Localisation under Large Appearance Change



Matthew Robert Church

Department of Electrical Engineering
University of Cape Town
Rondebosch, Cape Town
South Africa

October 2023

MSc.(Eng.) thesis submitted in partial fulfilment of the requirements for the degree of MSc. in Electrical Engineering in the Department of Electrical Engineering at the University of Cape Town

Keywords: localisation; agriculture; ROS; GAN

University of Cape Town

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

Declaration

I, Matthew Robert Church, hereby:

1. grant the University of Cape Town free licence to reproduce the above thesis in whole or in part, for the purpose of research;
2. declare that:
   (a) this thesis is my own unaided work, both in concept and execution, and apart from the normal guidance from my supervisors, I have received no assistance except as stated in my bibliography and acknowledgements.
   (b) neither the substance nor any part of the above thesis has been submitted in the past, or is being, or is to be submitted for a degree at this University or at any other university.

Signed by candidate

Matthew Robert Church

Department of Electrical Engineering
University of Cape Town

Saturday 7th October, 2023

Abstract

Localisation under Large Appearance Change

Matthew Robert Church

Saturday 7th October, 2023

Localisation is a foundational building block for more complex robot applications, and thus if low-cost localisation solutions can be found, the number of activities a robot can undertake will increase. However, appearance-based localisation systems in the past have required frequent traversals of the environment in order to sufficiently capture the change indicative of that environment. There are applications such as agriculture in which this frequent data collection is not appropriate.

This thesis presents an appearance-based localisation system that combines generated and recorded data in the form of experience-localiser pairs combined to create an experience based network that can be used for localisation. The inclusion of generated data reduces the requirement for frequent data collection, provided an adequate generation model can be trained. The experience, which is a collection of images and transforms describing a traversal of an environment is the primary means through which this generation of data can influence the network.

The images contained in the generated experiences were created from two parent experiences capturing two specific times of the day. The network trained learns a mapping from the two parent experiences creating intermediate domains that represent times between the parents, effectively filling in the gaps left by sparse data collection.

While the performance of the generation network narrows the functional scope of the system, within that narrow scope, experiences generated from recorded outings outperform the recorded counterparts provided the parent does as well, such that an experience generated from a recording collected at 10:00 and made to mimic the conditions at 14:00 will outperform the recording collected at 14:00. Should a version be used such that all recorded experiences are utilized as a collective, the system outperforms that of a system making use of just recorded data.

Acknowledgements

It is with sincere gratitude that I thank the people that supported me during the undertaking of this thesis.

I firstly would like to thank my family, when I told them my wish to resign my job and pursue a masters degree full time they not only celebrated but committed to any financial assistance I might require. When I needed more than financial support, they provided that as well.

I would like to thank my supervisor Dr Paul Amayo for taking a chance on an Aeronautical Engineer with the big dream of being a roboticist. Not only guiding the process of the research but also doing it in an easygoing manner.

I am grateful to Professor Rui Gong at ETH Zurich for access to the single source and target version of his DLOW network and to the University of Cape Town's High-Performance Cluster for the resources to train that network, particularly Mr Andrew Lewis for his support in getting up to speed.

Lastly, I am grateful to my friends not only did they make the transition from Pretoria seamless but provided the necessary distraction in times of stress and meals during busy weeks.

There is most likely a number of individual people that I have forgotten, for it takes a village to raise a thesis and I could most certainly not complete this on my own, this is an acknowledgement of them.

Contents

Declaration i

Abstract ii

Acknowledgements iii

1 Introduction 1

1.1 Motivation for Research 1

1.2 Background 2

1.3 Research Problem 4

1.4 Aims 4

1.5 Significance 4

1.6 Limitations 4

1.7 Structure Overview 5

2 Preliminaries 6

2.1 Introduction 6

2.2 Camera Model 6

2.2.1 Pinhole Camera Model 6

2.2.2 Depth Estimations 8

2.3 Visual Odometry 9

2.4 Generative Adversarial Networks 12

3 Localiser 14

3.1 Introduction 14

3.2 Vocabulary 14
3.3 Chow-Liu Tree 15
3.4 Sample Descriptors 16
3.5 Implementation 16

4 Experience **21**
4.1 Introduction 21
4.2 Background 21
4.3 Experience Generation 23
4.4 Experience Structure using Keyframes 24

