University of Trento

Department of Industrial Engineering

Master's degree in
Mechatronics Engineering

STATE AND POSE ESTIMATION OF A RACING VEHICLE FROM ONBOARD CAMERA FOOTAGE USING COMPUTER VISION

Supervisor UniTrento
Francesco Biral

Co-supervisor UniTrento
Edoardo Pagot

Student:
Francesco Mazzoni 224840

Academic year:
2022/2023

Dedicated to my family, that supported me during these five years
Dedicated to myself for my dedication to studies

Acknowledgments

I want to thank Professor Francesco Biral and Edoardo Pagot who supported me during my thesis work allowing me to reach this important step in my life.

I would also thank all the students with which I shared these five years: Giovanni Poletti, Andrea Zatelli, Andrea Cardone, Federico Marchi, Gianmarco Lavacca, Sara Degodenz, Marco Canova, Alessandro Castellaz, Mattia Sittoni, Svevo Fuser, Emanuele Mazza, Alessandro Berthod, Sara Malpaga, Achraf Bouzid, Ayman Barakat. The projects developed together and the time spent at University with you provided the opportunity to learn something from each of you. Any technical and moral support helped me a lot during the bachelor's and master's courses at the university.

Moreover, I want to thank my family who supported me psychologically, morally and economically. Thank you Maria Luisa for the talks about my studies and yours. I think we have learned something one from the other. And we will learn even more in the future.

I want also to mention the Inter fan community "Ermes Messaggero Nerazzurro Supporters". I spent some time with you during the last years on Telegram and your mentality, which is somehow scientific, helped me grow from a methodological point of view.

Last but not least, I want to thank the members of the Facebook community "Ingegneria del suicidio" who constantly help students and engineers with their issues.

Abstract

Vehicle state estimation in 3D can be useful for various situations and applications, from the autonomous driving side to the racing performance optimization. Position and orientation are usually provided with sensors that can measure the distance from objects reconstructing the scene around the vehicle. It is also true that sensors mounted on cars impact on their cost production. One way to provide the state of a vehicle is the analysis of camera images mounted on top of it. To get useful information from pixels and transfer them into 3D pieces of information one fundamental process is the camera calibration which allows one to get the parameters of the camera as well as its orientation and position with respect to the world reference frame. It is the set of parameters embedded in the so-called extrinsic camera matrix that allows one to estimate where the image sensor is located and, in turn, where the car on which the camera is mounted is positioned, given a certain reference frame. Various techniques in computer vision allow finding the proper set of parameters, from the classical least-square-error minimization procedure between known world-pixel correspondences to vanishing point techniques as well as deep learning approaches.

In racing scenarios, one kind of application of trajectory estimation is the reconstruction of the vehicle path along the track during a qualifying lap. Achieving a well-defined trajectory allows people working in the racing field to do proper data post-processing. For example, the modelling of a trajectory can be later modified by optimizing some objectives which can be tyre consumption reduction, fuel consumption reduction or simply getting a faster time lap. Formula 1 is a well-known racing world cup competition. The technologies involved are very complex and fascinate lots of engineers. One of the elements that are still unknown is the camera characteristics mounted on the t-shape structure over the roll bar. Focal length, lens distortion and other parameters characterizing this kind of sensor are still unknown and a procedure for a proper characterization has to be proposed. A better knowledge of an onboard camera in F1 scenarios may allow future work based on image measurements. For example, localization can be performed by image information fused with GPS data.

The presented work aims at studying a racing scenario from onboard camera images for a characterization of the image sensor together with the car position estimation. Camera calibration is a well-studied technique, but it performs well in controlled environments. Typical procedures require checkerboards with images taken by an off-the-shelf camera. In this case, no camera is handled and only video images are analyzed. This can provide some issues which are reflected in the camera state estimation and the car pose estimation too. Choosing the right pattern as well as the proper number of points and images impacts the final result describing the car's state. Indeed, it will be shown how the change of one pattern changes the effect on camera calibration and, consequently, on the estimation of car position.

Modelling different types of cameras has an impact as well. The pinhole camera

model is the most used one in computer vision applications but it does not deal with non-idealities such as lens distortion. The additional modelling of these effects gave various results of the car position, suggesting not much robustness provided by the chosen sets of points. Ultimately, using the classical camera calibration procedure by choosing the key points for parameter estimation gave some issues. The best result achieved in terms of camera localization was provided by the calibration given the chicane pattern and three radial distortion coefficients, with a reprojection error of 5 px. On the other hand, using the starting grid as a pattern gave low uncertainties on optimization variables, but poor camera localization was achieved. The reasons for such different results, possible errors made and what can be improved for future improvements will be pointed out in the presented work.

To solve problems regarding the robustness of the method and the uncertainties associated with the parameters involved, an alternative approach to the classical matching procedure is proposed. Instead of searching standalone points forming a pattern, a dense group of pixels describing lines is preferred. A higher number of points than the one used in the presented techniques can probably increase the precision of parameter estimation, providing more robust results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Car trajectory estimation refers to the process of determining the path followed by a car over a specific period of time. It is an essential aspect of various applications, ranging from civil scenarios like autonomous driving and collision avoidance systems to racing scenarios like performance analysis and optimization. The goal is to accurately estimate the car's position, velocity, and orientation to enable effective decision-making and control.

In both civil and racing scenarios, car trajectory estimation typically relies on several sensors and algorithms to process and interpret the available information. Let's explore the key components and techniques involved in car trajectory estimation in each scenario.

1. Civil Scenario: In civil scenarios, car trajectory estimation is crucial for autonomous driving systems, advanced driver assistance systems (ADAS), and traffic management. Various sensors are used, including radar, lidar, cameras, and GPS, to gather information about the car's surroundings, its own motion, and the environment. Sensor fusion algorithms combine data from multiple sensors to obtain a more accurate and comprehensive understanding of the car's position, velocity, and orientation. Techniques like Kalman filtering and particle filtering are commonly employed. Other algorithms involved are localization algorithms that utilize GPS data, map-matching techniques, and sensor fusion to determine the car's precise position within a map. Car trajectory estimation also involves tracking other objects on the road, such as pedestrians, cyclists, and other vehicles. Tracking algorithms help estimate the future positions of these objects for collision avoidance and planning purposes. Moreover, various mathematical models, such as kinematic and dynamic models, are used to represent the car's motion and predict its trajectory based on sensor measurements. These models take into account factors like acceleration, deceleration, and turning dynamics.

2. Racing Scenario: In racing scenarios, car trajectory estimation plays a vital role in performance analysis, driver assistance systems, and optimizing lap times.

Typically, racing cars are equipped with Inertial Measurement Units (IMUs), which are sensors consisting of accelerometers and gyroscopes employed to measure the car's linear and angular accelerations, as well as rotational rates. Racing cars also have data logging systems that record various parameters like speed, throttle position, brake pressure, steering angle, and tire slip. This data is used to reconstruct the car's trajectory during performance analysis. A possible implementation of an estimated trajectory estimated as input data is lap time optimization: in racing, trajectory estimation helps in optimizing lap times by analyzing the car's motion and identifying areas for improvement. By comparing optimal racing lines and braking points, drivers can enhance their performance.

Overall, car trajectory estimation is crucial in both civil and racing scenarios. While civil applications primarily focus on safety, efficiency, and autonomy, racing applications emphasize performance analysis and lap time optimization. One possible way to achieve position estimation is to combine GPS positioning with visual information provided by onboard cameras. The usage of cameras and, more in general, optical sensors for measurements is becoming fundamental in nowadays technology applications. In the automotive field, for example, recognizing a pedestrian or a traffic sign can enhance vehicle safety. Cameras are already used in parking assistance and can be further developed for automatic parking systems as well. Their implementation on cars does not only have a meaning in terms of measurement accuracy but it is also important from an economic point of view. Equipping cars with sensors has an impact on the final cost and using cameras, that have a low price, can help reduce it. Also in racing applications, giving visual information can provide additional data to the ones already returned by GPS and transponders. GPS resolution suffers indeed from physical obstacles and a camera can correct the lack of information by estimating the car position from pixel data.

Usually, car position estimation is achieved by using already-known optical sensors with known parameters modelling points projection on the sensor. In some other cases, the camera is unknown and a calibration procedure has to be performed. This is usually done by managing the camera and performing intrinsic parameter estimation in controlled environments with known patterns shown to it. There are also cases in which no sensor is provided at all. So camera calibration has to be performed with the available measurements of objects shown to the user through the sensor itself. For example, F1 cars are equipped with several onboard cameras for TV broadcasting purposes. None of them is completely described in the technical regulations, so no focal length, optical axes, and lens distortion coefficients are known in advance. It would be interesting to characterize this kind of sensor so that measurements of the real world can be performed by F1 teams for enhancing performances.

In this work, some methods for camera calibration and consequent vehicle trajectory estimation will be pointed out. The main objective is to show how it is possible to obtain the state of an object given the onboard sequence of images from a camera

which is initially undetermined and that be thus calibrated. The two topics of position estimation and camera calibration are fused together, meaning that from camera calibration the state of the vehicle is attempted to be estimated. As said, a possible scenario where trajectory and movements can be reconstructed from image inputs is a Formula 1 racing lap. Specifically, video frames were taken from a recorded onboard video from the T-shape camera mounted on the car and the car position on the track tried to be estimated from pixel information. It is the set of correspondences between world points and image points that allows to perform parameter estimation. In particular, world points can be projected on the image plane by applying first a roto-translation to express them in the camera reference frame and then projecting the obtained coordinates onto the image plane. This last projection is performed through the so-called intrinsic parameters of the camera, whereas the first transformation is performed by the extrinsic camera parameters. The problem of finding the intrinsic and the extrinsic parameters is called camera calibration and can be usually performed through the sum of square error minimization, where the error is computed between the projected points on the image plane and the corresponding pixel. This methodology is a classic one provided by Zhang [8] and it is used in various applications. Typically, a chessboard for camera calibration is used, but in the case study presented any camera cannot be handled. As the only available source is a video we can only rely on a given pixel position without any control over the environment surrounding the camera. What is pointed out, is how road pattern elements can be exploited for identifying the camera matrix elements and, as a consequence, the car position and orientation estimated from the camera extrinsic matrix. Alternatively, one can map a certain configuration that repeats on the road several times ignoring the real car movement and considering only the relative distance between points on that pattern. In this case, the true extrinsic parameters are recomputed later. Choosing the correct pattern, as well as the choice of the camera non-idealities model impacts the final position estimation. Various results can be addressed to the fact that the methodology involved considers a low number of points and images. So a higher quantity of points has to be selected. Moreover, classical calibration approaches do not constrain the parameter values involved, whereas the pitch camera angle as well as its height must be fixed.

The presence of few available points led to the formulation of an alternative methodology for camera calibration and vehicle position estimation. In particular, the new focus is to solve a constrained minimization problem taking into account the parametric curves that fit the pixels and the real-world points. The curves are clothoid splines passing through the points of interest and can be used to model complex profiles, like the road lanes in the specific case. The real-world clothoids have to correspond to the ones passing through the visible pixels in the image. The perfect match is provided with the optimal values of the intrinsic and the extrinsic parameters, the unknowns of this optimization problem. Together with input image data, one can also estimate the car position on the circuit providing an initial guess of the camera extrinsic parameters by exploiting the partial telemetry available. Such an

alternative method can overcome possible problems during parameter estimation that can be addressed to the low quantity of data collected. Indeed, fitting a curve provides potentially infinite points that can be used to perform matching.

This work can be divided into the following parts. First, some state-of-the-art works will be shown, explaining what has been done for trajectory estimation as well as for camera calibration. Next, the main methodology used in the work will be shown with a description of the camera calibration procedure, the point projection method and the reference frames involved in the work. After that, the experimental setup applying the classical approach will be provided together with the results. This part deals with the classic camera parameters estimation and the camera calibration problem where Zhang's method is applied in two circuit scenarios, providing certain results that will be discussed. In the final part instead, a possible improvement of the first method is proposed, providing much more points than before and introducing the telemetry data in the analysis.

Chapter 2

Related works

In this chapter, some state-of-the-art studies regarding vehicle state estimation and camera parameter identification are proposed. It is also interesting to see how the same problem, or a similar problem, has been solved in other studies or how different approaches to camera calibration can be used for camera pose estimation (and vehicle pose estimation as well since the two objects are bounded).

2.1 Car trajectory estimation

As previously stated, vehicle trajectory estimation plays a fundamental role according to the application scenario. New technologies regarding sensors, data analysis and neural network techniques allow the development of autonomous vehicles whose state estimation is relevant for decision makings regarding manoeuvres for example. Cameras can be used in this scenario together with other sensors, like GPS. This was what Sadli et al. [6] proposed. They worked on a system able to perform lane-level localization using only a vision sensor and a GPS for comparison. From visual information, they extracted the lateral car estimation using LaneNet and provided the longitudinal car direction by matching GPS coordinates and reference coordinates. Although camera intrinsic parameters are not needed, car position is incomplete since they don't provide any information about the yaw angle of the vehicle. Moreover, lateral distance is based on the fact that the testing road has a constant width. In a racing scenario, the width may vary along the track and multiple lines can affect the correct detection of lanes. The usage of all camera parameters provides an overall understanding of the sensor employed as well as its orientation.

Another related work concerning car state estimation in of autonomous vehicle studies is the one provided by Jean-Alix David in his Mater thesis [1]. In his work, he uses Open Street Map data instead of Google Earth like Sadli for reference point collection. Moreover, lane detection in this case is carried out by means of ridge detection, which is a peculiar computer vision technique that exploits the Laplacian of pixels. The difference with respect to the previous work is that in this

case a bird eye view is used for map matching. Also in this case, no information about yaw angle is provided and experiments show that non-highway scenarios may cause problems in position estimation since the ridge detection does not work well in the case of non-straight lanes. Differently, in the presented thesis the steps that determine the characteristics of the image sensor are described whereas David considers the camera used already calibrated. It also exploits data coming from additional sensors like IMU, which is not present in the presented case study.

Cameras in these kinds of problems can be off-the-shelf and mounted on the vehicle for tests or can be cameras already attached to the car, as in the F1 cars. Whatever the scenario, a localization of the camera can in turn provide the localization of the car itself knowing the relative position between the camera and the GPS or the car's centre of mass. A useful article that can be exploited for further studies on camera localization is the overview provided by Yihong Wu et al. [7]. This review describes all possible camera localization techniques that have been studied according to different scenarios concerning the knowledge of the environment surrounding the camera. In the case of known environments, PnP solutions are suggested although for large environments they are not enough and more sensors are needed. In the case of unknown environments instead, the best solution to achieve camera pose estimation is SLAM. This is becoming a widely used technique together with neural network solutions and custom SLAM algorithms. For the specific application presented, it will be sufficient to estimate extrinsic parameters in a closed-form solution once the intrinsic parameters are determined.

