

# **Camera Calibration through Camera Projection Loss**

by

**Talha Hanif Butt (118-1864)**

**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. We propose a new strategy to foresee camera parameters utilizing a picture pair. As far as our knowledge is concerned, this is the first method to jointly estimate these parameters via a multi-task learning methodology. Different from current methods, we don't design an end-to-end solution to directly estimate the desired parameters, we propose another portrayal that joins camera model conditions as a neural network to perform various tasks. We gauge the ideal parameters through novel *camera projection loss* (CPL) that utilizes the camera model neural network to remake the 3D point cloud and uses the recreation loss to appraise 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 methodology performs better compared 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

|   |           |
|---|-----------|
| <b>List of Tables</b>                           | <b>ix</b> |
| <b>List of Figures</b>                          | <b>x</b>  |
| <b>1 Introduction</b>                           | <b>1</b>  |
| 1.1 Geometry based Camera Calibration . . . . . | 1         |
| 1.2 CNN based Camera Calibration . . . . .      | 2         |
| 1.3 Loss balancing . . . . .                    | 3         |
| 1.4 Contributions . . . . .                     | 3         |
| <b>2 Literature Review</b>                      | <b>5</b>  |
| 2.1 Single Input . . . . .                      | 5         |
| 2.2 Multiple Inputs . . . . .                   | 7         |
| 2.3 Multi-task learning . . . . .               | 9         |
| <b>3 Camera Calibration Parameters</b>          | <b>10</b> |
| 3.1 Intrinsic Parameters . . . . .              | 12        |
| 3.1.1 Focal length . . . . .                    | 13        |
| 3.1.2 Principal point . . . . .                 | 16        |
| 3.1.3 Geometric distortion . . . . .            | 18        |
| 3.2 Extrinsic Parameters . . . . .              | 20        |
| 3.2.1 Position and Orientation . . . . .        | 20        |
| 3.2.2 Rotation . . . . .                        | 21        |
| 3.2.3 Translation . . . . .                     | 22        |

|                   |  |           |
|-------------------|--|-----------|
| <b>4</b>          | <b>Methodology</b>                             | <b>23</b> |
| 4.1               | Camera Model . . . . .                         | 23        |
| 4.2               | Parameterization . . . . .                     | 24        |
| 4.3               | Camera Projection Loss . . . . .               | 27        |
| 4.4               | Disentangling sources of loss errors . . . . . | 28        |
| <b>5</b>          | <b>Datasets</b>                                | <b>30</b> |
| 5.1               | Synthetic Data . . . . .                       | 30        |
| 5.1.1             | Dataset Format . . . . .                       | 30        |
| 5.1.2             | Expert Demonstrator . . . . .                  | 35        |
| 5.2               | Real Data . . . . .                            | 37        |
| 5.2.1             | Dataset Overview . . . . .                     | 38        |
| <b>6</b>          | <b>Experiments and Results</b>                 | <b>40</b> |
| 6.1               | Implementation Details . . . . .               | 40        |
| 6.2               | Comparative Analysis . . . . .                 | 40        |
| 6.2.1             | Experimental Setup . . . . .                   | 40        |
| 6.2.2             | Error Analysis on generated data . . . . .     | 41        |
| 6.2.3             | Error Analysis on real data . . . . .          | 42        |
| <b>7</b>          | <b>Conclusion and Future Work</b>              | <b>43</b> |
| <b>References</b> |  | <b>44</b> |

# List of Tables

|     |   |    |
|-----|---|----|
| 2.1 | Overview of some recent configurations for different aspects of Camera Calibration . . . . .  | 6  |
| 6.1 | Table showing MAE in predicted parameters on synthetic test set comprising of 23,796 images. . . . .  | 41 |
| 6.2 | 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. . . . . | 42 |

# List of Figures

|   |    |
|---|----|
| 1.1 Our technique gauges outward (pattern, dissimilarity, pitch and interpretation) and characteristic (central length and chief point offset) boundaries using pre-trained Inception-v3 Szegedy et al. (2016) and the proposed Camera Projection Loss. . . . . | 2  |
| 3.1 An overview of Camera Calibration Parameters based on transformations. MathWorks (2021) . . . . .   | 10 |
| 3.2 Components of Camera Calibration. MPHY0026 (2019) . . . . .   | 11 |
| 3.3 Camera Intrinsic Matrix. MathWorks (2021) . . . . .   | 12 |
| 3.4 Intrinsic Parameters - an overview. MathWorks (2021) . . . . .  | 12 |
| 3.5 Pixel Skew. MathWorks (2021) . . . . .  | 13 |
| 3.6 The focal point F and focal length f of a positive (convex) lens, a negative (concave) lens, a concave mirror, and a convex mirror. Wikipedia (2021c)   | 14 |
| 3.7 Thick lens diagram. Wikipedia (2021c) . . . . .   | 16 |
| 3.8 The geometry of a pinhole camera. Wikipedia (2021d) . . . . .   | 16 |
| 3.9 The geometry of a pinhole camera as seen from the X2 axis. Wikipedia (2021d) . . . . .  | 17 |
| 3.10 Geometric Distortion - an overview. ScienceDirect (2021) . . . . .   | 18 |
| 3.11 Extrinsic Camera Parameters. MathWorks (2021) . . . . .  | 20 |
| 3.12 Camera Extrinsic Matrix. Wikipedia (2021a) . . . . .   | 20 |
| 3.13 Classic Euler angles geometrical definition. The xyz (fixed) system is shown in blue, the XYZ (rotated) system is shown in red. The line of nodes (N) is shown in green. Wikipedia (2021b) . . . . .   | 21 |
| 3.14 Rotation Matrix. Medium (2019) . . . . .   | 22 |

|     |  |    |
|-----|--|----|
| 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. . . . . | 29 |
| 5.1 | Some representative images from the synthetic and real datasets. (a) Town 1 - CARLA (b) Town 2 - CARLA (c-d) Tsinghua-Daimler . . . . .  | 31 |
| 5.2 | Dataset directory structure. <a href="#">carla simulator (2018)</a> . . . . .  | 32 |
| 5.3 | System Overview. <a href="#">carla simulator (2018)</a> . . . . .  | 33 |
| 5.4 | Street maps of towns from CARLA <a href="#">Dosovitskiy et al. (2017)</a> Simulator (a) Town 1. (b) Town 2. . . . .  | 35 |
| 5.5 | Cyclist scale distribution of the cyclist dataset. <a href="#">Li et al. (2016)</a> . . . . .  | 37 |
| 5.6 | Outline of the cyclist discovery dataset. (a) Cyclist tests; (b) Test pictures with explanations: green, blue and yellow jumping boxes show cyclists, people on foot and different riders separately. <a href="#">Li et al. (2016)</a> . . . . .   | 38 |
| 5.7 | Statistics of Tsinghua-Daimler Cyclist Benchmark. <a href="#">Li et al. (2016)</a> . . . . .   | 38 |

# Chapter 1

## Introduction

Camera alignment manages tracking down the five inherent ( central length, picture sensor configuration, and chief point) and six outward (pivot, interpretation) boundaries of the particular 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, straight lines and evaporating focuses which can help to get the factors [Caprile and Torre \(1990\)](#); [Deutscher et al. \(2002\)](#). However, MVG-based methods lack generalizability because of pictures taken in unstructured conditions.