5 Experience with Style Transfer **25**
5.1 Introduction 25
5.2 Background 25
5.3 Domain Flow [DLOW] 27
5.4 Training Tests 30
5.5 Training Regime 33
5.6 Experience Generation 33

6 Experience Based Network **34**
6.1 Introduction 34
6.2 Background 34
6.3 Experience Based Network 35
6.4 Implementation and Performance Metrics 36
6.4.1 Experience-Localiser Pair 36
6.4.2 Control Hub Implementation 37

7 Evaluation **38**
7.1 Introduction 38
7.2 Environments 38
7.2.1 Old Zoo 38
7.2.2 Lindenhof 39

7.2.3 Backsburg 41
7.3 Research Platform 42
7.4 Experiments 42
7.4.1 Network Testing 43
7.4.2 Style Transfer 43
7.4.3 Localiser Testing in Agricultural Regions 46
7.4.4 Localiser Testing with Style Transfer 49

8 Conclusions **55**

List of Figures

1.1 Examples of the appearance change due to time of day. 2

1.2 An overview of the proposed system. 5

2.1 The geometry of the pinhole camera model. Image credit: Hartley *et al.*[22] . . . 7

2.2 The geometry of the stereo camera pair as it is used in triangulation. [23] 8

2.3 A graphic of the disparity calculated per pixel across a pair of stereo images. . . 9

2.4 The visual odometry pipeline receives a processed left and right image in a pair and applies a feature descriptor. This information is then used in conjunction with the previous pair in a sequential matching process to determine the points of interest for the triangulation step, after which an iterative loop ensures an accurate motion estimate. 10

2.5 Circular feature matching implemented by Geiger *et al.* [24] 11

2.6 A graphical definition of Epipolar geometry by Chotrov *et al.*[30] 11

2.7 The cycle consistency loss is a comparison of real and recovered images and is specific to each of the two generators. Image credit: [16] 13

3.1 Example of a vocabulary word. The left side of the image details the word itself and the right side of the image shows the word in context. Image credit: Cummins *et al.*[14] 15

3.2 Implementation process of openFABMAP 16

3.3 Displaying the relationship between the number of vocabulary words and the number of images in the training set. 17

3.4 Displaying the performances of different vocabularies on the same localising object. 18

3.5 Inlier calculation pipeline 18

3.6 A few examples of false positives returned by the localiser. In each instance, the inliers were larger than the threshold value. 19

3.7 A few examples of true positives returned by the localiser. In each instance, the inliers were larger than the threshold value. 20

4.1 A graphic illustration of the data collection of experiences in the presence of others. Image credit: Churchill *et al.* [2] 22

4.2 An example of the experience structure of Churchill. Image credit: Churchill [2] . 22

4.3 The experience generation workflow. 23

4.4 The process of reducing the nodes retains the transformation information between nodes. Where N depicts a node and where $N1$ through $N3$ might be a corner, and $N3$ through $N6$ would represent a straight section in the trajectory. 24

5.1 A comparison of the improvement in matching performance experienced where the coloured lines represent matches between the images. 26

5.2 Representation of DLOW moving an image from the source to the target domain. Image adapted from [18] 27

5.3 Statistical representation of the process of moving an image from the source to the target domain. Image adapted from [18] 27

5.4 Graphic depiction of the DLOW network with a single source and target domain. Additional loss functions are also included. Image adapted from [18] 28

5.5 A comparison of the effect of training epochs on the number of average good matches 31

5.6 An extreme example of the extreme domain shift that occurs at a domainess value of approximately 0.5 31

5.7 An example of artefacts introduced by DLOW 32

5.8 Output of the network trained in two phases, an initial general train and the second a fine-tuning phase conducted on a few select image pairs. 32

5.9 The process of creating a generated experience from an existing one. 33

6.1 Graphic depiction of an experience-based network as developed by Churchill *et al.* The green dashed lines show the links between nodes. Note the different numbers of nodes in each experience instance. Image credit: Churchill *et al.* [2] . 35