2.2 Camera calibration

Camera calibration is an important step if one wants to take measurements from this kind of sensor. In the presented work, the camera calibration procedure is exploited for camera pose and vehicle state estimation. It is true that given the intrinsic parameters, one can focus on the computation of only the extrinsic parameters, but since the intrinsic camera parameters are unknown, then the overall estimation procedure can be considered without isolating intrinsic parameters' estimation from the extrinsic ones. Zhang's method [8] is a possible implementation of the camera calibration procedure. It is a two-step process that exploits the flatness of the world surface on which key points lie. Together with that Heikkila uses the same approach introducing lens distortion. The detailed procedure will be shown in the next chapter.

These procedures are somehow classic approaches, that take into consideration world-image correspondences. What if world points are unknown or are affected by noise or, more in general, have lots of uncertainty associated? In this case, relying on the mentioned method is not advisable. Instead, there are other techniques that return the intrinsic parameters alone. Then these intrinsic parameters can be used to define the extrinsic ones as the real world is precisely known. At least, the sensor can be characterized. The first alternative approach for camera parameters

identification exploits the so-called vanishing points, straight lines that in image representation meet together on the vanishing line defining the horizon. This is what E. Guillou et al. [3] did. Using only one single image and assuming some constraints dealing with the sensor itself (for example optical centre in the image centre, the presence of at least two vanishing points and so on) it is possible to determine the focal length. Their work also defines a methodology for camera pose estimation by placing some virtual objects in the image. What the work does not consider is the presence of lens distortion. Indeed, the results were good thanks to the presence of high-quality images with low distortion. It will be shown that in the scenario presented in this work, distortion is present and a method based on vanishing points cannot precisely work.

An alternative approach based on only image information relies on deep learning. In particular, image features can be exploited to derive sensor characteristics like focal length or distortion coefficients via neural networks. Lopez et al. [5] proposed a deep learning approach for single image calibration exploiting the radial distortion affecting the true representation of the scene. Given a high-quality dataset based on SUN360 panorama images, a neural network was fed to provide focal length, radial distortion coefficients, tilt angle and roll angle of the camera. The network is trained on high-quality images. It can be used in the proposed work to have an initial guess of the intrinsic camera parameters but accuracy has to be verified with the available dataset extracted by the onboard camera images since they do not achieve the same quality as the dataset used by Lopez. Anyway, it can be a possible future work development for the F1 onboard camera characterization, since no available technical data are reported on the net.

Chapter 3

Methods and theoretical aspects

Vision systems are very important nowadays as they are able to perform measurements and estimations at low cost, being a valid alternative to LIDARs and other sensors for position estimation. One of the key aspects of performing measurements from an image sensor is the process of camera calibration. This problem consists in finding the so-called intrinsic and extrinsic parameters of the camera. While intrinsic parameters model its lens and sensor behaviour, extrinsic parameters define the roto-translation matrix that expresses coordinates in the real world projected in the camera reference frame. This last set of parameters can be exploited for an estimation of a car's position and orientation on the track, as the camera moves accordingly to the car's trajectory. Thus, given a set of frames with the known world points position and the corresponding pixel coordinates in each frame we can find the state estimation of the vehicle as the set of extrinsic parameters at each analyzed frame. Since camera calibration in the classic sense requires a pattern to be detected and analyzed, it is necessary to choose the correct set of points. At the same time, the proper camera model has to be taken into consideration.

3.1 Reference frames and model description

The environment in which the state and pose of the car are estimated can be described by the following reference systems:

- Global reference frame: given a circuit of interest, its road elements can be associated to a reference frame positioned in the starting line position with the triple of orthonormal vectors $\{\hat{i}, \hat{j}, \hat{k}\}$ representing the x, y and z direction. The z coordinate measures the height from the ground, the x is oriented with the longitude towards the east and the y component as the latitude towards the north. The representation of the world reference frame is shown in Figure 3.1.
- Local reference frame: for the specific case of patterns, Cartesian coordinates

defining their points' position are defined with respect to an origin placed inside the pattern. The choice of considering a local reference frame is because calibration has been performed isolating the patterns from the rest of the circuit. Moreover, in order to make road features resemble the typical chessboard calibration pattern the origin is placed on one of the detected points as in the case of chessboard applications. But one can apply a translation vector from the origin of the global reference frame to the local and express the camera location with respect to the global reference frame. The axes' orientation is the same as described in the case of the global reference. The only difference is that the local reference frame's origin is translated by a certain vector in the xy plane.

- Camera reference frame: for the sake of simplicity and due to the lack of information available, we can consider only the camera reference system as the moving object in the environment. It's not possible to precisely define where the camera is positioned with respect to the car's centre of mass, but once its position and orientation are defined we can have a good approximation of the car's state. The camera reference frame is defined again by a triple of orthonormal vectors $\{\hat{i_c}, \hat{j_c}, \hat{k_c}\}$ representing the x, y and z direction respectively. The z component is aligned with the optical camera axes, the y component points downwards and the x component points rightwards with respect to the camera. The camera is positioned in the T-shape part of the car over the roll bar and the z-x plane is not parallel to the x-y plane of the global reference frame. An illustration of the system is shown in Figure 3.3.
- Image reference frame: the image plane which is visible to the user is defined as a set of pixels with x and y components and for each pixel there is a set of RGB components defining the colour of that pixel. In practice, the sensor plane is nothing but a discretized set of squares with a three-component vector for each element defining the plane. Pixel coordinates are positive and the origin is positioned in the top left corner of the plane, with the x coordinate increasing from left to right and the y component increasing from the top to the bottom as shown in Figure 3.4.

Points in the world reference frame are mapped in the image plane on the onboard camera. In the next section, the camera model will be explained as well as how image points are obtained

3.2 Camera model and points projection

3.2.1 The pinhole camera model

The projection of real points and their rays on the camera image plane has to follow some geometrical rules that are defined by a camera model. The typical one used in computer vision applications is the pinhole camera model. It assumes that from each point of interest, it is possible to take a ray passing through a unique point,

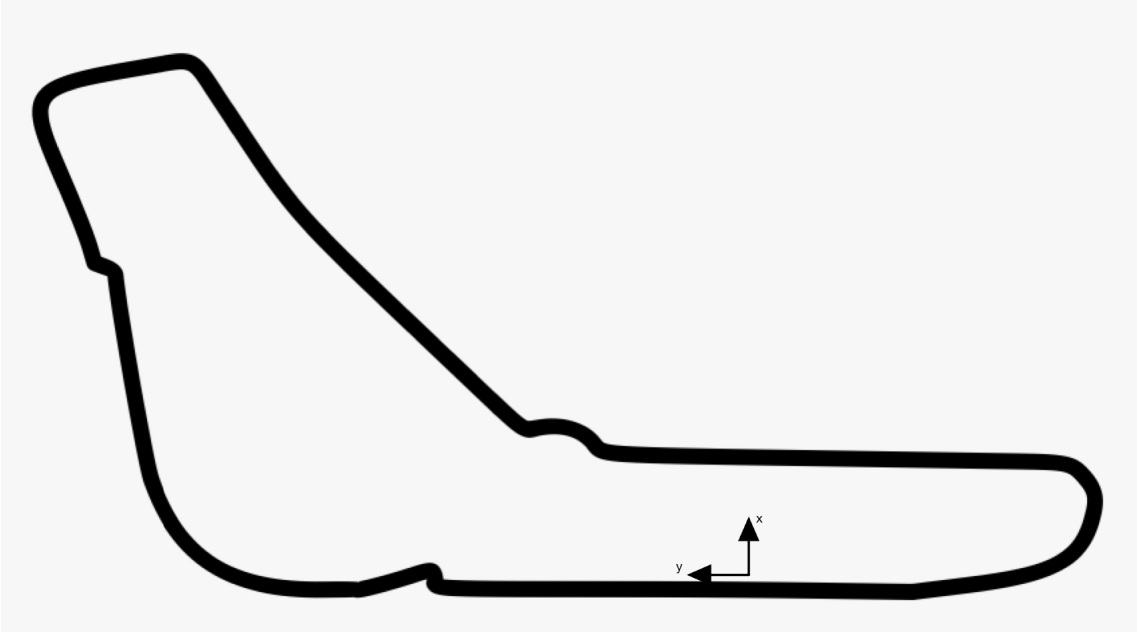

Figure 3.1: Global reference frame

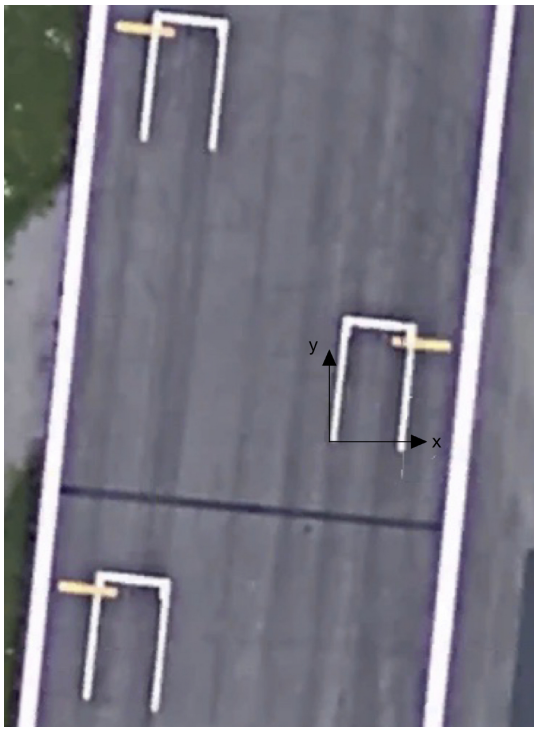

Figure 3.2: Local reference frame example

the pinhole, and then intersecting the image plane which has a distance f along the z axes of the camera called focal length. Suppose for the moment that the camera reference frame is rotated, such that the y-axes points upwards and the x-axes points to the left, and that the image reference frame is rotated accordingly and positioned in the image centre. This allows the reader to show the pinhole model described in Figure 3.5. As we can see, the point X has a ray passing through the image plane positioned in p in x as it goes towards the camera centre C. Of course, the model is an ideal one since the rays cannot be straight due to the presence of lenses that

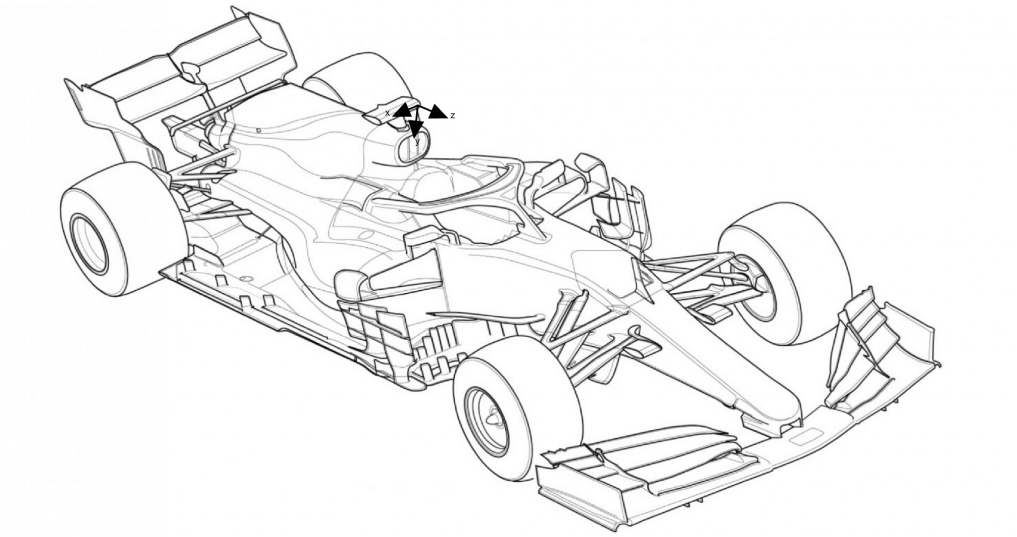

Figure 3.3: Camera reference frame

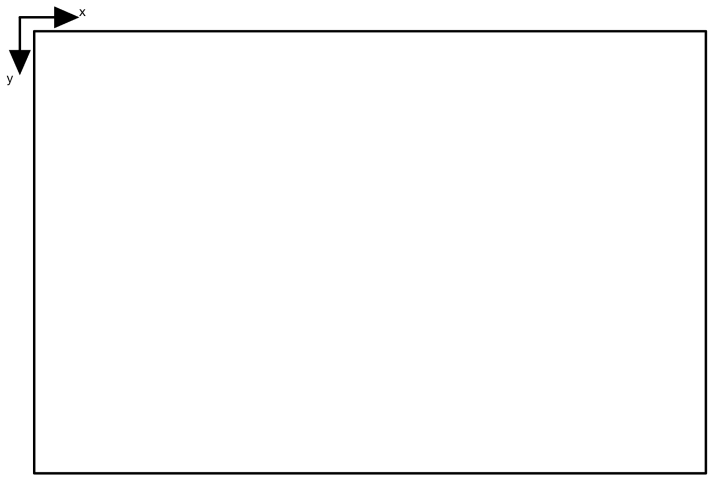

Figure 3.4: Image reference frame

cause radial and tangential distortion. Despite this approximation, it is still a good and simple model that is used quite a lot in vision applications.

3.2.2 Image coordinates determination

Another aspect to take into consideration is how image points are determined given the camera centre, the real-world coordinates and the distance between the optical centre C and p, which is the intersection between the image plane and the camera axes according to Figure 3.5. It is still supposed to use the reference frames of the camera and of the image as defined in the figure. Let's also consider for simplicity to take into consideration the zy plane. Tracing the straight line from the point of interest to the camera centre defines two triangles, the bigger one connecting the point to the camera centre and projecting the point itself on the Z axes, and the

smaller one connecting the camera centre, the intersection between the image plane and the Z axes and the intersection between the ray and the image plane. These two triangles are similar because they share the same angle in C and they have two orthogonal angles as projections of the points on the Z axes. The similarity is thus exploited to obtain the y coordinate of the point on the image plane resulting in

$$y = f\frac{Y}{Z} \tag{3.1}$$

The same holds for the x component:

$$x = f\frac{X}{Z} \tag{3.2}$$

As one can see from equations 3.1 and 3.2, after a projection of the point on the image plane in 2D information about the depth of the point is lost. In other words, knowing the X, Y and Z components of a point in 3D it is possible to project the point in 2D. On the other hand, going back from 2D to 3D is not possible as only 2 equations are available but 3 unknowns have to be determined. Another thing to notice is the role of Z in the point projection. As the point is far from the camera, the projected point will be closer to the camera centre. In other words, Z acts as a scaling factor, showing the object bigger or smaller on the image plane according to its distance from the camera.