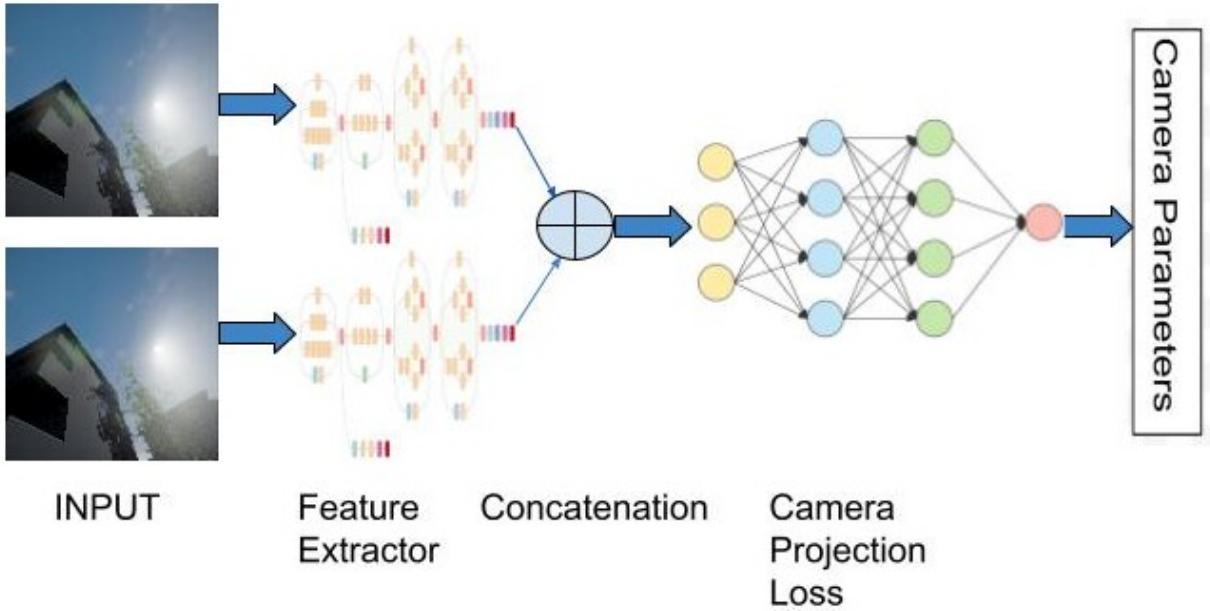


Figure 1.1: Our technique gauges outward (pattern, dissimilarity, pitch and interpretation) and characteristic (central length and chief point offset) boundaries using pre-trained Inception-v3 [Szegedy et al. \(2016\)](#) and the proposed Camera Projection Loss.

## 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 arrangement and propose a faster methodology 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.
- In contrast to existing strategies, rather than planning a start to finish arrangement to directly estimate the desired parameters DeTone et al. (2016), we proposed another portrayal that addresses camera model conditions as a neural organization in a perform multiple tasks learning (MTL) structure.

- We proposed a novel *camera projection loss* (CPL) that consolidates logical conditions in learning system. We utilize the proposed camera model neural organization to remake 3D point cloud and utilize the recreation misfortune to assess the camera boundaries.

The remainder of the report is structured as follows: Chapter 2 focuses on the literature, Chapter 3 sheds light on the Image formation process, Chapter 4 describes Stereo Vision, Chapter 5 talks about Camera Calibration Parameters, Chapter 6 explains Feature-based alignment, Chapter 7 discusses our Methodology, Chapter 8 describes our Datasets, Chapter 9 focuses on the Experiments and Results while Chapter 10 contains Conclusion and Future Work.

## Chapter 2

# Literature Review

Ongoing works have utilized the achievement of CNNs and proposed applying 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 for performing relapse of the perspective field on a camera based on pinhole, later zeroing in on identifying the skyline on pictures Workman et al. (2016), which is an intermediary for the roll and slant points of the camera given that the central length is available. Rong et al. (2016) utilize a grouping way to deal with single-factor spiral bending model from Fitzgibbon (2001a). Hold-Geoffroy et al. (2018) first joined extraneous and characteristic arrangement in a solitary network, anticipating the roll, slant and central length of a camera based on pinhole through a grouping methodology. They depended on upstanding 360 degree symbolism to artificially produce pictures of self-assertive central length, size and rotation, a methodology 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 factors, which are afterwards purified 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 of an ob-

| 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

scure 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 solitary picture utilizing a profound 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 methodology for evaluating the by and large stance 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 per factor. This situation is typically called learning multiple tasks [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 weighting total of the terms in the loss, varying on the way wherein the loads are set at the time of training: [Kendall et al. \(2018\)](#) utilize softmax and Gaussian probabilities (for relapse and grouping, separately) for assigning weight to the diverse loss terms as indicated by an errand subordinate vulnerability. As opposed to these vulnerability based strategies, [Chen et al. \(2017\)](#) decide weights values by changing the angle sizes related to every loss factor.

## Chapter 3

# Camera Calibration Parameters

The calibration algorithm calculates the camera matrix using the extrinsic and intrinsic parameters. The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system. The intrinsic parameters represent a projective transformation from the 3-D camera's coordinates into the 2-D image coordinates.

Some important definitions are as follows:

### Frame of reference

Estimations are made concerning a specific coordinate framework called the frame of reference.

### World Frame

A fixed coordinate framework for addressing objects (points, lines, surfaces, and so forth) in the world.

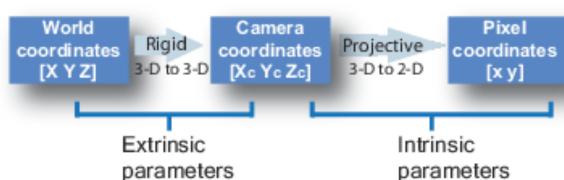


Figure 3.1: An overview of Camera Calibration Parameters based on transformations.  
[MathWorks \(2021\)](#)

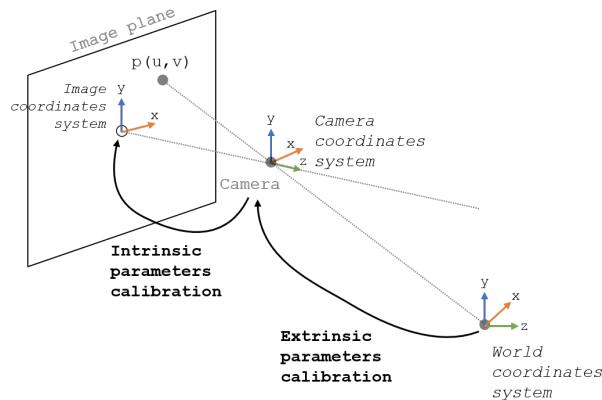


Figure 3.2: Components of Camera Calibration. [MPHY0026 \(2019\)](#)

### **Camera Frame**

Coordinate framework that utilizes the camera center as its beginning (and the optic pivot as the Z-axis).

### **Image or retinal plane**

Plane on which the picture is framed, note that the picture plane is estimated in camera coordinates (mm).

### **Image Frame**

Coordinate framework that calculates pixel areas in the image plane.

### **Intrinsic Parameters**

Camera boundaries that are inside and fixed to a specific camera/digitization arrangement.

### **Extrinsic Parameters**

Camera boundaries that are outer to the camera and may change regarding the world frame.

$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

Figure 3.3: Camera Intrinsic Matrix. [MathWorks \(2021\)](#)

|                       |  |
|-----------------------|--|
| $[c_x \ c_y]$         | — Optical center (the principal point), in pixels.                             |
| $(f_x, \ f_y)$        | — Focal length in pixels.  |
| $f_x = F/p_x$         |  |
| $f_y = F/p_y$         |  |
| $F$                   | — Focal length in world units, typically expressed in millimeters.             |
| $(p_x, \ p_y)$        | — Size of the pixel in world units.  |
| $s$                   | — Skew coefficient, which is non-zero if the image axes are not perpendicular. |
| $s = f_x \tan \alpha$ |  |

Figure 3.4: Intrinsic Parameters - an overview. [MathWorks \(2021\)](#)

### 3.1 Intrinsic Parameters

The intrinsic parameters include the focal length, the optical center, also known as the principal point, and the skew coefficient. The camera intrinsic matrix, K, is defined as shown in Figure 3.3.