6.2 A graphic representation of the control hub and the experience pairs 36

6.3 These plots show the probability that a robot will be lost for a given distance. In both cases the robot was lost for about 50% of the distance however the first robot was only lost for 10m per occasion. 37

7.1 Photos of the Old Zoo environment 39

7.2 A satellite image of the path utilised in the Old Zoo environment 39

7.3 Photos of the Lindenhof environment 40

7.4 A satellite image of the paths utilised in the Lindenhof farm environment. Where green is the evaluation path and yellow is the vocab path. Markers indicate the start point. 40

7.5 Photos of the Backenburg Farm environment 41

7.6 A satellite image of the paths utilised in the Backsburg farm environment. Where green is the evaluation path and yellow is the vocab path. Markers indicate the start point. 41

7.7 Photos of the Clearpath Husky at the Old Zoo 42

7.8 A visual depiction of the three timeslots used as domains to train the Old Zoo models 43

7.9 Output of the model trained to move images from source 04:30 to target 10:00 . 44

7.10 Output of the model trained to move images from source 10:00 to target 04:30 . 44

7.11 Output of the model trained to move images from source 10:00 to target 16:00 . 44

7.12 Output of the model trained to move images from source 16:00 to target 10:00 . 44

7.13 Output of the model trained to move images from source 04:30 to target 16:00 . 44

7.14 Output of the model trained to move images from source 16:00 to target 04:30 . 45

7.15 A visual depiction of the two domains used to train the Lidenhof model 45

7.16 Output of the model trained to move images from source 10:00 to target 16:00 . 46

7.17 Output of the model trained to move images from source 10:00 to target 16:00 . 46

7.18 Results of the localisation test conducted on the Lindenhof dataset 47

7.19 Example of localisation output of each vocabulary for road section, where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the size of vocab used. 47

7.20 Example of localisation output of each vocabulary for nectarine row section, where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the size of vocab used. 48

7.21 Results of the localisation test conducted on the Backsburg dataset 48

7.22 Example of localisation output of each vocabulary, where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the size of vocab used. 48

7.23 The effect of appearance change due to time of day on the performance of the localiser in the Old Zoo. 49

7.24 The effect of appearance change due to time of day on the performance of the localiser at Old Zoo in the presence of generated experiences. Where DLOW 10:00 are images created from the collection at 10:00 and transferred to either 16:00 or 04:30. DLOW 16:00 are images created from the collection at 16:00 being transferred to 10:00 and DLOW 04:30 are images created from the collection at 04:30 also being transferred to 10:00 50

7.25 A still of the localisation output for the experiment comparing all the experiences generated from the 10:00 data collection. 50

7.26 The lost probability of all the experiences generated from the data collection at 10:00. 51

7.27 The regions along the trajectory that were successfully localised to. The overlapping of points suggest a high number of successful localisations in those regions. 51

7.28 The probability plot of the individual experiences where data collected at 08:00 was used as the live stream. 52

7.29 The probability plot of the collection of DLOW experiences vs the parent experience 52

7.30 Example of localisation output where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the time of day mimicked and names containing *fineA* are generated. 53

7.31 Example of localisation output of the previous experiment, where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the time of day mimicked. Times in grey are recorded and times in orange are generated. 53

7.32 The lost probability of the experiment. Where DLOW represents the experiences as a collection, sunrise is the performance of the parent data collection and sunny is the performance of the collection closest in appearance to Live. 54

Chapter 1

Introduction

1.1 Motivation for Research

Localisation is the process by which a robot determines where it is relative to the environment in which it operates and is a foundational building block of robotic operations, along with mapping, the process by which a robot creates and saves a digital representation of the world. These building blocks provide a base for other more complex behaviours such as navigation, a robot's ability to successfully move around the environment autonomously, and offer robotic platforms the means to interact with the environment. Humans and animals easily perform these tasks, having only visited an area once. Glover *et al.* [1] deployed a system that simultaneously localises and maps out an environment, mimicking the process implemented by either a human or an animal and creating what they call an experience map.

