

NPL REPORT MS 30

CAMERA AND LIDAR SENSOR MODELS FOR AUTONOMOUS VEHICLES

RAHUL KHATRY AND ANDREW THOMPSON

APRIL 2021

Camera and Lidar sensor models for autonomous vehicles

Rahul Khatry and Andrew Thompson
Data Science

ABSTRACT

Camera and Lidar are important sensors in the automated vehicle sensing stack. In order to ensure safety of automated vehicles we need to understand the working of these sensors and verify their functioning in different edge cases so we can understand their characteristics in the operational design domain (ODD). In this study we will demonstrate how to apply the sensor modelling framework developed in [1] to model the functioning of these sensors.

© NPL Management Limited, 2021

ISSN 1754-2960

<https://doi.org/10.47120/npl.MS30>

National Physical Laboratory
Hampton Road, Teddington, Middlesex, TW11 0LW

This work was funded by the UK Government's Department for Business, Energy and Industrial Strategy (BEIS) through the UK's National Measurement System programmes

Extracts from this report may be reproduced provided the source is acknowledged and the extract is not taken out of context.

Approved on behalf of NPLML by
Louise Wright, Head of Science (Data Science).

CONTENTS

1	INTRODUCTION	1
2	CAMERA SENSOR	1
2.1	SENSOR FUNCTIONING	1
2.1.1	LENS FLARE	2
2.1.2	LENS DISTORTION	3
2.1.3	FOCUS BLUR.....	4
2.1.4	COMBINING THE EFFECTS.....	4
2.2	APPLYING THE SENSOR MODELLING FRAMEWORK	5
2.3	DISCUSSION	8
3	LIDAR SENSOR	9
3.1	SENSOR FUNCTIONING	9
3.2	APPLYING THE SENSOR MODELLING FRAMEWORK	10
3.3	LIDAR MEASUREMENTS AT NPL	12
3.4	DISCUSSION	12
4	SUMMARY	12
5	REFERENCES	13

1 Introduction

Understanding and modelling sensor characteristics is an important activity for ensuring autonomous vehicle safety. A typical sensing stack for an autonomous vehicle consists of Camera, Lidar, Radar, Sonar, GPS and IMU sensors. Camera and Lidar are important sensors in this sensing stack and are mainly used in the perception and localization functions. In order to ensure the safety of automated vehicles we need to understand the working of these sensors so we can understand their characteristics in the operational design domain (ODD). This ODD consists of the normal operating ranges of light conditions and weather conditions and also the edge cases where it becomes more difficult for the sensors to function properly. In order to check the working of the sensors in various conditions, the ability to simulate them and perform virtual testing can be of great help since it is costly to engineer the full range of scenarios physically. It is therefore also important to have reliable sensor models within such simulations.

In this study we demonstrate how to model the functioning of these sensors, and we use the sensor modelling framework to construct models of each of them, addressing several edge cases in the process.

Section 2 focuses on the camera sensor and Section 3 focuses on the Lidar sensor. Both sections have the same structure. We first introduce the sensor and give a brief description of the phenomena intended to be modelled in the report. We then follow the sensor modelling framework [12] in building models for these phenomena. We then illustrate the models by simulating them on appropriate datasets. We then conclude and briefly discuss areas for future work.

2 Camera Sensor

Cameras used in automated vehicles are a very important part of the sensing system and they are used for various purposes including pedestrian identification and sign detection [2]. These sensing activities are safety critical, and faults in performing these sensing activities properly could result in major accidents, such as the high-profile case of the Uber car crash [3]. Hence establishing the reliability of camera sensors through an understanding of how they operate is crucial.

2.1 Sensor Functioning

Modelling can either take a data driven or a physics based approach [1]. Because of the high degree of physical understanding around how cameras function, a physics based model is usually appropriate. The modelling process then involves determining the right mathematical equations and fitting parameters using available calibration data. In order to simulate a sensor we need to understand the various functioning parts and the different phenomena that occur in their operation. A digital camera consists of lenses, optical sensors and a digital image processor [4]. The light from the scene passes through the lens and interacts with the optical sensor which then captures an image signal. The signal is then processed through the digital image processor to provide the final output of the camera sensor. Some of the different phenomena adversely affecting image accuracy that can occur during this process of capturing images are lens flare, lens distortion, focus blur and motion blur. We next give a brief explanation of some of these effects.

2.1.1 LENS FLARE

Lens flare is caused by the interaction of light with the lenses. Lens flare can be understood by considering the process of reflection and internal reflection of the light passing through the lenses.

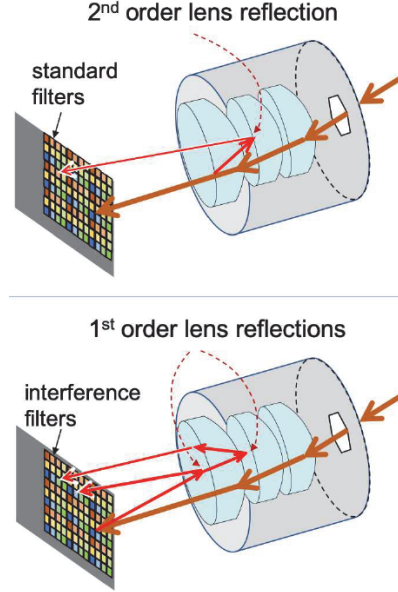


Figure 1 Primary and Secondary reflection inside lens causing lens flare [5]

Consider the case of a multi lens system as shown in Figure 1. The rays coming from the right (the scene) represent the primary beams which pass through the lenses. These primary beams are also reflected and refracted several times in primary and secondary internal reflections, and the reflected beams interact with the sensors as well. The beams with high intensity (e.g. direct rays from the sun) have a large enough first order and second order reflection component to be registered twice causing flare imprints on the image capture [5].