The above expressions of the image plane coordinates can be generalized to a more complex expression: the coordinates can be translated with respect to the top left image corner and the 3D points can be derived from the camera reference frame roto-translation with respect to the global reference frame. Moreover, image reference frame y-axes points downwards due to the fact that the camera model taken into account considers the pinhole as a global ray intersection point. This causes the image to be flipped on the image plane. The complete expression is described in the following sections in matrix form and it is used for solving the camera calibration problem.

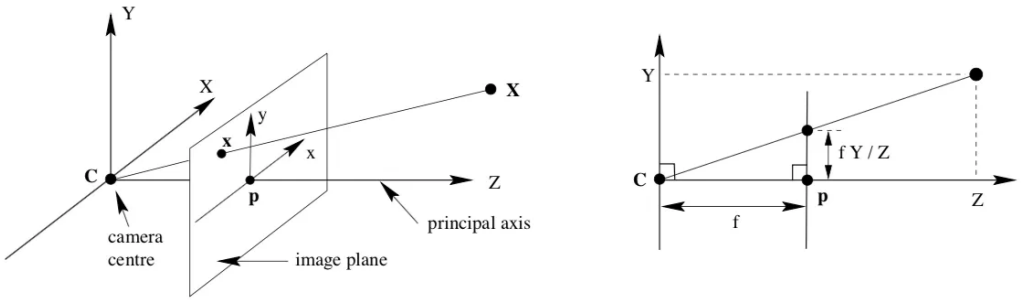

Figure 3.5: Pinhole camera model and geometry

3.3 Camera calibration problem

3.3.1 Introduction to the problem

The problem of camera calibration aims at finding the parameters of the following matrix form equation, describing how real points in 3D can be mapped on a 2D plane:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{I} & | & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.3}$$

Where

- $\mathbf{K}$ is the intrinsic matrix $\begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$ containing the focal length f_x in x and f_y in y in pixels and the optical center coordinates in x and y, i.e. o_x and o_y respectively
- $\begin{bmatrix} \mathbf{I} & | & \mathbf{0} \end{bmatrix}$ is a 3-by-4 matrix with a 3-by-3 identity submatrix and a zero column vector
- $\begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is the roto-translation matrix from the camera to the world reference frame, aiming at projecting world points in the camera reference frame. $\mathbf{R}$ denotes the rotation sub matrix, while $\mathbf{t}$ denotes the tranlation vector.
- $\begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ is the set of 3D coordinates in the global reference frame
- $\begin{bmatrix} x & y & w \end{bmatrix}^T$ is the set of homogeneous coordinates in the image plane representation with a scale factor w. Ideal pixel coordinates on the final 2D plane are thus scaled and the final values that can be measured from an image are $u = x/w$ and $v = y/w$.

Equation 3.3 considers the problem in an ideal scenario, assuming perfect square pixels, no skew of the image plane, no lens and tangential distortion. In real life, these elements are present. Radial lens distortion for example deviates straight lines from their ideal linear behaviour providing elements that appear curved. The distortion can be distinguished between pincushion distortion and Barrel distortion. The graphical difference is provided in Figure 3.6. As image points move away from the principal point (positive radial displacement), image magnification decreases and a pincushion-shaped distortion occurs on the image. As image points move toward the principal point (negative radial displacement), image magnification

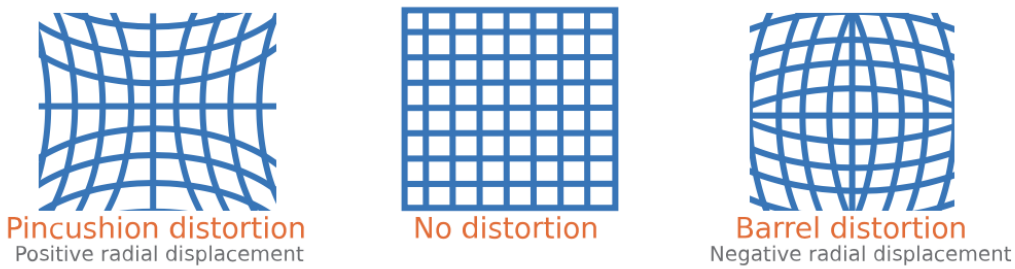

Figure 3.6: Radial lens distortion

increases and a barrel-shaped distortion occurs on the image. Mathematically speaking, the effect can be modelled as follows, considering no skew factor:

$$x_{distorted}^{rad} = x_n \cdot (1 + k_1 r^2 + k_2 r^4) \tag{3.4}$$

$$y_{distorted}^{rad} = y_n \cdot (1 + k_1 r^2 + k_2 r^4) \tag{3.5}$$

Equations 3.4 and 3.5 model the radial distortion due to the presence of lens via k_1 and k_2 coefficients. The model can be made more complex by adding another term $k_3 r^6$. In the equations:

- $r = \sqrt{x_n^2 + y_n^2}$
- x_n is the normalized undistorted x image component that without skew factor can be calculated as $x_n = (u - o_x)/f_x$
- y_n is the normalized undistorted y image component that without skew factor can be calculated as $y_n = (v - o_y)/f_y$

$x_{distorted}^{rad}$ and $y_{distorted}^{rad}$ are the final pixel values visible from the image side.

In addition to the radial distortion effect, one can also model the tangential distortion. This phenomenon occurs when the sensor plane is not orthogonal with the camera's optical axes as shown in Figure 3.7. This behaviour can be modelled by:

$$x_{distorted}^{tan} = x_n + [2 \cdot p_1 \cdot x_n \cdot y_n + p_2 \cdot (r^2 + 2 \cdot x_n^2)] \tag{3.6}$$

$$y_{distorted}^{tan} = y_n + [p_1 \cdot (r^2 + 2 \cdot y_n^2) + 2 \cdot p_2 \cdot x_n \cdot y_n] \tag{3.7}$$

where p_1 and p_2 are the tangential distortion coefficients.

The combination of radial distortion and tangential distortion is modelled by adding the two effects, so that

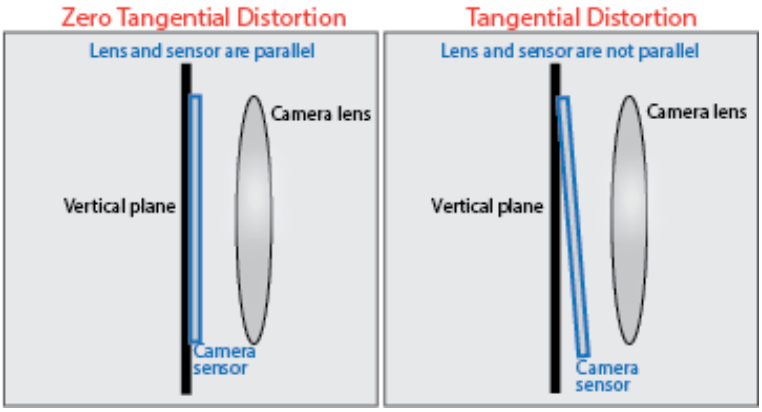

Figure 3.7: Tangential lens distortion

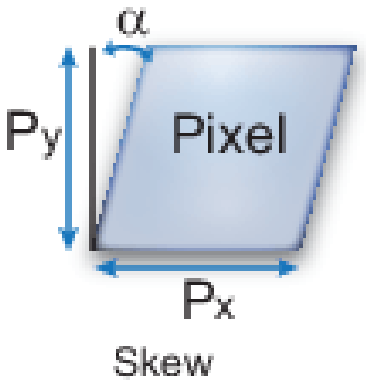

Figure 3.8: Skewed pixel representation

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x_{distortion}^{rad} + x_{distortion}^{tan} \\ y_{distortion}^{rad} + y_{distortion}^{tan} \end{bmatrix} \tag{3.8}$$

The last non-ideality that can be considered is the skew factor. Typically, the x and y sensor image axes are considered to be orthogonal. In reality, this may not happen and point projection can be affected by this. To model the skew it is sufficient to rewrite the camera intrinsic matrix as

$$\mathbf{K} = \begin{bmatrix} f_x & s & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

with the second element of the first row representing the skew factor. What s represents is given by $s = f_x tan\alpha$, being α the angle represented in figure 3.8. Since s defines the first value of pixel components in the absence of lens distortion, the normalized value that later defines the radial and tangential distortion coefficients has to be rewritten as $x_n = (u - o_x - s \cdot y_n)/fx$ with y_n being the same as before.

Finding the correspondences between real-world coordinates and distorted image coordinates provides the optimal values of the intrinsic parameters of the camera as well as lens distortion coefficients.

A camera calibration procedure can be seen as a first tentative to recover the parameters describing the state of the car. Indeed, given a certain number of frames, not necessarily in order, we can pick a set of image points from each image and their corresponding points in the real world. Since the camera is the same for all frames, the intrinsic parameters don't change, whereas the extrinsic parameters can describe the orientation of the vehicle as well as its position relative to the detected patterns.

Knowing the pixel values and the real-world point coordinates, the parameters have to make the difference between measured pixel and point projection equal to zero. Since it is impossible to achieve such a result, the problem is solved by taking more points than needed and solving an optimization problem. This means that the sum of square errors for all points and frames between the pixel x-component and the projected point x-component summed with the square error on the y side has to be minimized. The minimization of this objective function is provided by the intrinsic parameters, including the radial distortion components, and the extrinsic parameters as well. The complete procedure for solving the minimization is described in the following subsection

3.3.2 Zhang and Heikkila calibration procedure

The problem of camera calibration can be solved with various techniques, like the classic optimization problem to solve all the parameters involved or deep learning techniques or vanishing point techniques to find only the intrinsic parameters and later estimate the extrinsic parameters apart.

The procedure described here follows a two-step process: as a first step, intrinsic and extrinsic parameters are solved in closed form, assuming zero lens distortion; the second step takes the first results as an initial guess for the nonlinear least-squares minimization (Levenberg–Marquardt algorithm). The first step follows Zhang [8] procedure to find the parameters, while the second step involves also Hikkila work [4] for lens distortion effect that is not taken into consideration as an initial guess.

First step: find the parameters in closed form as an initial guess

Recalling equation 3.3 the problem can be simplified by assuming all points with $Z = 0$ since the points of interest are distributed on the road in the neighbourhood of the camera. This allows cancelling the third column of $\begin{bmatrix}\mathbf{R} & | & \mathbf{t}\end{bmatrix}$ and reduce the term $\mathbf{K}\begin{bmatrix}\mathbf{I} & | & \mathbf{0}\end{bmatrix}$ simply to $\mathbf{K}$. At this point, the homography matrix $\mathbf{H}$ is obtained. It represents points projection between the ideal image plane and one set of points lying on a 3D world plane, namely

$$\mathbf{H} = \lambda \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix} \tag{3.10}$$

The homography matrix is defined up to a scale factor λ. The following constraint equations can be imposed on the parameters to find a unique solution, thanks to the orthonormality of the first two columns of the reduced extrinsic matrix:

$$\mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 0 \tag{3.11}$$

$$\mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 \tag{3.12}$$

One can define $\mathbf{B} = \mathbf{K}^{-T}\mathbf{K}^{-1}$ up to a scale factor which is a symmetric matrix and can be defined by a 6D vector $\mathbf{b} = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix}^T$. So equations 3.11 and 3.12 can be rewritten in the following vectorial form for one image plane-world plane correspondence

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \tag{3.13}$$

where $\mathbf{v}_{ij} = \begin{bmatrix} h_{i1}h_{j1} & h_{i1}h_{j2} + h_{i2}h_{j1} & h_{i2}h_{j2} & h_{i3}h_{j1} + h_{i1}h_{j3} & h_{i3}h_{j2} + h_{i2}h_{j3} & h_{i3}h_{j3} \end{bmatrix}^T$, $\mathbf{h}_i = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} \end{bmatrix}^T$ and $\mathbf{h}_j = \begin{bmatrix} h_{j1} & h_{j2} & h_{j3} \end{bmatrix}^T$ being the i-th and j-th column of $\mathbf{H}$ respectively.

Given n images equation 3.13 can be enlarged becoming

$$\mathbf{V}\mathbf{b} = \mathbf{0} \tag{3.14}$$

with $\mathbf{V}$ being a 2n-by-6 matrix. The solution to 3.14 is well-known as the eigenvector of $\mathbf{V}^T\mathbf{V}$ associated with the smallest eigenvalue. The solution of the intrinsic parameters is:

$$o_y = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \tag{3.15}$$

$$\lambda = B_{33} - [B_{13}^2 + o_y(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \tag{3.16}$$

$$f_x = \sqrt{\lambda/B_{11}} \tag{3.17}$$

$$f_y = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \tag{3.18}$$

$$o_x = -B_{13}f_x^2/\lambda \tag{3.19}$$

Notice that the skew factor has not been considered in the intrinsic matrix. If one considers this value to be different from 0, then

$$s = -B_{12}f_x^2 f_y/\lambda \tag{3.20}$$

$$o_x = s \cdot o_y/f_x - B_{13}f_x^2/\lambda \tag{3.21}$$

Also, the initial guess of the extrinsic parameters is derived. Given the scale factor defined in this case as $\lambda = 1/\|\mathbf{K}^{-1}\mathbf{h}_1\|$ we get:

$$\mathbf{r}_1 = \mathbf{K}^{-1}\mathbf{h}_1 \tag{3.22}$$

$$\mathbf{r}_2 = \mathbf{K}^{-1}\mathbf{h}_2 \tag{3.23}$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \tag{3.24}$$

$$\mathbf{t} = \lambda\mathbf{K}^{-1}\mathbf{h}_3 \tag{3.25}$$

Now that the solutions in closed form of the intrinsic and the extrinsic parameters are obtained they can be used to refine the final solution by solving an optimization problem.