The pixel skew is defined as shown in Figure 3.5.

To map the camera directions to pixel facilitates (to plan virtual articles in reality), we want to track down the inborn camera boundaries. Figure 3.2 shows a portrayal of the components engaged with camera adjustment.

Intrinsic parameters include the following:

- Scale factor (often equal to 1)

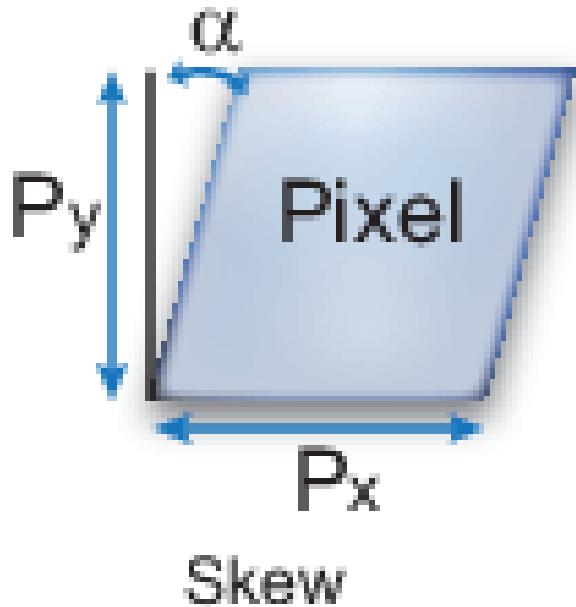


Figure 3.5: Pixel Skew. [MathWorks \(2021\)](#)

- Focal length (distance between the centre of projection an the image plane)
- Principal point (assume pixel 0,0 to be at the centre of the image)
- Skew (when the principal point is not exactly at the centre of the image plane)
- Geometric distortion (due to the lens)

### 3.1.1 Focal length

The central length of an optical structure is an extent of how decidedly the system meets or meanders light; it is the converse of the system's optical power. A positive central length shows that a structure meets light, while a negative central length exhibits that the system meanders light. A system with a more restricted central length turns the pillars even more unequivocally, getting them to a middle a more restricted distance or isolating them even more quickly. For the exceptional occurrence of a thin point of convergence in air, a positive central length is the distance over which at first collimated (equivalent) radiates are brought to a focus, of course a negative focal length exhibits how far before the point of convergence a point source ought to be arranged to shape a collimated column. For more wide optical structures, the central length has no normal importance; it is only the regressive of the system's optical power.

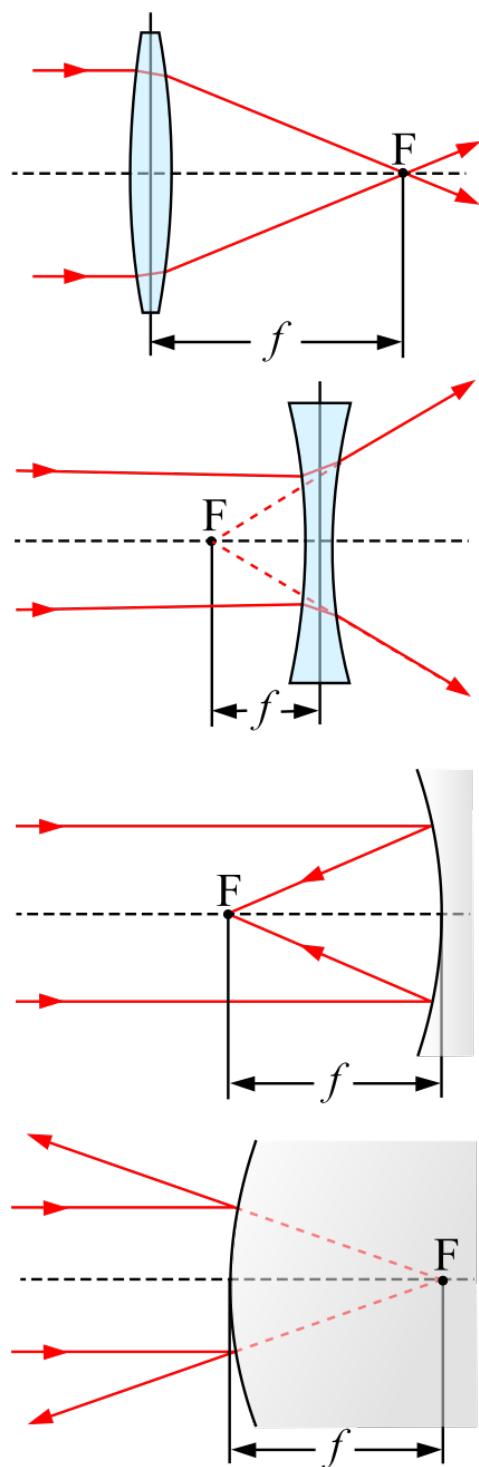


Figure 3.6: The focal point  $F$  and focal length  $f$  of a positive (convex) lens, a negative (concave) lens, a concave mirror, and a convex mirror. [Wikipedia \(2021c\)](#)

In most photography and all telescopic vision, where the subject is essentially vastly far away, longer central length (lower optical power) prompts higher enhancement and a more modest perspective; then again, more restricted central length or higher optical power is connected with lower intensification and a more broad perspective. On the other hand, in applications, for instance, microscopy in which intensification is cultivated by bringing the thing close to the point of convergence, a more restricted central length (higher optical power) prompts higher enhancement considering the way that the subject can be conveyed closer to the point of convergence of projection.

### **Thin lens approximation**

For a modest point of convergence in air, the central length is the partition from the point of convergence of the point of convergence to the essential foci (or focal characteristics) of the point of convergence. For a joining point of convergence (for example a raised point of convergence), the central length is positive and is the distance at which a light discharge light will be locked in to a single spot. For a meandering point of convergence (for example an internal point of convergence), the central length is negative and is the distance blunt from which a collimated bar appears, apparently, to isolate right after going from the perspective.

At the moment that a point of convergence is used to shape an image of some thing, the partition from the thing to the point of convergence  $u$ , the division from the point of convergence to the image  $v$ , and the central length  $f$  are associated by the accompanying:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$$

The focal length of a thin raised point of convergence can be conveniently assessed by using it to shape an image of a far away light source on a screen. The point of convergence is moved until a sharp picture is outlined on the screen. For this situation  $\frac{1}{u}$  is irrelevant, and the central length is then given by:

$$f \approx v$$

Concluding the central length of a bended point of convergence is genuinely more irksome. The central length of such a point of convergence is seen as where the spreading light outflows would meet before the point of convergence in case the point of convergence were not there. No image is outlined during such a test, and the focal length ought to be directed by passing light (for example, the light of a laser bar) accord-

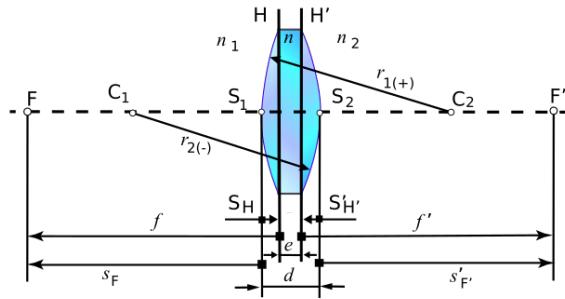


Figure 3.7: Thick lens diagram. [Wikipedia \(2021c\)](#)

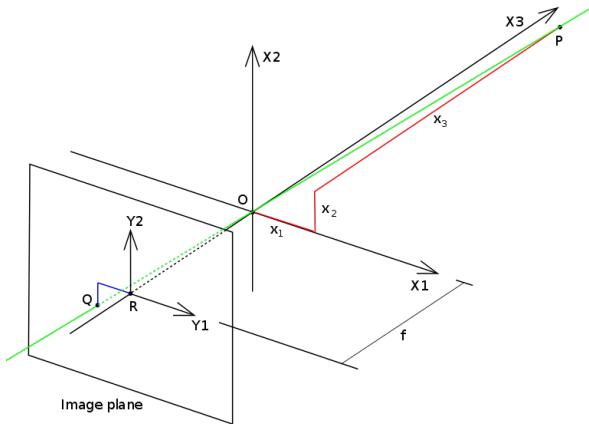


Figure 3.8: The geometry of a pinhole camera. [Wikipedia \(2021d\)](#)

ing to the viewpoint, examining how much that light becomes dissipated/bowed, and following the light outflow in converse to the point of convergence's of combination.