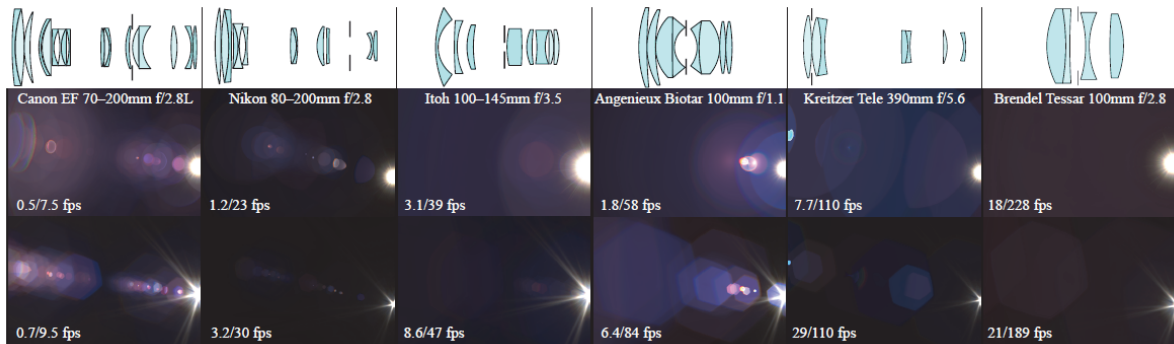


Figure 2 Lens flares patterns in different lens configurations [6]

Illustrations of these effects can be seen in Figure 2, which shows the flare components of lenses for different lens configurations. The output intensity of a single reflection flare component can be obtained using the approximation below [5].

$$F_{lens} \approx M(2M - 1)I_{source}R_{lens}^2$$

Here the output intensity F_{lens} depends on the number of lens elements M , the source intensity I_{source} and the reflectivity of the lens surface R_{lens} . The details of the reflective glare elements themselves can be simulated by ray tracing and integrating the appropriate rays over the sensor, which requires heavy computation [6]. An alternative, more computationally light, approach is to make use of flare templates, such as those shown in Figure 3 [7].

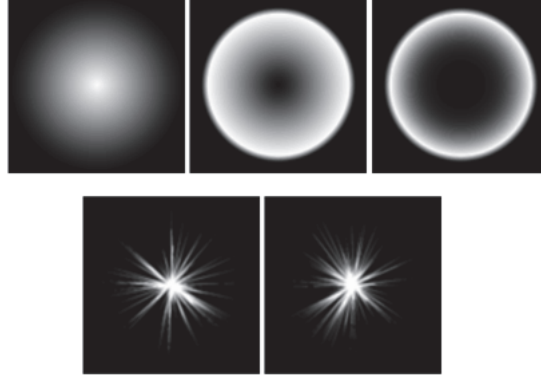


Figure 3 Different flare templates [7]

These flare templates can be superimposed onto images to obtain simplistic versions of lens flares obtained from the previous models. We have used this approach in the current study.

2.1.2 LENS DISTORTION

Another phenomenon that it is important to model is lens distortion. Lens distortion is caused by the curvature of the lens while capturing the images. Light is bent while passing through the lens and this occurs unevenly at different regions in the lens. This uneven bending of light for different regions of the captured scenery resulting in lens distortion. The lens distortion phenomenon is illustrated in Figure 4.

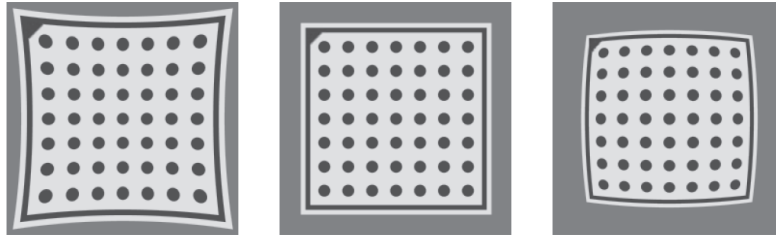


Figure 4 Lens distortion effect [8]

In [8], lens distortion is modelled using a polynomial model

$$\begin{aligned} x_u &= x_d + x_d(K_1 r_d^2 + K_2 r_d^4 + K_3 r_d^6) + 2P_1 x_d y_d + P_2(r_d^2 + 2x_d^2) \\ y_u &= y_d + y_d(K_1 r_d^2 + K_2 r_d^4 + K_3 r_d^6) + 2P_1 x_d y_d + P_2(r_d^2 + 2x_d^2). \end{aligned}$$

Here x_u, y_u are the undistorted coordinates of the centre of a given pixel and x_d, y_d are the distorted coordinates, and $r_d = \sqrt{x_d^2 + y_d^2}$ denotes the distance from the so-called principal point (the optical centre of the lens). The model includes several parameters: K_1, K_2, K_3 are the radial distortion parameters and P_1, P_2 are the decentering distortion parameters. Calibration testing, using known ground-truth calibration targets, is generally used to determine the parameters K_1, K_2, K_3 and P_1, P_2 [8].

