_{trans} + \mathcal{L}_{rot}$:

$$\mathcal{L}_{trans} = \sum_{i=1}^{N}\sum_{j=2}^{T} ||t_{i,j} - t_{i,j-1}||^2, \quad \mathcal{L}_{rot} = \sum_{i=1}^{N}\sum_{j=2}^{T} ||R_{i,j}^{\top} R_{i,j-1} - R_z(\frac{2\pi}{T})||^2$$
$$\text{subject to} \quad R^{\top}R = \mathbf{I}, det(R) = 1$$

where $R_z(\theta)$ denotes the rotation matrix around the z-axis by an angle of θ. In practice, unit quaternion is used to parameterize rotation during the optimization.
Metric scale calibration. Using the provided GPS coordinates (lat/lon/alt) of the GSV panoramas, we further geo-registered the *up-to-scale* SfM reconstruction by optimizing a 3D similarity transformation between the reconstructed model and the target coordinate frame. In this case, the target coordinate frame is the Earth-Centered-Earth-Fixed (ECEF) Cartesian-based coordinate system obtained from GPS. As a result, our final 3D reconstruction of the scene is in *metric scale*. The road plane equation is estimated by fitting a plane to the set of 3D points whose 2D pixel locations are lying on the *road* obtained from off-the-shelf semantic segmentation method [10].

3.2 Camera Localization

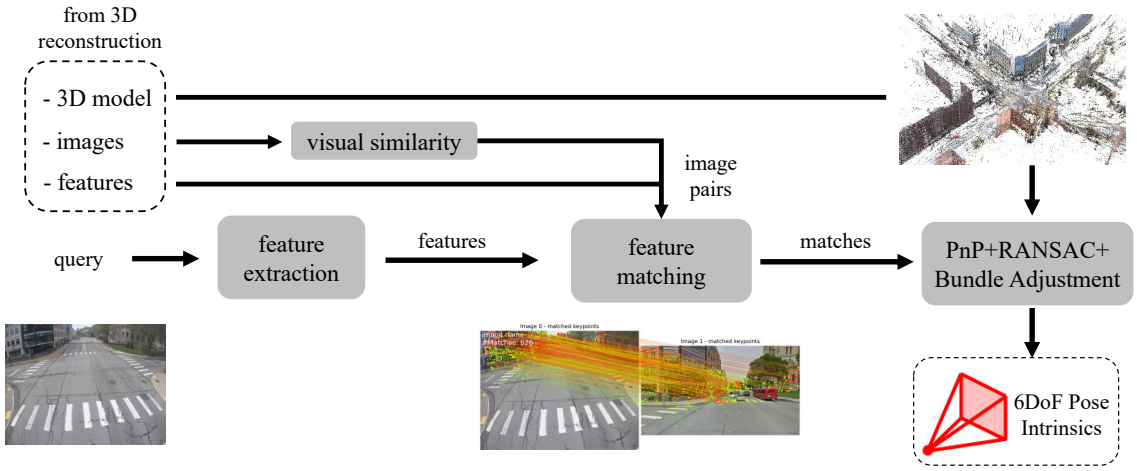


Figure 3.4: Overview of the Camera Localization pipeline. Given a query image from a traffic camera stream, our objective is to obtain the camera's intrinsic parameters and its 6DoF pose (rotation and translation) with respect to the 3D scene.

The camera localization step aims to determine the intrinsic and extrinsic parameters of the traffic camera with respect to the 3D scene. As depicted in Figure 3.4, we adopt a visual localization pipeline that involves localizing the query image (a frame from the traffic camera stream) within the 3D reconstruction constructed using Google Street View (GSV) images.

For every input query image, we retrieve the top-k similar database images from the vocabulary tree built in the reconstruction step. After this, we then match the query image with the retrieved database images ($k = 40$ images in our case) to establish 2D-3D correspondences and obtain the initial intrinsics as well as 6DoF camera pose using RANSAC+PnP. In this approach, we follow *hloc* [60] by using learned feature matching method SuperGlue [61] with SuperPoint [13] features descriptors to match the query image with the database images. Given the correspondences and initial estimates of intrinsics and extrinsics parameters, we perform an extra bundle adjustment step to refine the parameters. It is worth noting that the use of learned feature matching is crucial in this matching step, as it has been shown to outperform hand-crafted feature descriptors and matching methods, particularly in cases where the viewpoint of the traffic camera (often from the top) differs significantly from that of the Google Street View (GSV) images (captured from driving viewpoints). The learned feature matching approach, specifically utilizing SuperPoint [13] and SuperGlue [61], enables the generation of a large number of accurate matches between the query image and the comprehensive GSV database images. This rich set of matches allows for robust recovery of both the intrinsic and extrinsic camera parameters (as shown in Figure 3.5). Lastly, more recent advances in local feature matching such as LightGlue [44] can also be used to improve efficiency.

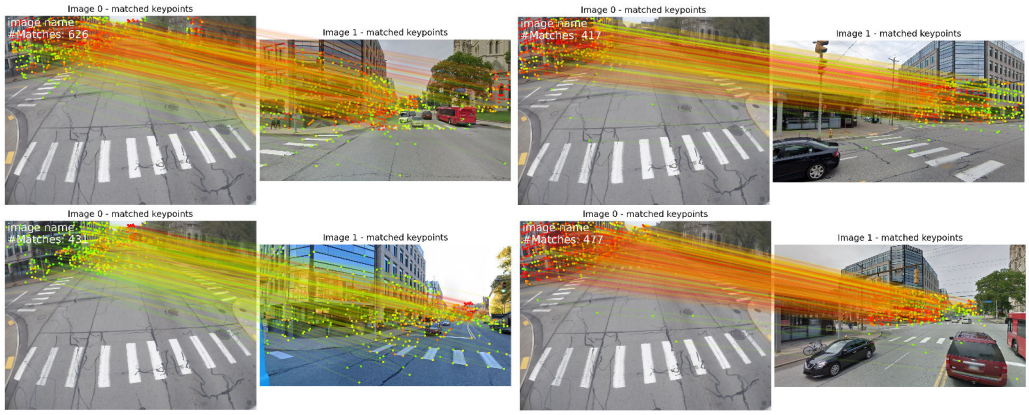


Figure 3.5: Examples of feature matching between the traffic camera image and GSV images. Despite the significant differences in viewpoint and illumination, learned methods like SuperPoint [13] and SuperGlue [61] are capable of retrieving a large number of correspondences.

3.3 Ablation Analysis and Results

Table 3.1: Comparison between our approach vs. checkerboard-based calibration.

Parameters	Checkerboard (pix.)	Our approach (error)
F_x	2707 ± 3.99	2721 (0.52%)
F_y	2708 ± 4.66	2721 (0.48%)
C_x	2032 ± 4.40	2021 (0.54%)
C_y	1443 ± 2.68	1464 (1.45%)
(radial) k_1	-0.304 ± 0.002	-0.281 (7.56%)
(radial) k_2	0.156 ± 0.007	0.152 (2.56%)

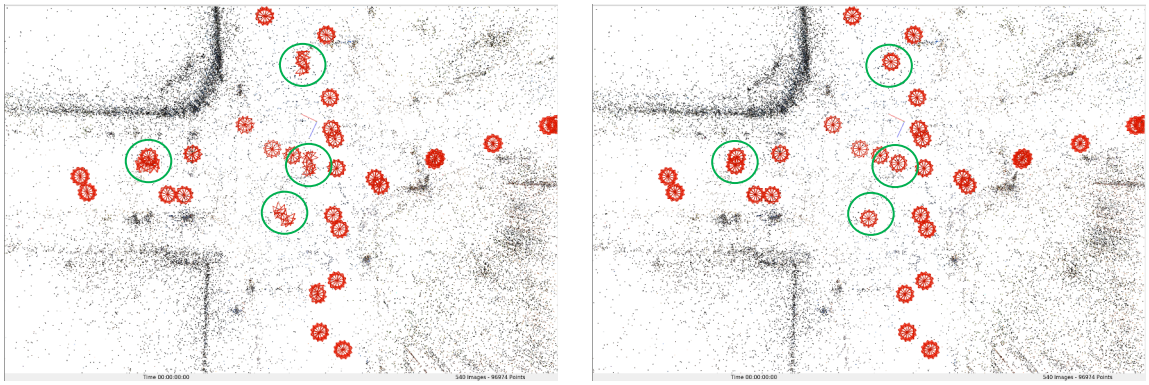
Table 3.2: Mean calibration error between measured vs. estimated distances.

Method	Mean Error (%)
DeepVPCalib [35]	12.2
Ours	**3.7**

Camera Calibration Accuracy: As we have access to 7 cameras that we mounted in Pittsburgh, we evaluate the accuracy of camera calibration obtained using our method:

- **Intrinsics Parameters**: In Table 3.1, we compared our results with checkerboard-based calibration (using `OpenCV` calibration for `ChArUco` board). The error of our method for the focal length was about 10 pixels, which is less than 0.5 percent. Additionally, the error margin of radial distortion parameters k_1, k_2 was also in an acceptable range.
- **Extrinsics Parameters**: Following the common evaluation protocol described in DeepVPCalib [35], we manually measured some distances between pairs of points on the road plane along with their pixel positions in the images (e.g., lane marking, crosswalks, etc.). We then computed the differences of two different measurements, defined as $r_i = \frac{|\hat{d}_i - d_i|}{\hat{d}_i}$, where $\hat{d}_i$ is the i-th ground-truth distance measurement and d_i is the i-th measurement based on the ray-plane intersection using the estimated intrinsics matrix and ground-plane equation. Since DeepVPCalib [35] does not compute the metric scale, we scale the estimated distances from DeepVPCalib with the ground-truth scale computed for an arbitrary measurement. As shown in Table 3.2, our method outperforms existing SOTA method DeepVPCalib [35] by a large margin, demonstrating the accuracy of our camera calibration as well as estimated scene geometry.

