Camera Calibration through Camera Projection Loss

by

Talha Hanif Butt (l18-1864)

Masters of Science in Data Science

A thesis submitted under the supervision of Dr. Asif Mahmood Gilani and Dr. Murtaza Taj in partial fulfillment of the requirements for the degree of Masters of Science in Data Science at the National University of Computer & Emerging Sciences

Department of Computer Science
National University of Computer & Emerging Sciences

Lahore, Pakistan, 2021

© Talha Hanif Butt 2021

Abstract

Estimation of camera parameters is known as Camera Calibration with 3D reconstruction, robot interaction and autonomous driving as some of its applications. In this work we propose a novel method to predict extrinsic (baseline, disparity, pitch, and translation), intrinsic (focal length and principal point offset) parameters using an image pair. To the best of our knowledge, ours is the first method to jointly estimate these parameters via a multi-task learning methodology. Unlike existing methods, instead of designing an end-to-end solution to directly estimate the desired parameters, we propose a new representation that incorporates camera model equations as a neural network in a multi-task learning (MTL) framework. We estimate the desired parameters via novel *camera projection loss* (CPL) that uses the camera model neural network to reconstruct the 3D point cloud and uses the reconstruction loss to estimate the camera parameters. We also propose a novel dataset using CARLA Dosovitskiy et al. (2017) Simulator to train and test our proposed method. Empirically, we demonstrate that our proposed approach achieves better performance with respect to both deep learning-based and traditional methods on both real and synthetic data.

Acknowledgements

I would like to thank **Dr. Murtaza Taj** for his continuous support for the last 2 years and 9 months. He provided me the opportunity to work on my own ideas and helped me in every possible manner wherever I got stuck. I can never forget our long discussions about what to do next. Actually, before locking Camera Calibration as my thesis topic, I had worked on Cross view Image Retrieval and Cyclist Detection but whenever I asked him that we can try another idea, he always stood by me. I cannot thank him enough for his guidance but will always remember it.

I would like to thank **Numan Bhai** for being the root cause of my first publication and guiding me through the process of publishing and always trying to guide me in a brotherly manner and allowing me to work on my own ideas without ever scolding me of what was possible if I had continuously worked on Cross View Retrieval. I will always be grateful for all those discussions and experiences.

Now, the legendary **Mohbat Bhai**, the first person I started working with after my joining. He was always available for help and tried his best to help me understand his previous work on Cross View Retrieval. From writing in latex to drawing figures, I am grateful for all his help and guidance.

Now comes the culprit for infusing in me the passion to work on my own ideas instead of current projects in the lab, the legend, **Faisal Bhai**, a hardworking man with high goals and the ability to pay the price for his dreams. I don't know how a person can suddenly decide to go for Medical School while working all his life in Electronics but as they say, When there is a will, there is a way. He was the perfect example of that. Thank you Ustaad ji, will always remember your contribution.

Fezan and Wahab were responsible for me realizing that we can manage multiple things at a time while performing reasonably well at all of them.

I would like to thank **Dr. Asif Gilani** for his support and guidance throughout the process of my Thesis. I remember him politely telling me that you are technically good but lack good writing capabilities and always motivating me regarding my abilities.

Now comes the most important partner in my journey, **my mother**, I want to thank her for always believing in me and never asking anything and just relentlessly supporting me. I still remember when I told her that I want to go to Islamabad for an year for research and she agreed instantly without asking anything. Basically, I believe that I can do anything I want and my mother is the root cause for this.

Dr. Sibt is the cause I am trying to perform research today. I still remember his introductory lecture on Digital Image Processing and his lines that **We will show you the sky and will teach you to fly and then it's up to you how high you can go**. He was the most passionate teacher I came across and was a great influence in shaping

my career path. I worked with him for about 2 years and the experience was amazing. I basically loved him and thought that I would have married him if he was not a man.

Some of the people that I firmly believe have contributed in my journey are **Sir Shahroz, Sir Ateeq, Madam Rafia, Sir Abbas, Sir Abdul Ali, Ali, Ammar, Hassam, Mian Saad, Zaid, Abdullah, Asim, Ahmed Nadeem, Muneeb Adil** and many more. Thank You all for your contributions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Camera calibration deals with finding the five intrinsic (focal length, image sensor format, and principal point) and six extrinsic (rotation, translation) parameters of the specific camera. Camera calibration is useful in many computer vision tasks such as image alignment and 3D reconstruction which are building blocks for many important applications including self driving cars, augmented reality, 3D pose estimation.

1.1 Geometry based Camera Calibration

In computer vision, extensive research has been performed for understanding the process of image formation Hartley and Zisserman (2003a), which is the underlying driver of good camera alignment given that enough mathematical requirements are met to fit the camera model. Multi-view geometry (MVG) based methods typically find corresponding points between images to generate enough constraints to solve the camera model equations and thus calculate the calibration parameters. In the course of recent years, various calibration strategies have been proposed Unnikrishnan and Hebert (2005); Geiger et al. (2012); Levinson and Thrun (2013); Pandey et al. (2012); Taylor and Nieto (2015). However, by far most of these procedures rely upon explicit calibration targets like checkerboards, and require critical measures of manual exertion Unnikrishnan and Hebert (2005); Geiger et al. (2012). The process of image correspondence is often automated via solid signals, for example, evaporating focuses and straight lines that can be utilized to recuperate the camera parameters Caprile and Torre (1990); Deutscher et al. (2002). However, MVG-based methods lack generalizability because of pictures taken in unstructured conditions.

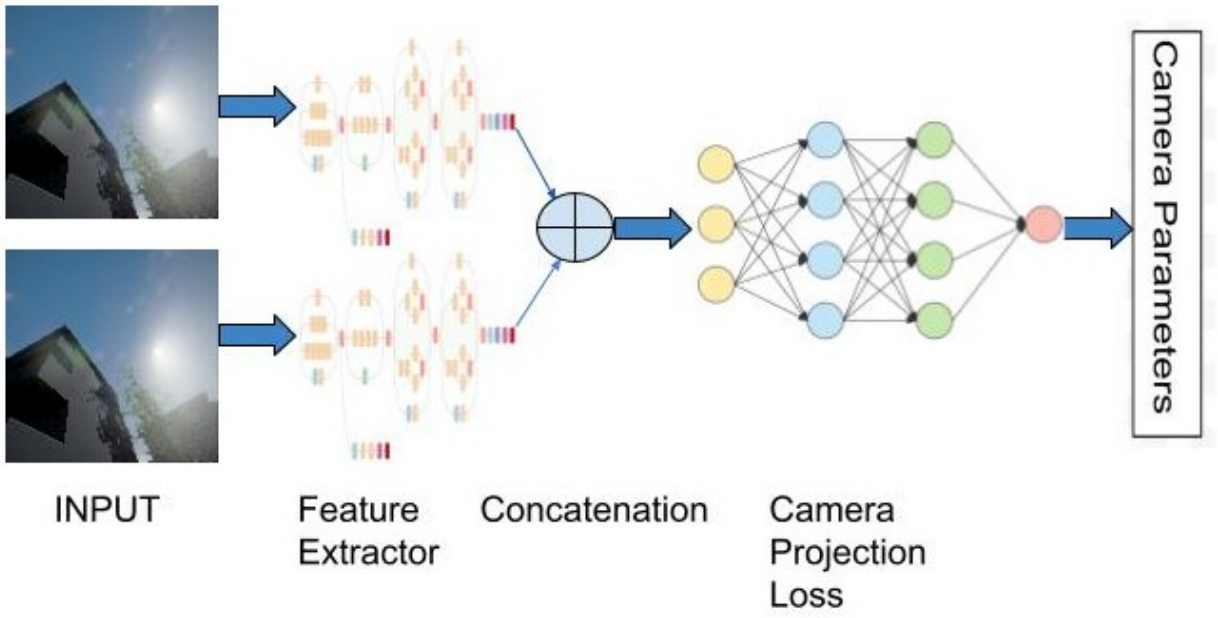


Figure 1.1: Our method estimates extrinsic (baseline, disparity, pitch and translation) and intrinsic (focal length and principal point offset) parameters using pre-trained Inception-v3 Szegedy et al. (2016) and the proposed Camera Projection Loss.

1.2 CNN based Camera Calibration

There is a basic requirement for programmed alignment strategies which can significantly expand the adaptability and flexibility of camera-based systems. The recent advancements in deep machine learning particularly convolutional neural networks (CNNs) have resulted learning-based methods for the estimation of camera parameters Workman et al. (2015); Rong et al. (2016); Hold-Geoffroy et al. (2018); Lopez et al. (2019); Zhai et al. (2016); DeTone et al. (2016); Bogdan et al. (2018); Zhang et al. (2020). CNN has been utilized to relapse the field of perspective on the camera by Workman et al. (2015). The overall goal of this approach was to estimate the horizon line in the image Workman et al. (2016). Instead of regression, Rong et al. (2016) proposed a classification-based method to estimate radial distortion model. Similarly, Hold-Geoffroy et al. (2018) also used a classification network. They artificially produce pictures of discretionary size, focal length, and rotation to obtain training data. Exemplary and learned strategies can likewise be joined by using a learning method to get an earlier appropriation on the camera parameters . These assessments are then refined utilizing exemplary strategies. More recently, Deep-Calib Bogdan et al. (2018); Lopez et al. (2019) proposed a monocular image-based solution to estimate the distortion parameter Barreto (2006). A summary of the various inputs used by the existing

methods to estimate some of the calibration parameters is shown in Table 2.1. Most of the existing ones usually ignore the underlying mathematical formulation of the camera model and instead propose a start to finish system to straightforwardly assess the ideal parameters. Thus they are difficult to interpret for real-world applications and has so far been ready to chiefly appraise the focal length of the camera via single image only DeTone et al. (2016); Bogdan et al. (2018); Zhang et al. (2020).