Simplifications of the above model which ignore decentering distortion and ignore higher order terms in r are commonly used [9]. Neglecting terms above second order gives

$$\begin{aligned} x_u &= x_d(1 + K_1 r_d^2) \\ y_u &= y_d(1 + K_1 r_d^2). \end{aligned} \quad (1)$$

An alternative to this model was proposed in [10], namely

$$\begin{aligned} x_u &= x_d/(1 - K_1 r_d^2) \\ y_u &= y_d/(1 - K_1 r_d^2). \end{aligned}$$

It is argued in [10] that this model gives just as good an approximation, whilst making expressions for certain quantities of interest, such as the fundamental matrix of a camera, more tractable. It is shown in [9] that, for this model, it is possible to obtain a closed-form expression for r_d in terms of $r_u = \sqrt{x_u^2 + y_u^2}$, namely

$$r_d = \frac{-1 + \sqrt{1 + 4K_1 r_u^2}}{2K_1 r_u}.$$

We will demonstrate the use of the model given in (1) in this report. All the models described here can be used to simulate distortion effects and can also be used to invert the model and remove lens distortions from images as well.

2.1.3 FOCUS BLUR

The third phenomenon that we will look at is focus blur. When the lens is not able to focus on certain objects they appear blurred. Blurring is demonstrated in Figure 5 [11] where we can see that the light from an off-axis point is spread by the lens so that the rays do not pass through a single focal point and the camera sensor receives the light across several pixels rather than at a single point. Figure 5 shows effective bundling of the red rays from an object but inability to bundle the green rays.

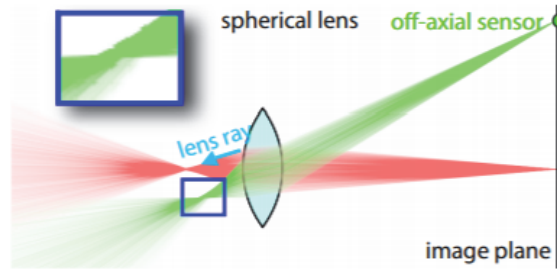


Figure 5 Image blurring effect [10]

This phenomenon can be simulated by applying Gaussian blur, in which the image is convolved with a 2D Gaussian blur kernel. Generally this phenomenon is corrected for in inbuilt processing before feeding the image to the next stages but some extreme cases like high flare and high speed moving targets may still lead to such distortions, and hence it might be useful to understand this phenomenon in order to deal with these edge case.

2.1.4 COMBINING THE EFFECTS

Combining all the phenomena explained above we can obtain the overall image output from a camera scene capture. Onboard camera post-processing can rectify a lot of these effects, but for the purpose of this study we are 'opening the box' and not simply assuming that post-processing has been done to rectify the outputs. Simulations can then replicate all these

phenomena and generate what we will get from the resultant image of a simulated scene. A flowchart of a model combining all these phenomena (and more) was given in [12] and is reproduced in Figure 6.

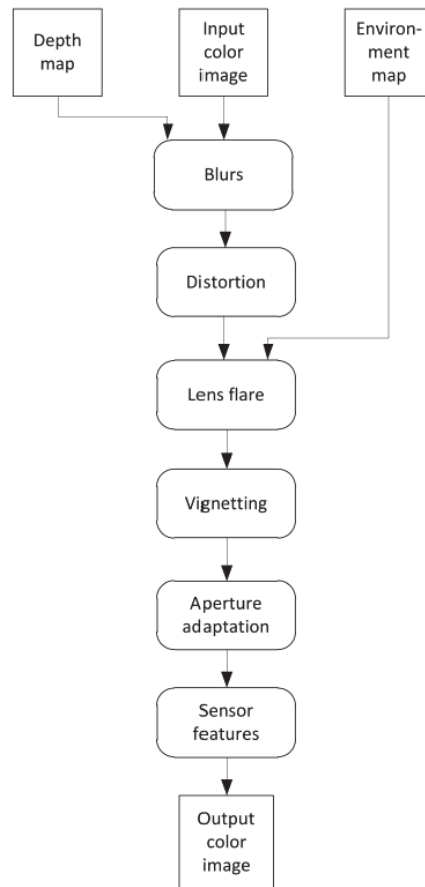


Figure 6 Camera distortion effects flowchart [12]

2.2 Applying the Sensor Modelling Framework

We apply the principles of the sensor modelling framework [1] to create the camera model. A good sensor model can be used to imitate the way a camera captures images in normal operating conditions as well as the edge cases of glaring light, bad weather and bad lighting.

To follow the sensor modelling framework we will go through the steps of the sensor modelling flowchart, which is reproduced in Figure 7. In the case of the physics based models considered for camera sensors, this process consists of a scoping exercise, followed by deciding on a model, and then the parameters of the model are fitted to desired results, and then finally an evaluation is carried out to check whether the model was able to produce the necessary outputs.

Step 1: Scoping exercise

We will first determine the effects we want to be able to capture by the sensor model. The effects we will be considering for this sensor model are

- 1) Lens flare
- 2) Lens distortion
- 3) Focus blur.