Second step: optimization procedure

The optimization procedure aims at refining the initial guesses obtained with the above procedure by reducing the error between actual pixel values and projected points via roto-translation and intrinsic parameters. Here the effect of lens distortion described by 3.4, 3.5, 3.6 and 3.7 is taken into consideration. The goal is to minimize the error function

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\|p_{ij} - \hat{p}(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{P}_j)\|^2 \tag{3.26}$$

where:

- n is the number of images
- m is the number of points in the i-th image
- p_{ij} is the measured j-th pixel in the i-the image
- $\hat{p}$ is the projected point $\mathbf{P}_j$ in the i-th image via $\mathbf{K}$, $\mathbf{R}_i$ and $\mathbf{t}_i$, namely the intrinsic matrix, the i-t rotation matrix and the i-th translation vector. The point has to be passed to the equation 3.8 modelling lens distortion

The unknowns that minimize equation 3.26 are 4 intrinsic parameters, 2 additional parameters for radial distortion and $12\times m$ total extrinsic parameters. The problem can be solved with the Levenberg-Marquardt algorithm with the initial guesses obtained at the previous step to avoid falling into local minima.

3.4 Calibration methodologies and pattern choices

It has been stated that the goal is to estimate where a camera is located and at the same time define its parameters, so it is necessary to define a methodology for its calibration. The approaches can be different:

- one can assume to provide a fixed pattern, like in the classic camera calibration using a chessboard, and capture the frames showing different orientations of this pattern with respect to the camera; in other words, one can pretend to deal with a camera, in the specific case the one mounted on the car, capturing the different ways this pattern appears to the image sensor. The movement of the car is ignored and once the intrinsic parameters are obtained, the position of the camera can be estimated by calculating the remaining extrinsic parameters.
- the other way is based on the usage of a sequence of frames showing a wider pattern developing along the path described in the video frames; in this case, the extrinsic parameters provide directly the position and orientation of the car with respect to the analyzed pattern, without computing it apart.

The procedure described in the first point aims at referring all points describing a pattern to a unique local reference frame. The detailed reason for this mapping choice instead of a more extended pattern will be provided in the section ”Final comments on results” of Chapter 4.

3.4.1 Fixed pattern methodology

As described previously, the first approach consists in calibrating the camera by estimating all the parameters and considering then only the intrinsic ones as relevant. The pattern is considered ”fixed” while the car moves around it. In other words, what matters is not where the pattern is really positioned in the global reference frame, but how its points are positioned locally. For this kind of calibration, the resulting extrinsic parameters won't be real and have to be recalculated given the intrinsic ones, which are instead true whatever the relative position between the car and the pattern.

As already mentioned, no camera is handled and no checkerboard can be positioned around, so one can search for a repeated pattern on the track. One possibility is to exploit the repeated sequence of starting grid positions. They cover a large sensor area compared with kerbs or other circuit elements. On the other hand, their sequence is presented in a poor variety of orientations since the car goes only in one direction. One possible solution to that is to exploit the alternation of starting grids and perform the correspondences as if the car goes in the opposite direction. This increases the number of frames and makes the dataset richer as pixels cover a larger sensor area. In the end, what matters is the distance between points forming a pattern, that does not change in space. The mapping on the image must be coherent as well.

Once the intrinsic parameters are computed, the true estimation of camera position and orientation is provided by recomputing the extrinsic parameters in closed form with the additional knowledge of computed intrinsic parameters. By doing so, a real visualization of the camera position on the track can be visualized.

Figure 3.9: Example of starting grid pattern in some video frames

3.4.2 Moving pattern methodology

The second approach proposes an alternative pattern to be detected. This pattern is not considered fixed but rather extended in the space. Frame after frame, the car moves near its elements and calibration is performed by matching visible points in the frame with the corresponding world points. The final results of camera positions and orientations frame per frame define directly the state of the camera and o the vehicle as well. For this kind of calibration, the kerbs defining the chicanes have been chosen. Keypoints of such patterns appear in different orientations and positions on the sensor. Although not (almost) all points appear in all the processed images covering a large area like in the case of starting grids, this kind of pattern may provide a complete calibration as well.

The difference with the previous methodology is that camera extrinsic parameters directly provide the position and orientation of the camera sparing additional computation time.

Figure 3.10: Example of chicane kerbs pattern in some video frames

Chapter 4

Experimental setup and results

In the following sections the dataset, the practical implementation and the results obtained for state estimation with camera calibration procedure are shown.

4.1 Dataset extraction

In order to perform the camera calibration it is necessary to define the data to analyze. As written previously, different patterns have been presented for camera calibration and pose estimation. The circuit of Monza is considered for these kinds of tests and the different patterns described will be picked from it for the world points data. For what concerns the image dataset, a video showing the qualifying lap of a driver during the F1 Italian Gran Prix in 2022 will be considered. For the presented tests, the Q1 session of Leclerc has been considered. For each frame, the key points lying on the starting grid or on the kerbs defining the chicane will be taken into consideration. Road elements in general are taken into consideration since the real-world measurements will be performed via Google Earth, showing the circuit from the above. Let's see in detail the dataset extraction for world points and image points.

4.1.1 World points extraction

The focus is on world points taken from lateral kerbs along the track, the starting grid position and the first chicane in Monza. Reference points lie on the corners of coloured kerbs or on starting position corners which are clearly visible from the sensor and from the satellite image provided by Google Earth. From the mentioned website it is possible to manually mark points of interest and create a pattern. The element containing all possible points is a ".kml" file containing geodetic coordinates.

Files in ".kml" format provide the latitude and longitude of the points of interest but they need to be analyzed in Cartesian coordinates. For this reason, one of the points belonging to the pattern will be marked as the origin according to which the Cartesian coordinates of all the other points will be calculated. In this way, a local reference frame with axes pointing in the north and east direction will be defined. Figure 4.1 and 4.2 show the interface through which is possible to collect the points of interest and the final 2D plot in Cartesian coordinates. Notice that in Figure 4.1 some markers do not correspond to the coloured edges of kerbs because the video was recorded in 2022 and shows slightly different coloured kerb lengths with respect to Google Earth satellite image. Therefore, world points are collected according to the visible kerbs in the video.

One important remark is that it is supposed that all points lie on a common plane with no difference in height. So we associate each point with the altitude revealed in Google of the origin point. This is an approximation since some elements like kerbs are not perfectly planar.

Figure 4.1: Google Earth interface with marked keypoints

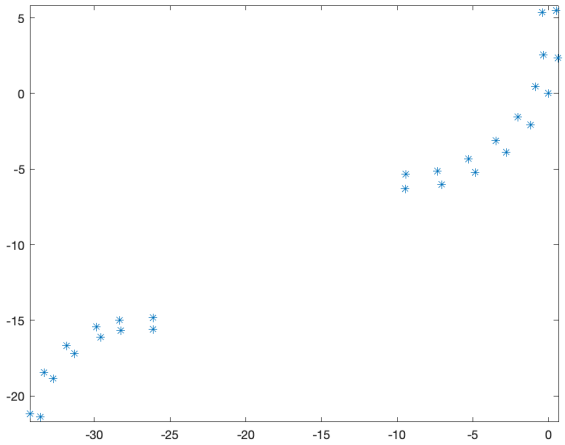

Figure 4.2: World points Cartesian coordinates

For the specific case of starting grids, key points do not only lie on the corners. Indeed, to augment the set of points and possibly have a good number of them some points on the lateral lanes near the starting grids have been considered. Providing an initial set of 12 points, the additional points increase the set to 18. Each couple of additional points results from the intersection between the line of the border and the one passing through two upper or lower edges defining the starting grid rectangle. The set is further augmented by selecting the 18 points and adding the mean position between couples of them. Couples with an additional point in the middle are the ones on lateral lanes and the white segments defining a single starting grid.

4.1.2 Image dataset and points extraction

From image points, the corresponding Cartesian world points which are visible inside the image can be collected. In some frames extracted from a video, the car is in the neighbourhood of the pattern of interest, being it a kerb, a sequence of starting grids or a chicane. It is not necessary that frames are examined in order. What matters is the correct correspondence between pixels and real-world coordinates. Pixels are manually selected from each image by picking the pixels' coordinates in proper order according to how world points have been collected. Images are taken from a screen recording of a qualifying video taken from the "F1TV" website, with a final resolution of 1034×1840. The order is checked for each image by marking the collected pixels and their order. Some examples are shown in Figure 4.3. For the specific case of starting grids, pixels on lateral lanes are marked if none of the points on the edges are occluded, for example by one of the front car wheels, following the line that passes through the marked pixels on the image. Then, the remaining points are obtained as midpoints of the starting grid segments. Another comment on this kind of pattern regards the selection of key points in images. Since some images show starting grids with alternating positions, there will be a forward mapping following the order left-right-left and a backward mapping when

an image shows a right-left-right sequence. In this case, to pretend the camera looking at a different angle the order of points collection should follow an inverse sequence.

Figure 4.3: Example of points selection using chicane kerbs

4.2 Matlab software for analysis

Matlab is the chosen software for the analysis since it provides a fair high number of functions for analysis. Starting with the world points dataset, it is possible to convert the ".kml" files containing geodetic coordinates into Cartesian ones. This is performed with an ad-hoc external Matlab function that applies the conversion. The collection of world and image points is still provided by Matlab with useful structure variables that can contain pixel coordinates and world coordinates. Structures can handle various types of variables so one can also add some flags for point inclusion or exclusion. Once the full dataset is achieved the calibration procedure can be taken into consideration with the equations from 3.3 to 3.26. The technique is embedded in the Matlab function "estimateCameraParameters" of the Computer Vision Toolbox and, given a set of image points data structure and of world points data structure, it outputs the intrinsic camera parameters and the extrinsic ones for each image. It is also possible to visualize the error associated with each estimation and perform some plots, in order to understand the goodness of the calibration. In particular, a reprojection error plot can be visualized, telling how much the calibration fails in points reprojection as a mean result in terms of pixels per image. Another useful plot is the visualization of the camera with respect to the observed pattern to see how well it is positioned with respect to it.

Camera calibration in Matlab can be performed providing the number of coefficients to estimate the radial distortion of lenses, the possibility to estimate the tangential lens distortion, estimate the skew factor and some initial guess parameters like the intrinsic matrix.

Several trials have been performed, by changing the number of radial distortion parameters from 2 to 3, by adding the tangential error estimation and the skew factor.

4.3 Results

4.3.1 First method: starting grids

The calibration procedure of the starting grid position is performed by analyzing 30 points per 16 images. Points are mapped so that the camera points to the pattern ignoring where it is actually located on the track, but how its points are mapped one with respect to the other in the local reference frame associated. The best calibration results based on starting grid position detection are given by the following settings: three radial distortion coefficients are used, tangential distortion is estimated and all images are considered without any removal of outliers. Results show a mean reprojection error of 7.90 pixels and intrinsic parameter values stored in table 4.1.

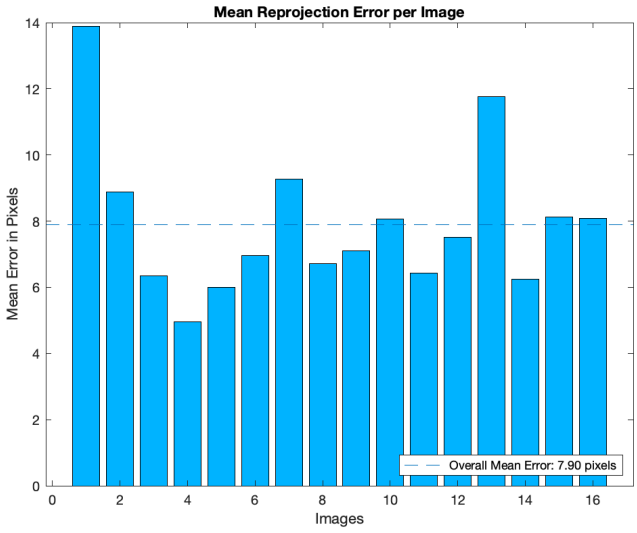

Figure 4.4: Histogram of reprojection error in pixels after calibration with the starting grids

Table 4.1: Intrinsic parameters for starting grid: 30 points, 16 images, 3 radial distortion coefficients

Intrinsic parameters	Nominal value	Uncertainty associated	relative error (%)
f_x [pixels]	875.3722	43.4681	4.9657
f_y [pixels]	99.6011	4.9609	4.9808
o_x [pixels]	911.3089	8.2161	0.901571
o_y [pixels]	364.5416	3.2577	0.893643
k_1	0.2099	0.0326	15.5312
k_2	-0.1960	0.0454	23.1633
k_3	0.0398	0.0135	33.9196
p_1	0.1063	0.0112	10.5362
p_2	0.0112	0.0021	18.7500

Focal length and optical centre values show low relative errors, while parameters determining the distortion coefficients show a higher relative error in percentage with respect to the other. Removing the first image which shows the highest reprojection error causes the mean value of it to lower down to 7.06 px and a slight change in the intrinsic parameters with a little increase of the relative errors. For what concerns other types of configurations, decreasing the distortion coefficients to 2 causes the mean reprojection error to increase to 8.05 pixels while some parameters increase their relative error and others decrease it. Providing no tangential distortion coefficient estimation causes the mean reprojection error to increase over 40 px. Moreover, intrinsic parameters show very high relative errors and totally different nominal values with respect to table 4.1. From these quick observations, one can notice that some parameters are not completely determined with the tests performed as they show higher relative errors with respect to the others reported. Possible sources of uncertainty can be the low number of images and points or the fact that points are captured more or less with the same relative angle position and they lie on the same area of the sensor. This does not properly define distortion coefficients, that can be properly tuned with points near the edges of the image sensor.