1.3 Loss balancing

In this work, we likewise investigate the issue of adjusting loss components for camera alignment and propose a quicker approach dependent on camera projections, similar to DeepCalib Lopez et al. (2019) but using proxy variables through 2D to 3D projection which are dependent on each other instead of using intermediary factors (that are straightforwardly apparent in the picture, for example, skyline and the separation from the focal point of the picture to the skyline).

1.4 Contributions

In this work we not only estimate all the 8 out of 11 calibration parameters, we also estimate baseline and disparity. Furthermore, propose a learning-based methods that rely on the underlying mathematical equations of pinhole camera model. The major contributions of our work are as follows:

- As far as we could possibly know, this work is the principal learning-based technique to mutually assess both intrinsic and extrinsic camera parameters including camera baseline, disparity, pitch, translation, focal length and principal point offset.
- The existing learning based approaches DeTone et al. (2016); Workman et al. (2015); Zhang et al. (2020) have not been applied to the estimation of all 10 camera parameters due to lack of any dataset. We addressed this limitation by generating a synthetic dataset from two towns in CARLA Dosovitskiy et al. (2017) simulation consisting of 48 different camera settings.
- Unlike existing methods, instead of designing an end-to-end solution to directly estimate the desired parameters DeTone et al. (2016), we proposed a new representation that represents camera model equations as a neural network in a multi-task learning (MTL) framework.

- We proposed a novel *camera projection loss* (CPL) that combines analytical equations in learning framework. We use the proposed camera model neural network to reconstruct 3D point cloud and use the reconstruction loss to estimate the camera parameters.

The remainder of the report is structured as follows: Chapter 2 focuses on the literature, Chapter 3 explains the datasets, Chapter 4 discusses our methodology, Chapter 5 focuses on the experiments and results while Chapter 6 contains conclusion and future work.

Chapter 2

Literature Review

Ongoing works have utilized the achievement of convolutional neural networks and proposed utilizing learned strategies to appraise camera parameters. Various parts of the issue of alignment utilizing pictures have been considered previously. Workman et al. (2015) prepared a CNN to perform relapse of the field of perspective on a pinhole camera, later zeroing in on identifying the skyline on pictures Workman et al. (2016), which is an intermediary for the slant and roll points of the camera if the focal length is known. Rong et al. (2016) utilize a grouping way to deal with single-parameter spiral bending model from Fitzgibbon (2001a). Hold-Geoffroy et al. (2018) first joined extraneous and characteristic calibration in a solitary network, anticipating the slant, roll and focal length of a pinhole camera through a grouping approach. They depended on upstanding 360 degree symbolism to artificially produce pictures of self-assertive size, focal length and rotation, an approach that has also been used to generate training data Lopez et al. (2019). Exemplary and learned strategies can be consolidated. In Zhai et al. (2016), learned techniques are utilized to acquire an earlier appropriation on the conceivable camera parameters, which are then refined utilizing exemplary strategies, speeding up the execution time and vigor concerning completely mathematical strategies. We don't follow such a methodology in this work. Nonetheless, the forecast created by our technique can be utilized as an earlier in such pipelines. More recently, Deep-Calib Bogdan et al. (2018); Lopez et al. (2019) remember the distortion parameter for the single picture self-calibration issue by creating contorted pictures utilizing the unified projection model Barreto (2006).

2.1 Single Input

Lee (2012) remake the projective design to discover camera parameters including focal length, position, and direction given a solitary picture of a scene square shape

Reference	Input	Estimated Parameters
Lopez et al. (2019)	RGB Image	Tilt, Roll, Focal length, Radial distortion
Schöller et al. (2019)	RGB Image, Projected Radar Data	Tilt, Pan, Roll
Iyer et al. (2018)	RGB Image, Raw LiDAR point cloud	Rotation, Translation
Poursaeed et al. (2018)	Stereo Image pair	Fundamental Matrix
En et al. (2018); Melekhov et al. (2017); Charco et al. (2018); Hansen et al. (2012)	RGB Image pair	Rotation, Translation
Bogdan et al. (2018)	RGB Image	Focal length, Distortion
Bhardwaj et al. (2018); Xiang et al. (2017); Nakajima and Saito (2017); Komorowski and Rokita (2012); Kendall et al. (2015)	RGB Image	Rotation, Translation
Hold-Geoffroy et al. (2018)	RGB Image	Tilt, Roll, Focal length
Workman et al. (2015)	RGB Image	Focal length
Shalnov and Konushin (2017)	Head Detections, Focal length	Rotation, Translation
Lee (2012)	RGB Image	Focal length, Position, Orientation
Ranftl and Koltun (2018)	Putative matches	Fundamental Matrix

Table 2.1: Overview of some recent configurations for different aspects of Camera Calibration

of an obscure perspective proportion and size. Workman et al. (2015) investigate the utilization of a profound convolutional neural network, prepared on normal pictures got from Internet photograph assortments, to straightforwardly gauge the focal length utilizing just crude pixel forces as info highlights. Hold-Geoffroy et al. (2018) straightforwardly construe camera alignment parameters from a single image using a deep convolutional neural network. Bhardwaj et al. (2018) takes advantage of profound figuring out how to separate chosen central issue highlights from vehicle pictures in the video and utilizes a novel sifting and conglomeration calculation to naturally deliver a vigorous gauge of the camera calibration parameters from only many examples. Xiang et al. (2017) present PoseCNN, another Convolutional Neural Network for 6D item present assessment. PoseCNN gauges the 3D interpretation of an item by confining its middle in the picture and anticipating its separation from the camera. The 3D turn of the article is assessed by relapsing to a quaternion portrayal. Nakajima and Saito (2017) propose a clever technique for vigorous camera present assessment utilizing numerous component descriptor information bases produced for each apportioned perspective, wherein the element descriptor of each keypoint is practically invariant. Komorowski and Rokita (2012) presents a technique for outward camera calibration (assessment of camera revolution and interpretation networks) from a succession of pictures. It is expected camera inherent network and bending coefficients are known and fixed during the whole arrangement. Kendall et al. (2015) present a hearty and continuous monocular six level of opportunity relocalization framework. The framework prepares a convolutional neural network to relapse the 6-DOF camera present from a solitary RGB picture in a start to finish way without any need of extra designing or chart advancement. Bogdan et al. (2018) consequently appraises the natural parameters of the camera (focal length and distortion parameter) from a solitary info picture. To prepare the CNN, they influence the lot of omnidirectional pictures accessible on the Internet to consequently create an enormous scope dataset made out of millions of wide field-of-see pictures with ground truth inherent parameters. Lopez et al. (2019) propose a definition for spiral mutilation that is more qualified for learning than straightforwardly anticipating the contortion parameters. Also, foreseeing extra heterogeneous factors compounds the issue of loss adjusting. They propose another loss dependent on direct projections so that they don't have to adjust heterogeneous loss terms.

2.2 Multiple Inputs

Schöller et al. (2019) propose the primary information driven strategy for programmed rotational radar-camera alignment without devoted calibration targets. Their methodology depends on a coarse and a fine convolutional neural network. They utilize a boosting-enlivened preparing calculation, where they train the fine network on the

remaining residual error of the coarse network. Iyer et al. (2018) propose a mathematically managed profound network able to do naturally assessing the 6-DoF inflexible body change between a 3D LiDAR and a 2D camera continuously. CalibNet lightens the requirement for calibration targets, in this way bringing about critical savings in calibration endeavors. Poursaeed et al. (2018) propose novel neural network designs to appraise central lattices in a start to finish way without depending on point correspondences. New modules and layers are acquainted all together with safeguard numerical properties of the major lattice as a homogeneous position 2 framework with seven levels of opportunity. En et al. (2018) addresses the undertaking of relative camera present assessment from crude picture pixels, through profound neural network. The proposed RPNet network accepts sets of pictures as info and straightforwardly induces the relative postures, without the need of camera natural/extraneous. While cutting edge frameworks dependent on SIFT + RANSAC, can recuperate the interpretation vector simply up to scale, RPNet is prepared to deliver the full interpretation vector, in a start to finish way. Melekhov et al. (2017) presents a convolutional neural network based approach for assessing the overall posture between two cameras. The proposed network takes RGB pictures from the two cameras as info and straightforwardly creates the overall turn and interpretation as yield. The framework is prepared in a start to finish way using move learning from an enormous scope arrangement dataset. Charco et al. (2018) proposes to utilize a profound learning network engineering for relative camera present assessment on a multi-see climate. The proposed network is a variation engineering of AlexNet to use as regressor for forecast the general interpretation furthermore, turn as yield. The proposed approach is prepared from scratch on an enormous informational index that takes as info a couple of pictures from a similar scene. Hansen et al. (2012) present a calculation for persistent, online sound system outward re-alignment working just on inadequate sound system correspondences on a for each casing premise. They get the 5 level of opportunity outward posture for each casing, with a proper pattern, making it conceivable to display time-subordinate varieties. The underlying extraneous evaluations are found by limiting epipolar errors, and are refined through a Kalman Filter (KF). Perception covariances are gotten from the Cramer-Rao lower bound of the arrangement vulnerability. Shalnov and Konushin (2017) propose a clever strategy for camera present assessment dependent on individuals perception in the info video caught by static camera. Rather than past procedures, their technique can manage bogus positive recognitions and erroneous restriction results. In particular, the proposed technique doesn't make any supposition about the used article locator and accepts it as a parameter. Also, they donot require a gigantic named dataset of genuine information and train on the engineered information as it were. They apply the proposed method for camera present assessment dependent on head perceptions.