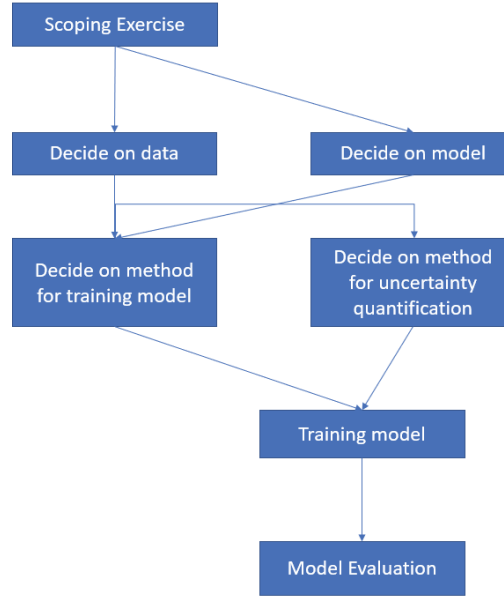


Figure 7 Sensor modelling Framework [1]

Step 2: Deciding on the model

The model we are planning to make consists of the following sets of equations for each of the respective phenomena being modelled.

Lens flare. The lens flare templates are applied to images and visual inspection of the results is used to find the one which fits closest to the observed flare behaviour.

Lens distortion. We use equation (1) to model radial distortion; see the description in Section 2.1.2. We describe a method for determining the distortion parameter K_1 in (1), given some distorted and undistorted images. Given the coordinates (x_u^i, y_u^i) of n reference points ($i = 1, \dots, n$) in an image before distortion and the coordinates (x_d^i, y_d^i) of the same reference points after distortion, minimising the least squares error

$$\sum_{i=1}^n \left[x_u^i - x_d^i \left(1 + K_1 (r_d^i)^2 \right) \right]^2 + \left[y_u^i - y_d^i \left(1 + K_1 (r_d^i)^2 \right) \right]^2$$

with respect to K_1 gives the closed-form expression for K_1 of

$$K_1 = \sum_{i=1}^n \frac{(r_d^i)^2 [x_d^i x_u^i + y_d^i y_u^i - (r_d^i)^2]}{(r_d^i)^6}.$$

Focus blur. A Gaussian kernel

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

is applied on the out of focus region of the image to obtain the distorted measurements of the image.

Step 3. Deciding on the data

The data we use for demonstrating the model consists of images which contain the respective edge cases and we seek to replicate the effect through the model equations.

Lens flare. We produce lens flare effects similar to a flared image using the flare templates approach described above and combining it with a clear image.

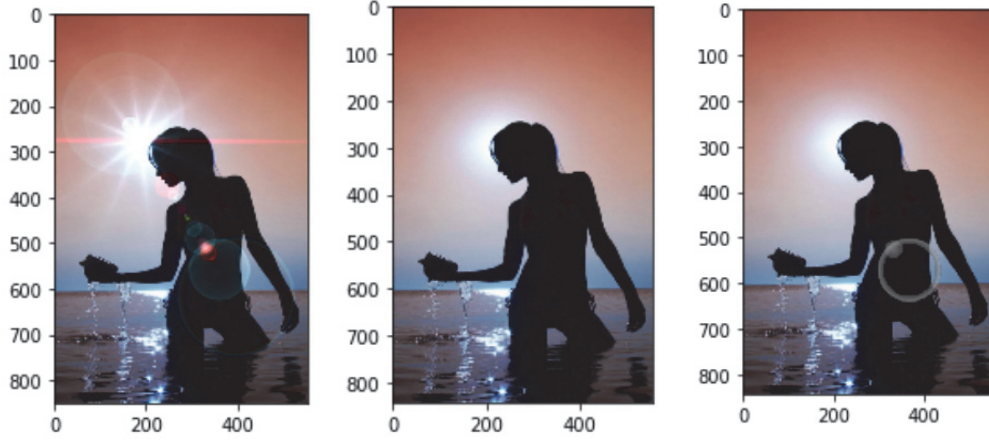


Figure 8 Flare (left) and clear image (middle) and simulated flare (right)

We can see in Figure 8 that we are able to produce flare images similar to the flared image on the left hand side and flare templates in the first row on right and left most in Figure 3. We apply the following pixelwise threshold

$$\text{if } i_{flare}(R/G/B) > 150 : i_{resultant} = 0.4 \times i_{flare} + 0.4 \times i_{image}$$

Lens distortion. We will take undistorted check image and distort them using the given model equations. The effect can be seen in Figure 9.

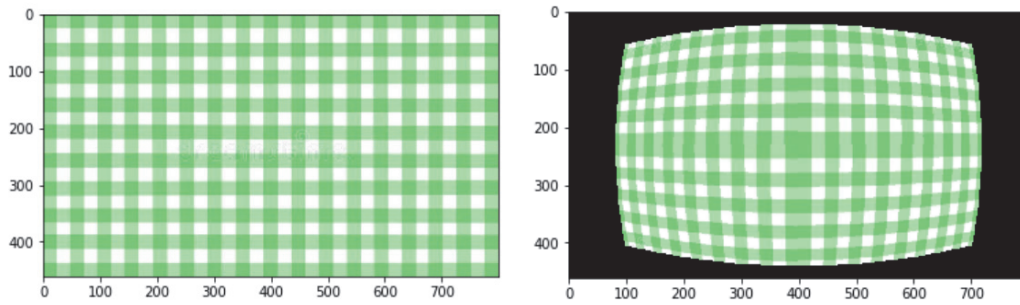


Figure 9 Lens distortion model (distorted and undistorted images)

We also take undistorted and distorted images of a scene. Then we replicate the distorted capture using the model equations to generate artificially distorted images. The effect can be seen in Figure 10. The leftmost image is the undistorted image, the middle image is the image obtained from a lens with distortion and the image on the right is produced by the sensor model trying to emulate the same effect.