Enforcing the known relative pose between frames from the same panorama improves the accuracy of reconstruction: As shown in Figure 3.6, this constraint



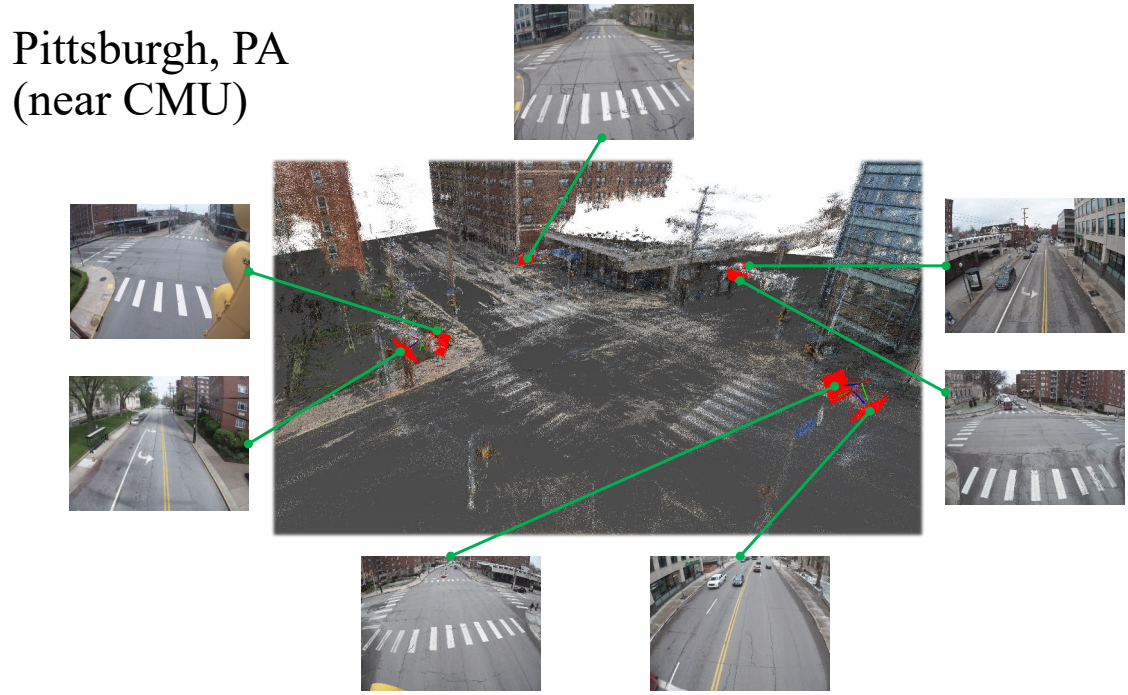
BEFORE enforcing relative pose **AFTER** enforcing relative pose

Figure 3.6: **Before** and **After** enforcing the known relative pose between images sampled from the same panorama. The correction of erroneous poses is highlighted in the green circles.

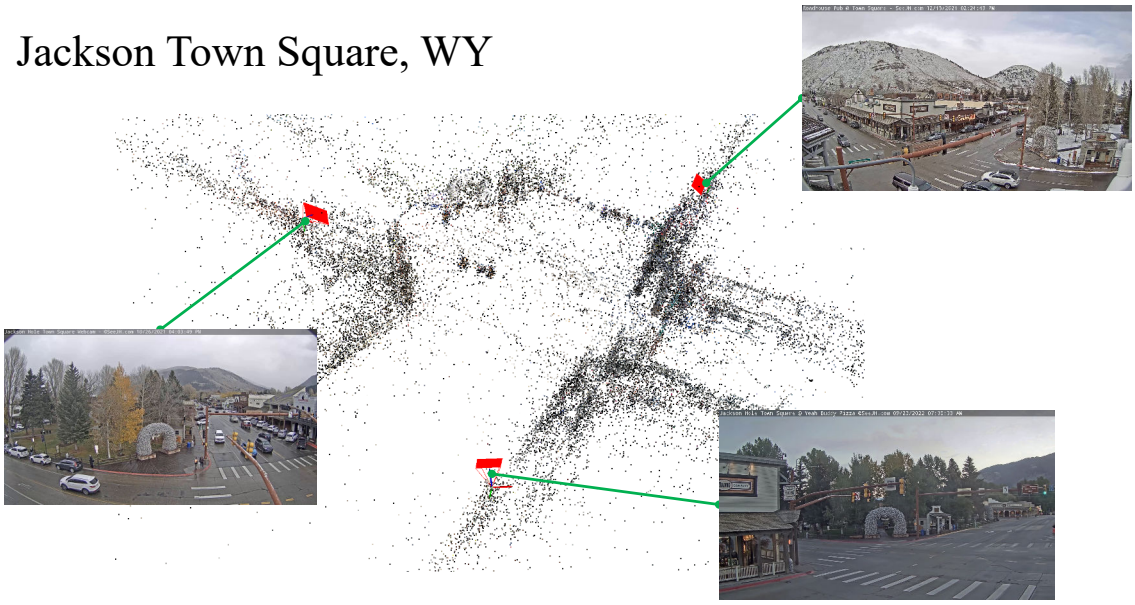
helps correct erroneous camera poses. Our observation is that a few erroneous camera poses do not significantly affect the quality of the reconstruction when we have a large number of cameras. However, this constraint is especially helpful when we try to reduce the number of images being used for reconstruction. Empirically, on 10 different intersections, if we reduce the number of images by 70%, the reprojection error (measured in pixels) reduces by more than 30%.

Qualitative Results: It is important to highlight that our camera localization approach can be applied to any camera in-the-wild, given sufficient coverage of the location by Google Street View (GSV). In Figure 3.7, we present detailed reconstructions and localizations of two different intersections, showcasing the effectiveness of our method. Additionally, in Figure 3.8, we provide examples of several locations where we successfully ran our pipeline and obtained both the scene reconstruction and camera localization. These qualitative results demonstrate the capability of our framework to accurately reconstruct the scene and determine the camera's position within it, showcasing its potential for various applications in smart city environments.

Reproducibility: Code will be made publicly available to the community, whereas the data is subject to Google Street View's license.



(a) Fifth-Craig Intersection, Pittsburgh, PA



(b) Jackson Town Square, WY

Figure 3.7: Results of 3D scene reconstruction and camera localization for in-the-wild cameras.

Figure 3.8: Additional examples demonstrating the robustness of our method in reconstructing scenes and localizing cameras.

Chapter 4

Amodal Object Understanding

Our goal is to automatically generate 3D amodal supervision data and learn amodal 2D/3D representations using a novel framework. The pipeline involves several steps, as depicted in Figure 4.1:

- **Occlusion Category Classification (OCC)**: The pipeline starts by utilizing an OCC network on a stream of data. This network performs two tasks: localizing 2D keypoints and categorizing the occlusion status of objects within the scene. The objective is to identify unoccluded objects, which are those not occluded by other objects or the scene itself.
- **Generating 3D Amodal Supervision Data**: Once the unoccluded objects are identified, we leverage the camera's calibration parameters and the scene constraints obtained from Chapter 3 to perform 3D spatio-temporal reconstruction of these objects. By placing the unoccluded objects back into their original positions, we generate various occlusion configurations as 3D ground-truth supervision data. This data is used to train the model for predicting amodal 2D/3D representations.
- **Learning 2D/3D Amodal Representations**: Using the generated amodal 2D/3D data as ground-truth supervision, we propose a novel architecture to learn 2D/3D amodal representations and recover the 3D pose of the object by disentangling each layer of occlusion in a network (see Fig. 4.4).

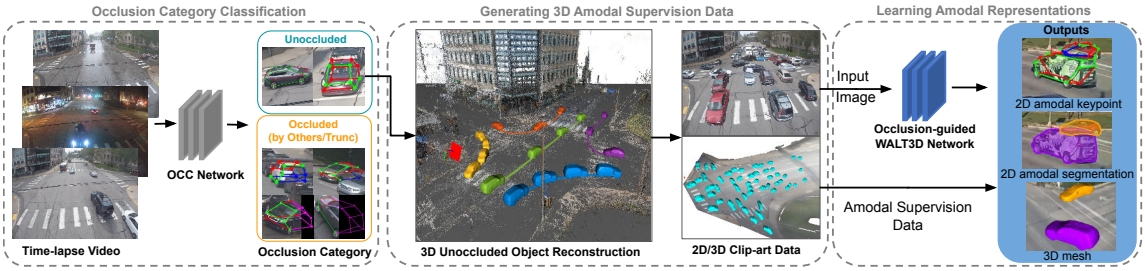


Figure 4.1: **Overall Framework:** We illustrate our framework for mining unoccluded objects to generate 3D amodal supervision data which is then used to learn 2D/3D amodal representations. The key idea is to use the Occlusion Category Classification (OCC) network on a stream of data to mine for unoccluded objects. We then perform 3D spatio-temporal reconstruction of these mined unoccluded objects following [42] to get 3D shape and poses (composited into 3D background scene reconstruction). These unoccluded 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 using occlusion-guided WALT3D network.

4.1 Occlusion Category Classification

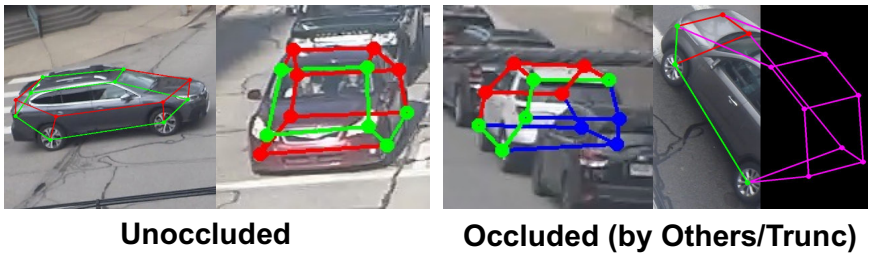


Figure 4.2: Occlusion Category Hierarchy. We classify each instance into *Unoccluded* (left) or *Occluded* (by Others/Truncation) (right) based on per-keypoint occlusion type: visible, self-occluded, occ-by-others, and occ-by-truncation.

Occlusion categories are more nuanced than simply represented using 2 classes -"occluded" or "unoccluded". Consider the example images in Fig. 4.2. At an object-instance level, we can consider the left examples to be of *unoccluded* cars, and the right examples to be of cars that are either *occluded* by other objects or truncated. But, interestingly, at the object-part level, each keypoint can be either visible, self-occluded, occ-by-others, and occ-by-truncation. From this definition,

object-instance level *Unoccluded* category can be obtained from part-level categories if all the keypoints are either visible or self-occluded.

In this work, we show that this nuanced categorization is crucial in two important ways: (1) Object-instance level categorization is used to mine individual objects that are *Unoccluded*. Evidently, many downstream vision tasks such as segmentation, tracking, and reconstruction work well for such Unoccluded objects. (2) Part-level or keypoint-level categorization is crucial to provide visibility constraints for 3D reconstruction. Note that the shape and the pose of the object determine the visibilities of different keypoints (via raycasting) from a camera. Having visibility constraints can thus prevent large errors in the object's shape and pose.

Learning Occlusion Categories: Human labeling of the above occlusion categories are much more accurate than the localization of invisible keypoints [57]. Thus, we collected a large dataset with keypoints' location and occlusion category labeled manually (details in Section 4.4). The 4-way supervised classifier of keypoints, called the Occlusion Category Classification (**OCC**) module is shown in Fig. 4.1.

Mining Unoccluded Objects: Given a stream of time-lapse data from a camera, we use the OCC network on each detected object instance to estimate keypoints with occlusion category classes. In each instance, if all keypoints are either visible or self-occluded, the object instance is classified as *Unoccluded* (see Fig. 4.2). Otherwise, it is classified as *Occluded (by others/truncation)*. Using this strategy, even conservatively choosing only high confidence and low recall samples, we are able to mine for thousands of unoccluded objects from time-lapse data per camera. Each of the mined objects comprises of bounding box, segmentation mask, and 2D keypoint locations. These mined unoccluded objects and the 2D predictions are then used for generating 2D/3D amodal supervision data.