Focusing on the problem of determining a position for navigation in environments that undergo large appearance changes over greater distances and timeframes, Churchill *et al.* [2] created a system that records outings of the environment. When the robot fails to determine where it is based on what was previously recorded, a new recording is conducted. Thus the first outing would record the entire traversal, and subsequent outings would only record when the robot cannot determine its location with reference to the previous outings or, as they define, experiences. Thus the system would inherently record more data in areas with the most drastic change in appearance and therefore require many outings to the environment. An example of the changes an environment can experience with just reference to time can be seen in Figure 1.1, this change is only compounded with changes in weather and seasons. The large number of outings for the solution proposed by Churchill *et al.* [2], which might be applicable for certain applications such as training autonomous vehicles on the roads, is not appropriate when applied to an environment like a farm where fundamental activities are disrupted during data collection.

The question is, can a vision-based localisation system be implemented that provides robust localisation in unstructured or semi-structured environments, such as that of a farm, and that only requires sparsely collected data? The benefit of a solution like this is two-fold, firstly

the requirement of only sparse data collection ensures an unobtrusive avenue to investigate the automation of tasks, and secondly, due to the nature of a vision-based localisation system, the platform itself can be relatively simple, as the only input the system would require is a sequence of frames captured.



Figure 1.1: Examples of the appearance change due to time of day.

The following sections of this chapter introduce the research conducted regarding this question and the solution devised. Starting with a brief overview of existing research in the field of vision-based localisation, followed by a formalised question, the research aims specific to the solution are then devised, including the questions and objectives and ends with an expansion on why this work is important before concluding with the limitations of the work.

1.2 Background

The problem of localisation can be approached in several ways by any number of sensors or by a fusion of multiple sensors. The choice of sensor suite will determine the approach to localisation. This work uses cameras as the primary sensor for localisation and focuses on an appearance-based localisation approach. The use of appearance or vision-based systems for localisation is not a novel concept and has been successfully implemented by other research groups [2]–[8]. The

localisation problem has also been solved in the implementation of Simultaneous Localisation and Mapping [SLAM] systems [1], [9]–[12]. Given an area, a robot implementing SLAM would systematically move from the starting location building a map as it traverses the unknown environment. How the robot moves through the area and knows when it is fully mapped depends on the algorithms implemented.

Much like the problem of SLAM, appearance-based localisation has been addressed before, and a common and fundamental aspect of such systems is the robot's ability to place the currently viewed image in the built map of the environment that it has. This ability to approximate location from an image is called visual place recognition Lowry *et al.* [13] and the approximation of robot orientation to that frame is called pose estimation. An important development in this localisation component of visual place recognition with respect to appearance-based localisation is that of FABMAP Cummins *et al.* [14]. FABMAP is a system developed for visual place recognition and is used to determine whether the robot has returned to a previously visited location. It does this efficiently by implementing look-up procedures and fast bail-out methods. Other the visual place recognition systems include SeqSLAM [10] and NetVLAD [15].

FABMAP system acts as the basis of the works described by Glover *et al.* [1], which shows success during different times of the day, and by Churchill *et al.* [2], which presents a system that has been able successfully to navigate for incredibly long distances by implementing the concept of experiences and an experience network. Building on the work of Churchill *et al.* [2], Porav *et al.* have been able to improve the performance of the system by altering the input images to match those of conditions previously seen [3], making use of cycleGAN [16] which allows an image to image translation to be learnt using unpaired datasets. In this way, the input image, which might have failed to localise due to a large change in the environment's appearance, is altered to match environmental conditions already recorded and can allow the robot to localise. The caveat here is that in order to localise in environments that undergo large appreance change requires frequent outings to record those changes.

A limitation of cycleGAN [16] implemented by Porav *et al.*, however, is that it can only learn a single source to target mapping, such that only experienced changs can be generated. To address this limitation, Almahairi *et al.* developed Augmented cycleGAN [17]. They found that by introducing a variable x in to the latent space of the model during the training phase, they are able to adapt cycleGAN to learn a many-to-many mapping, opening the ability for cycleGAN to be applied to more complex domain adaption tasks. Based on the work of Almahairi *et al.*, Gong *et al.* created DLOW [18], which in a sense, does one-to-many mapping but is also able to create intermediate frames between the source and single or multiple target domains by employing an introduced domainness variable. The work of Gong *et al.* facilitates the ability to create images that are a ratio of the source and target domains where in the past it had to be one or the other, perhaps improving the ability of an appearance-based localisation system, primed with these generated images, to localise in an environment that has experienced a large appearance change.