2.3 Multi-task learning

When preparing a convolutional neural network for alignment, the loss is a total of a few losses, one for every parameter. This situation is typically known as multi-task learning Caruana (1997). Multi-task learning needs to manage the test of preparing a network to play out a few assignments with discrete losses. The majority of these methodologies depend on a weighted total of the loss terms, varying on the way wherein the loads are set at training time: Kendall et al. (2018) utilize Gaussian and softmax probabilities (for relapse and grouping, separately) to weight the diverse loss terms as indicated by an errand subordinate vulnerability. As opposed to these vulnerability based strategies, Chen et al. (2017) decide the worth of the weights by changing the angle sizes related to every loss term.

Chapter 3

Datasets

3.1 Synthetic Data

We trained and evaluated our proposed approach by generating a new dataset using Town 1 and Town 2 of CARLA Dosovitskiy et al. (2017) Simulator. The dataset consists of 48 camera configurations with each town having 24 configurations. The parameters modified for generating the configurations include fov, x, y, z, pitch, yaw, and roll. Here, fov is the field of view, (x, y, z) is the translation while (pitch, yaw, and roll) is the rotation between the cameras. The total number of image pairs is $79,320$, out of which $18,083$ belong to Town 1 while $61,237$ belong to Town 2, the difference in the number of images is due to the length of the tracks.

3.1.1 Dataset Format

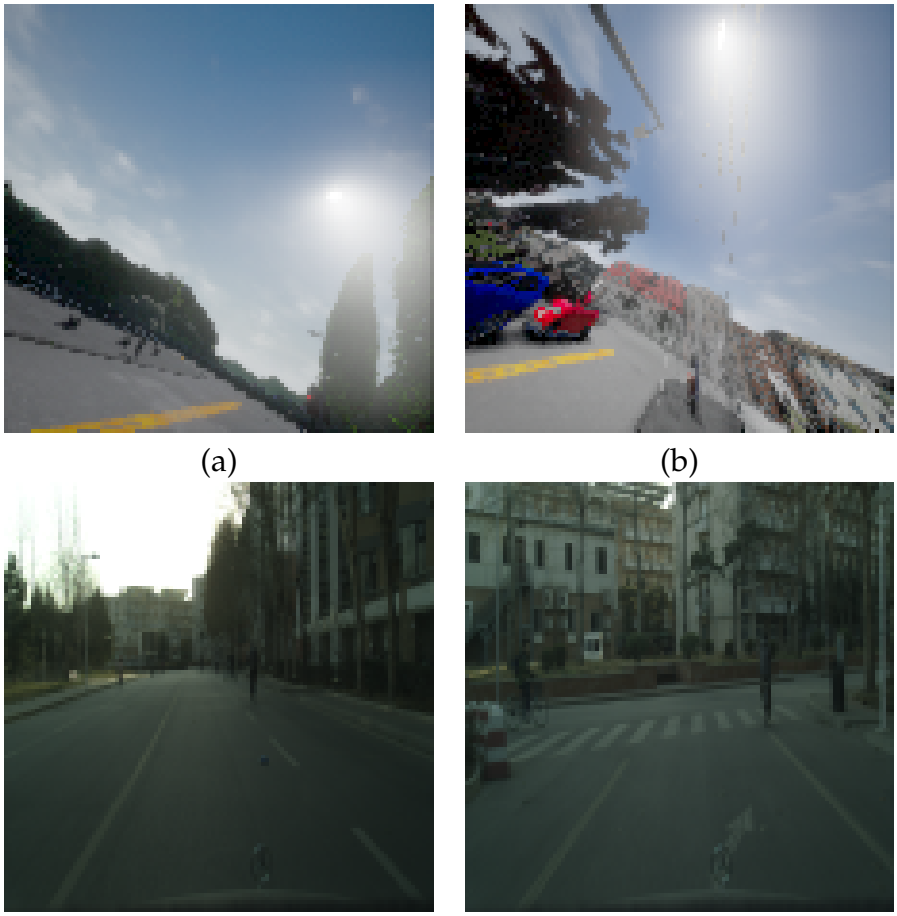
Dataset is collected using a hardcoded expert driver navigating towards a goal.

The dataset directory should look like the structure in Figure 3.2.

Dataset Metadata

Each dataset contains a metadata file with the following information:

- Cameras used.
- The FOV, and image size for the cameras.
- The range of the number of cars to be used in each episode.



(a) (b) (c) (d)

Figure 3.1: Some representative images from the synthetic and real datasets. (a) Town 1 - CARLA (b) Town 2 - CARLA (c-d) Tsinghua-Daimler.

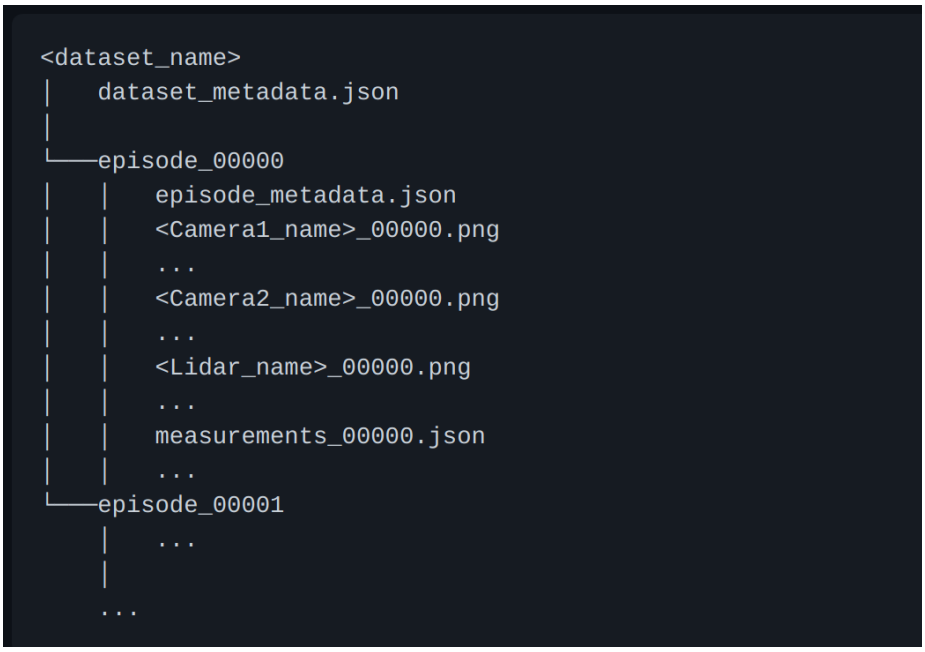


Figure 3.2: Dataset directory structure. carla simulator (2018)

- The range of the number of pedestrians to be used in each episode.
- The percentage of lateral noise.
- The percentage of longitudinal noise.
- The set of weathers to be sampled from.

Episode Metadata

Each episode is stored on a different folder. For each collected episode they generate a json file containing its general aspects that are:

- Number of Pedestrians: the absolute number of produced people on foot.
- Number of Vehicles: the absolute number of generated vehicles.
- Spawned seed for pedestrians and vehicles: the irregular seed utilized for the CARLA object bringing forth process.
- Weather: the climate of the scene.

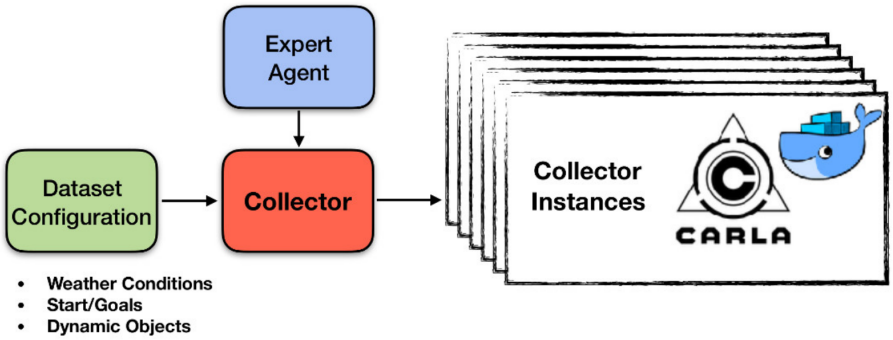


Figure 3.3: System Overview. carla simulator (2018)

Every scene keeps going from 1-5 minutes divided in reenactment steps of 100 ms. For each progression, they store information separated into two distinct classes, sensor dat put away as PNG pictures, and estimation information put away as json records.