4.2 Generating 3D Amodal Supervision Data

In this section, we will describe how to exploit the mined unoccluded objects to generate amodal 2D and 3D supervision signals. We use image based compositing to generate input image and their corresponding 2D and 3D representations. Following the nomenclature of [58], we call the generated data as *clip-art 2D* and *3D data*.

Generating Unoccluded Object Reconstructions: Each of the mined unoccluded

Figure 4.3: **Automatically generated 2D and 3D Clip-Art to supervise our 3D amodal network:** Unoccluded objects are first mined using time-lapse imagery of the WALT dataset [58]. Randomly sampled and non-intersecting unoccluded objects are composited back into the background image in their respective original positions to maintain correct appearances. The resulting 3D Clip-Art images and their respective amodal segmentation masks, keypoint locations, and their occlusion categories, depth maps and 3D meshes are shown. The clip-art method generates realistic appearances and 3D from any camera with diverse viewing geometry, weather, lighting and occlusion configurations. See more examples in the Supplementary.

objects is reconstructed following the approach described in [42]. Starting with the predicted 2D keypoint locations of each object from the OCC network, we initialize the 3D poses using the EPnP algorithm [40, 57] by considering only the visible keypoints. Since unoccluded objects have accurate 2D localization of keypoints, the resulting reconstruction is precise. Subsequently, we perform joint optimization of the predicted 2D keypoints and the initialized 3D mean shape of all the mined unoccluded objects to obtain object poses and mean shape coefficients. To ensure physically plausible reconstructions, we enforce the constraint that all the objects detected in the camera should lie on the same plane. This additional optimization constraint ensures physically plausible object poses. It is important to note that we utilize the accurate camera calibration obtained using our method described in Chapter 3. Ultimately, the mined unoccluded object image with its corresponding segmentation

mask, keypoint locations, and 3D poses are used in a clip-art based framework to generate 3D supervision data in severe occlusions as illustrated in Fig. 4.1.

Amodal data generation using clip-art: As mentioned earlier, getting amodal supervision for occluded objects is a challenging problem. By using the clip-art based image augmentation approach described below, we automatically generate a large number of realistic supervision signals in severe occlusions. Using the 3D poses of the mined unoccluded objects from the previous section, we composite the object's image and its corresponding amodal 2D and 3D representations (see Fig. 4.3). Non-intersecting 3D objects are randomly sampled and pasted back into the background image from the farthest away from camera to the closest. Note that our 3D-based approach differs from simply compositing 2D images [58] which often leads to objects that could be intersecting in the real-world. Thus, our method generates physically accurate and realistic occlusion configurations. Each such generated clip-art image is accompanied by amodal segmentation masks, amodal 2D/3D bounding box, 3D poses, and per-keypoint occlusion type (via raycasting). This kind of supervision signal provides complete scene understanding and will play a major role for deciphering different layers of occlusions for learning downstream tasks such as tracking and reconstruction. The accurate geometry of the scene (i.e., camera localization and ground plane constraints) obtained automatically as described in Chapter 3 also allows us to generate inter-category occlusion configurations (e.g., vehicle occluded by people).

4.3 Learning 2D/3D Amodal Representations

We have generated a large clip-art image dataset with corresponding amodal 2D/3D ground-truth representations. Using these as supervision, we will recover the 3D pose of the object by disentangling each layer of occlusion in a network (see Fig. 4.4). 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. The loss between the predicted bounding box and the Ground-Truth Amodal Bounding box is given as L_{AB}. The ROI features are also passed through the 3D prediction network.

Learning 2D Amodal Representations: For computing the amodal features of an object, it is essential to learn different occlusion layers in the amodal bounding

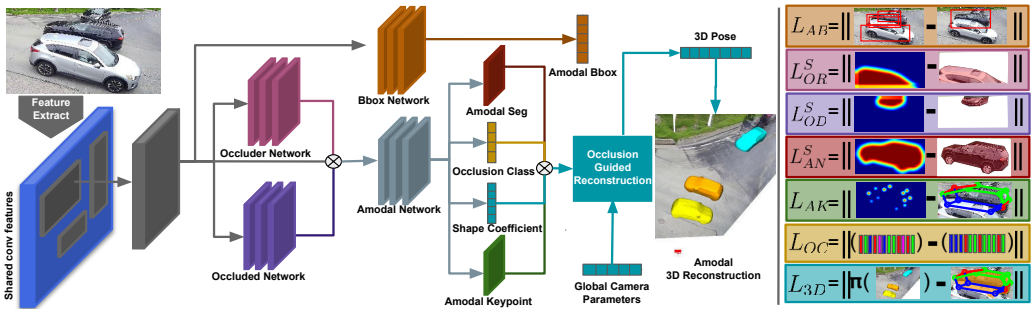


Figure 4.4: **WALT3D Network:** Given the Amodal Clip-Art Image and the corresponding 2D/3D representations of the objects from the occlusion-aware supervision, we illustrate the network used to train to predict 3D pose and shape of the object. The input image is passed through a backbone to extract ROI features. These features are passed through an occluder and occluded networks which help disentangle objects' occlusion types. 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.

box. Inspired from [33, 58], we learn the occluder-occluded-object interaction which helps us distinguish each object interaction in the bounding box, where the occluder is the layer occluding the amodal object, while the occluded is the layer occluded by the amodal object of interest. Our generated clip-art based data readily provides supervision for each layer. We train each of these components using the binary cross-entropy loss function L:

$$L_M^T = -W_T[G_M^T \log(F_M^T) + (1 - G_M^T) \log(1 - F_M^T)] \tag{4.1}$$

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 feature map F with the weights given by W. The features from both the occluded and occluder layer are concatenated with the input ROI feature to produce an occlusion-robust amodal feature vector for each object. This combined amodal feature is used to compute the segmentation mask, keypoint locations, shape coefficients, and per-keypoint 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:

$$L_{AS} = L^S_{OD} + L^S_{OR} + L^S_{AN} \tag{4.2}$$

Here L^S_{OD}, L^S_{OR}, L^S_{AN} are binary cross-entropy losses for occluded, occluder, and amodal segmentation maps.

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

$$L_{AK} = \sum_{k \in K} L^k_{OD} + L^k_{OR} + L^k_{AN} \tag{4.3}$$

Here L^k_{OD}, L^k_{OR}, L^k_{AN} 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:

$$L_{OC} = -\sum_{k \in K} \sum_{c \in M} y^k_c log(p^k_c) \tag{4.4}$$

where y^k_c is the ground-truth binary indicator if keypoint k belongs to occlusion category class $c \in \{$visible, self-occluded, occ-by-others, occ-by-truncation$\}$ while p^k_c is the predicted probability observation of class c for keypoint k from the network. This helps us predict multiple objects and their visibility accurately. Note that the supervision for training the amodal representations is given from clip-art based data generation method as shown in Fig. 4.1.

Learning 3D Amodal Reconstruction: Since we have generated the 2D amodal representations of the objects, now we can regress for 3D representation from these features. 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 [8, 9] with occlusion category supervision. 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 $L_{3D} = L_{Reproj} + L_{OC}$, with L_{Reproj} being defined as:

$$L_{Reproj} = \frac{1}{2}\sum_{i=1}^{N} ||w_i \circ (\pi(RX_i + t) - x_i)||^2 \tag{4.5}$$

L_{Reproj} represents 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, and X_i and x_i represent the 3D and 2D predicted points. L_{OC}, as described above, is the occlusion category consistency term which enforces that the occlusion configuration of the predicted 3D object should be as similar as possible to the predicted occlusion type, preventing large errors in the reconstructed object pose.

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:

$$L = L_{AB} + L_{AS} + L_{AK} + L_{3D} \tag{4.6}$$

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.

4.4 Dataset and Implementation Details

There are multiple vehicle keypoints datasets [41, 55, 69, 76] but none provide detailed occlusion categories. They also lack the appearance diversity to perform well on in-the-wild evaluation data. Thus, 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 traffic cameras. The dataset captures a large number of appearance variations including day/night, weather, and seasons. It contains of 7,018 images with 42,547 vehicle instances (90/10% train/test split) with annotations of 12 semantic keypoints for each vehicle and the corresponding occlusion category. Of these, 5,384

Figure 4.5: Sample images from our proposed Occlusion Category Classification (OCC) Dataset. The dataset contains a wide range of appearance variations including day and night and various traffic scenarios, accompanied by human-annotated keypoint locations and occlusion type (color-coded).

instances are marked as Occluded-by-Others and 1,467 instances as Occluded-by-Truncation (see Fig. 4.11). The dataset is used for finetuning and evaluation and will be publicly released.

WALT Dataset [58]: This dataset contains images from 20 cameras in 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 resulting in approximately 3.3 million car instances for our experiments. We use the WALT raw dataset to generate the Clip-Art and Stationary WALT dataset for evaluation.

Clip-Art WALT Dataset: From the WALT dataset, we mine for Unoccluded objects resulting in 2.1 million objects. We generate supervision data as described in Sec. 4.2 by pasting them back into the scene with different backgrounds resulting in 10000 training and 500 testing images per camera. The resulting Clip-Art dataset covers occlusion categories in different lighting and weather conditions.

Stationary WALT Dataset: From the WALT test set, we mine unoccluded stationary objects by clustering objects detected at the same location. The unoccluded

amodal 2D/3D predictions of the stationary object is used as ground truth to compare predictions when the object is occluded by another object at different times. This strategy extracted 536 stationary objects observed over 60k frames for evaluation.

Camera and Scene Parameter Estimation: Using our method as described in Chapter 3, we obtain the 3D scene geometry as well as the camera's intrinsics and extrinsics parameters.

Metrics: We follow the Mean Average Precision (IoU=0.5) [43] 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: All baselines MaskRCNN [28], Occ-Net [56], 3DRCNN [37], WALT [58], and our proposed method are pre-trained on available vehicle keypoints datasets (Carfusion, PASCAL3D+, KITTI3D, ApolloCar3D)[41, 55, 69, 76] and finetuned on the same Clip-Art WALT dataset. Note that all the baselines either use only visible regions or 2-class categorization (visible or not).

4.5 Ablation Analysis and Results

Occlusion Category Analysis: We train Occ-Net [56] and OCC module on a combination of [41, 55, 69, 76] and further finetune on OCC dataset, then evaluate the accuracy of 2D keypoint localization and per-keypoint occlusion classification on OCC testing data. First, we observe an improvement of 20% in keypoint localization accuracy (66.41% to 80.12% on PCK@0.1) compared to Occ-Net[56], demonstrating the importance of our OCC dataset for providing more diverse data for vehicle understanding. In terms of per-keypoint occlusion category classification, we achieve 86.18% precision for binary visibility classification (visible vs. occluded). The OCC dataset allows us to further classify occlusion type, where we achieve 80.80%, 61.74%, and 63.01% for self-occluded, occ-by-others, and occ-by-truncation category respectively.