In order to produce these distorted images we start with the undistorted image on the left. We then apply equation (1) to it with distortion factor of 1.4×10^{-6} and with principal point shifted

upwards by 50 pixels from the centre of the image. No information concerning the principal point was available, and so this choice was made after manual experimentation.

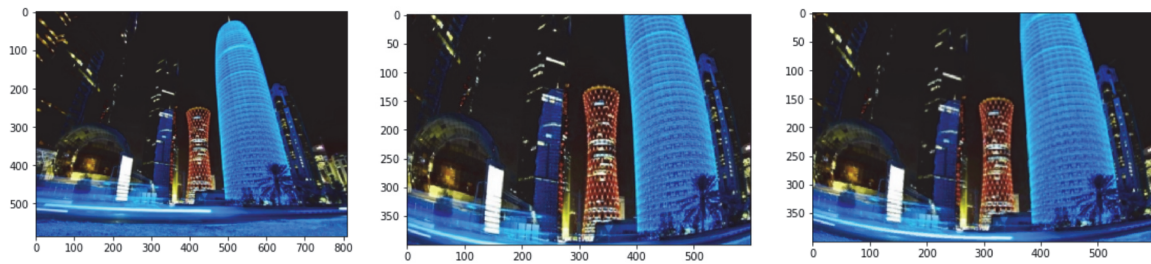


Figure 10 Lens distortion model used to simulate distorted images from undistorted images

Focus blur. To simulate focus blur we take a blurred image and a clear image and try to replicate the blurred image using our Gaussian blur effect.

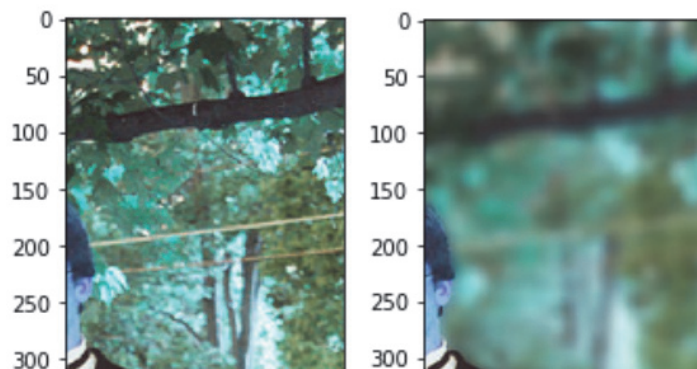


Figure 11 Clear and blurred images

An example of a clear image and a blurred image can be seen in Figure 11. We take a section of the two images and apply a Gaussian kernel to the clear image. The result can be seen in Figure 12.

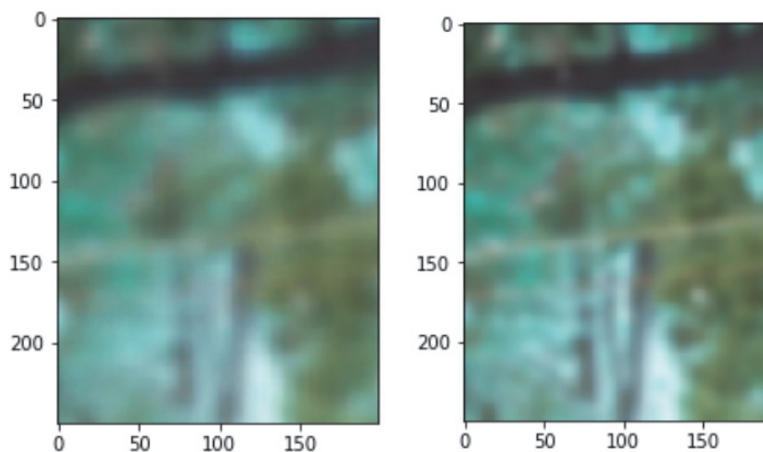


Figure 12 Blurred image (left) and Gaussian blur (right)

2.3 Discussion

We were able to programmatically produce effects that approximated the desired effects simulating the phenomena of interest. The model described above can hence be used to

simulate these edge cases. Further development to introduce more complex effects and realistic outputs and to combine it in an image capture process can be carried out as an extension of this study.

3 Lidar Sensor

Lidars are considered a very important part of the autonomous vehicle sensor suite in some sensor stacks e.g. Volvo [18], Waymo [19] etc. There are others like Tesla that are trying to steer away from using Lidar because of the cost of the sensor [20]. In conditions where cameras don't function properly, such as darkness, the Lidar becomes essential for being able to operate the automated vehicle.

3.1 Sensor Functioning

Lidar operates by projecting lasers and sensing the reflected beams. The phase of the reflected beam and the time difference between projecting the laser and obtaining the reflection are used to measure obstacle characteristics such as distance and colour [13]. The lasers are scanned across the scene to collect data about the volume surrounding the vehicle.