### 3.1.2 Principal point

The geometry related to the mapping of a pinhole camera is illustrated in the figure 3.8. The figure contains the following basic objects:

- A 3D symmetrical organize framework with its starting point at O. This is additionally where the camera opening is found. The three tomahawks of the organize framework are alluded to as X<sub>1</sub>, X<sub>2</sub>, X<sub>3</sub>. Hub X<sub>3</sub> is pointing in the survey course of the camera and is alluded to as the optical pivot, head hub, or chief beam. The plane which is crossed by tomahawks X<sub>1</sub> and X<sub>2</sub> is the front side of the camera, or chief plane.

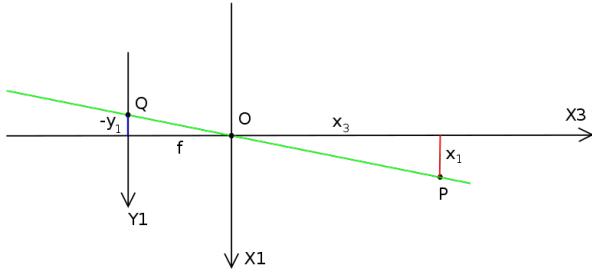


Figure 3.9: The geometry of a pinhole camera as seen from the X2 axis. [Wikipedia \(2021d\)](#)

- A picture plane, where the 3D world is projected through the gap of the camera. The picture plane is corresponding to tomahawks  $X_1$  and  $X_2$  and is situated at distance  $f$  from the beginning  $O$  the negative way of the  $X_3$  hub, where  $f$  is the central length of the pinhole camera. A reasonable execution of a pinhole camera suggests that the picture plane is found with the end goal that it converges the  $X_3$  pivot at organize -  $f$  where  $f > 0$ .
- A point R at the convergence of the optical pivot and the picture plane. This point is alluded to as the chief point or picture focus.
- A point P some place on the planet at arrange  $(x_1, x_2, x_3)$  comparative with the tomahawks  $X_1$ ,  $X_2$ , and  $X_3$ .
- The projection line of point P into the camera. This is the green line which goes through point P and the point O.
- The projection of point P onto the picture plane, indicated Q. This point is given by the convergence of the projection line (green) and the picture plane. In any useful circumstance we can accept that  $x_3 > 0$  which implies that the convergence point is clear cut.
- There is additionally a 2D organize framework in the picture plane, with beginning at R and with tomahawks  $Y_1$  and  $Y_2$  which are corresponding to  $X_1$  and  $X_2$ , individually. The directions of guide Q relative toward this organize framework is  $(y_1, y_2)$ .

The pinhole opening of the camera, through which all projection lines should pass, is thought to be vastly little, a point. In the writing this point in 3D space is alluded to as the optical (or focal point or camera) focus.

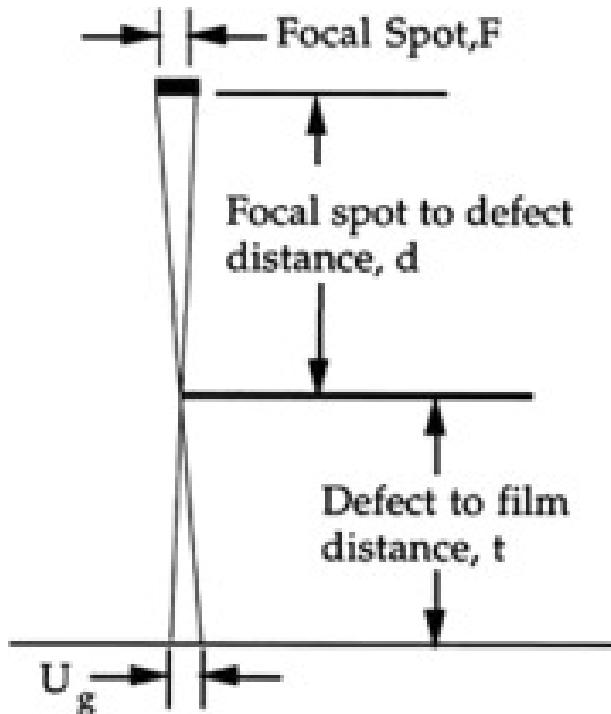


Figure 3.10: Geometric Distortion - an overview. [ScienceDirect \(2021\)](#)

### 3.1.3 Geometric distortion

Mathematical twisting can be isolated into three parts: optical bending, locator mutilation, and arrangement subordinate contortion. The optical part of the contortion can be portrayed as any twisting because of the HST+FOC optical setup, i.e., all that which starts upstream from the intensifier tube. It is controlled by beam following models of the optical framework. The organization subordinate twists incorporate those angles which contrast from one configuration to another, for example, the output contortion examined previously. The majority of these twists result from occasions which happen downstream from the intensifier tube. At last, the finder twists are those which happen inside the cylinder itself.

The precision of the mathematical mutilation rectification of FOC information is presently restricted by the coarseness and size of the reseau lattice (60 pixel dividing). Ostensibly, the precision of the mathematical revision is awesome - around 0.2%. That is, the genuine relative partition between focuses after mathematical revision is great to around 0.5 pixels over a detachment of around 250 pixels. Be that as it may, there are a few conditions on this which should be fulfilled.

The mathematical twisting is just unequivocally characterized at the places of the reseau marks present in the organization. The genuine mathematical amendment is the

consequence of a 2-D polynomial fit between the noticed reseau marks and a reference framework, henceforth the pixel-by-pixel bending is the aftereffect of insertion between these known focuses with the understanding that the variety of the contortion has a moderately long scalelength and is both smooth and well-behaved.

Framing frameworks, for example, cameras utilized for flying photography, give a quick "preview" perspective on the Earth from straightforwardly overhead. The essential mathematical mutilation in vertical elevated photos is because of help dislodging. Protests straightforwardly beneath the focal point of the camera focal point (for example at the nadir) will have just their tops noticeable, while any remaining items will seem to incline away from the focal point of the photograph with the end goal that their tops and sides are apparent. In case the articles are tall or are far away from the focal point of the photograph, the twisting and positional blunder will be larger.

The calculation of along-track scanner symbolism is like that of an airborne photo for each output line as every identifier basically takes a "preview" of each ground goal cell. Mathematical varieties between lines are brought about by irregular varieties in stage height and mentality along the course of flight.

Images from across-track filtering frameworks display two primary sorts of mathematical bending. They also display alleviation uprooting (A), like elevated photos, yet in only one bearing corresponding to the course of output. There is no uprooting straightforwardly underneath the sensor, at nadir. As the sensor examines across the area, the top and side of items are imaged and seem to incline away from the nadir point in each output line. Once more, the removal increments, moving towards the edges of the area. Another mutilation (B) happens because of the turn of the examining optics. As the sensor examines across each line, the separation from the sensor to the ground increments further away from the focal point of the area. Albeit the checking mirror turns at a steady speed, the IFOV of the sensor moves quicker (comparative with the ground) and sweeps a bigger region as it draws nearer to the edges. This impact brings about the pressure of picture highlights at focuses away from the nadir and is called extraneous scale distortion.