Occlusion Categories help Mining Unoccluded Objects: To detect occluded

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 4.1: Accuracy of our OCC module compared with baseline using bbox IOU threshold δ [58] in detecting Occluded objects.

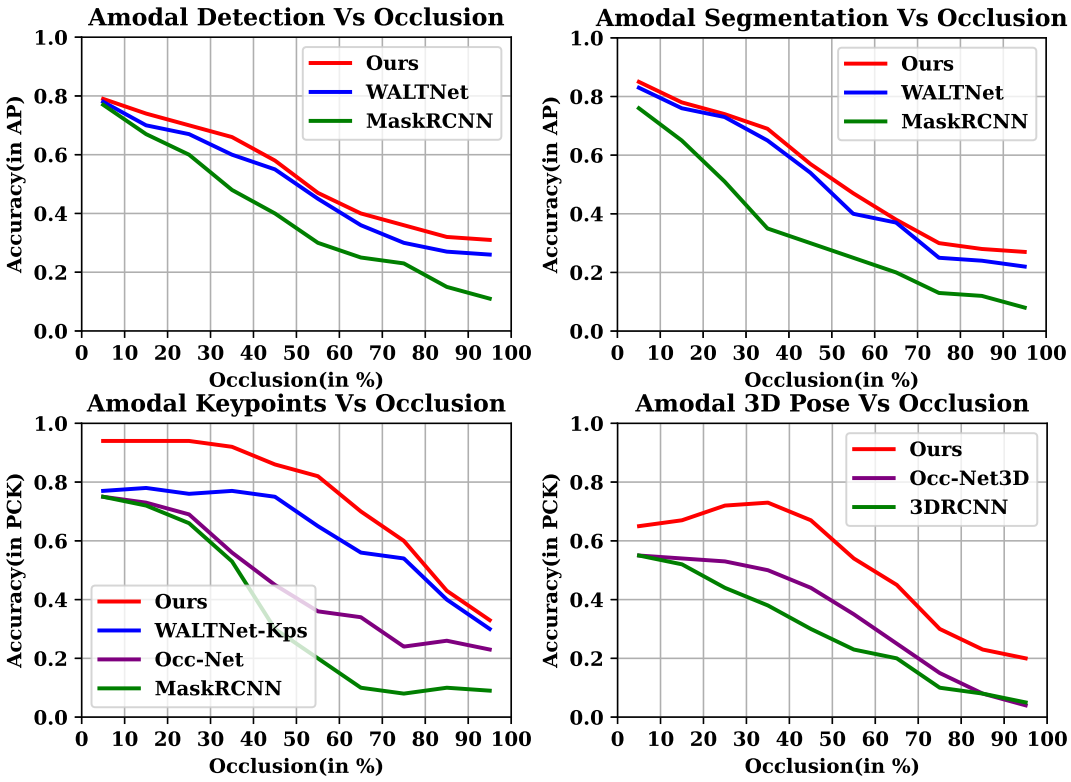


Figure 4.6: We show the accuracy of our method with respect to increasing percentage of occlusion on multiple tasks like amodal detection, segmentation, keypoint and 3D pose estimation. Observe that our method consistently performs better than other baselines showing robustness to increasing occlusion percentage. The baselines, WALTNet and Occ-Net, use only visible vs. occluded classes and 3DRCNN uses visible only.

	AN + Reproj			+OR+OD			+OR+OD+OC		
	AK	AS	Both	AK	AS	Both	AK	AS	Both
Seg	✗	72.3	72.5	✗	76.3	**76.9**	✗	76.4	76.5
Kps	73.5	✗	73.8	74.3	✗	81.2	85.1	✗	**85.3**
3D	55.4	42.3	56.5	58.5	46.9	58.3	62.3	50.3	**63.4**

Table 4.2: Accuracy analysis of each network component with different representations, i.e. keypoints and segmentation. We show the accuracy of segmentation, keypoint localization and 3D pose for a combination of network (AN, OD, OR) and representation type (AK and AS).Observe that with the addition of each constraint, the accuracy of 3D pose estimation improves. Specifically, adding OR and OD network helps improve the accuracy of segmentation and keypoints, while adding the occlusion category loss show improvement in the 3D pose estimation.

	Clip-Art WALT dataset			Stationary WALT dataset		
	Seg (AP)	Kps (PCK)	3D (AP)	Seg (AP)	Kps (PCK)	3D (AP)
3DRCNN [37]	✗	✗	56.5	✗	✗	76.8
WALTNet[58]	76.1	✗	✗	91.7	✗	✗
Occ-Net [56]	✗	73.8	55.4	✗	88.8	87.3
Ours	**76.5**	**85.3**	**63.4**	**93.5**	**93.2**	**91.7**

Table 4.3: Accuracy comparison of our method to baselines on both the composited data and the real world stationary WALT dataset. We consistently perform better than the baselines for amodal tasks compared to just learning visible vs occluded classification.

objects, WALT [58] used a simple heuristic where an object is classified as Occluded if its bounding box IOU with other objects (in the same category) is greater than δ. In Table 4.1, we compare this heuristics baseline (using different thresholds of δ) with our OCC network in detecting Occluded objects. We show that our OCC module is significantly more effective compared to the naive heuristic especially in inter-category occlusion scenarios (e.g., vehicle occluded by people or background objects), allowing us to effectively filter out unwanted occluded objects in the training dataset, thus simultaneously reduce training time and improve training data's purity.

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 4.9. We

Figure 4.7: We show qualitative results of our method on multiple sequences of the WALT dataset. The input image to the pipeline produces amodal segmentation mask and keypoint locations. Our method predicts 3D poses of the objects using an end-to-end differentiable optimization to produce the 3D poses of the objects. We show the reconstructed 3D poses of the objects from two views. We observe accurate reconstruction of vehicles in wide-ranging poses and different occlusion configurations. Further, we show results on different level and types of occlusions like truncation **(row 1)**, occlusion by vehicles **(row 1 and 2)**. Also observe that our method is able to disentangle multiple layers of occlusion where people and vehicles occluded the purple vehicle in **(row 4)**.

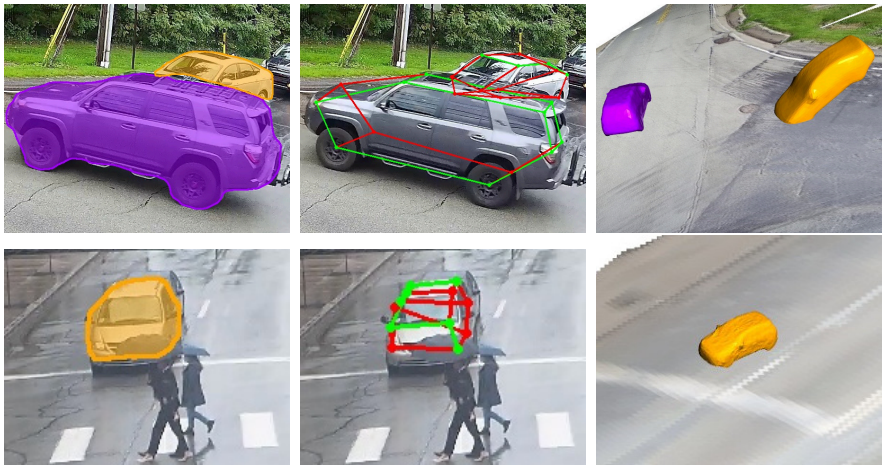
observe specifically that the 3D recovery consistently outperforms other baselines both in the case of self-occlusion and occlusion-by-others.

Dissecting the Network: We analyze the advantages and disadvantages of different network choices in Table 4.2. 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%.

Segmentation vs. Keypoints for Amodal 3D: We show analysis of using different representations, i.e., segmentation and keypoints for 3D recovery in Table 4.2. 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.

Binary visibility vs. multiple occlusion categories: We analyze our method by comparing to baselines which only use binary visibility in Table 4.3 and show qualitative results in Fig. 4.8. We observe marginal improvement over WALTNet for segmentation and bounding box 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 learning framework for handling both the self-occlusion and occlusion-by-others cases. Results of our method can be seen in Fig 4.7).



WALTNet[58] **Occ-Net[56]** **3DRCNN[37]**

Figure 4.8: Comparisons showing that our occlusion categorization (last two rows of Fig. 4.7) improves 2D/3D predictions compared to SOTA. While WALTNet and Occ-Net use visible vs. occluded classes, 3DRCNN uses visible only. Observe that the 3D fit to visible points shows large rotation error **(row 1)** or even misses objects **(row 2)** in severe occlusions. We are able to detect and reconstruct heavily occluded objects (80% occlusion) compared to previous baselines.

4.6 Additional Materials

Dataset	Image source	Appearance diversity in terms of				# images	# car instances	Occ. kpt. anns.	Per-kpt. 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
Carfusion	Handheld	No	No	No	No	53,000	100,000	Yes	No
ApolloCar3D	Self-driving	No	No	No	No	5,277	60,000	No	No
OCC	Handheld Self-driving Traffic cameras	Yes	Yes	Yes	Yes	7,018	42,547	Yes	Yes

Table 4.4: Summary and comparison of our OCC dataset to other publicly available datasets with vehicle keypoint annotations.

4.6.1 Network Architecture

Occlusion Category Classification (OCC) Network Given an input ROI from any off-the-shelf object detector, our goal is to infer the locations as well as the visibility status (visible/self-occluded/occluded-by-others/occluded-by-truncation) of 12 predefined vehicle semantic keypoints. Specifically, we discard detected objects (bounding boxes) with confidence score less than 0.3. In terms of network architecture, we utilized the top-down keypoint regression network from Occ-Net [56] with HRNet [71] as the backbone and added a simple classification head where we associate each keypoint with one of the four labels mentioned above. To handle the imbalance in the number of training samples between four categories, we used the weighted cross-entropy loss with a ratio of 1:1:5:8 (vis:self-occ:occ-oth:occ-trunc). The network is trained with a batch size of 16 using Adam [34] optimizer with a learning rate of 10^{-4} that halved every 10 epochs. We train the network end-to-end for 30 epochs using ground-truth keypoint location and category supervision data and report the best epoch on the corresponding dataset's validation set. Additional results from our OCC network are shown in Fig. 4.11.