Let's now focus on the true extrinsic camera parameters estimation. For the following purpose, intrinsic parameters are considered to be fixed and extrinsic parameters are thus calculated in closed form. In order to calculate them, the world points position of some starting grids sequence was taken as well as their true mapping on images. This gives the camera position with respect to the shown pattern for every image. For the moment, it is sufficient to see what happens for a few images to see whether the results are reliable or not. Figure 4.5 shows the detected pattern, the reprojected world points relative to each frame and the estimated camera position and orientation given the intrinsic parameters calculated. Qualitatively speaking, the position is wrong compared to the one seen on the left side, in particular the position in the forward direction of the camera. There are some meters of difference between the position of the camera of the left frame compared with the one obtained after intrinsic parameters calculation. Such a difference can be probably justified by the number of points collected, which may be low, by the quality of

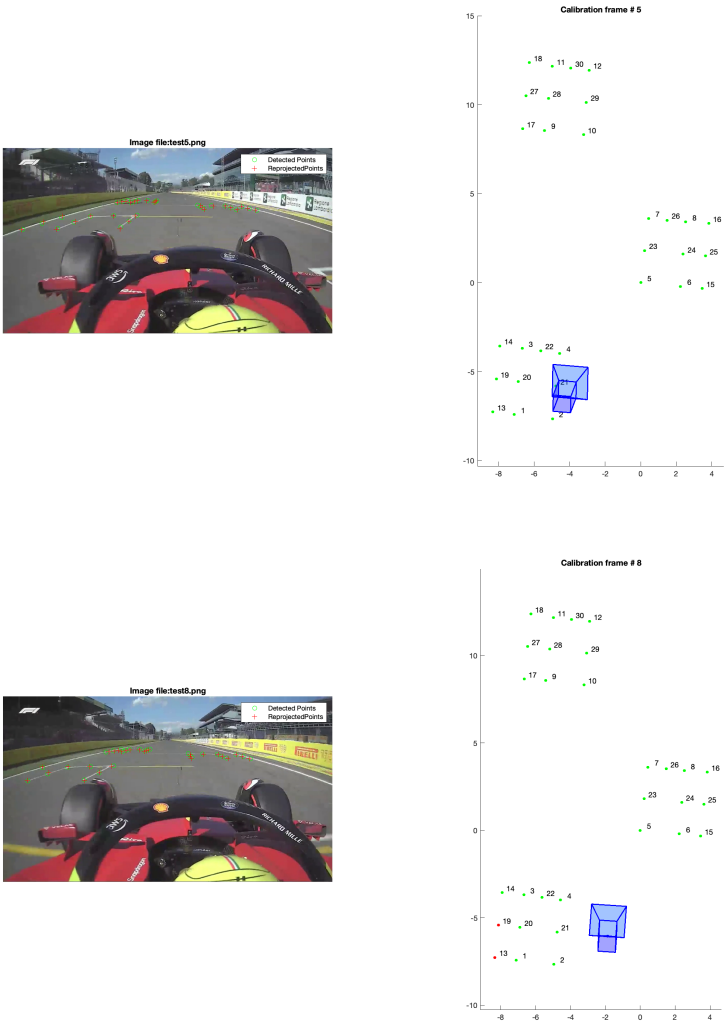

Figure 4.5: Camera pose given the intrinsic parameters and starting grids

the collection, which was performed manually both on Google Earth and on the image analyzed, and by the intrinsic parameters obtained. Other tests have been performed: changing the model of the camera lens distortion may provide bad reprojection errors, but the camera position may result more similar to the real one.

An increase in the number of points, as well as a change in the pattern, may help increase the accuracy both of the intrinsic parameters and the camera position, which in turn determines a quite good estimation of the car position on the track.

4.3.2 Second method: chicane kerbs

Let's now focus on another pattern. This refers to a sequence of coloured kerbs of the first chicane in Monza circuit. For this kind of calibration, the car moves through an extended pattern with some points visible in certain frames and others occluded because the car has not reached them (or because it left some of them

back). The relative movement between the pattern and the car provides a direct estimate of the intrinsic parameters as well as of the extrinsic ones at the same time, instead of recomputing the position and orientation of the camera once the intrinsic parameters are given. For this kind of calibration, let's focus on the best results achieved in terms of reprojection error. Point selection in this case is performed by looking at the corners of coloured kerbs and their position on Google Earth. This returns 28 points that can be seen in 22 different images. Three radial distortion coefficients, no tangential distortion estimation and no skew factor provide the results in table 4.2.

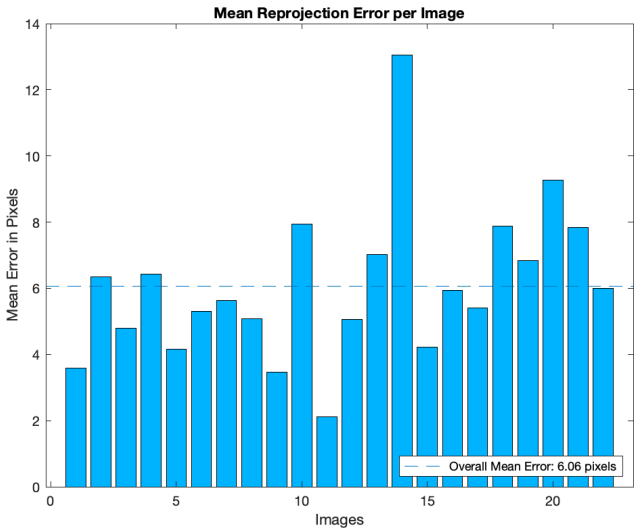

Figure 4.6: Reprojection error after calibration using the chicane pattern

Table 4.2: Intrinsic parameters chicane pattern, 28 points, 22 images, 3 radial distortion coefficients

Intrinsic parameters	Nominal value	Uncertainty associated	relative error (%)
f_x [pixels]	921.3689	36.6920	3.980
f_y [pixels]	398.9531	35.2066	8.820
o_x [pixels]	545.2952	36.1003	6.620
o_y [pixels]	470.1436	23.2992	4.960
k_1	-0.1152	0.0243	21.09
k_2	0.0005	0.0124	2480
k_3	0.0010	0.0017	170.0

Reported results show a mean reprojection error of 6.06 px, which is lower than the previous case, and a low relative error in the first four parameters. Compared with the ones in table 4.1 there is a high variation in the nominal values for f_y, o_x and o_y as well as an increase in the relative errors. Radial distortion coefficients instead present totally different nominal values than before and the uncertainty is too high. Removing one outlier, like image 14 for example, the mean reprojection error goes down to 5.73 px and nominal values remain more or less the same. The only exception is given by k_2 which increases its relative error almost three times. The

huge uncertainty on radial distortion parameters may be due to the fact that points on the corners do not completely cover the edges of the images where distortion is more present. More points for a clearer determination of these parameters have to be collected, but this encounters the limits of the image itself. Namely, the presence of the shape of the car limits the collection of more points and the fact that these elements lie only on the lower side of the image provides a lack of information for the upper part of the image, that cannot be mapped. Moreover, let's provide some comments on results obtained with different calibration settings with respect to the initial presented configuration. Tangential distortion provides totally different nominal values with high uncertainty associated. Also, the mean reprojection error increases over 80 px. This divergence is probably due to the fact that the number of points in the upper and lower parts of the image is not sufficient, as in the case of the radial distortion coefficient. Introducing the skew factor estimation provides more or less the same results that have been presented here with an additional value of the skew of 81.4335 ± 102.5574. Again the high uncertainty associated is probably due to the low number of points detected. The combination of tangential distortion estimation and skew estimation provides different values than the ones presented in 4.1 since tangential distortion is included.

Once the results regarding intrinsic parameters are provided it is time to have a look at the camera positions obtained. Differently than before, it is not necessary to recalculate the extrinsic parameters given the intrinsic ones since the position of the camera is consistent with the actual movement of the car, thus the extrinsic parameters estimated already provide the approximate state of the car. The considered intrinsic parameters are the one of table 4.2.

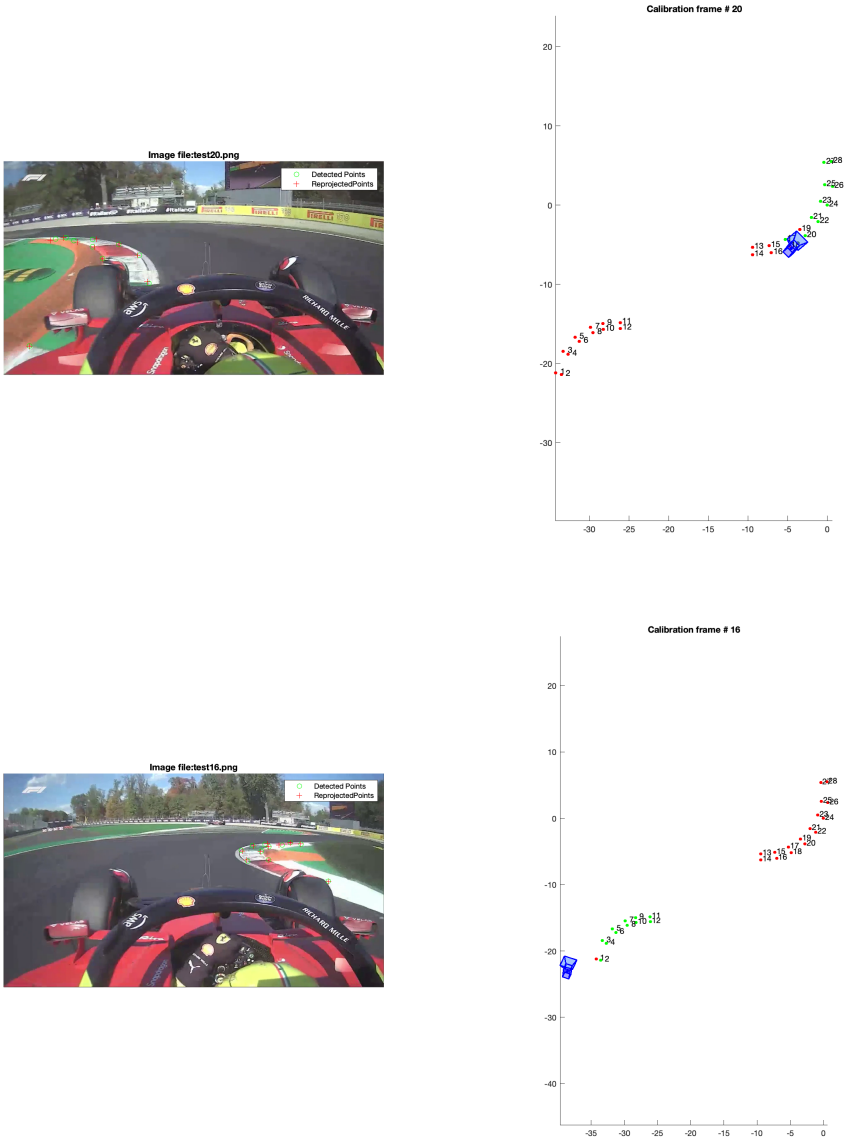

Figure 4.7: Comparison between an image and its associated state estimation on a 2D map given the chicane

Figure 4.7 shows where the camera is located with respect to the observed pattern in the image. Qualitatively speaking, it can be seen that the camera is close to its actual position in the image. The distance from the lateral line and the one from the first coloured kerb edge is in the order of a few meters. It is also true though that camera's relative orientation seems too tilted with respect to the road and, in some cases, the camera seems closer to the target than expected. By the way, no huge difference can be detected with respect to the previous case study dealing with starting grids and the possible error can be in the order of some centimetres. Since the whole estimated camera extrinsic parameters reconstruct the car position per each frame analyzed it is possible to plot all the camera poses together. Figure 4.8 represents this: the "S" shape formed in the selected track provides a quite good estimate of the car trajectory since the points representing the camera follow the distance from the kerbs given by each frame. In other words, the proximity of the

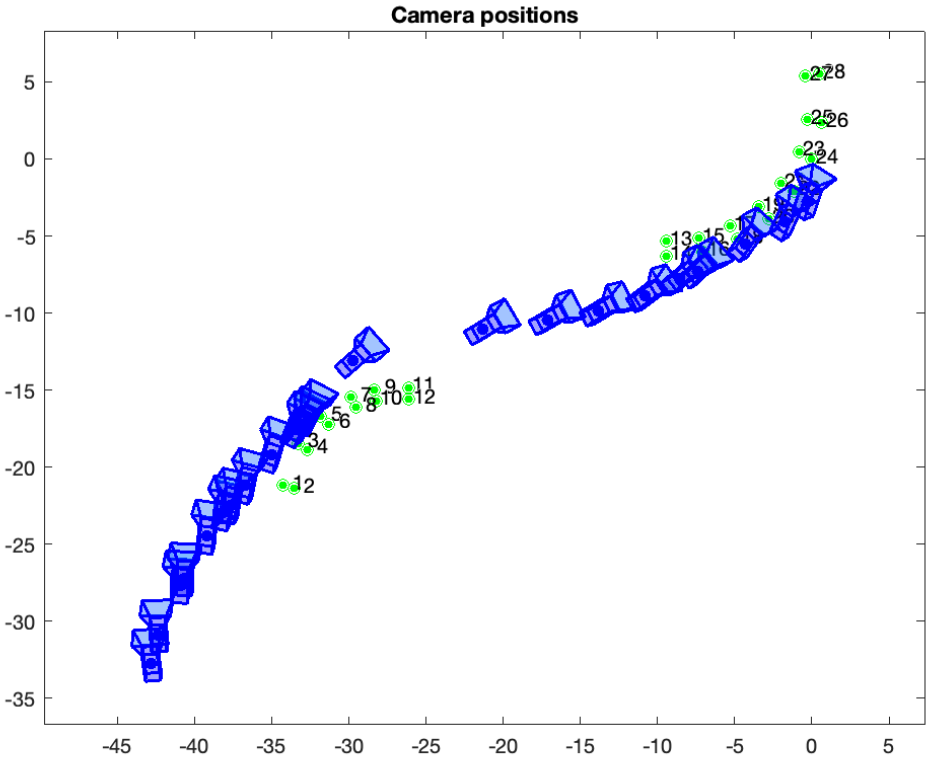

Figure 4.8: Overall representation of camera poses in the chicane pattern

image sensor to the kerbs is visually provided also by the corresponding frame, with the location of the camera near the kerbs' borders.

4.4 Final comments on results

Some different techniques for simultaneous camera calibration and pose estimation have been proposed. The first approach consists in detecting a pattern without considering the real pattern position on the track global reference frame but only how the key points are presented to the camera sensor and the relative position between them. The second approach considers a larger pattern with the camera moving near its elements. The difference in the final results is that with the first approach, one can estimate the intrinsic parameters considering a limited space in which points are positioned but true extrinsic parameters have to be recomputed. With the second one instead extrinsic parameters are already provided after the calibration procedure, but a more extended pattern has to be considered.

Starting grid pattern calibration and position estimation showed the following: intrinsic parameters showed relatively lower uncertainties with respect to the second approach but, once the intrinsic parameters are considered fixed, the final position of the camera tested with some images was not precise enough. Grids are elements that occupy a large part of the image sensor, but they are mainly distributed in the central part of the image. This means that intrinsic parameters modelling lens

distortion measure a low effect and do not provide a complete estimation of these parameters. Moreover, it seems that the relatively good results for the intrinsic parameters in terms of uncertainty shift the underdetermination of parameters towards the extrinsic ones, resulting in bad camera positioning and thus bad car pose estimation.