All pictures are defenseless to mathematical twists brought about by varieties in stage solidness remembering changes for their speed, elevation, and disposition (rakish direction concerning the ground) during information securing. These impacts are most articulated when utilizing airplane stages and are mitigated generally with the utilization of satellite stages, as their circles are somewhat steady, especially comparable to their separation from the Earth. In any case, the toward the east pivot of the Earth,during a satellite circle makes the scope of checking frameworks cover a region somewhat toward the west of each past examine. The resultant symbolism is hence slanted across the picture. This is known as slant bending and is normal in symbolism got from satellite multispectral scanners.

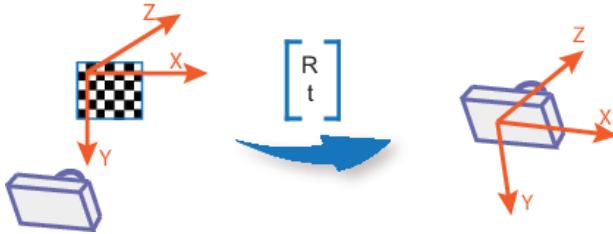


Figure 3.11: Extrinsic Camera Parameters. [MathWorks \(2021\)](#)

$$\begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix}_{4 \times 4}$$

Figure 3.12: Camera Extrinsic Matrix. [Wikipedia \(2021a\)](#)

## 3.2 Extrinsic Parameters

The extraneous boundaries comprise of a turn, R, and an interpretation, t. The beginning of the camera's organize framework is at its optical focus and its x-and y-hub characterize the picture plane.

In Figure 3.12, R and T are the extraneous boundaries which mean the facilitate framework changes from 3D world directions to 3D camera organizes. Proportionally, the outward boundaries characterize the situation of the camera place and the camera's going in world directions. T is the situation of the beginning of the world organize framework communicated in directions of the camera-focused arrange framework.

The extraneous grid appears as an inflexible change network: a 3x3 turn framework in the left-block, and 3x1 interpretation section vector morally justified as displayed

underneath:  $[R \mid t] = \left[ \begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array} \right]$

### 3.2.1 Position and Orientation

The situation of an article can be addressed as an interpretation of the item from the origin.

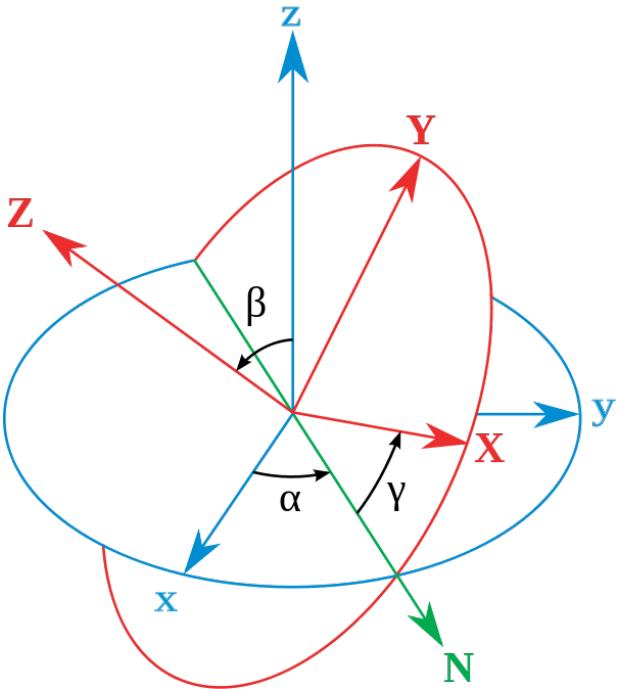


Figure 3.13: Classic Euler angles geometrical definition. The  $xyz$  (fixed) system is shown in blue, the  $XYZ$  (rotated) system is shown in red. The line of nodes ( $N$ ) is shown in green. [Wikipedia \(2021b\)](#)

The direction of an article can be addressed as a revolution of an article from its unique unrotated direction.

### 3.2.2 Rotation

Outward revolutions are natural turns that happen about the tomahawks of the decent organize framework  $xyz$ . The  $XYZ$  framework pivots, while  $xyz$  is fixed. Beginning with  $XYZ$  covering  $xyz$ , a creation of three extraneous pivots can be utilized to arrive at any objective direction for  $XYZ$ . The Euler or Tait–Bryan points ( $\alpha, \beta, \gamma$ ) are the amplitudes of these natural turns. For example, the objective direction can be reached as follows (note the switched request of Euler point application):

- The  $XYZ$  framework turns about the  $z$  pivot by  $\gamma$ . The  $X$  pivot is presently at point  $\gamma$  as for the  $x$  hub.
- The  $XYZ$  framework pivots once more, however this time about the  $x$  hub by  $\beta$ . The  $Z$  pivot is presently at point  $\beta$  regarding the  $z$  hub.

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure 3.14: Rotation Matrix. [Medium \(2019\)](#)

- The XYZ framework pivots a third time, about the z hub once more, by point  $\alpha$ .

In total, the three basic pivots happen about z, x and z. For sure, this grouping is frequently meant z-x-z (or 3-1-3). Sets of pivot tomahawks related with both appropriate Euler points and Tait–Bryan points are regularly named utilizing this notation.

Rotation network can likewise be addressed as displayed in Figure 3.14.

Three kinds of revolutions are needed as extraneous boundaries which are pitch, roll and yaw.

- Pivot around the front-to-back hub is called roll.
- Pivot around the side-to-side hub is called pitch.
- Turn around the upward hub is called yaw.

### 3.2.3 Translation

An interpretation vector is a sort of change that moves a figure in the facilitate plane starting with one area then onto the next. At the end of the day, an interpretation vector can be considered as a slide with no turning. The slide will not change the shape or size of the figure, and with no revolution, the direction will not change either.

Three esteems relating the interpretation in x,y and z aspects are required.

## 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:

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

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

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

The  $3 \times 4$  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  $4 \times 4$  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  $3 \times 4$  lattice  $\mathbf{A}$  addresses the inherent parameters of the camera: the scale factors  $\alpha_u$  and  $\alpha_v$ , the directions  $u_0$  and  $v_0$  of the head point, and the point  $\theta$  between the picture toma-hawks.

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  $(\theta_p)$ , 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:

$$\begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.3)$$

A 2D point in picture facilitate framework is projected to camera organize and afterward to world arrange framework and the process can be explained by combining (Eq. 4.1) and (Eq. 4.2) as:

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

Combining (Eq. 4.1) , (Eq. 4.2) and  $\mathbf{D}$  as:

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

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

The image to camera transformation can be performed as follows:

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

Assuming skew = 0

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

$$y_{cam} = \frac{u}{f_x} - \frac{u_0}{f_x} = \frac{u - u_0}{f_x} \quad (4.9a)$$

$$z_{cam} = \frac{v}{f_y} - \frac{v_0}{f_y} = \frac{v - v_0}{f_y} \quad (4.9b)$$

$$x_{cam} = 1 \quad (4.9c)$$

Similarly, for camera to world transformation we have:

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

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

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

$$Y = y_{cam} + y \quad (4.12b)$$

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

Finally, for camera-to-camera transformation:

$$xW = \frac{f_x * b}{disparity} \quad (4.13a)$$

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

$$zW = -xW * \frac{v - v_0}{f_y} \quad (4.13c)$$

To project a point from image to camera coordinate:

$$x_{cam} = f_x * b / disparity \quad (4.14a)$$

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

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

$x_{cam}$  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:

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

$$Y = y_{cam} + t_y \quad (4.15b)$$

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

$X$  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  $\omega$  and predicted  $\hat{\omega}$  using:

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

## 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):

$$\begin{aligned} L_{f_x} &= L((f_x^{GT}, f_y^{GT}, u_0^{GT}, v_0^{GT}, b^{GT}, d^{GT}, \theta_p^{GT}, \\ &\quad t_x^{GT}, t_y^{GT}, t_z^{GT}, X^{GT}, Y^{GT}, Z^{GT}), \omega) \\ L_{f_y} &= L((f_x^{GT}, f_y^{GT}, u_0^{GT}, v_0^{GT}, b^{GT}, d^{GT}, \theta_p^{GT}, \\ &\quad t_x^{GT}, t_y^{GT}, t_z^{GT}, X^{GT}, Y^{GT}, Z^{GT}), \omega) \\ &\dots \\ L_Z &= L((f_x^{GT}, f_y^{GT}, u_0^{GT}, v_0^{GT}, b^{GT}, d^{GT}, \theta_p^{GT}, \\ &\quad t_x^{GT}, t_y^{GT}, t_z^{GT}, X^{GT}, Y^{GT}, Z), \omega) \\ L^* &= \frac{L_{f_x} + L_{f_y} + L_{u_0} + \dots + L_Z}{13} \end{aligned} \quad (4.17)$$

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:

$$L^* = \alpha_{f_x} L_{f_x} + \alpha_{f_y} L_{f_y} + \alpha_{u_0} L_{u_0} + \dots + \alpha_Z L_Z \quad (4.18)$$

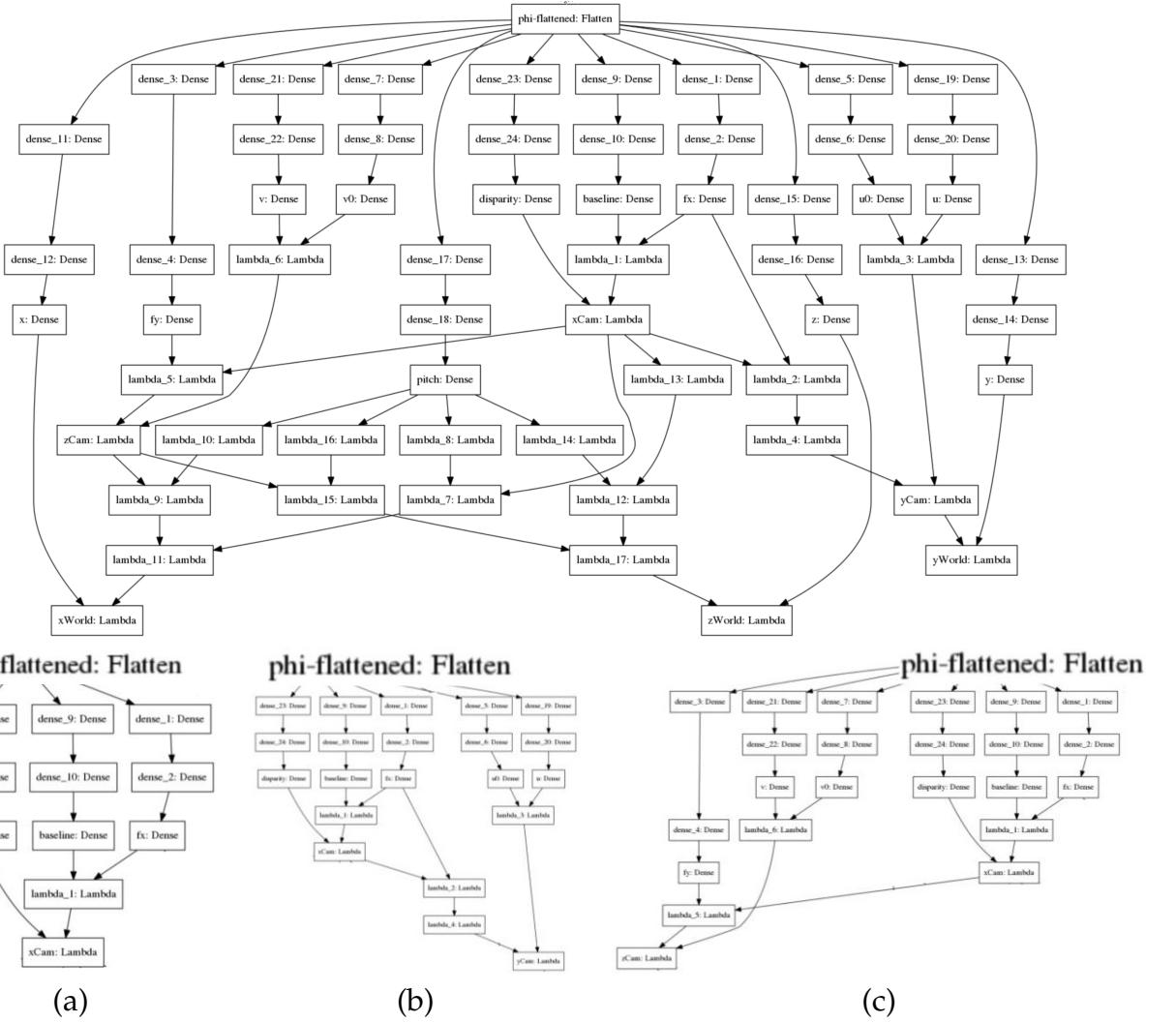


Figure 4.1: Camera Projection Loss (CPL) in the form of Lambda layers. Lambda layers have been used to implement the loss using (Eq. 4.14a - Eq. 4.15c). (a-c) are sub components CPL showing Lambda layer representation of  $x_{cam}$ ,  $y_{cam}$  and  $z_{cam}$  respectively. This depicts the implementation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

## Chapter 5

# Datasets

### 5.1 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.

#### 5.1.1 Dataset Format

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

The dataset directory should look like the structure in Figure 5.2.

#### 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.

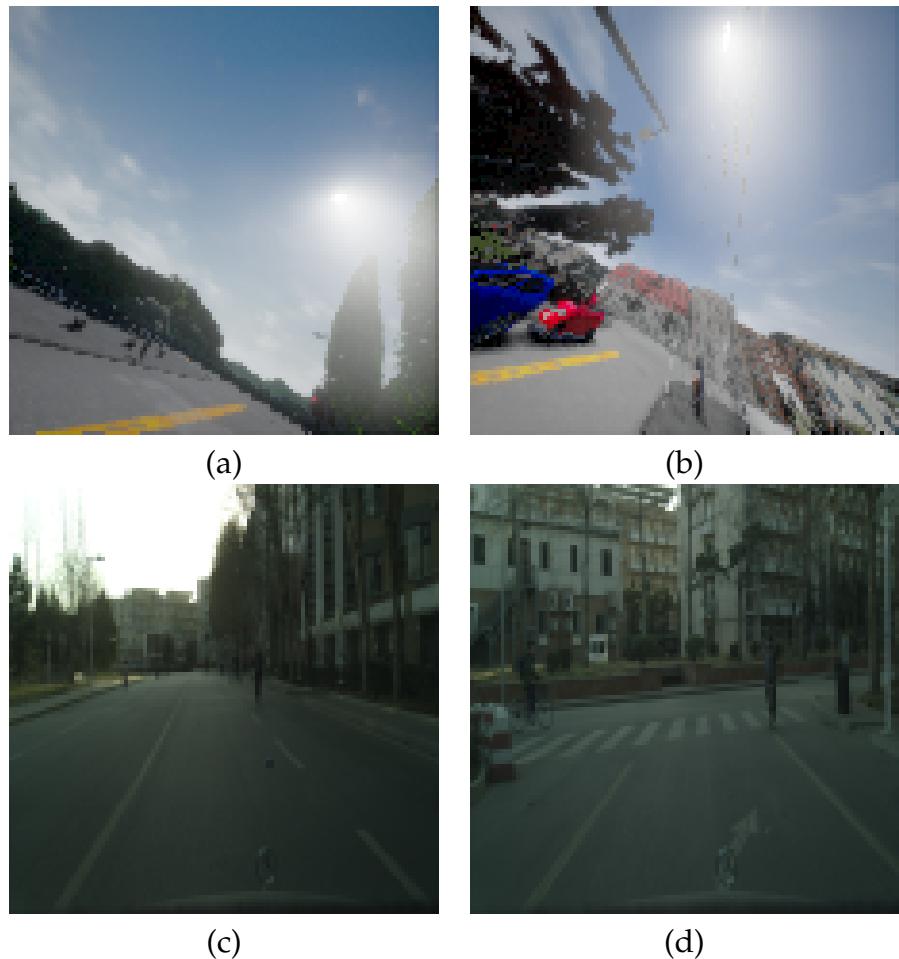


Figure 5.1: Some representative images from the synthetic and real datasets. (a) Town 1 - CARLA (b) Town 2 - CARLA (c-d) Tsinghua-Daimler.