Amodal 3D Reconstruction Network We use the Detectron2-based [75] codebase to train the network. We replicate the MaskRCNN Head for each of the proposed heads, i.e. Occluder Head, Occluded Head and Amodal Object Head. From the ROI, we compute feature maps of 3 layers i.e. first layer is $14 \times 14 \times 256$, second layer is $14 \times 14 \times 256$, third layer is $28 \times 28 \times 256$. Finally we do a softmax to produce the mask heatmap of $28 \times 28 \times c$, where c is the number of classes. Similarly, we follow the

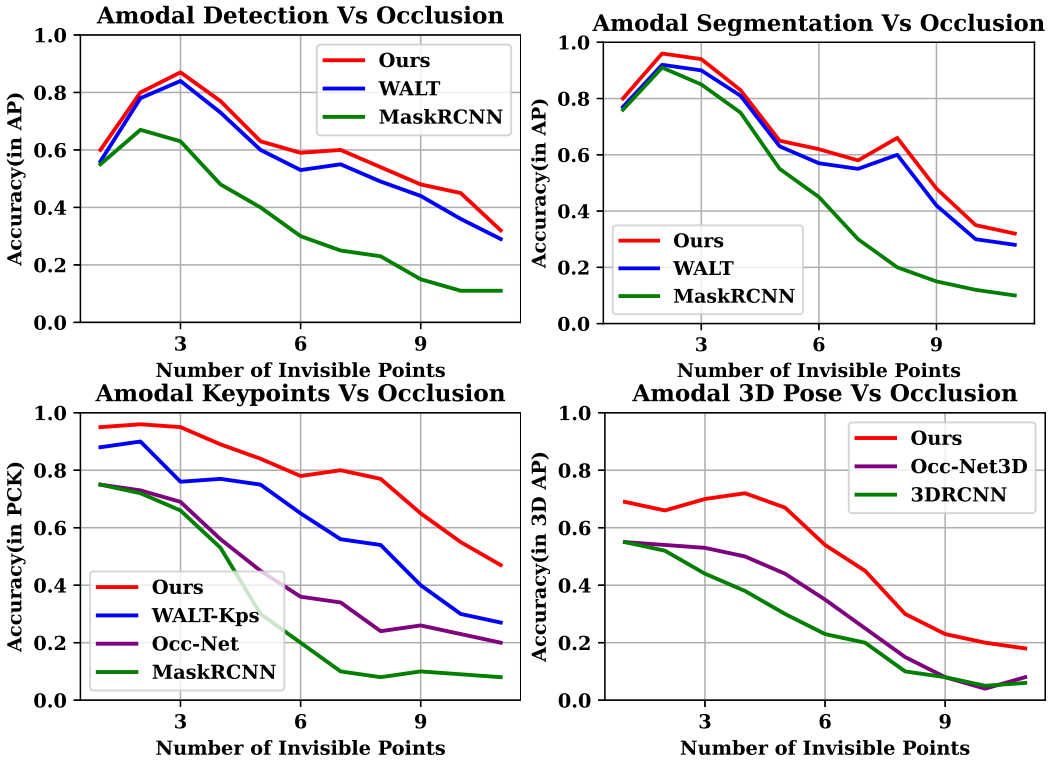


Figure 4.9: We show the accuracy of our method with respect to increasing number of occluded keypoints on multiple tasks like amodal detection, segmentation, keypoint and 3D pose estimation. Observe that our method consistently performs better than other baselines showing robustness to increasing occlusion percentage.

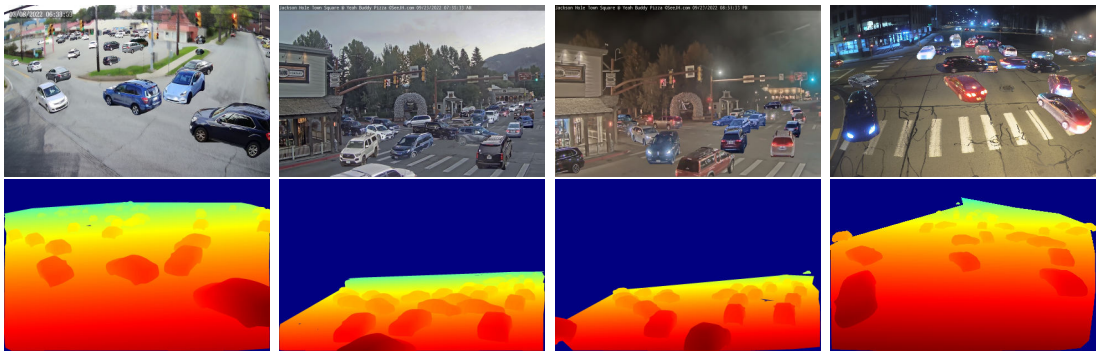


Figure 4.10: We show amodal depth computation on the 3D clip-art dataset. Observe that the depth information is accurate and can be used to train amodal depth networks as well.

same paradigm for the keypoint heatmap. Once we have the 2D amodal keypoints and segmentations, we use an occlusion-aware variant of BPnP[8] to learn for the 3D poses. The shape basis and mean shape are computed using [56] for keypoints and [32] for segmentation masks. We train the network using 4 2080Ti GPUs with a batch size of 11 for 12 epochs for all the trained models in the paper. We used 0.001 learning rate to train the network. We generate the 3D Clip-art automatically while training which are extensively dependent on the CPU computation for superimposing the objects and generating ground-truth.

4.6.2 Comparison to Keypoint Occlusions

We show the accuracy of the amodal segmentation, keypoint estimation and 3D recovery with respect to number of invisible/ occluded keypoints in Fig. 4.9. We improve over baseline methods like Occ-Net [56] and MaskRCNN [28] on occluded keypoint localization and amodal segmentation tasks respectively. We further show accuracy boost in 3D pose estimation and reconstruction compared to previous state-of-the-art like 3DRCNN [37].

4.6.3 Dataset Annotations

To increase the dataset diversity, we prioritized the number of different cameras and viewpoints rather than the number of images per camera. A summary and comparison

of our OCC dataset with other publicly available datasets are detailed in Table 4.4. On average, we extracted 120 images per camera source for more than 60 different cameras spanning a wide variety of viewpoints, appearances, sensor types, etc. For each image, we run an off-the-shelf object detector to extract the car instances with high confidence score. This set of car instances are manually annotated by the trained annotators from a commercial annotation service. We utilized a web-based interface annotation tool from DeepLabCut [49] where the annotators were asked to select 12 keypoint locations and its corresponding occlusion category for every car. Note that we also asked the annotators to filter out erroneous instances such as bad quality images and/or wrong detections. As of the time of paper submission, we have annotated a total of 42,547 car instances in 7,018 images.

4.6.4 2D/3D Clip-Art Data

We show the accurate amodal depth supervision from our automatically generated 3D clip-art data in Fig. 4.10. More examples from our 2D/3D Clip-Art amodal supervision data, including the clip-art image with corresponding amodal segmentation, keypoints, and 3D object reconstruction, are shown in Fig. 4.12.

4.6.5 Additional Qualitative Results:

We show additional results in Fig. 4.13 with different occlusion categories like self-occlusion, Truncation and occluded by others.

Importance of occlusion categories: We have shown in Figure 7 of the main paper comparisons to a baseline method [57] which classifies the keypoints into visible vs. occluded only. We show sample qualitative results in Figure 9 of the main paper that demonstrate clear improvements over baseline methods on 2D/3D reconstruction tasks. With additional occlusion categories, our method is capable of explicitly modeling object-to-object occlusion configurations, allowing a significant performance boost, especially in heavy occlusion scenarios.