1.3 Research Problem

It has already been shown that long-term navigation in the presence of large appearance change can be achieved with the framework developed by Churchill *et al.* [2], provided the framework has the ability to collect data frequently. It has also been shown that altering the input images to match what has already been seen can improve the performance of such a system [3]. However, combining these two solutions and applying them to the problem of large appearance change concerning visual localisation in unstructured or semi-structured settings leaves much to be desired. The fusion of the two solutions will still require high availability of data to capture the appearance change, and the choice of target for the style transfer needs to be determined beforehand to alter the input images. If we were to change the input images for, say, a time-of-day experiment, appropriate models would need to be trained for each condition. What if the environment doesn't match a pre-trained model?

1.4 Aims

We aim to create a system that facilitates the successful localisation of a robot in an unstructured or semi-structured environment by augmenting the framework developed by Churchill *et al.* [2] with images generated from previous outings to the environment. These outings should be the minimum required to appropriately capture the large visual appearance change indicative of the time of day. Therefore facilitating the ability of more complex automation tasks, such as a robot performing autonomous navigation in rowed crops.

1.5 Significance

We demonstrate a possible real-world application and benefit of style transfer networks to applications where only sparse data collection is appropriate. We present a potential solution that is cost-effective and can be deployed not only in an agricultural context, the automation of which is gaining popularity [19]–[21] but also for applications where conventional global localisation techniques, such as using GPS, are not applicable, like that of a mine.

1.6 Limitations

The core of this thesis is focused on the development of a vision-based localisation system in an attempt to mitigate large appearance change. Therefore the creation of anything more than a loss function for the already developed style transfer network created by Gong *et al.* [18] is outside the scope of this thesis. Other limitations include access to agricultural land, compute power available on the utilised platform, and the choice of input sensors.

1.7 Structure Overview

The thesis begins with the introduction chapter and then proceeds to lay the groundwork for the work in Chapter 2. Chapter 2 describes the properties of the camera model being used and how it is possible to estimate motion from a sequence of left and right image pairs. Chapter 2 also briefly discusses Generative Adversarial Networks as an introduction to the work conducted later in the thesis. In Chapter 3, we discuss the localiser object, a core component of this thesis, by first discussing the associated background and inputs to the system before concluding with how it was trained and implemented for each environment. Chapter 4 introduces the concept of an experience, how it was previously utilised in [2] and [3] and how it is utilised in this work. Chapter 5 discusses another core component of the work in the form of the Generative Adversarial Network that is used to synthesise or "hallucinate" experiences in an attempt to mitigate large appearance change. Chapter 6 discusses the culmination of all the parts in the form of the Experience Based Network, similar to that implemented in [2], and how the network determines which of the experiences should be presented to the systems built on top of the localiser. A general overview of how the system takes a stereo pair of images to outputs presented to the navigator can be seen in Figure 1.2. Chapter 7 describes the environments and research platform used for the experiments, development of the network, and experiments themselves, along with the results. The thesis concludes with Chapter 8, which details the findings of the work and suggests avenues for future research.

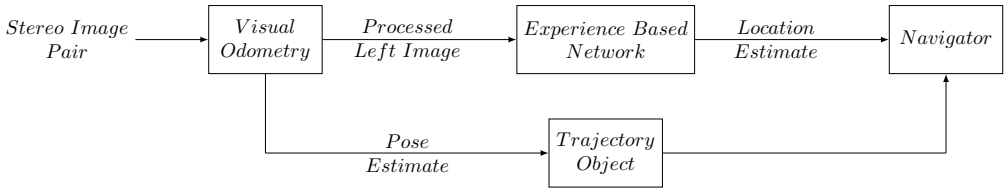


Figure 1.2: An overview of the proposed system.