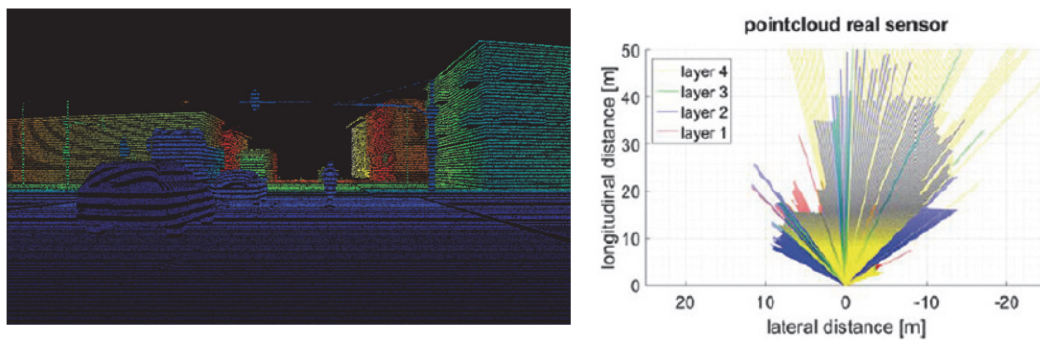


Figure 13 Lidar measurements [12]

The Lidar then forms an occupancy grid of the sensed obstacles which can then be used by the perception and localization modules. This is also used in further sensor data fusion and in forming global occupancy grid maps and path planning.

The measurement of any obstacle happens when the beam reflects off it and is read by the sensor. This phenomenon of being reflected and measured is best modelled probabilistically and several possible models have been proposed in the literature [14, 15]. The approach we will consider in this study is taken from [15] and involves constructing a forward model in which it is assumed that the probability of a beam being reflected decreases as the point to be measured moves further away. This effect occurs because the laser is scattered by the air (and particularly by the moisture in the air) so the intensity of the received reflected signal decreases with distance. The left-hand panel of Figure 15 shows three squares representing objects, and the right-hand panel depicts the detection probabilities of the two squares in the range of the sensor measurement.

The occupancy grid map is modelled as a log likelihood map where the log likelihood of an obstacle being present is updated by the measurements taken by the Lidar sensor. As the model runs it updates the log odds of the occupied cells and also updates the log odds of the unoccupied cells, as can be seen in Figure 16. The probabilistic approach also quantifies uncertainty due to signal strength of the presence of objects in cells as a natural by product. More precisely, the log odds of each cell is updated as

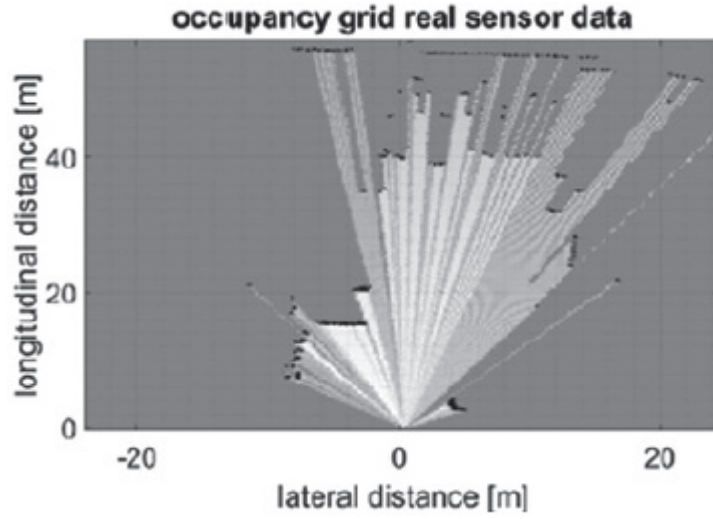


Figure 14 An example of an occupancy grid

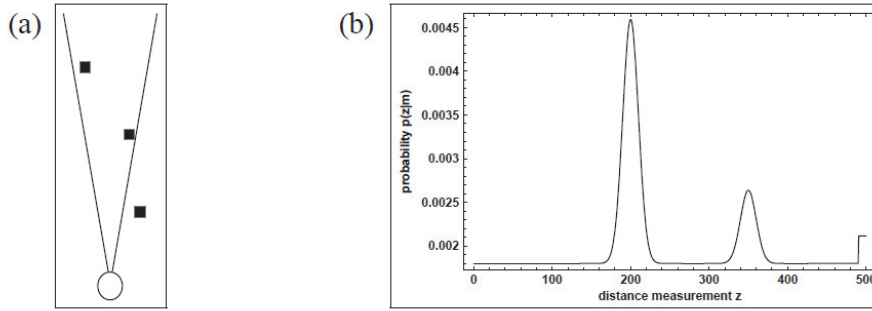


Figure 15 Forward model for probability of measurement [15]

$$\log odds = \log odds + \frac{\text{confidence}}{1 - \text{confidence}}$$

where $\text{confidence} = 0.9$ if an object is detected and $\text{confidence} = 0.3$ if not. These parameters can be chosen based upon an understanding of the physics. The probability of occupancy, here denoted occupancy , is then updated to be

$$\text{occupancy} = 1 - \frac{1}{e^{\log odds}}.$$

A separate dataset of Lidar measurements is used to understand the ray characteristics of a Lidar, which is then used to calibrate the probabilistic measurement model over the measurement range.