The pattern taken from the chicane can be seen from multiple views with points covering a limited portion of a single image, but they overall reach all possible measurable parts of the sensor (with measurable meaning that correspondence in the world reference frame can be found with the instruments available). This results in a worse determination of intrinsic parameters as they show an overall bigger uncertainty with respect to the previous case. The final position of the camera, on the other hand, turns out to be good given one set of intrinsic parameters described in the dedicated section. This is probably due to the fact that key points of a pattern are visible at a more variable distance from the sensor, providing a more complete characterization of the pose in 3D space of the camera and of the car as well.

From the general point of view, it has been observed that the two proposed methods suffer from underdetermination. It has never been observed an overall set of results providing good camera intrinsic parameters with high accuracy and the same for extrinsic parameters. A possible source of error that has produced such not robust results is the low number of points and images. Usually, for camera calibration in controlled scenarios, 10-30 images and 20-30 points each are good enough to determine the whole set of camera parameters. Zhang's method is demonstrated to be able to calibrate a camera with just two images. Moreover, there are cases in which camera calibration can be performed from a single image or with just a few points in the scene forming a polygon. In this case study, despite the number of points and images being in line with the ones suggested in online software documentation, high uncertainty is present. This means that for the specific case study, more points and images are needed. If one wants to increase the dataset the only possible solution to the problem is to take into consideration elements with high collectable points. Taking into consideration lanes defining the borders of the track can provide a rich dataset of points to be used in an optimization problem with the corresponding measurements in the world reference frame. This idea is developed in the next section as a possible solution for the encountered problems for future works.

One note about the local reference frames: each detected pattern showed points with coordinates referring to a point belonging to the pattern itself. This was done for the following reasons. First of all, the objective was to recreate as most as possible a classic camera calibration procedure with a checkerboard-like pattern. Typical procedures with cameras pointing to checkerboards position the origin of the world reference frame inside the pattern, thus the quantities describing point coordinates are in the order of a few centimetres. The problem can be scaled to the presented case study with points distant from one another in the order of meters. Positioning the origin should not be a problem, since coordinates will be scaled accordingly,

but some initial tests showed a divergence of the final camera positions of several meters, even in the order of 10^2. This leads to the second motivation for why a local reference frame has been chosen. Indeed, choosing for example the origin of a pattern considering the global reference frame could provide non-realistic results. The reason for such divergence can be caused by the low amount of points and images that provide low robustness and thus the possible divergence of the camera positions. Choosing a higher number of points can solve also this issue.

It is true that a poor set of points has been considered in each approach. This could have negatively contributed to the underdetermination of parameters, but there is another important thing to say about reprojection errors. In fact, while this parameter has been presented as good if it is between 5-8 pixels, compared with the huge numbers obtained by changing the parameters, it is true that state-of-the-art calibration procedures produce much better results. A reprojection error is considered good if it is below 1 px! This means that what has been proposed is not precise enough compared with the state-of-the-art techniques. Again, the solution can be the one proposed above, which is a huge increase in the dataset. But this is only a possible solution.

Other possible sources of uncertainty can indeed affect parameter accuracy and high reprojection error. First of all the dataset selection, regarding manual points selection via Google Earth and manual pixel selection on the image. Satellite images can be zoomed in for a particular selection of points, but the resulting image does not show sharp features or clear edges. This turns out to introduce some uncertainty for every selected point defining the pattern. The same can be said for pixel collection from video frames. The video collected for image analysis is a screen recording, that introduces some compression rate in addition to the one already existing. Indeed, the original video source does not show a high definition. This produces blocking artifacts on images and the original low definition provides a bad definition of features to be detected, which in turn appear blurred. Points selection on images and on Google Earth can thus be considered a big source of uncertainty. Kerbs, in particular, have been considered flat, whereas they show some holes occluding real kerb borders and changing the real coordinates of edges. In addition to that, paintings defining kerbs may change year after year introducing false correspondences between the recorded satellite image and the recorded video of a qualifying session. The novel alternative proposed in the next chapter can mitigate the problem since kerbs are avoided and lane borders remain unaltered through the years.

A final observation of the images and corresponding points has to be made. Indeed, the only measurable points in the world reference frame are road elements, with quite good reliability on latitude and longitude, while altitude is not precise enough. This means that the only possible correspondences are between road elements on the image and road elements on Google Earth. But this introduces a quite evident limit to the problem resolution. Since the road lies mainly in the lower middle part of the image, pixels in the upper part are unused, while they can certainly add more

data for the optimization procedure and consequently provide a much better sensor characterization. This can motivate the change of f_y and its uncertainty from one pattern to the other. Another element in images that limits the collection of pixels is the shape of the car, which is fixed frame after frame occluding some useful key points for parameter estimation. In addition to this, the camera calibration procedure suffers the mapping of patterns whose normal vector to the plane forms an angle greater than 45° with respect to the camera optical axes. This causes the final camera orientation to have some pitch angle value, but in reality, its real orientation is not known.

Generally speaking, issues emerged due to bad dataset selection and wrong methodology, but it is quite evident that such image analyses, with this definition, without any possible element to be measured apart from the road and with world points correspondences affected by uncertainty, introduce relevant limitations to the problem. In the next chapter, a possible alternative that can overcome part of these problems by significantly increasing the dataset is presented, together with some other possible variants and future works related.

Chapter 5

State estimation via curve matching

This chapter focuses on the second possible methodology with which is possible to estimate the pose and the trajectory of the vehicle and try to overcome the issues described in the last section of the previous chapter. The method focuses on the usage of telemetry of the car for a rough estimate of the extrinsic camera parameters. Given a set of frames and the associated telemetry data as an initial guess, the mapped points on the road borders should be projected in such a way that the back projection from the camera image plane to the camera reference frame of the same pattern coincides. This method implements lateral road lines as a pattern to be detected since these elements can be considered a source of a huge quantity of matching points. Their coordinates are mapped along a part of a circuit and can be detected on the image as well by some computer vision algorithm implementations. The matching between curves defined by pixels and curves defined by world points is provided by proper intrinsic and extrinsic parameters. The difference with the previous method is that in this case there is an initial guess of car position and orientation and, in turn, of the camera. Moreover, it is possible to set some constraints on the parameters, like the maximum height of the camera or its pitch angle, and it is possible to collect a fair high number of points and images in an automatic procedure. This reduces the data collection effort and it should be able to reduce the variance associated with the optimization variables. In general, the role of the variables and the general camera model is more explicit than before. For the moment, the method can be considered as an initial stage for possible state estimation since further improvements have to be made in order to get results close to reality. In the first part, the theoretical aspects of the overall procedure will be pointed out, explaining the elements to understand the various parts of the state estimation. Then, the resolution of the optimization problem is highlighted, with the initial guesses and the minimization procedure. Final issues and possible improvements are discussed.

5.1 General procedure algorithm

The parameter identification is performed by following a detailed procedure or algorithm. This allows one to fuse the telemetry data collected with the image information available. Initial data consists of the following: the telemetry data, which are Cartesian coordinates of the vehicle's GPS (probably projected on a projection line) and a video stream from which one can collect the video time steps of interest with starting reference time given by the beginning of the recorded launched lap. For each time instant, it is also possible to collect the associated video frame. Given the dataset, it is possible to estimate the state per each frame as an initial guess. The time the car approaches the track of interest is known by the video and its time sequence. Time telemetry data doesn't have the same interval as the video timeline, so a linear interpolation must be performed considering the time interval which includes the video time instant of interest. The car's possible position is then used to search for a possible yaw angle. This is provided by a clothoid list of the circuit of interest describing the midline of the track. The search of the nearest point of the fitted clothoid to the estimated position guesses returns the arc length and angle associated. From that, one can search the right and left border points of interest and associate a yaw angle guess to the car. This leads to the build of an initial roto-translation matrix from the ground reference frame to the camera reference frame. On the other hand, from image analysis, one can extract the borders seen by the camera. A Canny edge detection algorithm and a line policy selection can help detect the borders in a specified region of interest. The detected pixels, back-projected in the camera reference frames, must match the world points seen by the camera. This matching is provided by proper camera parameters, given as initial guess, and extrinsic parameters as well. The algorithm 1 shows the detailed scheme for parameter identification.

5.2 Theoretical elements for dataset extraction

The following methods are involved in the data collection of this procedure, namely lines detection via computer vision algorithms and clothoid fitting. Having a general knowledge of them helps understand the procedure described above.

5.2.1 Edge detection via Canny algorithm and selection

The Canny edge detection algorithm is a well-known technique in computer vision for image edge extraction. The core is to exploit the change of the gradient in the image area, thresholding the borders via weak and strong edges and returning a binary image. The Canny algorithm process is defined by following these steps:

- Apply Gaussian filter to smooth the image in order to remove the noise
- Find the intensity gradients of the image

Algorithm 1 Camera parameters identification procedure

Require: Set of images from video frames I
Require: Clothoid lists R, L and M
Require: Video time steps t_i
Require: Image feature extractor procedure $borders(\cdot)$
Require: Telemetry tuple $T = \{t_{tlm}, x_{tlm}, y_{tlm}, v_{tlm}\}$
Require: Initial guess of camera intrinsic matrix K_g
  while Frame $k \in$ Set I **do**
    Extract frame time step t_k
    Search the time interval t^k_{tlm} inside t_{tlm} such that $t_k \in t^k_{tlm} = [t_{tlm,lb}, t_{tlm,ub}]_k$
    Count the number of times n the interval t^k_{tlm} is called
    Estimate x position: $\hat{x}^k_{tlm} = x_{tlm}(t_{tlm,lb}) + \frac{x_{tlm}(t_{tlm,ub}) - x_{tlm}(t_{tlm,lb})}{(t_{tlm,ub}) - (t_{tlm,lb})} \cdot (t_i - t_{tlm,lb})$
    Estimate y position: $\hat{y}^k_{tlm} = y_{tlm}(t_{tlm,lb}) + \frac{y_{tlm}(t_{tlm,ub}) - y_{tlm}(t_{tlm,lb})}{(t_{tlm,ub}) - (t_{tlm,lb})} \cdot (t_i - t_{tlm,lb})$
    Call the mean clothoid $M(\hat{x}^k_{tlm}, \hat{y}^k_{tlm}) = s^k_{tlm}, \psi^k_{tlm}$
    Define ROI for k
    Extract features $F = borders(k, ROI)$
    Project F with respect to camera frame: $K_g^{-1}F$
    Estimate the extrinsic matrix $[R|t]_g(\hat{x}^k_{tlm}, \hat{y}^k_{tlm}, \psi^k_{tlm})$
    Project $K_g^{-1}F$ in ground reference frame via $[R|t]_g^{-1}$ getting $\{x^w_L, y^w_L\}$ and $\{x^w_R, y^w_R\}$
    Fit the clothoid passing through $\{x^w_L, y^w_L\}$ and $\{x^w_R, y^w_R\}$
    sample the points after fitting $(\bar{x}^w, \bar{y}^w)^R$, $(\bar{x}^w, \bar{y}^w)^L$
    Call $M(s^k_{tlm} + s_{offset}) = (X^R, Y^R.X^L, Y^L)$, fit the obtained points and sample
    Calculate the error for frame k $e_k = \sum_i (X^R - x^w_R)^2 + (X^L - x^w_L) + (Y^R - y^w_R) + (Y^L - y^w_L)$
  end while
  Minimize the global error $J = \sum_k e_k$

- Apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection
- Apply double threshold to determine potential edges
- Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

Gaussian filtering is first applied to remove image noise by applying a convolution filter on it. Edge detection is performed by analyzing all possible gradient directions and can be performed through other filtering techniques (Sobel, Roberts or Prewitt). To account for remaining spurious responses, it is essential to filter out edge pixels with a weak gradient value and preserve edge pixels with a high gradient value. This is accomplished by selecting high and low threshold values. If an edge pixel's gradient value is higher than the high threshold value, it is marked as a strong edge pixel. If an edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value, it is marked as a weak edge pixel. If an edge pixel's gradient value is smaller than the low threshold value, it will be suppressed. The two threshold values are empirically determined and their definition will depend on the content of a given input image.

5.2.2 Clothoid curves and fitting

Clothoids are curves whose curvature changes linearly with its curve length (the curvature of a circular curve is equal to the reciprocal of the radius). They are also commonly referred to as Euler spirals or Cornu spirals. The usage of clothoids can have various advantages with respect to the classical polynomial fitting procedure, for example in scenarios where sharp edges or discontinuities are present. The principle of linear variation of the curvature of the transition curve between a tangent and a circular curve defines the geometry of the Euler spiral:

- Its curvature begins with zero at the straight section (the tangent) and increases linearly with its curve length
- Where the Euler spiral meets the circular curve, its curvature becomes equal to that of the latter

A general clothoid in parametric form must satisfy the following differential equations:

$$x'(s) = cos\theta(s) \tag{5.1}$$

$$y'(s) = sin\theta(s) \tag{5.2}$$

$$\theta'(s) = k(s) \tag{5.3}$$

The curvature is a linear function of s, namely $k(s) = \kappa_0 + \kappa'(s)$.

The solution of equations 5.1, 5.2 and 5.3 is

$$x(s) = x_0 + \int_0^s cos\Big(\frac{\kappa'\tau^2}{2} + \kappa_0\tau + \theta_0\Big)d\tau \tag{5.4}$$

$$y(s) = y_0 + \int_0^s cos\Big(\frac{\kappa'\tau^2}{2} + \kappa_0\tau + \theta_0\Big)d\tau \tag{5.5}$$

where (x_0, y_0) is the base point where the clothoid originates, θ_0 and κ_0 are respectively the angle and the curvature at the base point, κ is the curvature change rate or sharpness, s is the curvilinear abscissa. Notice that $\theta(s)$ and $\kappa(s)$ are, respectively, the angle and the curvature at the abscissa s. In the presented application, clothoids are used to interpolate points and create splines of clothoids. There are two ways of interpolation using clothoids, the G^1 and G^2 Hermite interpolation problems. Let's see the second one, which will be used during the procedure.

Definition. *(G^2 Hermite interpolation problem) Given two points (x_0, y_0) and (x_1, y_1), two angles θ_0 and θ_1 and two curvatures κ_0 and κ_1, the G^2 Hermite Interpolation Problem with clothoids asks to find the solution of*

$$G(s_k) = \mathbf{p}_k$$
$$\lim_{s\to s_k^+} G'(s) = \lim_{s\to s_k^-} G'(s)$$
$$\lim_{s\to s_k^+} G''(s) = \lim_{s\to s_k^-} G''(s)$$

with the additional constraints on the curvature:

$$k'(s) = u(s)$$
$$k(0) = \kappa_0$$
$$k(L) = \kappa_1$$

where $L > 0$ is the length of the curve, $u(s)$ is a piecewise constant function to be determined that can be interpreted as a control variable.