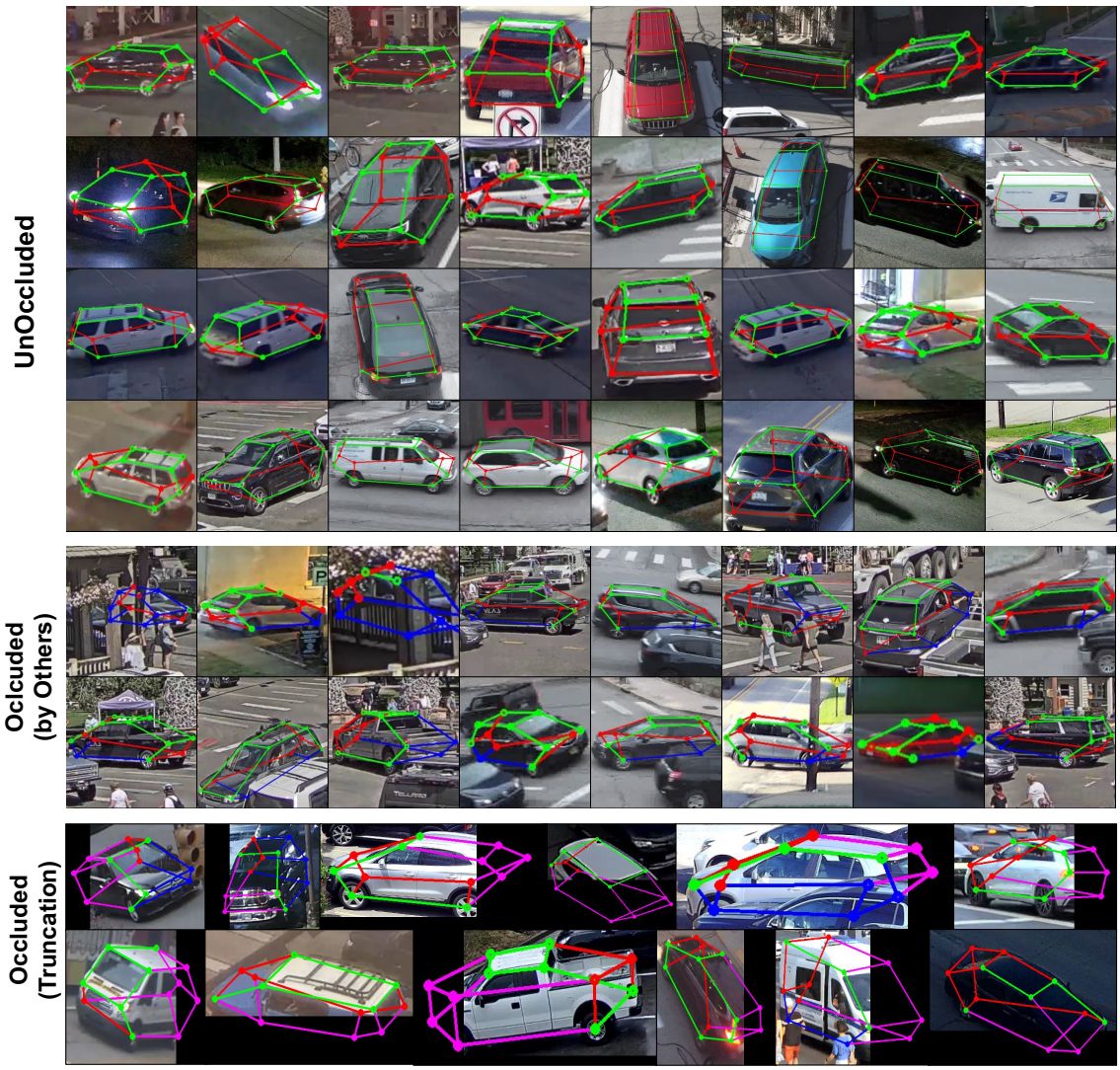
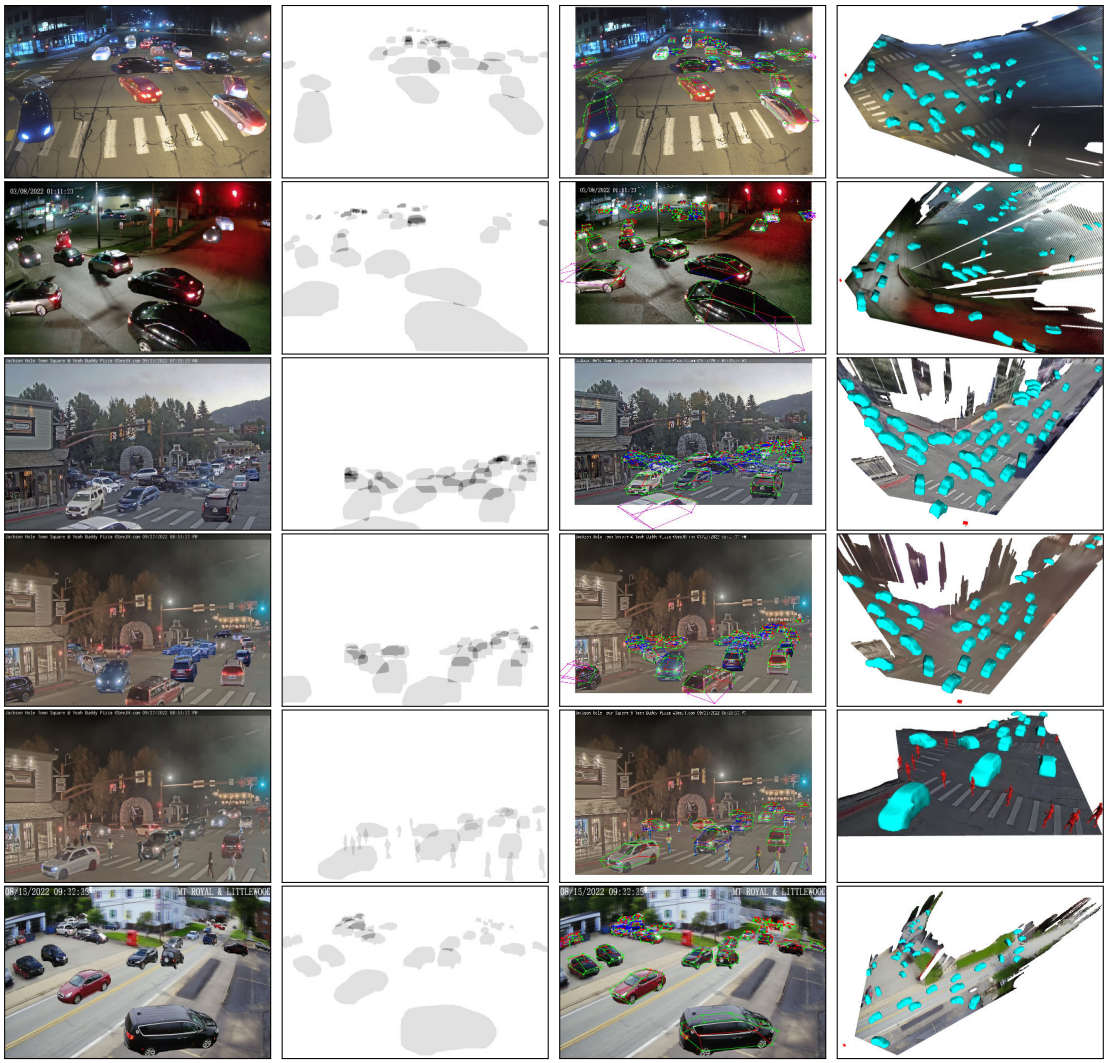


Figure 4.11: Additional results from our OCC network. Observe that our network is able to reliably localize keypoint locations as well as per-keypoint occlusion category in many complex configurations. (Per-keypoint occlusion type: visible, self-occluded, occ-by-truncation, and occ-by-others)



Clip-Art Image **Segmentation** **Keypoints** **3D Reconstruction**

Figure 4.12: **Automatically generated 2D and 3D Clip-Art to supervise our 3D amodal network:** Unoccluded objects are first mined using time-lapse imagery of the WALT dataset [58]. Randomly sampled and non-intersecting unoccluded objects are composited back into the background image in their respective original positions to maintain correct appearances. The resulting 3D Clip-Art images and their respective amodal segmentation masks, keypoint locations, and their occlusion categories and 3D meshes are shown. The clip-art method generates realistic appearances and 3D from any camera with diverse viewing geometry, weather, lighting, and occlusion configurations.

Figure 4.13: We show additional qualitative results 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

Chapter 5

Applications

The authorities of Shaler Township[1], located in Pennsylvania near Pittsburgh, has undertaken a project to leverage automated traffic analytics for studying traffic and pedestrian behavior. The aim is to enhance mobility and safety in the area. As part of this project, cameras have been installed at strategic locations along Mount Royal Boulevard, chosen for their significance in understanding the corridor's activity.

The captured visual data is subjected to analysis using custom algorithms specifically designed for vehicle detection, tracking, and computation of various analytic information. These analytics encompass vehicle counts, vehicle direction of travel, vehicle speed estimates, and vehicle classification. By extracting these insights from the data, the authorities gain valuable information that can inform decision-making processes related to traffic management and safety measures within Shaler Township.

Visual Data Information: The six camera locations in Shaler Township are depicted in Figure 5.1.

Camera Localization and Calibration: Leveraging our automated reconstruction and calibration pipeline, we successfully estimated various parameters for the cameras, including height from the ground, pitch, roll, horizontal field of view, and vertical field of view. The estimated values are presented in Table 5.1.

Vehicle Speed Estimates and Vehicle Activity Analytics: Figure 5.2 and Figure 5.3 provide vehicle speed estimates and activity heatmaps for the different cameras.

[1] https://goo.gl/maps/kY2Vrak4VN5XZ2gu7

Camera	Latitude	Longitude	Height	Pitch	Roll	HFOV	VFOV
1	40.524806	-79.962040	4.16 m	21.6°down	negligible	74°	45.7°
2	40.524637	-79.962152	4.71 m	19.6°down	negligible	74°	45.3°
4	40.514347	-79.959257	4.78 m	14.2°down	negligible	71°	44°
5	40.514347	-79.959257	4.26 m	19.6°down	negligible	72°	44°
6	40.516671	-79.959172	5.75 m	18.3°down	negligible	71°	44°

Table 5.1: Estimated Camera Parameters.

Activity Heatmap: Heatmaps are utilized to visualize the level of vehicle activity at each camera location. These heatmaps are generated by aggregating the tracks of all vehicles over the entire data acquisition period. Each pixel in the image accumulates a count based on the number of times a vehicle passes through that specific area. The accumulated values are then normalized by the maximum count, resulting in a value ranging from 0 to 1. Dark blue represents areas with no vehicular activity (0), while dark red indicates the highest level of vehicular activity (1). The color scale is unique to each heatmap, meaning that a value of 1 in one heatmap does not correspond to a value of 1 in another heatmap. To enhance visualization, a color scale is applied to the value range and smoothed using a Gaussian function. Finally, the heatmap is overlaid onto an image of the scene.

Vehicle Speed: Vehicle speed was estimated by using the camera calibration and localization methods previously discussed to estimate the ground plane yielding approximate speed calculations in 3D space. Rather than average the speed of the vehicle within the camera's field of view, a specific region of interest was defined for individual cameras. These virtual speed traps permitted estimates of speed a vehicle crossed over the region of interest, which was defined as a line on the road. Therefore, any reported speed estimates are instantaneous speed estimates. Estimated speed estimates are not linked to any personally identifying information and are reported only as aggregate findings.