Sensor Data

All images collected are stored as png. All lidar sensors collected are store as PLY files.

Measurements

Measurements represent all the float data collected for each simulation step. Each measurement is associated with the respective sensor data. The units of the measurements are on SI format, otherwise we specify the format. All measurements are stored in json files and contain the following information:

- Step Number: the quantity of the current reproduction step, begins at nothing and is increased by one for each recreation step.
- Game Timestamp: the time that has elapsed since the recreation has begun. Communicated on miliseconds.
- Position: the world situation of the self image vehicle. It is communicated as a three dimensional vector (x,y,z) in meters.
- Orientation: the direction of the vehicle concerning the world. Communicated as Euler points (line, pitch and yaw).

- Acceleration: the speed increase vector of the self image vehicle concerning the world.
- Forward Speed: a scalar communicating the straight forward speed of the vehicle.
- Intentions: a sign that is relative such that the powerful articles on the scene are having on the conscience vehicle activities. We utilize three distinct expectation signals: halting for walkers, halting for vehicles and halting for traffic signals. For instance, a goal of 1 for halting for person on foot implies that the inner self vehicle is completely halted for a walkers that is under 5 meters away. An expectation of the very class of 0.5 implies that the master saw a people on foot and has diminished its speed somewhat. A goal of 0 methods there are no walkers close by in the field of perspective on the master.
- High Level Commands: the general order expressing what the sense of self vehicle ought to do in the following convergence: go straight, turn left, turn right, or couldn't care less (the personality vehicle could pick any choice). These orders are encoded as a whole number. 2 is couldn't care less, 3 for turn left, 4 for turn right, 5 for go straight.
- Waypoints: a set containing the 10 future places of the vehicle.
- Steering Angle: the current point of the vehicle's directing wheel. Standardized from - 1 to 1.
- Throttle: the current tension on the choke pedal. Standardized from 0 to 1..
- Brake: the current tension on the brake pedal.Normalized from 0 to 1.
- Hand Brake: in the event that the hand brake is enacted.
- Steer Noise: the current guiding point in the vehicle considering the clamor work.
- Throttle Noise: the current strain on the choke pedal considering the commotion work.
- Brake Noise: the current tension on the brake pedal considering the clamor work.

For every last one of the non-player specialists (walkers, vehicles, traffic signal), the accompanying data is given:

- Unique ID: an interesting identifier of this specialist.
- Type: in case it is a walker, a vehicle or a traffic signal.

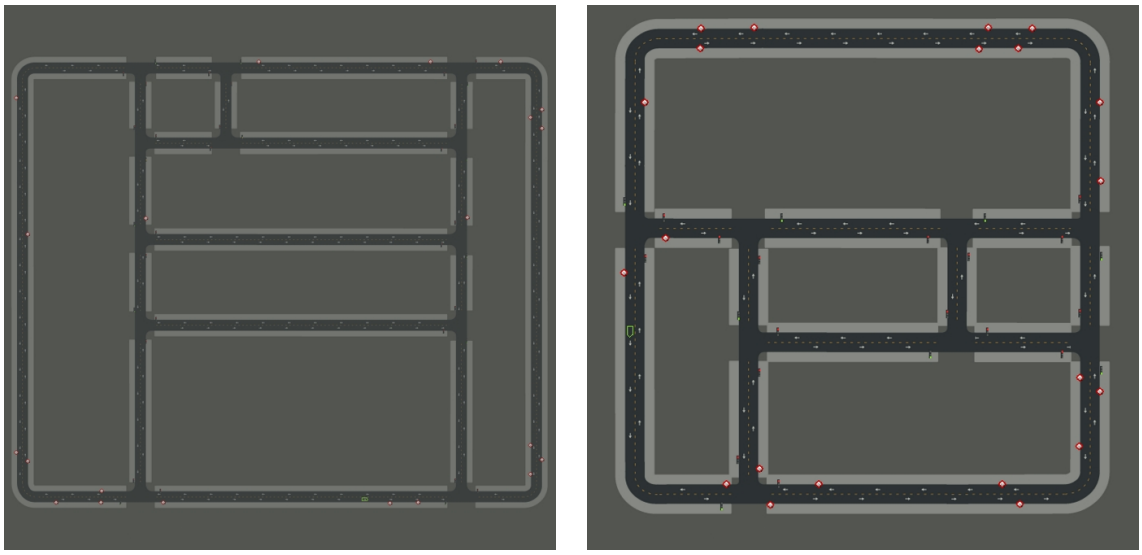


Figure 3.4: Street maps of towns from CARLA Dosovitskiy et al. (2017) Simulator (a) Town 1. (b) Town 2.

- Position: the world situation of the specialist. It is communicated as a three dimensional vector (x,y,z) in meters.
- Orientation: the direction of the specialist regarding the world. Communicated as Euler points (column, pitch and yaw).
- Forward Speed: a scalar communicating the direct forward speed of the specialist.
- State: just for traffic signals, contains the condition of the traffic signal, in case it is either red, yellow or green.

3.1.2 Expert Demonstrator

The master approaches favored data about the recreation state, including the specific guide of the climate and the specific places of the inner self vehicle, any remaining vehicles, and walkers.

Local/Global Planner

The master route has the accompanying attributes.

- The way determined by the master is determined utilizing an organizer.
- This organizer utilizes an A* calculation to decide the way to arrive at a specific objective.
- This way is then changed over into waypoints.
- A PID regulator produces the choke, brake, and directing dependent on the waypoints.
- The master drives on the focal point of the path.
- The master keeps a steady speed of 35 km/h when driving straight and lessens the speed when making goes to around 15 km/h.

Obstacle Avoidance

The agent has the following obstacle avoidance behaviours:

- Reduce speed for pedestrians that are from 5 to 15 meters away
- Completely stop for pedestrians closer than 5 meters away.
- Follows lead vehicle, imitating the lead vehicle speed.
- Stop for red traffic lights.

To compute all the obstacle avoidance behaviors the position and orientation of the ego-car on the map are used. The position of all the other objects; vehicles, pedestrians and traffic lights; are also used.

Noiser

To further develop variety, authenticity, and increment the quantity of visited state-activity sets, we add commotion to the personality vehicle controls. This diminishes the contrast between disconnected preparing and web based testing situations. The commotion mimics a steady float away from the ideal direction of the master. Nonetheless, for preparing, the float isn't utilized as sign to copy, just the responses performed by the master.

The information assortment is partitioned into a bunch of scenes performed by a specialist driver described in section 3.1.2.

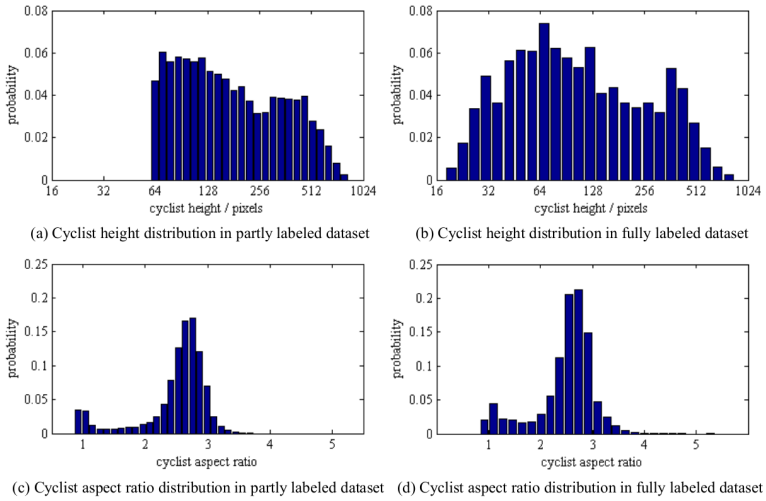


Figure 3.5: Cyclist scale distribution of the cyclist dataset. Li et al. (2016)

The client can arrange a dataset design record. This document contains a bunch of start/end positions, climates and number of dynamic items to show up on each datum assortment scene. Further, the client likewise arrange a CARLA settings object containing every one of the sensors that will be put away as a dataset.

This design document and the master demonstrator are utilized inside an authority module that can be repeated on a few docker examples. This occasions produce a dataset on with a few estimations and sensors put away on a particular organization as described in section 3.1.1.

3.2 Real Data

We have used a recent Cyclist Detection dataset Li et al. (2016) for evaluating our approach on real world data to get better understanding of the working model on unseen scenarios. We have used the test set provided by the authors containing 2,914 images by first deriving the right image using left and disparity images and then use the pair as input to compare different methods.