One implementation of clothoid splines is the search for the nearest point on a curve from a point of interest. If the point of interest is an initial guess position for the car and camera, it's possible to extract the distance of that point from the clothoid representing the midline and find the minimum one with the corresponding point on the clothoid. The calculation of the distance of a point from a clothoid is provided by Bertolazzi and Frego in [2].

5.3 Ojective function definition and minimization

The goal of this procedure is to obtain the final states of the camera and, consequently, of the car. The parameters that define the states are the unknowns of the objective function to be minimized. What has to be achieved is the error minimization between corresponding points, where the points are the projected points from the world to the camera and the ones from the image to the camera. Let's see in detail the parts that define the objective function.

5.3.1 From world points to camera points

World points can be seen by the camera with the following sequence of matrices inverted:

$$\mathbf{M}_{CW} = \left(\left[\mathbf{R}_{Z,\psi^k} \mid \mathbf{t}_k \right] \left[\mathbf{R}_{Y,\pi/2} \mid \mathbf{0} \right] \left[\mathbf{R}_{Z,-\pi/2} \mid \mathbf{0} \right] \left[\mathbf{R}_{X,\theta_{cam}} \mid \mathbf{0} \right] \right)^{-1} \quad (5.6)$$

where

- $\left[\mathbf{R}_{Z,\psi^k} \mid \mathbf{t}_k \right] = \begin{bmatrix} cos(\psi^k) & -sin(\psi^k) & 0 & x^k \\ sin(\psi^k) & cos(\psi^k) & 0 & y^k \\ 0 & 0 & 1 & z_{cam} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and represents the translation of the camera by a quantity given by $\mathbf{t}_k$, with fixed height and the other coordinates changing frame per frame, and a rotation around the z-axes of a yaw angle ψ^k changing frame per frame
- $\left[\mathbf{R}_{Y,\pi/2} \mid \mathbf{0} \right]$ and $\left[\mathbf{R}_{Z,-\pi/2} \mid \mathbf{0} \right]$ are needed so that the z-camera axes points out of the lens, the y-axes points downwards and the x-axes points rightwards with respect to the camera
- $\left[\mathbf{R}_{X,\theta_{cam}} \mid \mathbf{0} \right]$ introduces a small pitch angle tilt of the camera with respect to the road

As an initial guess for ψ^k, x^k and y^k the estimated position of the car can be considered, x^k_{tlm} and y^k_{tlm}, with respect to the track based on telemetry data and on a midline clothoid model of the circuit. The linear interpolation between time sampling instances, given the video frame rate, provides an estimate of the position of the car per frame. Then, given the midline clothoid of the circuit, it is possible to calculate the nearest point to the estimated position and find a guess for the yaw angle ψ^k_{tlm}. On the other hand, z_{cam} and θ_{cam} are constant at every frame. The height can be assumed to be no more than 1m since F1 cars cannot be higher than that while the pitch can be a few degrees as an initial guess. These guess values will be denoted as $z_{cam,g}$ and $\theta_{cam,g}$.

The unknowns for this part of the optimization will be the deviation from the guess parameters, so that for a common frame k

$$\psi^k = \psi^k_{tlm} + \delta_{\psi^k}$$

$$x^k = x^k_{tlm} + \delta_{x^k}$$

$$y^k = y^k_{tlm} + \delta_{y^k}$$

$$z_{cam} = z_{cam,g} + \delta_{z_{cam}}$$

$$\theta_{cam} = \theta_{cam,g} + \delta_{\theta_{cam}}$$

with the errors in the form $\delta_{(\cdot)}$ are initially set to 0.

The transformation to be performed from world reference frame to camera reference frame is thus defined in the following way

$$\mathbf{x}_{cam} = \mathbf{M}_{CW} \cdot \mathbf{x}_{world} \tag{5.7}$$

with $\mathbf{M}_{CW}$ defined previously. The vector $\mathbf{x}_{world}$ expresses the world points of the right and left border of the circuit collected at a given frame. Since all points can be approximately considered as lying on a plane the z coordinate can be set to 0, leading to the reduced expression

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}^{camera} = \tilde{\mathbf{M}}_{CW} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}^{world} \tag{5.8}$$

Since $Z^{world} = 0$ the 3rd column of $\mathbf{M}_{CW}$ can be truncated becoming $\tilde{\mathbf{M}}_{\mathbf{CW}}$. The final expression of 5.8 provides a new set of points that can be used for dataset creation. Specifically, once the plane of interest is defined the clothoid passing through the points of that plane allows to get an infinite set of points that can be sampled at a fixed distance. So, the final dataset for each frame to be used inside the minimization procedure is the set of points lying on a clothoid C with distance $s = \ell$ between one couple of samples and the following one.

This procedure has to be performed both for the right and left border providing the sampled points

- $\{P_i = (\hat{x}^R_i, \hat{y}^R_i) | P_i \in C^R, d(P_i, P_{i+1}) = \ell, i = N\}$
- $\{P_i = (\hat{x}^L_i, \hat{y}^L_i) | P_i \in C^L, d(P_i, P_{i+1}) = \ell, i = M\}$

with C^R the clothoid passing through the points P_i on the right side, C^L the clothoid passing through the points P_i on the left side, $d(\cdot)$ denoting the distance on the clothoid between two points and N and M being the points that satisfy the requirements inside one clothoid arc on the right and left side respectively.

5.3.2 From image points to camera points

The second part of the objective function is the projection of image points in a 3D real plane with respect to the camera reference frame. Image points can be back-projected on another plane by inverting the intrinsic matrix K. This provides a set of points in the camera xy plane. Consider also the fact that camera lenses affect the projection of rays from the real points in 3D to image points in 2D. So a correction has to be performed. In this case, for simplicity, it is possible to consider only the effect of radial distortion. So, the process of pixel inversion is given first by the multiplication between the inverse of matrix K and the array containing the pixel values times a scale factor. Then the resulting array is normalized by the polynomial modelling the radial distortion coefficient. In other words

$$\begin{cases} x_n = w \cdot \frac{u - o_x}{f_x} \\ y_n = w \cdot \frac{v - o_y}{f_y} \end{cases} \tag{5.9}$$

provides the back projection of pixels. Then

$$\begin{cases} x_{bp} = \frac{x_n}{1 + k_1 \cdot r^2 + k_2 \cdot r^4} \\ y_{bp} = \frac{y_n}{1 + k_1 \cdot r^2 + k_2 \cdot r^4} \\ r^2 = x_n^2 + y_n^2 \end{cases} \tag{5.10}$$

undistorts the pixels. Coefficients k_1 and k_2 are initially set to 0 as an initial guess. Also in this case points are sampled along a clothoid that fits the projected lanes. If the length of the sample is set to $s = \ell$ as before the set of left and right points to be sampled will be:

- $\{p_i = (\tilde{x}^R_{bp,i}, \tilde{y}^R_{bp,i}) | p_i \in C^R_p, d(p_i, p_{i+1}) = \ell, i = N\}$
- $\{p_i = (\tilde{x}^L_{bp,i}, \tilde{y}^L_{bp,i}) | p_i \in C^L_p, d(p_i, p_{i+1}) = \ell, i = M\}$

where the clothoids passing through the points will be C^R_p and C^L_p.

5.3.3 Final objective function

Once the procedure for point extraction is defined the final objective function is given by

$$J = \sum_{k=1}^{N_f} \Big(\sum_{i=1}^{N_k} (\tilde{x}_{bp,i}^{R} - \hat{x}_{i}^{R})^2 + (\tilde{y}_{bp,i}^{R} - \hat{y}_{i}^{R})^2 + \sum_{i=1}^{M_k} (\tilde{x}_{bp,i}^{L} - \hat{x}_{i}^{L})^2 + (\tilde{y}_{bp,i}^{L} - \hat{y}_{i}^{L})^2 \Big) \quad (5.11)$$

where N_f is the number of frames analyzed, N_k is the minimum between the number of samples for the right side from world to camera and the samples from image to camera and M_k is the minimum between the number of samples for the left side from world to camera and the samples from image to camera. Optimization variables should provide a meaningful result, so an upper and lower boundary should be provided. This means that the general formulation can be written as

$$\begin{aligned} &\min_{x} \quad J(x) \\ &\text{s.t.} \quad lb \leq x \leq ub \end{aligned} \quad (5.12)$$

where lb and ub express the lower and upper bounds of the variables of interest. The unknowns are embedded in x which accounts for 6 intrinsic camera parameters, 2 fixed extrinsic parameters and $3 \times N_f$ extrinsic parameters that vary frame per frame.

5.4 Implementation

In this section, the practical implementation of the proposed solution is shown. The possible software, data sources and possible implementation of the parts described above will be pointed out.

5.4.1 Telemetry data extraction, video source, and circuit data

First of all, it is necessary to provide the circuit of interest and the car telemetry for a particular session. Again, one can focus on the Monza circuit and on Charles Leclerc's qualifying session of 2022. As presented in the previous chapter, videos can be taken from the "F1TV" website, but there are also some of them on YouTube with slightly higher quality. One can verify if a higher video resolution can help determine the parameters, although it is still compressed with all the issues associated. The last observed video regards the Q3 qualifying session of 2022 in Monza so it is necessary to search for the proper telemetry data.

For this purpose, the "Fast F1" Python library can be useful since it provides all telemetry data for all Gran Prix sessions in a season and also for past seasons. Data have been captured during racing events using the official data service "F1 live timing". So by means of a Python script it is possible to search for the proper qualifying session and for the proper driver. Telemetry data include speed, lap timing and a set of Cartesian coordinates that probably refer to a projection of the real car position on an imaginary line of projection. This indeed should provide

the approximate track position of a driver for TV broadcasters, with no need to be extremely precise. By the way, it can still be a good set of data to start with as an initial guess. A ".csv" file can be exported from a Python script including timing, speed and Cartesian coordinates of the car.

From Google Earth, it is possible to collect a fine set of midpoint data and measure the distance between the borders and midline points. A fitting clothoid can be built interpolating these midpoints and for every abscissa value of this curve, one can access the Cartesian coordinates of the left and right borders. After a proper translation between the reference systems of the Fast F1 telemetry data and the Google Earth world coordinates one can access the desired portion of the circuit, that is the first chicane for example.

5.4.2 Video and frame analysis

For what concerns the video analysis, a Matlab script can be implemented for video reading. Matlab video analysis provides the possibility to capture the time instances per frame and other related data like its frame rate. Knowing the frame at which the car crosses the starting line it is possible to extract the video time instances at which the car approaches the circuit track of interest. Consequently, a vector of instances can be saved. The corresponding frames show the car approaching and passing through the first chicane in Monza and a total of 135 frames describe the car's motion in that portion of the track.

Then, these frames have to be processed in order to extract the features of interest. These are the pixels separating the white line of the circuit lanes from the grey asphalt. The Canny Edge detection algorithm can help extract the borders, with 0.16 as the lower threshold and 0.32 as the upper threshold via Matlab implementation, but this is not the only part to be done.

Indeed, once the Canny Edge detection is applied, what is provided is groups of pixels connected. Two of them define part of the border we are searching for. In order to easily find the points of interest, one can search in a region of interest of the picture defined as the set of pixels between $y = 290$ and $y = 540$, that is the area above the shape of the car and a bit lower than the horizon. The ROI defined in this way is split into a left and a right side as the most probable regions where left and right road lanes can be detected. This first filtering removes the unnecessary pixel detection of the foreground. Moreover, each sub-ROI is split into an upper and lower part since the lane is more likely to spread from the bottom right or left side of the image towards the centre due to perspective. As a selection policy for border detection, in the sub-sub-ROI, the group of pixels showing the longest path has to be selected. Moreover, the sub-sub-ROI is defined as a percentage of the split original ROI, for the moment set to 70%. If a group belongs to a bigger one in the left or right sub-ROI, then select the group on the left or right side accordingly. This procedure provides the lanes in a very specific region such that two curves can be detected. The policy works quite well for all 135 frames but some exceptions may occur. Due to poor road lane definition, tyre signs on the lanes and outdoor

illumination, Canny Edge detection may find borders with short lines. This causes the procedure to find pixel groups that do not belong to lanes but to other elements showing a sharp change in the threshold. Another issue creating false positives is the union between lines of pixels of different elements. This can be due to the fact that pixel features remain the same even though they belong to different elements. Another aspect to consider is the presence of the F1 car. In lane detection processes in the automotive section, the car shape is almost absent, allowing the user to entirely see the road. In this case, the front part of the car covers a great percentage of the image area, not providing a complete detection of the pixels in that part of the sensor. The ROI selection is performed also for excluding car contours detection, which can interfere with road lines adding unusual shapes. Canny edge detection was applied on the whole image, but from the binary mask obtained, the edges falling inside the car shape area are excluded. A special mask is created via manual segmentation, following the car shape profile. Although it is an approximation (tyre orientation and antenna vibration may cause some parts to exit the car mask) many of the unwanted lines are removed. Figure 5.1 shows an example of Canny Edge detection application while Figure 5.2 shows only the final border lines.

Figure 5.1: Canny edge detection application

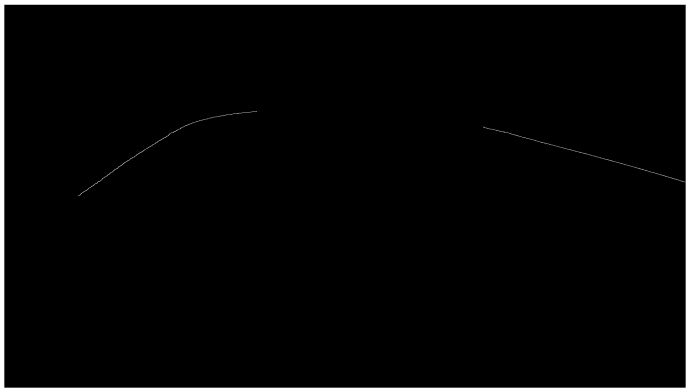

Figure 5.2: Example of final borders extracted

5.4.3 Raw data creation and world points sampling

A raw data structure can be built in Matlab, containing the estimated position per frame knowing the time instances of the recorded video and the time instances of the telemetry data. The estimated position along x and y is provided by the algorithm 1. Moreover, the same algorithm defines the procedure, per each frame, for yaw angle estimation. This is achieved by calling the mean clothoid in the nearest point to the estimated position for a given frame. These are the main data to collect and that will be used in the optimization procedure as initial guesses.