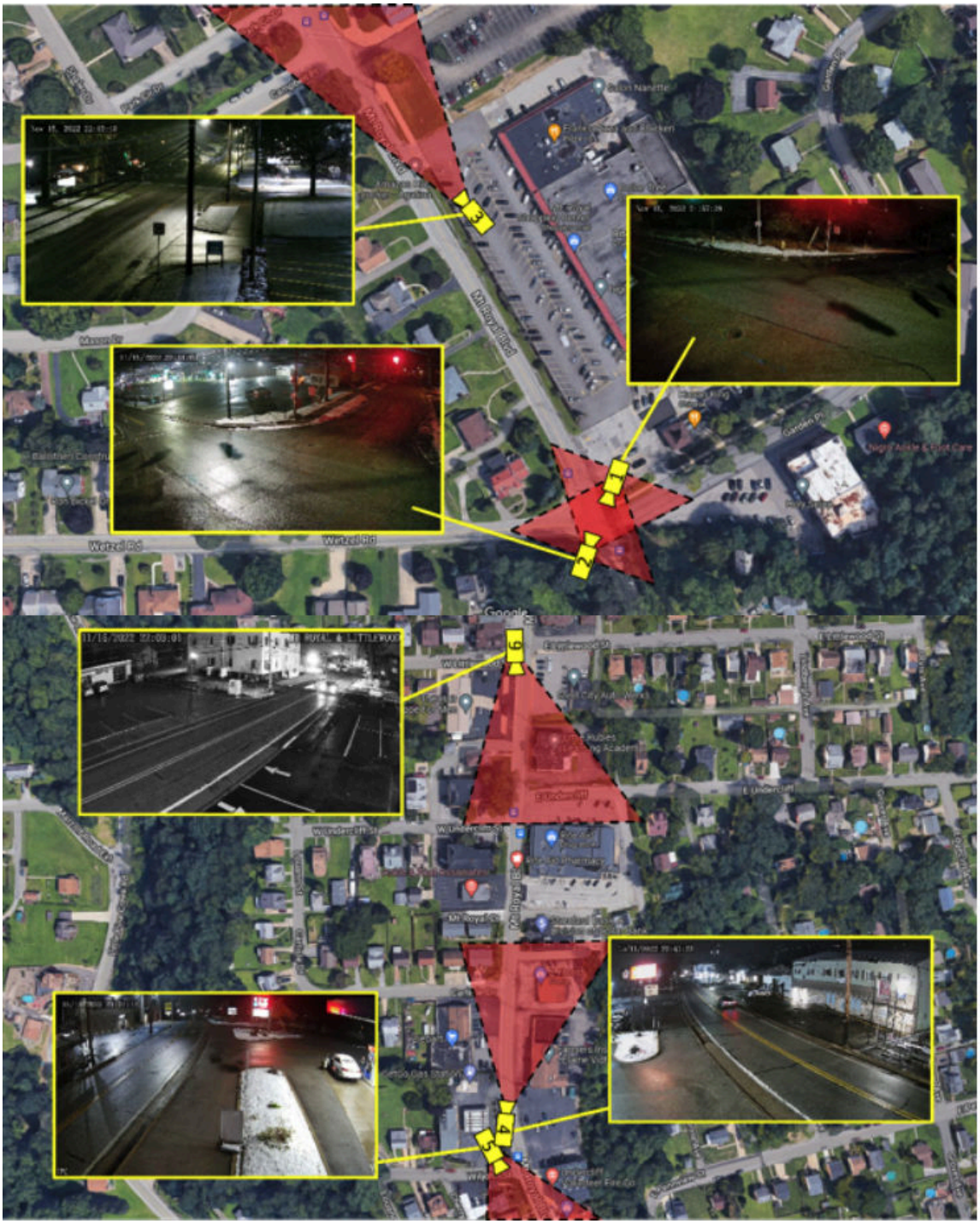
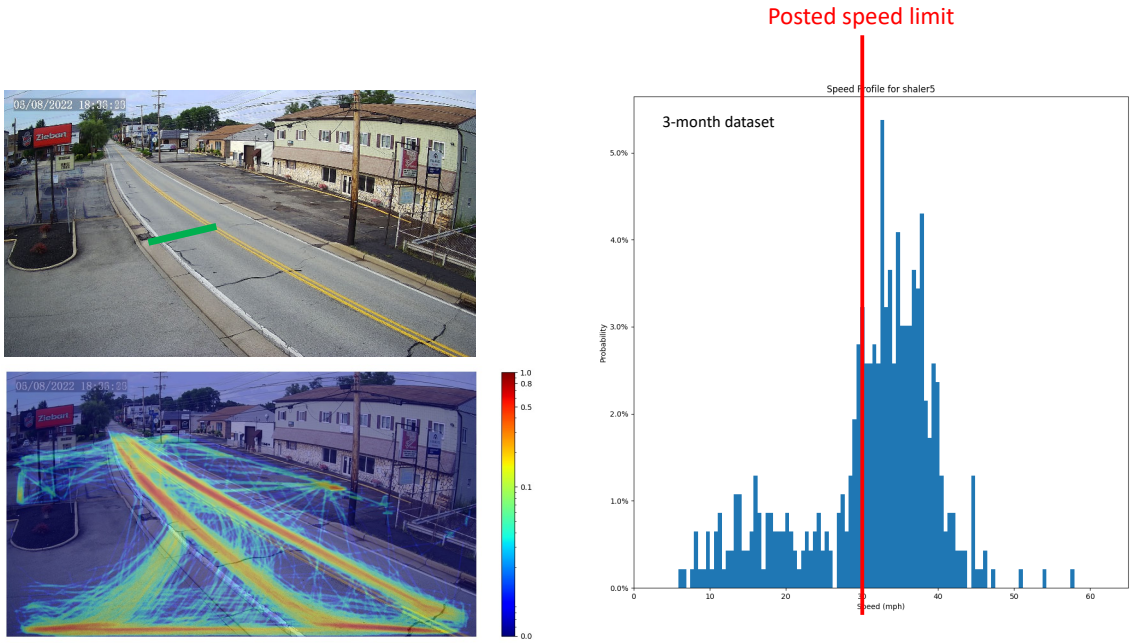
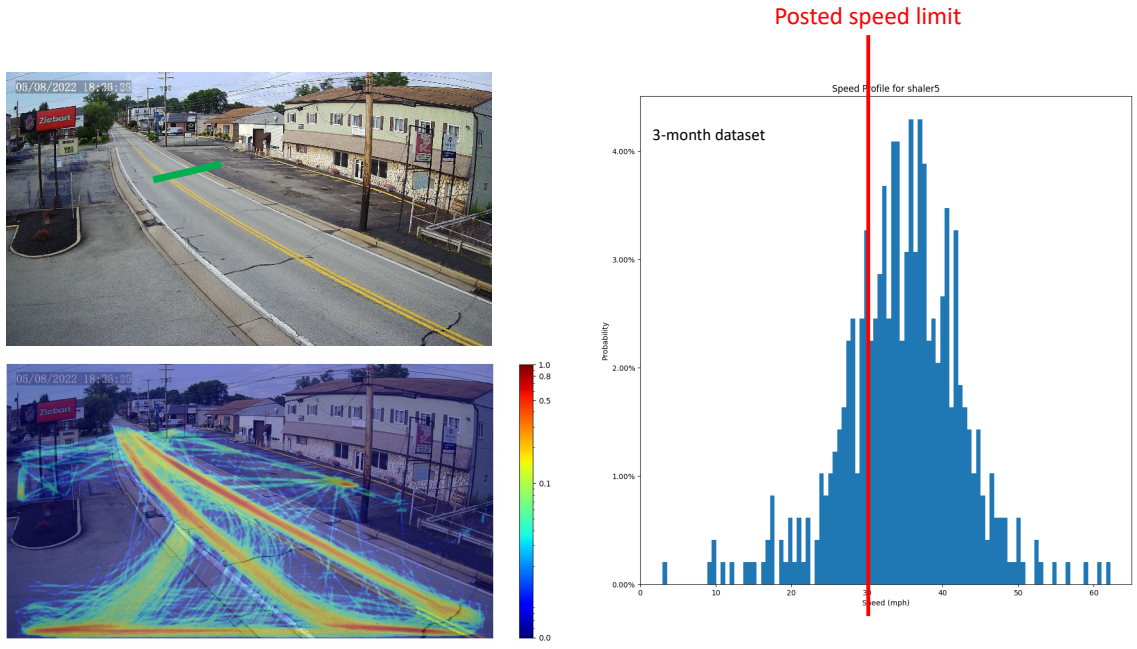


Figure 5.1: Location of 6 cameras that were installed along Mount Royal Boulevard. Each camera is illustrated with approximate viewing angle and field of view as shown in example image captures.

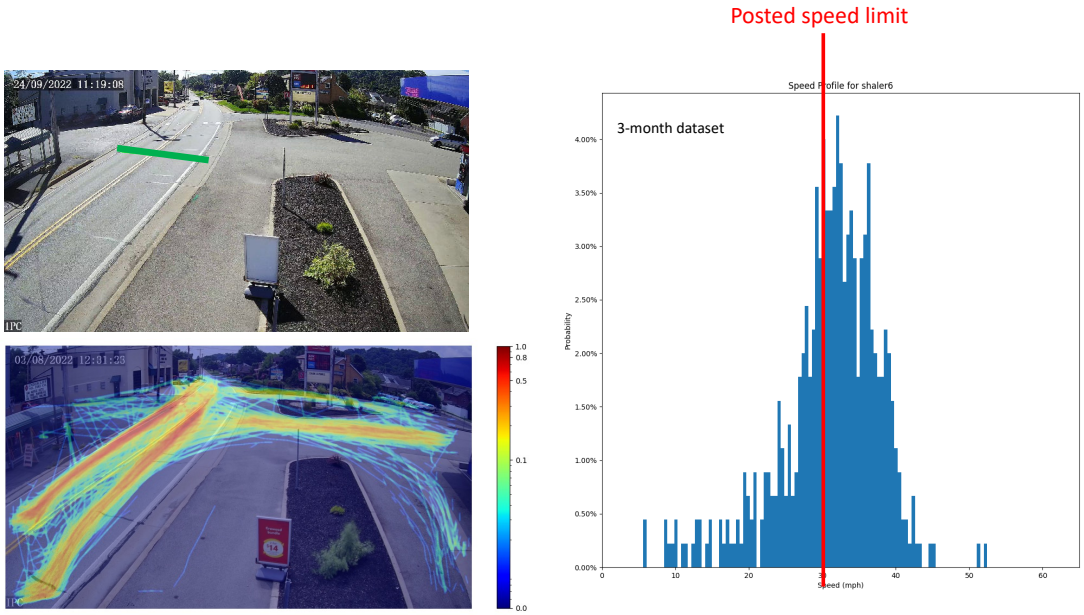


(a) Shaler 5a

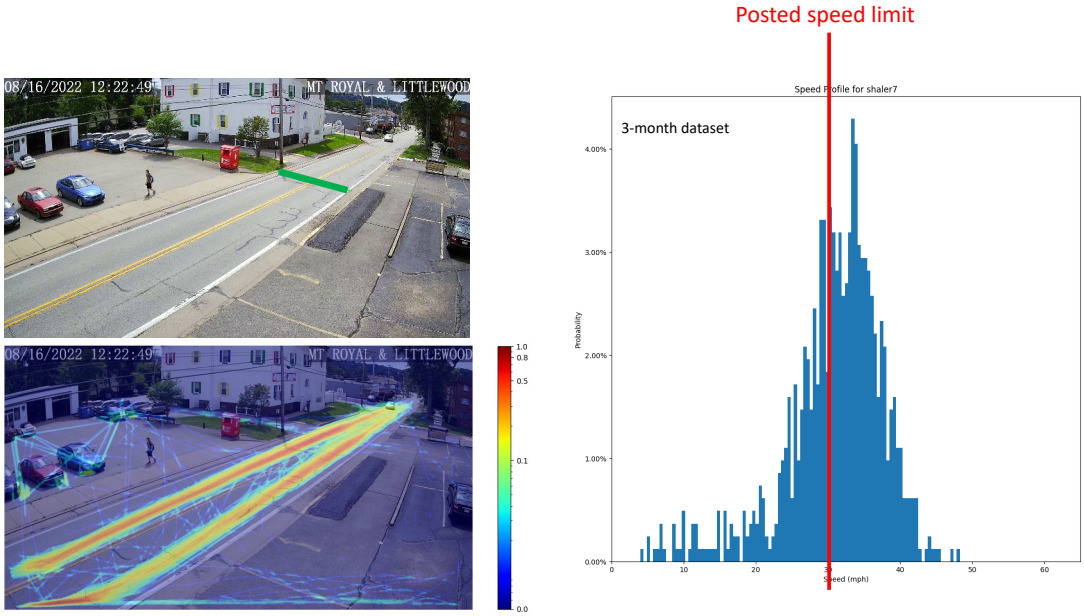


(b) Shaler 5b

Figure 5.2: Speed estimates and activity heatmap for two different virtual speed traps for the same camera. The virtual speed trap is visually represented by a green line.



(a) Shaler 6



(b) Shaler 7

Figure 5.3: Speed estimates and activity heatmap for two different cameras. The virtual speed trap is visually represented by a green line.