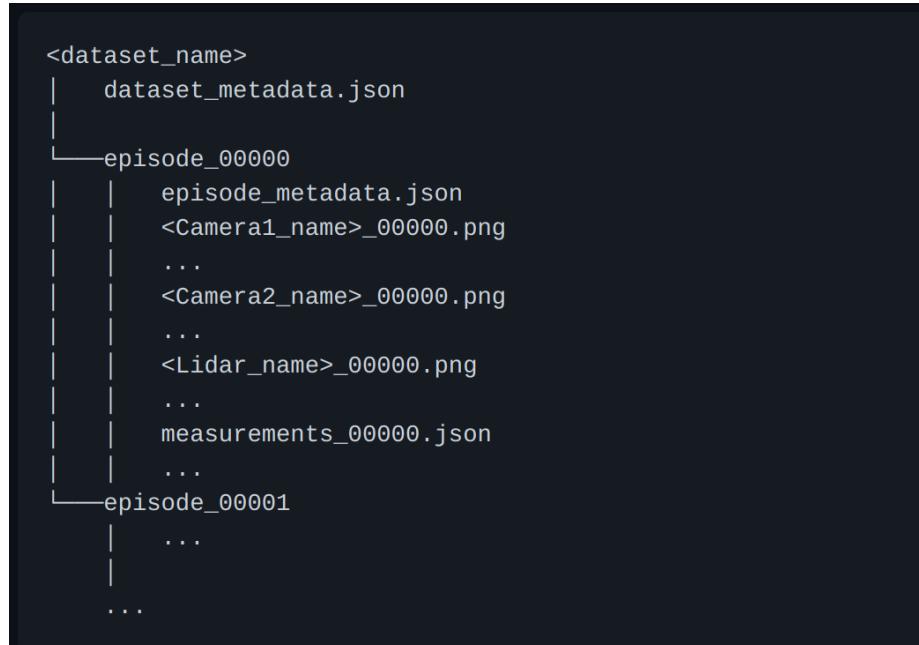


Figure 5.2: Dataset directory structure. [carla simulator \(2018\)](#)

- 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.
- Produced seed for walkers and vehicles: the sporadic seed used for the CARLA object delivering process.
- Weather: the climate of the scene.

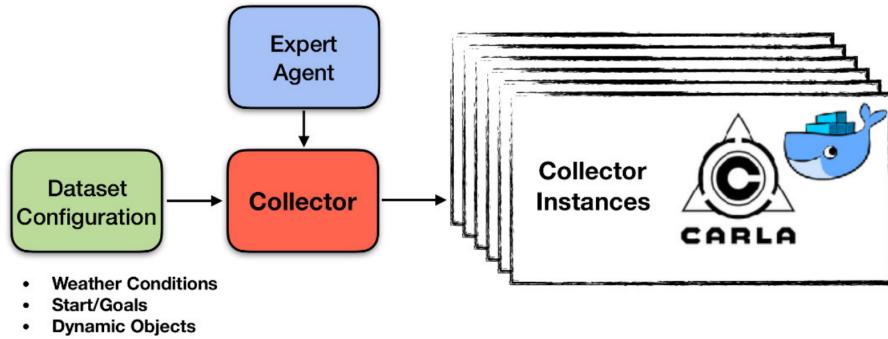


Figure 5.3: System Overview. [carla simulator \(2018\)](#)

Each scene continues to go from 1-5 minutes isolated in reenactment steps of 100 ms. For every movement, they store data isolated into two particular classes, sensor information set aside as PNG pictures, and assessment data set aside 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 amount of the current multiplication step, starts at nothing and is expanded by one for every diversion step.
- Game Timestamp: the time that has passed since the entertainment has started. Imparted on milliseconds.
- Position: the world circumstance of the mental self portrait vehicle. It is conveyed as a three dimensional vector (x,y,z) in meters.
- Direction: the bearing of the vehicle concerning the world. Conveyed as Euler focuses (line, pitch and yaw).

- Speed increase: the speed increment vector of the mental self view vehicle concerning the world.
- Forward Speed: a scalar conveying the straight forward speed of the vehicle.
- Goals: a sign that is relative with the end goal that the amazing articles on the scene are having on the inner voice vehicle exercises. We use three unmistakable assumption signals: stopping for walkers, ending for vehicles and ending for traffic lights. For example, an objective of 1 for ending for individual by walking infers that the internal identity vehicle is totally stopped for a walkers that is under 5 meters away. An assumption for the actual class of 0.5 suggests that the expert saw a group by walking and has decreased its speed to some degree. An objective of 0 strategies there are no walkers nearby in the field of viewpoint on the expert.
- Undeniable Level Commands: the overall request communicating what the ability to be self aware vehicle should do in the accompanying combination: go straight, turn left, turn right, or can't muster the energy to care (the character vehicle could pick any decision). These orders are encoded overall number. 2 is can't muster enough willpower to care, 3 for turn left, 4 for turn right, 5 for go straight.
- Waypoints: a set containing the 10 future spots of the vehicle.
- Guiding Angle: the current reason behind the vehicle's coordinating wheel. Normalized 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: if the hand brake is ordered.
- Steer Noise: the current directing point in the vehicle considering the racket work.
- Choke Noise: the current strain on the gag pedal considering the disturbance work.
- Brake Noise: the current pressure on the brake pedal considering the uproar work.

For each and every one of the non-player subject matter experts (walkers, vehicles, traffic light), the going with information is given:

- Novel ID: a fascinating identifier of this subject matter expert.

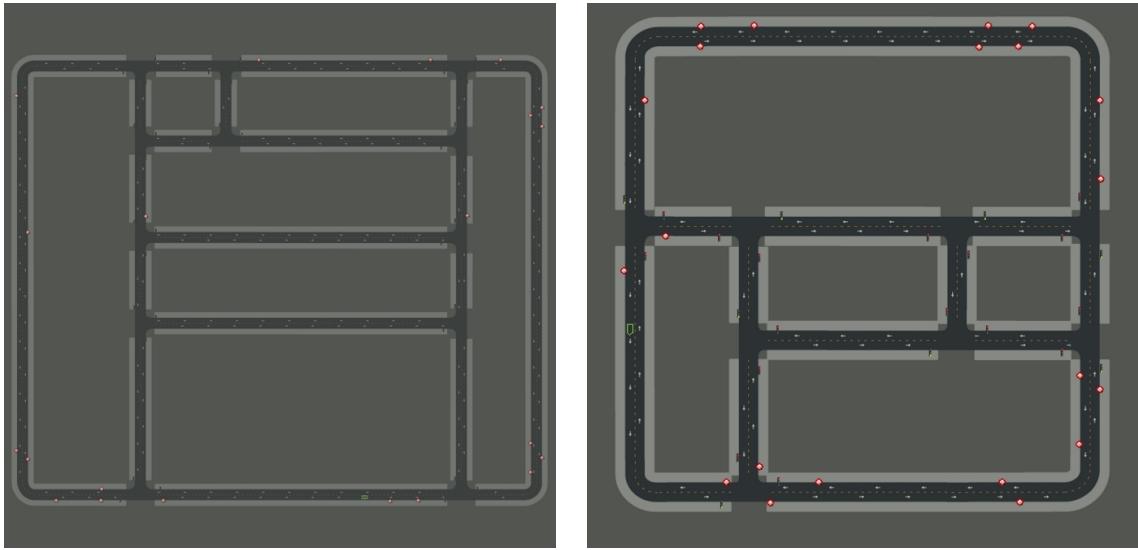


Figure 5.4: Street maps of towns from CARLA Dosovitskiy et al. (2017) Simulator (a) Town 1. (b) Town 2.