Another point to consider is the collection of world points. In the previous subsection, it has been mentioned the fact that pixels are collected within a region of interest. This ROI has a world correspondence. Knowing where the lower ROI boundary crosses the car shape on the image and where the upper limit of the ROI crosses the road, one can project some rays going from the t-shape camera to the ground passing through the upper part of the front wheels. Knowing the mean dimensions of an F1 car it is possible to estimate a lower boundary of about 4.67 m from the camera and an additional 17.50 m length from this lower bound. Inside these boundaries, frame after frame, one can estimate the proper world points to collect.

5.4.4 Optimization procedure

The cost function can be solved with a constrained minimization solver. Indeed, it is possible to provide physical limits to the variables of the optimization. For example, the variable z_{cam} cannot be higher than 1 m and at the same time, it can't be lower than the roll bar. The same can be thought for other optimization variables, from the position and orientation delta values to the camera pitch angle. To solve the minimization procedure, a Matlab implementation can be defined in the following way:

- first, solve the problem by finding an initial guess; this will find an initial value of the optimization parameters with a relatively low-cost function value. This procedure can be performed by applying the Matlab functions "patternsearch" or "particleswarm" to the cost function.
- the solution provided by the mentioned solvers can be refined by the "fmincon" application with upper and lower bounds

Since the process takes a long time, and some border feature detection may fail or can be not enough precise, it is suggested to take fewer frames than the original 135, avoiding the ones that can cause problems. The number of points collected should in any case be sufficient for a proper determination of the parameters involved.

For what concerns the optimization variables, some of them are provided as variations to be added to the nominal value. This regards extrinsic parameters especially. The initial guess values for a first-tentative optimization procedure are reported in table 5.1. Upper and lower bounds can be provided for each variable to give a first

Table 5.1: Initial guess values for first-tentative optimization

Optimization parameters	Initial guess
f_x [pixels]	1500.0000
f_y [pixels]	1900.0000
o_x [pixels]	960.0000
o_y [pixels]	540.0000
k_1	0.0000
k_2	0.0000
w	1
δ_z [m]	0
δ_θ [rad]	$1.0000 \cdot 10^{-7}$
$\delta_{x,k}$ [m]	0
$\delta_{y,k}$ [m]	0
$\delta_{\psi,k}$ [rad]	0

Table 5.2: Upper and lower bounds for optimization variables

Optimization parameters	Lower bound	Upper bound
f_x [pixels]	$-\infty$	∞
f_y [pixels]	$-\infty$	∞
o_x [pixels]	$-\infty$	∞
o_y [pixels]	$-\infty$	∞
k_1	$-\infty$	∞
k_2	$-\infty$	∞
δ_θ [rad]	0	$\pi/10$
δ_z [m]	-0.05	0.03
w	1	1
$\delta_{x,k}$ [m]	-10	10
$\delta_{y,k}$ [m]	-10	10
$\delta_{\psi,k}$ [rad]	$-\pi/5$	$\pi/5$

constraint to the final result, as reported in table 5.2. Notice that the scale factor is formally set to 1 since during some optimizations the change of this parameter provided some issues. Moreover, an initial matching guess between curves is provided by $w = 1$.

5.5 Results, comments and possible modifications

For the moment, the first results of this proposed method are obtained with a slight modification of the described procedure. In the above lines, it was mentioned to take border lines and pixel lines projected in camera reference frames, fit them with a clohoid curve each and sample every $s = \ell$, where $\ell = 1$ in this case. This is computationally expensive and gives some issues during optimization, especially when dealing with pixel fitting. The resulting transformation of them produces sharp displacements between them and an undersampling is necessary. Anyway,

Table 5.3: Some optimization variables obtained

Optimization parameters	Nominal value
f_x [pixels]	1499.8124
f_y [pixels]	2024.0358
o_x [pixels]	958.9216
o_y [pixels]	539.6002
k_1	-1.8151
k_2	7.3345
w	1

this didn't solve the problem at all. So, in order to provide some initial quick results the following modifications have been done:

- call only the mean clothoid
- search every $s = 1$ on the mean clothoid the points to collect
- find the points on the right and left borders with the proper shift associated with the curvilinear abscissa obtained
- transform the points in the camera reference frame
- transform the pixels in the camera reference frame
- search the matching points of pixels: take the distance between world points projected and find the same distance on the pixel sequence with respect to the camera in relation to the total length of the sequence
- apply the minimization procedure considering the new sampled pixels and the world points transformed

This procedure provides some first-tentative results that have to be refined in future works. With $s = 1$ and 81 frames analyzed the function value after minimization is 141,9871. The function is not exactly 0 and it seems that every frame provides a single function value contribution of about 1,75. If one considers the function value per point in each frame it means that one matching point provides roughly 0,1 as a contribution to the overall objective function. Some optimisation variables obtained are reported in table 5.3.

It is also possible to report how some relevant extrinsic parameters, like position and yaw angle changed after optimization. Figures 5.3 and 5.4 show the change of these parameters after a first tentative optimization. Figure 5.5 shows the projected curves after a first tentative minimization for a specific frame.

Moreover, an additional modification regards the projection of world points in the camera reference frame. Since in an image the depth information is lost, the matching is between the pixels projected back with $\mathbf{K}^{-1}$ and the points' x and y coordinates in the camera reference frame normalized by their z coordinate. This is justified both graphically since points that are far from the camera converge

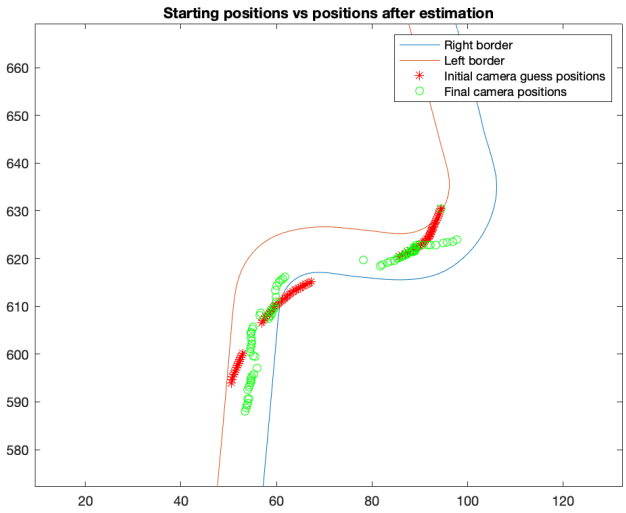

Figure 5.3: Final positions after optimization

Figure 5.4: Final yaw angles after optimization

towards the centre and mathematically by the pinhole camera model equations 3.1 and 3.2.

Comments on results

For the moment, results are partial and not complete as not all the frames have been analyzed. The objective function value is not enough close to zero even though the contribution of each point to the minimization function seems low. Additionally, optimization variables modelling the intrinsic parameters are totally different from the ones achieved in the previous sections. This confirms that previous models were not reliable enough and further refinements of these parameters have to be performed. Moreover, Figures 5.3 and 5.4 show a high change in the extrinsic parameters with respect to the initial guess. It seems that the position tends to move

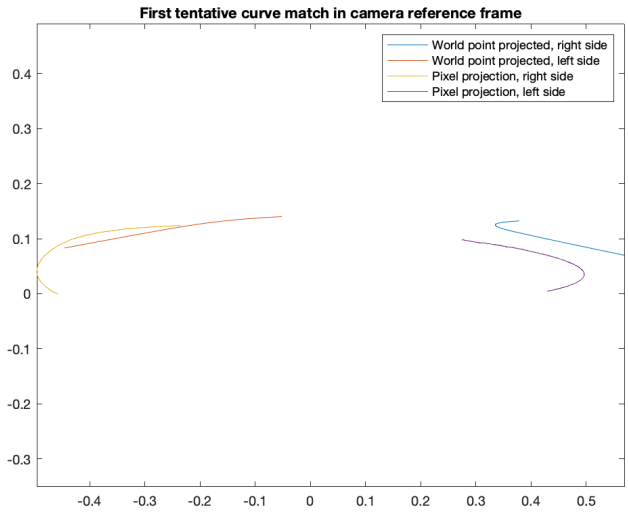

Figure 5.5: First tentative curve projection for frame 11

towards the midline value whereas the yaw angle shows an oscillatory behaviour towards the end of the chicane. An additional comment has to be done in Figure 5.5. Pixel curves projected in the camera still show high curvature due to lens distortion. This probably means that the values of k_1 and k_2 are not enough for performing a well undistortion process. Probably, a third component can help undistort the image.

These first-tentative results are thus probably due to the following reasons:

- Points selection: the points selection method may be inaccurate; using clothoids and searching points on them should provide a more precise method for points selection, but due to computational problems it was not possible to use it;
- Cost function construction: the overall procedure for the cost function computation may be revised. In particular, only 2 coefficients for radial distortion (or undistortion) have been used, but using three of them as well as an estimation of the skew and the tangential distortion can provide a better projection;
- initial guess values provide a local minimum and another value has to be found
- A better correspondence between ROI on the image and the ROI in the real world with upper and lower bounds has to be refined. A possible solution to the problem can be the introduction of these upper and lower bounds as additional optimization variables.
- an additional nonlinear constraint can help to limit the values of extrinsic parameters within certain limits and avoid certain behaviour. On the other hand, one has to take into consideration the fact that the function cost may not decrease or it can even increase, meaning that the global optimum values have not been reached.

All the listed hypotheses for problem correction can be revised in future works to

improve the methodology.

Possible modifications

For the moment, the procedure is a proposal from which further studies can be developed. Some issues are under investigation: clothoid curves applied to pixels are difficult to build due to the non-smoothness of the sample sequence. An undersampling technique has been performed to get a proper curve fitting, but the correct undersampling factor has to be properly tuned since it is not always working. This is one issue to be solved in future works, as it can provide a more coherent matching for error minimization. Another issue is the correct matching between world points samples and pixels samples detected inside the ROI. Indeed there is no guarantee that the world ROI is correct and a better search can be developed. For example, one can estimate the variation of upper and lower bounds as additional optimization variables to get the proper world points for the matching. In addition to this, some refinements can make the difference: reducing s for point choice can provide more fitting data and improve the results, as well as a refinement of the undistortion model. Indeed only radial distortion with two coefficients has been considered, but adding one extra term or introducing the tangential distortion can help improve the results. As stated previously, a possible modification one can add is the addition of linear or non-linear inequality constraints on variables in order to limit possible oscillations and provide a coherent trajectory reconstruction.

Chapter 6

Conclusions

Several techniques have been analyzed or proposed for the simultaneous characterization of a camera sensor and for car position estimation. In the first part, an analysis regarding the classical world-image matching for intrinsic and extrinsic camera parameters estimation was explained. The matching pattern can give the intrinsic parameters which are used to define the extrinsic ones. These elements are also used for an estimation of the car's position thanks to the mounting between the camera and the vehicle. Moreover, according to the calibration approach used, extrinsic parameters have been computed apart or have been directly provided after calibration. This depends on how the matching is performed. If one focuses only on the pattern ignoring the true movement of the car and where points are really positioned in the global reference frame then the true position and camera orientations have to be recalculated. On the other hand, if the point disposition is captured following the car movement and the true disposition of points in the real world, then the extrinsic camera parameters are already provided after calibration. Two types of patterns have been observed: one is the starting grid, a pattern that occupies mainly the central portion of the image, while the other one is the sequence of kerbs in a chicane, that is shown in the image in various parts of the image sensor.

Calibration procedures with these two patterns provided not stable results. The change of calibration settings in the optimization procedure produced very different results of the intrinsic parameters and a change in their uncertainty associated. Moreover, the best values obtained for mean reprojection error are quite high with respect to state-of-the-art calibration procedure results. In the case of calibration using the starting grid pattern intrinsic parameters seemed to be relatively accurate but extrinsic parameters provided wrong camera positions. Using the chicane instead provided more uncertainty on the intrinsic parameters' values but much better camera localization with respect to the observed pattern. This difference can be addressed to the localization of points on the image sensor during calibration: in the case of starting grids, points were more concentrated in the central part of the image where distortion is lower but they represent high relative distances within the same frame. On the contrary, chicane kerbs appear mainly closer to the sensor with

fewer points per frame spread in various parts of the image. This provides a higher uncertainty in the characterization of the intrinsic parameters and, in particular, for what concerns the parameters modelling lens distortion. At the same time, the proximity of represented points with respect to the sensor provided better sensor localization than before. In general, stability is not granted due to a high change in results after the addition or subtraction of optimization parameters.

Possible reasons for the poor results obtained can be:

- the quality of images, which are taken from screen recording with an original source showing low definition and thus showing blocking artefacts or blurred features;
- the position of measurable points on the image, which is mostly present in the middle part of the sensor and certainly limits the sensor characterization
- the number of points and images involved that seem to be too low as the fitting procedure ”shifts” towards intrinsic parameters or extrinsic ones according to the case study

An alternative method for car position estimation and sensor characterization is defined in the last chapter. The new methodology aims at improving the results obtained with the previous methods, providing more robustness to the optimization procedure. The augmented set of points can indeed improve the parameter identification by collecting many samples of the lanes both in the images and in the world reference frame. For the moment, the proposed method has to be refined providing a better procedure for defining the cost function.

Bibliography

[1] Jean-Alix DAVID. *Visual Map-based Localization applied to Autonomous Vehicles*. 2015 (cit. on p. 5).

[2] Marco Frego and Enrico Bertolazzi. "On the Distance between a Point and a Clothoid Curve". In: *2018 European Control Conference (ECC)*. 2018, pp. 1–6. DOI: 10.23919/ECC.2018.8550554 (cit. on p. 40).

[3] Erwan Guillou et al. "Using vanishing points for camera calibration and coarse 3D reconstruction from a single image". In: *The Visual Computer* 16 (2000), pp. 396–410 (cit. on p. 7).

[4] J. Heikkila and O. Silven. "A four-step camera calibration procedure with implicit image correction". In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997, pp. 1106–1112. DOI: 10.1109/CVPR.1997.609468 (cit. on p. 16).

[5] Manuel Lopez et al. "Deep Single Image Camera Calibration With Radial Distortion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on p. 7).

[6] Rahmad Sadli et al. "Map-matching-based localization using camera and low-cost GPS for lane-level accuracy". In: *Sensors* 22.7 (2022), p. 2434 (cit. on p. 5).

[7] Yihong Wu, Fulin Tang, and Heping Li. "Image-based camera localization: an overview". In: *Visual Computing for Industry, Biomedicine, and Art* 1.1 (2018), pp. 1–13 (cit. on p. 6).

[8] Z. Zhang. "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334. DOI: 10.1109/34.888718 (cit. on pp. 3, 6, 16).