3.2 Applying the Sensor Modelling Framework

Similar to the camera model we will again be applying the principles of the sensor modelling framework [1] to create the Lidar model. A good sensor model can be used to imitate the process of capturing readings and forming an occupancy grid map from the measured readings. The capturing process is treated as a probabilistic process hence a reading is not considered definitive but is used to increase the probability of presence of an obstacle on the occupancy grid.

Step 1: Scoping exercise

We first determine the effects we want to be able to capture in the sensor model. The effects we consider for this sensor model are

- 1) Capturing the readings
- 2) Probabilistically formation of the occupancy grid

Step 2: Deciding on the model

The model is defined by the set of equations indicated in Section 3.1.

Step 3: Deciding on the data

The data for the model forming the occupancy grid is taken from the Github repository [16]. In this code a robot moves and takes readings as it moves along a path. As the robot traverses the map and takes readings, the model updates the log odds of the occupied cells and unoccupied cells as seen in Figure 16.

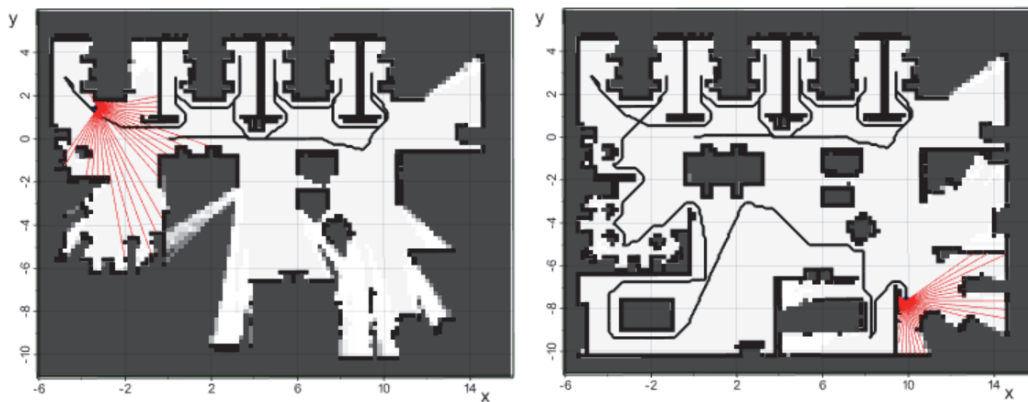


Figure 16 Creation of occupancy grid by updating log odds map

The final occupancy grid map generated can be seen below.

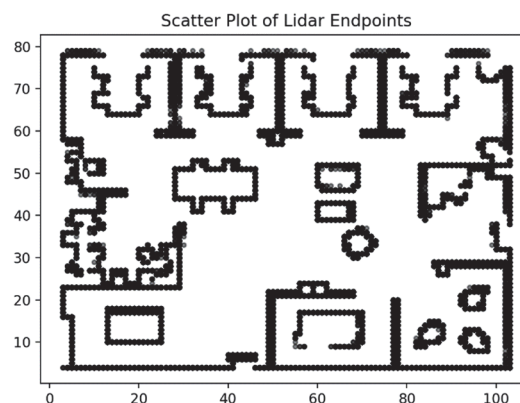


Figure 17 Final occupancy grid

3.3 Lidar measurements at NPL

The point clouds shown in Figure 18 were generated at NPL from Lidar measurements of a scene consisting of multiple flat surfaces. The point cloud is here generated by an inbuilt deterministic forward model within the sensing device.

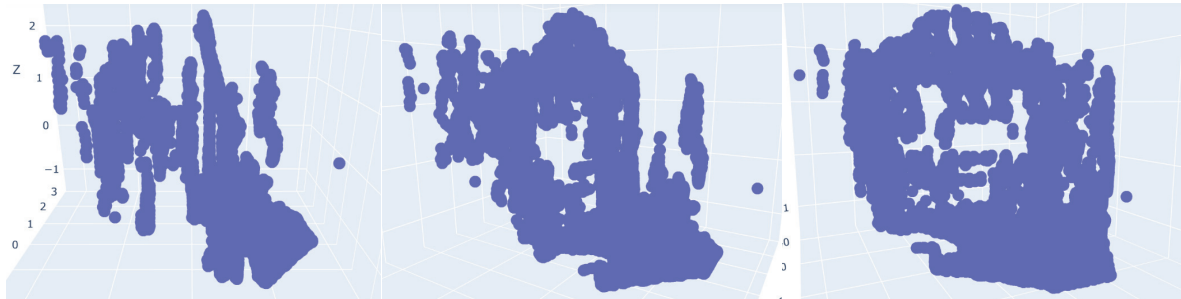


Figure 18 3D Lidar point cloud created in the NPL Lab

This gives us one snapshot of the Lidar's measurements and which corresponds to a single instantaneous measurement in the model described in Sections 3.1 and 3.2. A natural extension of the work would be to have the Lidar move along a path, taking multiple snapshots, with GPS data on the position of the Lidar being supplied. Integrating many such measurements using the approach described in Sections 3.1 and 3.2 would lead to a more comprehensive mapping of the occupancy grid of the whole room.