Figure 3.6: Overview of the cyclist detection dataset. (a) Cyclist samples; (b) Test images with annotations: green, blue and yellow bounding boxes indicate cyclists, pedestrians and other riders respectively. Li et al. (2016)

	Training			Test set	Total
	Set 1	Set 2	Non-VRU		
Total Frames	9741	5095	1000	14570	30406
Labeled Frames	9741	1019	1000	2914	14674
Total BBs	16202	3016	0	13143	32361
Cyclist BBs	16202	1301	0	4658	22161
Pedestrian BBs	0	1539	0	7380	8919
Other rider BBs	0	176	0	1105	1281

Figure 3.7: Statistics of Tsinghua-Daimler Cyclist Benchmark. Li et al. (2016)

3.2.1 Dataset Overview.

Around 6 recording hours were gathered from a vehicle-mounted sound system vision camera (picture goal of 2048×1024 pixels, gauge of 20 cm) at 25 Hz passing through normal metropolitan traffic in 5 distinct days. The recordings were gathered in the northern city of Beijing picked for their somewhat high convergence of cyclists and people on foot: Haidian District and Chaoyang District. Other than the pictures, IMU data, including speed, longitudinal speed increase and yaw rate, was caught simultaneously to offer valuable vehicle data for various application undertakings.

Around 14674 casings have been explained from more than 5 million pictures for a sum of 32361 marked weak street clients (VRUs), including cyclists, people on foot, tri-cyclists and engine cyclists and so on Figure 3.6 shows an extract from the new dataset. For every one of the physically marked pictures, we separated them into two bunches: somewhat marked and completely named.

In part of the way marked pictures, just ideal cyclists were commented on with two tight bouncing boxes, showing the full degree of the rider and the bicycle separately. Cyclists lower than 60 pixels, impeded or shortened over 10% or movement obscured were overlooked. Walkers, tri-cyclists and motorcyclists were likewise disregarded. The pictures were clarified each 10 edges in the halfway named set.

In completely named pictures, all VRUs were clarified as long as they were higher than 20 pixels, not blocked more than 80% and not shortened over 50%.

Figure 3.5 shows that a large portion of the named cyclists are concentrating somewhere in the range of 30 and 500 pixels. Plus, on the completely marked dataset, 87.91% cyclists are completely noticeable, 9.21% cyclists are to some degree impeded, furthermore, the left 2.88% are vigorously impeded. Just 1.23% cyclists are part of the way shortened by picture borders.

Chapter 4

Methodology

We train a convolutional neural network to foresee the extraneous and natural camera parameters. To accomplish this, we utilize subordinate regressors that share a typical network engineering as the component extractor. We use a Inception-v3 Szegedy et al. (2016) pretrained on ImageNet Russakovsky et al. (2015) as a feature extractor followed by the Lambda layers for loss computation with 13 regressors. Instead of training these regressors to predict the focal length, principal point, baseline, pitch, and translation, we use intermediary factors that are not apparent in the picture and are reliant upon one another. This allows us to directly relate our method with the mathematical foundations of multi-view geometry Hartley and Zisserman (2003b) resulting in better performance. The details of the model are shown in Figure 4.1 and are discussed next.

4.1 Camera Model

The camera model that we consider is the point of view projection model dependent on the pinhole camera Faugeras and FAUGERAS (1993). On the off chance that M has world directions (X, Y, Z) and ventures onto a point m that has pixel coordinates (u, v), the activity can be portrayed, in homogeneous directions, by the situation:

$$S \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{P} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.1}$$

where S is a scaling factor and the network $\mathbf{P}$ is in the design

$$P = \begin{pmatrix} p_1^T & p_{14} \\ p_2^T & p_{24} \\ p_3^T & p_{34} \end{pmatrix} \tag{4.2}$$

The 3×4 framework $\mathbf{P}$ is normally alluded to as point of view projection network and decayed into two frameworks: $\mathbf{P} = \mathbf{AD}$ where

$$\mathbf{D} = \begin{pmatrix} \mathbf{R} & t \\ 0_3^T & 1 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} \alpha_u & -\alpha_u \cot\theta & u_0 & 0 \\ 0 & \frac{\alpha_v}{\sin\theta} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The 4×4 network $\mathbf{D}$ addresses the planning from world directions to camera facilitates and accounts for six extrinsic parameters of the camera: three for the revolution $\mathbf{R}$ which is ordinarily determined by three pivot (Euler) points: R_x, R_y, R_z and three for the interpretation $t = (t_x,\ t_y,\ t_z)^T$. 0_3 addresses the invalid vector $(0,\ 0,\ 0)^T$. The 3×4 lattice $\mathbf{A}$ addresses the inherent parameters of the camera: the scale factors α_u *and* α_v, the directions u_0 *and* v_0 of the head point, and the point θ between the picture tomahawks.

In this work, we rely on 2D to 3D projection as a reference frame, leaving 13 free parameters: focal length $(f_x,\ f_y)$, principal point $(u_0,\ v_0)$, disparity (d), baseline (b), pitch (θ_p), translation $(t_x,\ t_y,\ t_z)$ and 3D coordinates (X, Y, Z).

Thus, the parameters to be recovered by the network are the focal length $(f_x,\ f_y)$, principal point $(u_0,\ v_0)$, disparity, baseline, pitch and translation $(t_x,\ t_y,\ t_z)$.

4.2 Parameterization

As revealed by previous work Workman et al. (2015, 2016); Hold-Geoffroy et al. (2018), a sufficient definition of the factors to anticipate can incredibly help assembly and last execution of the network. For the instance of camera calibration parameters, for example, the focal length or the slant points are hard to decipher from the picture content. All things considered, they can be better addressed as a substitute parameters that are straightforwardly noticeable in the picture. We use 2D to 3D projection as a proxy for our parameters.

$\mathbf{A}$ can also be written as:

$$\begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.3}$$

A 2D point in image coordinate system is projected to camera coordinate and then to world coordinate system and the process can be explained by combining (Eq. 4.1) and (Eq. 4.2) as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.4}$$

Combining (Eq. 4.1) , (Eq. 4.2) and **D** as:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \sim \left[\begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & t \\ 0_3^T & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{4.5}$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathbf{R} & t \\ 0_3^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{4.6}$$

The image to camera transformation can be performed as follows:

$$\mathbf{A}^{-1} = \begin{pmatrix} \frac{1}{f_x} & 0 & \frac{-u_0}{f_x} \\ 0 & \frac{1}{f_y} & \frac{-v_0}{f_y} \\ 0 & 0 & 1 \end{pmatrix} \tag{4.7}$$

Assuming skew = 0

$$\begin{pmatrix} y_{cam} \\ z_{cam} \\ x_{cam} \end{pmatrix} \sim \begin{pmatrix} \frac{1}{f_x} & 0 & \frac{-u_0}{f_x} \\ 0 & \frac{1}{f_y} & \frac{-v_0}{f_y} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{4.8}$$

$$y_{cam} = \frac{u}{f_x} - \frac{u_0}{f_x} = \frac{u - u_0}{f_x} \tag{4.9a}$$

$$z_{cam} = \frac{v}{f_y} - \frac{v_0}{f_y} = \frac{v - v_0}{f_y} \tag{4.9b}$$

$$x_{cam} = 1 \tag{4.9c}$$

Similarly, for camera to world transformation we have:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathbf{R} & t \\ 0_3^T & 1 \end{pmatrix} \begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \\ 1 \end{pmatrix} \tag{4.10}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \sim \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} + \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{4.11}$$

$$X = x_{cam} * \cos\theta + z_{cam} * \sin\theta + x \tag{4.12a}$$

$$Y = y_{cam} + y \tag{4.12b}$$

$$Z = -x_{cam} * \sin\theta + z_{cam} * \cos\theta + z \tag{4.12c}$$

Finally, for camera-to-camera transformation:

$$xW = \frac{f_x * b}{disparity} \tag{4.13a}$$

$$yW = y_{cam} * xW = -xW * \frac{u - u_0}{f_x} \tag{4.13b}$$

$$zW = -xW * \frac{v - v_0}{f_y} \tag{4.13c}$$

To project a point from image to camera coordinate:

$$x_{cam} = f_x * b/disparity \tag{4.14a}$$

$$y_{cam} = -(x_{cam}/f_x) * (u - u_0) \tag{4.14b}$$

$$z_{cam} = (x_{cam}/f_y) * (v_0 - v) \tag{4.14c}$$

x_{cam} works as a proxy for f_x, baseline and disparity while y_{cam} works as a proxy for f_x, u and u_0 and z_{cam} works as a proxy for f_y, v and v_0. Using x_{cam}, y_{cam} and z_{cam} from Eq. 4.14a, Eq. 4.14b and Eq. 4.14c respectively, points can be projected to world coordinate system using:

$$X = x_{cam} * \cos(\theta_p) + z_{cam} * \sin(\theta_p) + t_x \tag{4.15a}$$

$$Y = y_{cam} + t_y \tag{4.15b}$$

$$Z = -x_{cam} * \sin(\theta_p) + z_{cam} * \cos(\theta_p) + t_z \tag{4.15c}$$

X works as a proxy for pitch and t_x while Y works as a proxy for t_y and Z works as a proxy for pitch and t_z.