Chapter 2

Preliminaries

2.1 Introduction

This thesis is primarily interested in reducing the data requirement, or outing frequency, needed for successful localisation in specific environments. It approaches this problem by augmenting a localisation framework with generated data. This chapter introduces the underlying methods required to create a localisation framework and to achieve this augmentation. It will begin with a preliminary breakdown of the properties of a camera, which is the input sensor for our implementation, followed by the definition of both visual odometry and visual place recognition, as these are the building blocks of the systems described in the later chapters. The chapter concludes with an introduction to generative adversarial networks.

2.2 Camera Model

The primary sensor of an appearance-based localisation system is a camera. These localisation systems can be implemented in a monocular arrangement where only one camera is used or in a stereo configuration when a pair of cameras are deployed a known distance apart, this distance is termed the baseline. The baseline distance between the two cameras is then used to add information about the observed scene. This work utilises a stereo camera pair.

2.2.1 Pinhole Camera Model

To understand how a camera interacts with the world and the process of rendering a 3D scene into a format that can be used effectively, the pinhole camera model can be used. The model describes a simplified process of how a camera performs a transformation from 3D space to 2D space. Figure 2.1 shows a geometric representation of this transformation.

In Figure 2.1, C is the centre of the camera, X is the point in the 3D world that is projected

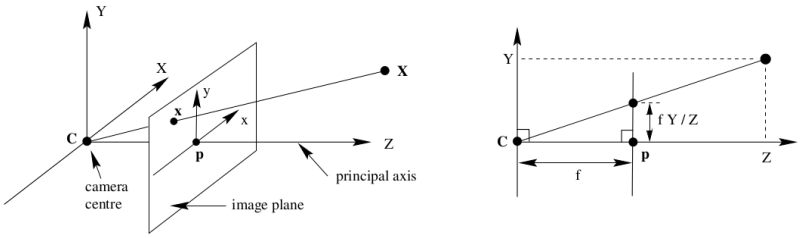


Figure 2.1: The geometry of the pinhole camera model. Image credit: Hartley *et al.*[22]

into x on the image plane, which is a distance, known as focal length f, from the camera centre. This mapping of 3D points to 2D points is achieved by using a projection matrix. Given by Equation 2.1 where P is the projection matrix, which consists of both a camera's intrinsic and extrinsic matrices as seen in Equation 2.2.

$$x = PX \tag{2.1}$$

$$P = k[R|t] \tag{2.2}$$

The intrinsic matrix k describes the geometric properties specific to the camera. This matrix is represented in Equation 2.3 where f_x is the focal length in the x direction, f_y is the focal length in the y direction, s is the axis skew. The principal point offsets x_0 and y_0 are due to the lens sensor interaction and are a property of the manufacture of the camera, in x and y, respectively.

$$k = \begin{pmatrix} f_x & s & x_o \\ 0 & f_y & y_o \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

The extrinsic matrix describes the camera's relationship to the world, such as the orientation. The extrinsic matrix consists of a 3x3 rotation matrix and a 3x1 translation vector normally represented in a 3x4 matrix as seen in Equation 2.4.

$$[R \,|\, \boldsymbol{t}] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right] \tag{2.4}$$

The intrinsic and extrinsic matrices are determined during the calibration procedure of the camera. By using these matrices, the pair of images can then be modified to remove the effects of the lenses in a process called undistortion and be transformed onto a common image plane in a rectification process.

2.2.2 Depth Estimations

Due to the camera's transformation from 3D to 2D space, the information about the distance of the point in front of the camera, or depth, is lost. To estimate the motion of a robot, this distance or depth information needs to be recovered. One way we can calculate the lost depth information is by using the overlapping properties of a stereo pair of cameras viewing the same scene through a process called triangulation. Figure 2.2 is a top-down diagram of how this process works.