The point cloud generation is inbuilt, and information concerning the model used is proprietary and so not available. However, if it is possible to compare a recovered occupancy grid with ground truth, it would still be possible to construct a physically accurate forward model of the sensor. The point clouds generated here are deterministic, and future work implementing a probabilistic forward model would allow for the quantification of uncertainties of the point cloud.

3.4 Discussion

We were able to programmatically create sensor models which simulated Lidar measurements using a probabilistic forward model to create point clouds and produced occupancy grids by updating the log odds based on those point clouds. A point cloud from a typical Lidar sensor has been obtained that could be used to inform development of the forward model. There remains scope to extend the work of this report to perform more detailed analysis of the functioning of Lidar sensors. The modelling approach considered here can be used in the development of higher fidelity Lidar sensors, and to take into account weather effects on the operation of the Lidar sensor by altering the forward model to include additional scattering.

4 Summary

In this study we have demonstrated how to model certain aspects of the functioning of cameras and Lidar sensors, using the sensor modelling framework [1] to construct the models, addressing several edge cases in the process. Sensor models are needed in order to simulate sensors' working both in normal conditions and upon edge cases within a virtual testing environment. The models developed here can generate edge cases within a virtual testing environment so that a vehicle AI is presented with realistically distorted images during testing and so that dangerous edge cases can be checked without creating physically dangerous real-world situations.

The ultimate aim of this work is to support the creation of standardised virtual testing frameworks within the autonomous transport industry. Such standardisation across the industry could result in a common language which would improve collaborative research and result in the development of high-quality testing standards and improved communication between industry and suppliers. One important piece in this puzzle is standardising approaches to sensor modelling, and the work in this study and in [1] is a first step in this direction.

Future work will now focus upon engaging stakeholders in order to gain a more informed understanding of the types of weather conditions, edge cases and distortion phenomena which are most relevant for the various sensors in the autonomous vehicle sensing stack. Eventually with such standardization in sensor modelling and interfacing, virtual testing can get one step closer to being adopted across the industry as a part of safety testing or type approval in automated vehicles. This would also be a step towards improvement in uncertainty reporting and estimation in automated driving systems which is an important component of safe operation in any system.

5 References

- [1] Khatry, R. and Thompson, A (2021). A sensor modelling framework for autonomous systems. NPL Technical Report (draft), available upon request.
- [2] Ponn, T., Kröger, T. and Diermeyer, F. (2020). Identification and Explanation of Challenging Conditions for Camera-Based Object Detection of Automated Vehicles. *Sensors* 20(13).
- [3] <https://www.bbc.co.uk/news/technology-54175359>
- [4] Chen, J., Venkataraman, K., Bakin, D., Rodricks, B., Gravelle, R., Rao, P. and Ni, Y. (2009). Digital camera imaging system simulation. *IEEE Transactions on Electron Devices* 56(11), pp. 2496-2505.
- [5] Sassoon, E., Schechner, Y. Y. and Treibitz, T. (2019). Flare in interference-based hyperspectral cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10174-10182.
- [6] Hullin, M., Eisemann, E., Seidel, H. P. and Lee, S. (2011). Physically-based real-time lens flare rendering. In *ACM SIGGRAPH 2011 papers*, pp. 1-10.
- [7] <https://www.opengl.org/archives/resources/features/KilgardTechniques/LensFlare>
- [8] HALCON, Solution Guide III-C 3D Vision, MVTech, Munich, Germany Copyright 2003-2018.
- [9] Park, J., Byun, S. C. and Lee, B. U. (2009). Lens distortion correction using ideal image coordinates. *IEEE Transactions on Consumer Electronics* 55(3), pp. 987-991.
- [10] Fitzgibbon, A.W. (2001). Simultaneous linear estimation of multiple view geometry and lens distortion. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 125-132.
- [11] Lee, S., Eisemann, E. and Seidel, H. P. (2010). Real-time lens blur effects and focus control. *ACM Transactions on Graphics* 29(4), pp. 1-7.
- [12] Kuciš, M. and Zemčík, P. (2012). Simulation of camera features. In *16th Central European Seminar on Computer*, pp. 117-123.
- [13] Schaermann, A. and Hanke, T. (2019). BMW Group: Generation and Validation of Sensor Models for Automated Driving Systems Using VIRET VTD, *Engineering Reality Magazine*, Volume IX.
- [14] Weston, R., Cen, S., Newman, P. and Posner, I. (2019). Probably unknown: Deep inverse sensor modelling radar. In *2019 International Conference on Robotics and Automation*, pp. 5446-5452.
- [15] Thrun, S. (2003). Learning occupancy grid maps with forward sensor models. *Autonomous robots* 15(2), pp. 111-127.
- [16] <https://github.com/peterstratton/LidarOccupancyGridMapping>
- [17] Appendix D of Performance Testing for Sensors in Connected and Autonomous Vehicles: Feasibility Study report, July 2020 from Transport Catapult and NPL.

- [18] <https://techcrunch.com/2020/05/06/volvo-to-use-luminars-lidar-in-production-vehicles-to-unlock-automated-driving-on-highways/>
- [19] <https://blog.waymo.com/2020/03/introducing-5th-generation-waymo-driver.html>
- [20] <https://www.therobotreport.com/researchers-back-teslas-non-lidar-approach-to-self-driving-cars/>