4.3 Camera Projection Loss

At the point when a solitary design is prepared to foresee parameters with various extents, exceptional consideration should be taken to gauge the loss terms to such an extent that the assessment of certain parameters don't overwhelm the learning system. We notice that for the instance of camera calibration, all things considered of enhancing the camera parameters independently, a solitary metric dependent on 2D to 3D projection of focuses can be utilized.

Given two images with known parameters $\omega = (f_x, f_y, u_0, v_0, b, disparity, \theta_p, t_x, t_y, t_z, X, Y, Z)$ and a prediction of such parameters given by the network $\hat{\omega} = (f_x^{'}, f_y^{'}, u_0^{'}, v_0^{'}, b^{'}, disparity^{'}, \theta_p^{'}, t_x^{'}, t_y^{'}, t_z^{'}, X^{'}, Y^{'}, Z^{'})$, we get the projected point in world coordinate system through

Eq. 4.14a - Eq. 4.15c. Loss is computed between actual ω and predicted $\hat{\omega}$ using:

$$L(\omega, \hat{\omega}) = (\frac{1}{n}) \sum_{i=1}^{n} MAE(\omega, \hat{\omega}) \tag{4.16}$$

4.4 Disentangling sources of loss errors

The proposed loss takes care of the errand adjusting issue by communicating various mistakes as far as a solitary measure. In any case, utilizing a few camera parameters to foresee the 3D focuses presents another issue during learning: the deviation of a point from its ideal projection can be credited to more than one parameter. All in all, a mistake from one parameter can backpropagate through the camera projection loss to different parameters.

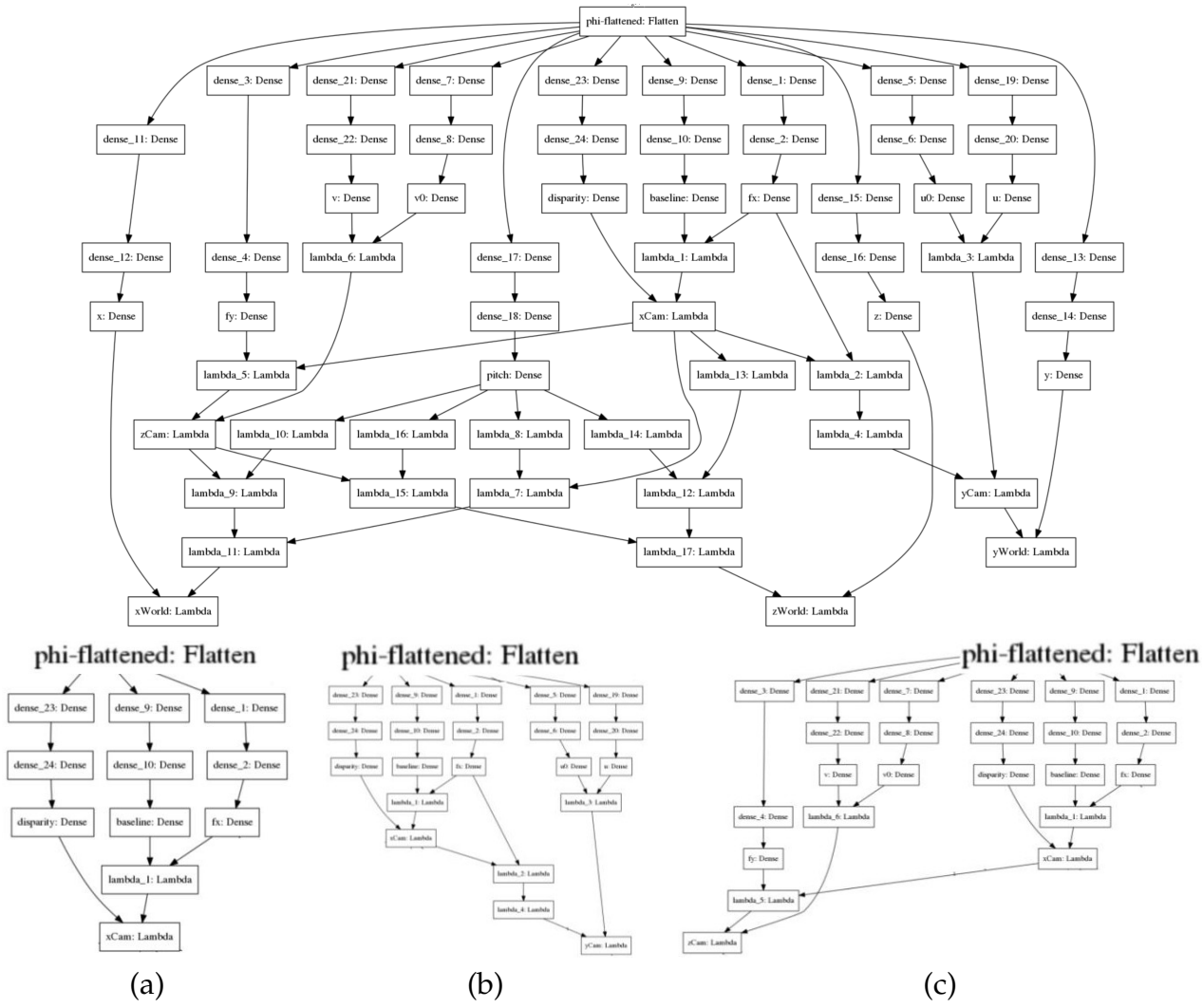
To stay away from this issue, we unravel the camera projection loss, assessing it separately for every parameter similar to Lopez et al. (2019):

$$\begin{aligned}
L_{f_x} &= L((f_x, f_y^{GT}, u_0^{GT}, v_0^{GT}, b^{GT}, d^{GT}, \theta_p^{GT}, \\
&\quad t_x^{GT}, t_y^{GT}, t_z^{GT}, X^{GT}, Y^{GT}, Z^{GT}), \omega) \\
L_{f_y} &= L((f_x^{GT}, f_y, u_0^{GT}, v_0^{GT}, b^{GT}, d^{GT}, \theta_p^{GT}, \\
&\quad t_x^{GT}, t_y^{GT}, t_z^{GT}, X^{GT}, Y^{GT}, Z^{GT}), \omega) \\
&\dots \\
L_Z &= L((f_x^{GT}, f_y^{GT}, u_0^{GT}, v_0^{GT}, b^{GT}, d^{GT}, \theta_p^{GT}, \\
&\quad t_x^{GT}, t_y^{GT}, t_z^{GT}, X^{GT}, Y^{GT}, Z), \omega)
\end{aligned}$$

$$L^* = \frac{L_{f_x} + L_{f_y} + L_{u_0} + ... + L_Z}{13} \tag{4.17}$$

The loss function is further normalized to avoid the unnecessary bias due to one or more error terms by introducing weights $alpha_i$ with each of the parameters. This bias is introduced due to heterogeneous ranges of various parameters. These weights $alpha_i$ are learned adaptively during the training process. The updated loss function is defined as:

$$L^* = \alpha_{f_x} L_{f_x} + \alpha_{f_y} L_{f_y} + \alpha_{u_0} L_{u_0} + ... + \alpha_Z L_Z \tag{4.18}$$



(a) (b) (c)

Figure 4.1: Camera Projection Loss (CPL) in the form of Lambda layers. Lambda layers have been used to implement the loss using (Eq. 4.14a - Eq. 4.15c). (a-c) are sub components CPL showing Lambda layer representation of x_{cam}, y_{cam} and z_{cam} respectively. This depicts the implementation of Eqs. 4.14a-4.14c as neural network respectively.

Chapter 5

Experiments and Results

5.1 Implementation Details

Our loss is implemented and trained using Keras Chollet et al. (2015), an open-source deep learning framework. All networks are trained on GeForce GTX 1050 Ti GPU for 200 epochs with early stopping using ADAM optimizer Kingma and Ba (2015) with Mean Absolute Error (MAE) loss function and a base learning rate η of 10^{-3} with a batch size of 16.

5.2 Comparative Analysis

5.2.1 Experimental Setup

We compared our proposed method with two state-of-the-art approaches namely Average field of view Workman et al. (2015) and Deep-Homo DeTone et al. (2016). Average field of view Workman et al. (2015) is a baseline approach, given an inquiry picture, it utilizes the normal field of perspective on the preparation set as the forecast Workman et al. (2015). Deep-Homo DeTone et al. (2016) estimates an 8-degree-of-freedom homography between two images. We have modified Deep-Homo DeTone et al. (2016) to predict the required 13 parameters for comparison purposes as by default, it only predicted 8 values corresponding to the four corners and then using 4-point parameterization and then convert it into the homography matrix. For the purpose of the ablative study, we also created three variants of our multi-task learning approach namely MTL-Baseline, MTL-CPL-U, and MTL-CPL-A. MTL-Baseline does not include any additional layers to incorporate camera model equations, instead, it is an end-to-end learning architecture based on mean absolute error (MAE). It has 13 regressors sharing a common

Table 5.1: Table showing MAE in predicted parameters on synthetic test set comprising of 23,796 images.