- Type: in the event that it is a walker, a vehicle or a traffic light.
- Position: the world circumstance of the trained professional. It is conveyed as a three dimensional vector ( $x,y,z$ ) in meters.
- Direction: the course of the expert with respect to the world. Conveyed as Euler focuses (section, pitch and yaw).
- Forward Speed: a scalar conveying the direct forward speed of the subject matter expert.
- State: only for traffic lights, contains the state of the traffic light, on the off chance that it is either red, yellow or green.

### 5.1.2 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 expert keeps a consistent speed of 35 km/h when driving straight and reduces 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.

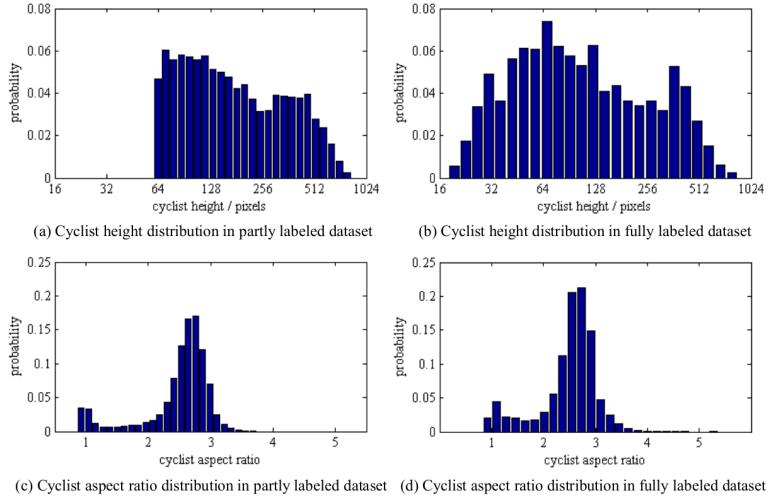


Figure 5.5: Cyclist scale distribution of the cyclist dataset. [Li et al. \(2016\)](#)

The information assortment is partitioned into a bunch of scenes performed by a specialist driver described in section 5.1.2.

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 5.1.1.

## 5.2 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 5.6: Outline of the cyclist discovery dataset. (a) Cyclist tests; (b) Test pictures with explanations: green, blue and yellow jumping boxes show cyclists, people on foot and different riders separately. [Li et al. \(2016\)](#)

|                 | Set 1 | Training Set 2 | Non-VRU | Test set | Total |
|-----------------|-------|----------------|---------|----------|-------|
| 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 5.7: Statistics of Tsinghua-Daimler Cyclist Benchmark. [Li et al. \(2016\)](#)

### 5.2.1 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 fairly high union of cyclists and individuals by walking: Haidian District and Chaoyang District. Other than the photos, IMU information, including speed, longitudinal speed increment and yaw rate, was gotten all the while to offer significant vehicle information for different application endeavors.

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 5.6 shows a concentrate from the new dataset. For all of the actually stamped pictures, we isolated them into two packs: to some degree checked and totally named.

In part of the way marked pictures, just ideal cyclists were remarked on with two tight ricocheting boxes, showing the full level of the rider and the bike independently. Cyclists lower than 60 pixels, blocked or abbreviated more than 10% or development clouded were neglected. Walkers, tri-cyclists and motorcyclists were moreover dismissed. The photos were explained every 10 edges in the midway named set.

In totally named pictures, all VRUs were explained as long as they were higher than 20 pixels, not hindered over 80% and not abbreviated more than 50%.

Figure 5.5 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 6

# Experiments and Results

### 6.1 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  $\eta$  of  $10^{-3}$  with a batch size of 16.

### 6.2 Comparative Analysis

#### 6.2.1 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 6.1: Table showing MAE in predicted parameters on synthetic test set comprising of 23,796 images.

| Method                         | $f_x$       | $f_y$       | $u_0$       | $v_0$       | $b$         | $d$         | $t_x$       | $t_y$       | $t_z$       | $\theta_p$  |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 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       | <b>1.01</b> | <b>1.02</b> | 1.51        | <b>0.17</b> | 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        | <b>4.22</b> | 4.12        | 3.97        | 0.65        | 0.25        | 2.42        | 0.62        | 2.42        | 5.69        |
| MTL-CPL-U-TL (Ours)            | <b>2.50</b> | 382.20      | 35.70       | 3.91        | <b>0.47</b> | 20.89       | 0.18        | <b>0.39</b> | 0.19        | 9.75        |
| MTL-CPL-A-TL (Ours)            | 21.92       | 128.92      | 185.29      | 31.95       | 0.65        | 2.14        | <b>0.10</b> | 1.96        | <b>0.17</b> | <b>2.53</b> |

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.

### 6.2.2 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 6.1 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 6.2: 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                                | $f_x$         | $f_y$         | $u_0$        | $v_0$        | $b$        | $d$         | $t_x$      | $t_y$      | $t_z$      | $\theta_p$  |
|---------------------------------------|---------------|---------------|--------------|--------------|------------|-------------|------------|------------|------------|-------------|
| Deep-Homo <b>DeTone et al. (2016)</b> | 2206.5        | 2205.5        | 986.6        | 474.4        | 2.3        | <b>6.43</b> | <b>0.6</b> | 3.3        | 0.8        | 64.6        |
| MTL-Baseline (Ours)                   | 1831.5        | 1803.4        | <b>759.8</b> | <b>436.3</b> | 19.4       | 35.7        | 12.3       | 16.2       | 14.7       | 498.5       |
| MTL-CPL-U (Ours)                      | <b>1355.9</b> | <b>1790.7</b> | 3680.9       | 3919.1       | 58.0       | 1223.1      | 15.5       | <b>2.2</b> | <b>0.2</b> | 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        | <b>2.2</b> | 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        | <b>29.0</b> |

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

### 6.2.3 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 6.2 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 7

# 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 frobncatable 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. [xi](#), [32](#), [33](#)
- 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>. [40](#)
- 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](#), [40](#), [41](#), [42](#)
- 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](#), [xi](#), [3](#), [30](#), [35](#)
- e-consystems (2018). What is a stereo vision camera?
- 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. [23](#)
- 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. [23](#)
- 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.
- Hirschmüller, 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](#), [24](#)
- 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*. [40](#)
- 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. [xi](#), [37](#), [38](#)
- 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](#), [28](#)
- 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.
- MathWorks (2021). What is camera calibration? [x](#), [10](#), [12](#), [13](#), [20](#)
- Medium (2019). Calibration. [x](#), [22](#)

- 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>.
- MPHY0026 (2019). Intrinsic camera parameters calibration. [x](#), [11](#)
- 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. [23](#)
- Schöller, C., M. Schnettler, A. Krämer, 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](#)

ScienceDirect (2021). Geometric distortion. [x](#), [18](#)

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 poly-cameras. *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](#), [23](#)

Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.

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.
- Wikipedia (2021a). Camera resectioning. x, 20
- Wikipedia (2021b). Euler angles. x, 21
- Wikipedia (2021c). Focal length. x, 14, 16
- Wikipedia (2021d). Pinhole camera model. x, 16, 17
- 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, 24, 40, 41
- Workman, S., M. Zhai, and N. Jacobs (2016). Horizon lines in the wild. *arXiv preprint arXiv:1604.02129*. 2, 5, 24
- 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](#), [43](#)
- 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.