Chapter 6

Conclusions and Future Work

In this thesis, we presented a comprehensive approach to tackle two significant technical challenges. Firstly, we proposed a scalable framework for in-the-wild scene reconstruction and accurate camera localization, providing a foundation for various downstream applications that rely on precise real-world distance measurements. Secondly, we developed a novel method to automatically generate realistic 3D amodal supervision data from time-lapse imagery. By leveraging occlusion category information and utilizing mixed 2D/3D amodal representations, we obtain accurate 3D amodal reconstruction under occlusion. 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 at night. Our framework can be used in a variety of smart city applications, providing valuable information for improving transportation systems and urban infrastructure.
Future Work: Firstly, we aim to scale up the number of cameras to expand the scope of analysis and explore additional applications for automated traffic analysis. Secondly, our amodal object reconstruction method currently applies to individual cameras, and there is a need for research to generalize it across multiple views. Furthermore, the method assumes the availability of a mean shape model for the object class, making it more challenging to apply to rare or unique objects. Addressing these limitations will enhance the applicability and effectiveness of our framework in real-world scenarios.

Bibliography

[1] William Agnew, Christopher Xie, Aaron Walsman, Octavian Murad, Caelen Wang, Pedro Domingos, and Siddhartha Srinivasa. Amodal 3d reconstruction for robotic manipulation via stability and connectivity, 2020. 2.2

[2] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets deep learning for car instance segmentation in urban scenes. In *British machine vision conference*, volume 1, page 2, 2017. 2.2

[3] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013. 3.1

[4] António Bandeira Araújo. Drawing equirectangular vr panoramas with ruler, compass, and protractor. *Journal of Science and Technology of the Arts*, 10(1): 15–27, 2018. 3.1

[5] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245. 3.1

[6] Romil Bhardwaj, Gopi Krishna Tummala, Ganesan Ramalingam, Ramachandran Ramjee, and Prasun Sinha. Autocalib: Automatic traffic camera calibration at scale. *ACM Transactions on Sensor Networks (TOSN)*, 14(3-4):1–27, 2018. 1.1

[7] Joseph R Cathey and Matthew A Dailey. Camera calibration using lane markings: An evaluation of vanishing point detection methods. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):124–133, 2005. 1.1, 2.1

[8] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-end learnable geometric vision by backpropagating pnp optimization. In *CVPR*, 2020. 4.3, 4.6.1

[9] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao 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. 4.3

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 3.1

[11] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer, 2016. 2.2

[12] Matthew A Dailey, Benjamin C Schoepflin, Juraj Sochor, and Michal Seman. Camera calibration for traffic scene analysis using vehicle motion. *IEEE Transactions on Intelligent Transportation Systems*, 1(1):43–50, 2000. 1.1, 2.1

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. (document), 1.1, 2.1, 3.1, 3.2, 3.5

[14] Carlos A Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, et al. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21383–21392, 2022. 1.2

[15] Hoang Dung Do and Reinhard Klette. Camera calibration for road scene analysis using vehicle motion and lane markings. *IEEE Transactions on Intelligent Transportation Systems*, 16(7):2700–2712, 2015. 2.1

[16] Katerina Dubska, Jiri Matas, Ondrej Holik, and Michal Seman. Camera calibration using vehicle motion. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):283–294, 2014. 1.1, 2.1

[17] Katerina Dubska, Jiri Matas, Ondrej Holik, and Michal Seman. Camera calibration for road scene analysis using vehicle motion with robust estimation of camera parameters. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):540–551, 2015. 2.1

[18] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6144–6153, 2018. 2.2

[19] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita 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)*, pages 430–446, 2018. 2.2

[20] Patrick Follmann, Rebecca König, Philipp Härtinger, Michael Klostermann, and Tobias Böttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1328–1336. IEEE, 2019. 2.2

[21] Rik Fransens, Christoph Strecha, and Luc 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)*, volume 1, pages 300–307. IEEE, 2006. 1.2, 2.2

[22] Tianshi Gao, Benjamin Packer, and Daphne Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR 2011*, pages 1361–1368. IEEE, 2011. 1.2, 2.2

[23] Google. Google Street View. https://www.google.com/streetview/. 2.1, 3.1

[24] Vasileios Grammatikopoulos, Manolis N Lourakis, and John K Tsotsos. Camera calibration for road scene analysis using vanishing points. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):134–144, 2005. 2.1

[25] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4022–4031, 2022. 2.2

[26] Ruiqi Guo and Derek Hoiem. Beyond the line of sight: labeling the underlying surfaces. In *European Conference on Computer Vision*, pages 761–774. Springer, 2012. 2.2

[27] Jiebo He and Nicholas Yung. Camera calibration for traffic scene analysis using lane markings. *IEEE Transactions on Intelligent Transportation Systems*, 8(3): 417–427, 2007. 2.1

[28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 4.4, 4.6.2

[29] Edward Hsiao and Martial Hebert. Occlusion reasoning for object detectionunder arbitrary viewpoint. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1803–1815, 2014. 1.2

[30] 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. 2.2

[31] Vladislav Ishimtsev, Alexey Bokhovkin, Alexey Artemov, Savva Ignatyev, Matthias Niessner, Denis Zorin, and Evgeny Burnaev. Cad-deform: Deformable fitting of cad models to 3d scans. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 599–628. Springer, 2020. 2.2

[32] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *European Conference on Computer Vision*, pages 515–532. Springer, 2020. 4.6.1

[33] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4019–4028, June 2021. 2.2, 4.3

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 4.6.1

[35] Viktor Kocur and Milan Ftáčnik. Traffic camera calibration via vehicle vanishing point detection. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V 30*, pages 628–639. Springer, 2021. 1.1, 3.2, 3.3

[36] Philipp Krähenbühl. Free supervision from video games. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2955–2964, 2018. 2.2

[37] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018. 4.4, **??**, **??**, 4.6.2

[38] Abhijit Kundu, Yin Li, and James M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018. 2.2

[39] Man Lan, Jiang Zhao, and Tiantian Zhu. Camera calibration for traffic scene analysis using multiple vanishing points. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1806–1817, 2014. 2.1

[40] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81 (2):155, 2009. 4.2

[41] Chi Li, M Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D Hager, and Manmohan Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5465–5474, 2017. 1.2, 2.2, 4.4, 4.4, 4.5

[42] Fangyu Li, N Dinesh Reddy, Xudong Chen, and Srinivasa G Narasimhan. Traffic4d: Single view longitudinal 4d reconstruction of repetitious activity using self-supervised experts. In *IEEE Intelligent Vehicles Symposium*, 2021. (document), 4.1, 4.2

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva

Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4.4

[44] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 3.2

[45] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. 3.1

[46] Luiz Henrique Luvizon, Marcelo de Souza, and Mohamed Bennamoun. Camera calibration for traffic scene analysis using vehicle motion. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):363–374, 2014. 2.1

[47] Luiz Henrique Luvizon, Marcelo de Souza, and Mohamed Bennamoun. Camera calibration for traffic scene analysis using vehicle motion with uncertainty estimation. *IEEE Transactions on Intelligent Transportation Systems*, 17(1): 270–281, 2016. 2.1

[48] Ricardo Maduro and Reinhard Klette. Camera calibration for road scene analysis using manual measurements. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):501–510, 2008. 2.1

[49] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 2018. URL https://www.nature.com/articles/s41593-018-0209-y. 4.6.3

[50] K. Muller, A. Smolic, M. Drose, P. Voigt, and T. Wiegand. 3-d reconstruction of a dynamic environment with a fully calibrated background for traffic scenes. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(4):538–549, 2005. doi: 10.1109/TCSVT.2005.844452. 1.1

[51] Angga Nurhadiyatna and Reinhard Klette. Camera calibration for road scene analysis using manual measurements. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1149–1159, 2013. 2.1

[52] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2.2

[53] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2.2

[54] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance

segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019. 1.2, 2.2

[55] N. Dinesh Reddy, Minh Vo, and Srinivasa 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. 2.2, 4.4, 4.4, 4.5

[56] N. Dinesh Reddy, Minh Vo, and Srinivasa 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. 1.2, 2.2, 4.4, 4.5, **??**, **??**, 4.6.1, 4.6.2

[57] N. Dinesh Reddy, Minh Vo, and Srinivasa G. Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7326–7335, 2019. 4.1, 4.2, 4.6.5

[58] N. Dinesh Reddy, Robert Tamburo, and Srinivasa 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)*, pages 9356–9366, June 2022. (document), 1.2, 2.2, 4.2, 4.3, 4.2, 4.3, 4.4, 4.1, **??**, 4.5, **??**, 4.12

[59] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 2.2

[60] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 3.2

[61] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020. (document), 1.1, 2.1, 3.2, 3.5

[62] Benjamin C Schoepflin and Matthew A Dailey. Camera calibration for traffic scene analysis using vehicle motion. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):111–120, 2003. 2.1

[63] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2.1, 3.1

[64] Samuel Schulter, Menghua Zhai, Nathan Jacobs, and Manmohan Chandraker. Learning to look around objects for top-view representations of outdoor scenes. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1.2,

2.2

[65] Hao Sheng, Keniel Yao, and Sharad Goel. Surveilling surveillance: Estimating the prevalence of surveillance cameras with street view data. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 221–230, 2021. 1

[66] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017. 2.2

[67] Mostafa Sina and Reinhard Klette. Camera calibration for road scene analysis using manual measurements. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1595–1604, 2013. 2.1

[68] Jakub Sochor, Roman Juránek, and Adam Herout. Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding*, 161:87–98, 2017. 1.1, 2.1

[69] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *CVPR*, 2019. 4.4, 4.4, 4.5

[70] Andrea Vedaldi and Andrew Zisserman. Structured output regression for detection with partial truncation. In *Advances in neural information processing systems*, pages 1928–1936, 2009. 2.2

[71] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020. 4.6.1

[72] Kunfeng Wang, Hua Huang, Yuantao Li, and Fei-Yue Wang. Research on lane-marking line based camera calibration. In *2007 IEEE International Conference on Vehicular Electronics and Safety*, pages 1–6. IEEE, 2007. 1.1

[73] Manchen Wang, Joseph Tighe, and Davide 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. 1.2

[74] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 9621–9630, 2019. 2.2

[75] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick.

Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4.6.1

[76] 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*, pages 75–82, March 2014. 4.4, 4.4, 4.5

[77] Xiaoding Yuan, Adam Kortylewski, Yihong Sun, and Alan Yuille. Robust instance segmentation through reasoning about multi-object occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11141–11150, June 2021. 2.2

[78] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, June 2020. 1.2

[79] Xingchen Zhang, Yuxiang Feng, Panagiotis Angeloudis, and Yiannis Demiris. Monocular visual traffic surveillance: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14148–14165, 2022. 2.1

[80] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 1.1, 2.1

[81] Xiaowei Zhou, Spyridon Leonardos, Xiaoyan Hu, and Kostas Daniilidis. 3d shape estimation from 2d landmarks: A convex relaxation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4447–4455, 2015. 2.2

[82] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2.2

[83] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1464–1472, 2017. 1.2, 2.2

[84] M Zeeshan Zia, Michael Stark, and Konrad Schindler. Towards scene understanding with detailed 3d object representations. *IJCV*, 2015. 2.2