Method	f_x	f_y	u_0	v_0	b	d	t_x	t_y	t_z	θ_p
Average Workman et al. (2015)	72.44	72.44	40.27	40.27	12.53	21.34	12.53	12.90	12.73	89.68
Deep-Homo DeTone et al. (2016)	28.51	28.52	**1.01**	**1.02**	1.51	**0.17**	1.51	1.32	1.23	22.48
MTL-Baseline (Ours)	20.90	23.98	14.63	13.95	1.06	1.35	0.89	1.01	1.01	20.02
MTL-CPL-U (Ours)	38.36	58.19	46.02	46.11	2.79	11.87	2.80	1.11	1.44	107.89
MTL-CPL-A (Ours)	4.79	**4.22**	4.12	3.97	0.65	0.25	2.42	0.62	2.42	5.69
MTL-CPL-U-TL (Ours)	**2.50**	382.20	35.70	3.91	**0.47**	20.89	0.18	**0.39**	0.19	9.75
MTL-CPL-A-TL (Ours)	21.92	128.92	185.29	31.95	0.65	2.14	**0.10**	1.96	**0.17**	**2.53**

feature extractor directly predicting the required values. MTL-Baseline is implemented to study the effect of proposed camera projection loss. We also used two variants of camera projection loss one with uniform weighting (MTL-CPL-U) in the loss function and the other with adaptive weighting (MTL-CPL-A) to balance the heterogeneous ranges of different calibrations parameters. MTL-CPL-U-TL employs MTL-CPL-U as the base network and transfer learning while ten different models for the required 10 parameters. MTL-CPL-A-TL is similar to MTL-CPL-U-TL except that the base network is MTL-CPL-A.

5.2.2 Error Analysis on generated data

We compare the mean absolute error (MAE) of each of the parameters by all the methods with our proposed approach. It can be seen from Table 5.1 that our proposed multi-task learning approach outperforms other methods on 7 out of 10 parameters. For principal point offset (u_0, v_0) and disparity (d), MTL based methods resulted in higher MAE values than Deep-Homo DeTone et al. (2016) due to bias in loss introduced as a result of the heterogeneous range of values among parameters. For all the remaining parameters our proposed multi-task learning approach resulted in minimum values for MAE. This indicates that incorporating camera model geometry in

Table 5.2: Table showing MAE in predicted parameters on Tsinghua-Daimler test set comprising of 2,914 images. For this experiment, we just did a forward pass without any transfer learning or training.

Method	f_x	f_y	u_0	v_0	b	d	t_x	t_y	t_z	θ_p
Deep-Homo DeTone et al. (2016)	2206.5	2205.5	986.6	474.4	2.3	**6.43**	**0.6**	3.3	0.8	64.6
MTL-Baseline (Ours)	1831.5	1803.4	**759.8**	**436.3**	19.4	35.7	12.3	16.2	14.7	498.5
MTL-CPL-U (Ours)	**1355.9**	**1790.7**	3680.9	3919.1	58.0	1223.1	15.5	**2.2**	**0.2**	3861.5
MTL-CPL-A (Ours)	2208.8	2206.7	987.8	475.7	3.0	6.44	3.0	3.1	0.9	51.6
MTL-CPL-U-TL (Ours)	2166.1	4160.6	896.3	470.4	**2.2**	27.0	2.1	3.4	1.0	30.8
MTL-CPL-A-TL (Ours)	3341.9	2215.4	985.9	474.3	2.7	27.8	1.2	4.5	2.1	**29.0**

the learning framework not only resulted in a more interpretable learning framework but it also outperforms the state-of-the-art methods.

5.2.3 Error Analysis on real data

For this experiment, we didn't trained on the Tsinghua-Daimler dataset but just performed a forward pass to test the generalizability and the results further strengthen our argument. It can be seen from Table 5.2 that our proposed multi-task learning approach outperforms other methods on 8 out of 10 parameters. For disparity (d) and translation in x-axis (t_x), MTL based methods resulted in higher MAE values than Deep-Homo DeTone et al. (2016) due to bias in loss introduced as a result of the heterogeneous range of values among parameters. For all the remaining parameters our proposed multi-task learning approach resulted in minimum values for MAE which further solidifies our argument of incorporating camera model geometry in the learning framework.

Chapter 6

Conclusion and Future Work

We proposed a method to predict extrinsic (baseline, pitch, and translation) and intrinsic (focal length and principal point offset) parameters. We proposed a parameterization for camera projection that is better for learning than directly predicting the calibration parameters. We proposed a new loss function based on camera projections. Our method outperforms several baselines, including CNN-based methods on both synthetic and real data which gives strength to our argument of incorporating camera model geometry for more interpretable and generalized learning framework. In the future, we will explore the application of the proposed parameterization to other problems while also increase our dataset to have more configurations and towns while also compare with methods like Deep-PTZ Zhang et al. (2020).

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Alpher, A. (2002). Frobnication. *Journal of Foo 12*(1), 234–778.

Alpher, A. and J. P. N. Fotheringham-Smythe (2003). Frobnication revisited. *Journal of Foo 13*(1), 234–778.

Alpher, A., J. P. N. Fotheringham-Smythe, and G. Gamow (2004). Can a machine frobnicate? *Journal of Foo 14*(1), 234–778.

Authors (2006a). The frobnicatable foo filter. ECCV06 submission ID 324. Supplied as additional material `eccv06.pdf`.

Authors (2006b). Frobnication tutorial. Supplied as additional material `tr.pdf`.

Barreto, J. P. (2006). A unifying geometric representation for central projection systems. *Computer Vision and Image Understanding 103*(3), 208–217. 2, 5

Bengio, Y., P. Simard, and P. Frasconi (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks 5*(2), 157–166.

Bevilacqua, M., A. Roumy, C. Guillemot, and M. L. Alberi-Morel (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding.

Bhardwaj, R., G. K. Tummala, G. Ramalingam, R. Ramjee, and P. Sinha (2018). Autocalib: automatic traffic camera calibration at scale. *ACM Transactions on Sensor Networks (TOSN) 14*(3-4), 1–27. 6, 7

Bogdan, O., V. Eckstein, F. Rameau, and J.-C. Bazin (2018). Deepcalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production*, pp. 1–10. 2, 3, 5, 6, 7

Brown, D. C. (1971). Close-range camera calibration. *Photogrammetric Eng. 37*(8), 855–866.

Caprile, B. and V. Torre (1990). Using vanishing points for camera calibration. *International journal of computer vision 4*(2), 127–139. 1

carla simulator (2018). data-collector. x, 12, 13

Caruana, R. (1997). Multitask learning. *Machine learning 28*(1), 41–75. 9

Charco, J. L., B. X. Vintimilla, and A. D. Sappa (2018). Deep learning based camera pose estimation in multi-view environment. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 224–228. IEEE. 6, 8

Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Chen, Z., V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich (2017). Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *arXiv preprint arXiv:1711.02257*. 9

Cheng, Z., Q. Yang, and B. Sheng (2015). Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 415–423.

Chollet, F. et al. (2015). Keras. https://github.com/fchollet/keras. 27

Claus, D. and A. W. Fitzgibbon (2005). A rational function lens distortion model for general cameras. In *Proc. CVPR*, pp. 213–219.

Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223.

Dalal, N. and B. Triggs (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Volume 1, pp. 886–893. IEEE.

DeTone, D., T. Malisiewicz, and A. Rabinovich (2016). Deep image homography estimation. *arXiv preprint arXiv:1606.03798*. 2, 3, 27, 28, 29

Deutscher, J., M. Isard, and J. MacCormick (2002). Automatic camera calibration from a single manhattan image. In *European Conference on Computer Vision*, pp. 175–188. Springer. 1

Devernay, F. and O. Faugeras (2001). Straight lines have to be straight. *MVA 13*, 14–24.

Dong, C., C. C. Loy, K. He, and X. Tang (2016). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence 38*(2), 295–307.

Dosovitskiy, A., G. Ros, F. Codevilla, A. Lopez, and V. Koltun (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16. i, x, 3, 10, 15

Eigen, D., C. Puhrsch, and R. Fergus (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pp. 2366–2374.

En, S., A. Lechervy, and F. Jurie (2018). Rpnet: An end-to-end network for relative camera pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0. 6, 8

Everingham, M., S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision 111*(1), 98–136.

Fattal, R. (2007). Image upsampling via imposed edge statistics. In *ACM Transactions on Graphics (TOG)*, Volume 26, pp. 95. ACM.

Faugeras, O. and O. A. FAUGERAS (1993). *Three-dimensional computer vision: a geometric viewpoint*. MIT press. 20

Fitzgibbon, A. W. (2001a). Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Volume 1, pp. I–I. IEEE. 5

Fitzgibbon, A. W. (2001b). Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proc. CVPR*.

Gatys, L. A., A. S. Ecker, and M. Bethge (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Geiger, A., P. Lenz, C. Stiller, and R. Urtasun (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research 32*(11), 1231–1237.

Geiger, A., F. Moosmann, Ö. Car, and B. Schuster (2012). Automatic camera and range sensor calibration using a single shot. In *2012 IEEE International Conference on Robotics and Automation*, pp. 3936–3943. IEEE. 1

Glasner, D., S. Bagon, and M. Irani (2009). Super-resolution from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 349–356. IEEE.

Glorot, X. and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, Volume 9, pp. 249–256.

Griffin, G., A. Holub, and P. Perona (2007). Caltech-256 object category dataset.

Hansen, P., H. Alismail, P. Rander, and B. Browning (2012). Online continuous stereo extrinsic parameter estimation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1059–1066. IEEE. 6, 8

Hartley, R. and A. Zisserman (2003a). *Multiple view geometry in computer vision*. Cambridge university press. 1

Hartley, R. and A. Zisserman (2003b). *Multiple View Geometry in Computer Vision* (2 ed.). New York, NY, USA: Cambridge University Press. 20

Hartley, R. I. and A. Zisserman (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049.

He, K., X. Zhang, S. Ren, and J. Sun (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

He, K., X. Zhang, S. Ren, and J. Sun (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

He, K., X. Zhang, S. Ren, and J. Sun (2016b). Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645. Springer.

Hirschmuller, H. (2007). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence 30*(2), 328–341.

Hold-Geoffroy, Y., K. Sunkavalli, J. Eisenmann, M. Fisher, E. Gambaretto, S. Hadap, and J.-F. Lalonde (2018). A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2354–2363. 2, 5, 6, 7, 21

Huang, G., Z. Liu, K. Q. Weinberger, and L. van der Maaten (2016). Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*.

Huang, J.-B., A. Singh, and N. Ahuja (2015). Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5197–5206.

Ioffe, S. and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Iyer, G., J. K. Murthy, K. M. Krishna, et al. (2018). Calibnet: Self-supervised extrinsic calibration using 3d spatial transformer networks. *arXiv preprint arXiv:1803.08181*. 6, 8

Johnson, J., A. Alahi, and L. Fei-Fei (2016). Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pp. 694–711. Springer.

Johnson, J., A. Karpathy, and L. Fei-Fei (2016). Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4565–4574.

Kendall, A., Y. Gal, and R. Cipolla (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491. 9

Kendall, A., M. Grimes, and R. Cipolla (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946. 6, 7

Kingma, D. P. and J. Ba (2015). Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 27

Komorowski, J. and P. Rokita (2012). Extrinsic camera calibration method and its performance evaluation. In *International Conference on Computer Vision and Graphics*, pp. 129–138. Springer. 6, 7

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105.

LeCun, Yann A an d Bottou, L., G. B. Orr, and K.-R. Müller (2012). Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer.

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation 1*(4), 541–551.

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*(11), 2278–2324.

LeCun, Y., K. Kavukcuoglu, and C. Farabet (2010). Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 253–256. IEEE.

Lee, J.-H. (2012). Camera calibration from a single image based on coupled line cameras and rectangle constraint. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 758–762. IEEE. 5, 6

Levinson, J. and S. Thrun (2013). Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*, Volume 2, pp. 7. 1

Li, X., F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila (2016). A new benchmark for vision-based cyclist detection. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1028–1033. IEEE. x, 17, 18

Liu, B., S. Gould, and D. Koller (2010). Single image depth estimation from predicted semantic labels. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1253–1260. IEEE.

Long, J., E. Shelhamer, and T. Darrell (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440.

Lopez, M., R. Mari, P. Gargallo, Y. Kuang, J. Gonzalez-Jimenez, and G. Haro (2019). Deep single image camera calibration with radial distortion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11817–11825. 2, 3, 5, 6, 7, 25

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, Volume 2, pp. 1150–1157. Ieee.

Mao, X.-J., C. Shen, and Y.-B. Yang (2016). Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv preprint arXiv:1606.08921*.

Martin, D., C. Fowlkes, D. Tal, and J. Malik (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Volume 2, pp. 416–423. IEEE.

Melekhov, I., J. Ylioinas, J. Kannala, and E. Rahtu (2017). Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 675–687. Springer. 6, 8

Mermin, N. D. (1989, October). What's wrong with these equations? *Physics Today*. http://www.cvpr.org/doc/mermin.pdf.

Nakajima, Y. and H. Saito (2017). Robust camera pose estimation by viewpoint classification using deep learning. *Computational Visual Media 3*(2), 189–198. 6, 7

Netzer, Y., T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, Volume 2011, pp. 5.

Oliva, A. and A. Torralba (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision 42*(3), 145–175.

Pajdla, T. and T. Werner (1997). Correcting radial lens distortion without knowledge of 3-d structure.

Pandey, G., J. R. McBride, S. Savarese, and R. M. Eustice (2012). Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 1

Poursaeed, O., G. Yang, A. Prakash, Q. Fang, H. Jiang, B. Hariharan, and S. Belongie (2018). Deep fundamental matrix estimation without correspondences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0. 6, 8

Ranftl, R. and V. Koltun (2018). Deep fundamental matrix estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 284–299. 6

Rong, J., S. Huang, Z. Shang, and X. Ying (2016). Radial lens distortion correction using convolutional neural networks trained with synthesized images. In *Asian Conference on Computer Vision*, pp. 35–49. Springer. 2, 5

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review 65*(6), 386.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision 115*(3), 211–252. 20

Schöller, C., M. Schnettler, A. Krämmer, G. Hinz, M. Bakovic, M. Güzet, and A. Knoll (2019). Targetless rotational auto-calibration of radar and camera for intelligent transportation systems. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3934–3941. IEEE. 6, 7

Sermanet, P., D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Shalnov, E. and A. Konushin (2017). Convolutional neural network for camera pose estimation from object detections. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 42*. 6, 8

Shi, W., J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883.

Simonyan, K. and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Srivastava, R. K., K. Greff, and J. Schmidhuber (2015). Highway networks. *arXiv preprint arXiv:1505.00387*.

Sun, J., Z. Xu, and H.-Y. Shum (2008). Image super-resolution using gradient profile prior. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8. IEEE.

Sun, L. and J. Hays (2012). Super-resolution from internet-scale scene matching. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pp. 1–12. IEEE.

Swaminathan, R. and S. Nayar (2000). Nonmetric calibration of wide-angle lenses and polycameras. *IEEE T-PAMI 22*(10), 1172–1178.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9.

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826. x, 2, 20

Tai, Y., J. Yang, and X. Liu. Image super-resolution via deep recursive residual network.

Taylor, Z. and J. Nieto (2015). Motion-based calibration of multimodal sensor arrays. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4843–4850. IEEE. 1

Telea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of graphics tools 9*(1), 23–34.

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation 3*(4), 323–344.

Tsai, Y. R. (1986). An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. CVPR*.

Ulyanov, D., A. Vedaldi, and V. Lempitsky (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Unnikrishnan, R. and M. Hebert (2005). Fast extrinsic calibration of a laser rangefinder to a camera. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09*. 1

Weng, J., P. Cohen, and M. Herniou (1992). Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (10), 965–980.

Workman, S., C. Greenwell, M. Zhai, R. Baltenberger, and N. Jacobs (2015). Deepfocal: A method for direct focal length estimation. In *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 1369–1373. IEEE. 2, 3, 5, 6, 7, 21, 27, 28

Workman, S., M. Zhai, and N. Jacobs (2016). Horizon lines in the wild. *arXiv preprint arXiv:1604.02129*. 2, 5, 21

Xiang, Y., T. Schmidt, V. Narayanan, and D. Fox (2017). Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*. 6, 7

Xiao, J., K. A. Ehinger, A. Oliva, and A. Torralba (2012). Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2695–2702. IEEE.

Xie, J., L. Xu, and E. Chen (2012). Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 341–349.

Yang, Z., K. Zhang, Y. Liang, and J. Wang (2017). Single image super-resolution with a parameter economic residual-like convolutional neural network. In *International Conference on Multimedia Modeling*, pp. 353–364. Springer.

Yin, X., X. Wang, J. Yu, M. Zhang, P. Fua, and D. Tao (2018). Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 469–484.

Zeiler, M. D. and R. Fergus (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer.

Zeyde, R., M. Elad, and M. Protter (2010). On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pp. 711–730. Springer.

Zhai, M., S. Workman, and N. Jacobs (2016). Detecting vanishing points using global image context in a non-manhattan world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5657–5665. 2, 5

Zhang, C., F. Rameau, J. Kim, D. M. Argaw, J.-C. Bazin, and I. S. Kweon (2020). Deepptz: Deep self-calibration for ptz cameras. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 1041–1049. 2, 3, 30

Zhang, Z. (1996). On the epipolar geometry between two images with lens distortion. In *Proc. ICPR*, pp. 407–411.

Zhang, Z., Y. Matsushita, and Y. Ma (2011). Camera calibration with lens distortion from low-rank textures. In *CVPR 2011*, pp. 2321–2328. IEEE.

Zioulis, N., A. Karakottas, D. Zarpalas, and P. Daras (2018). Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 448–465.