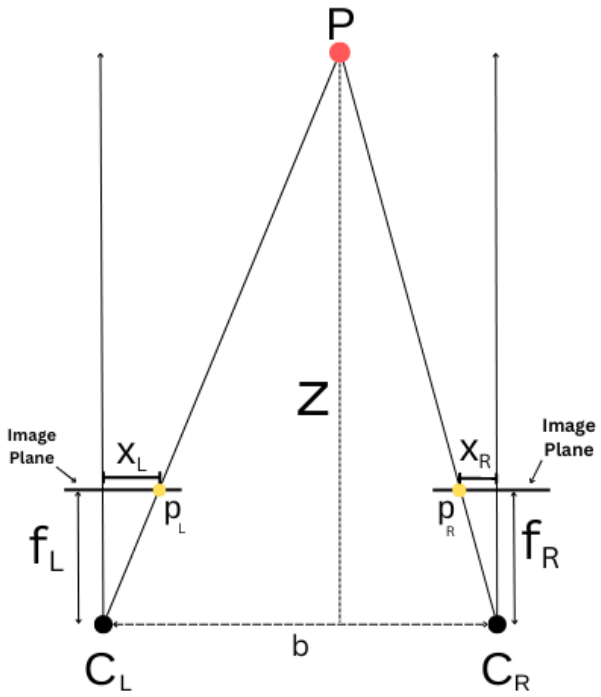


Figure 2.2: The geometry of the stereo camera pair as it is used in triangulation. [23]

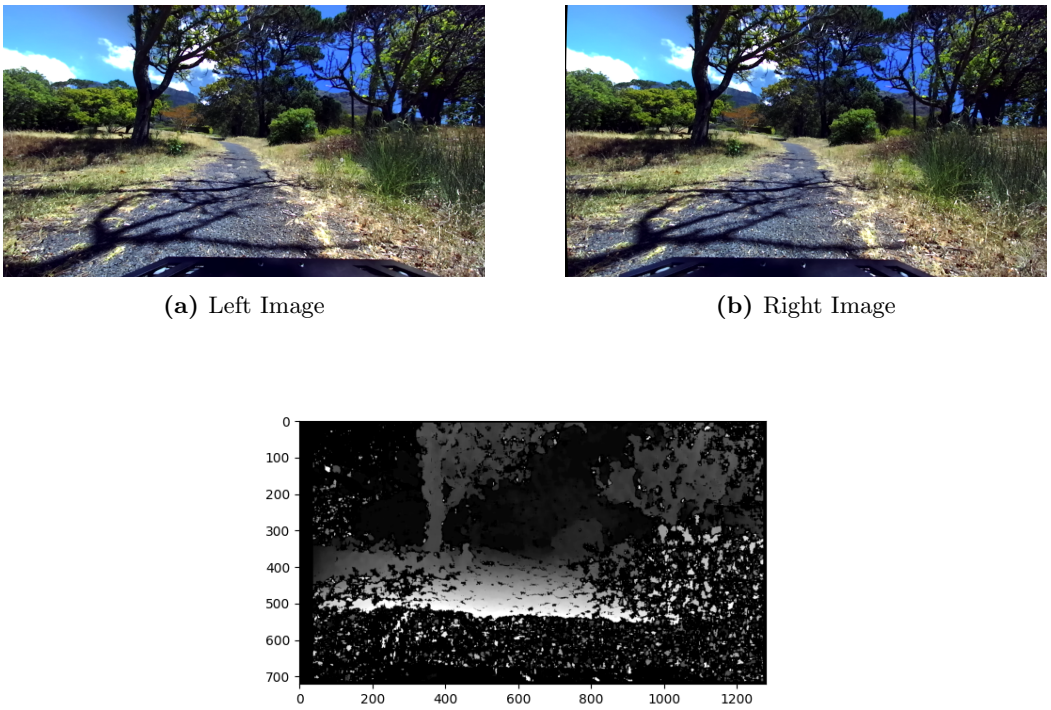
Where b in Figure 2.2 is the baseline C_l and C_r are the left and right cameras, respectively, p_l and p_r are the corresponding pixels of point P projected onto the respective camera image plane, and Z is the depth we are trying to recover. The point P viewed by each camera will have a difference in pixel mapping due to the physical relationship of the camera to that point. This difference in relative x value is known as the disparity and because of the physical relationship between the pair of cameras and P in space, the further the point is, the smaller the disparity between the two projections will be. This calculation with respect to the left image is shown in Equation 2.5 where x_l is the x-coordinate of the point in the left image plane, x_r is the x-coordinate of the point in the right image plane.

$$d = x_l - x_r \tag{2.5}$$

A disparity map is simply this information over all the pixels in an image. Figure 2.3 shows the

output of this calculation when plotted per pixel across the whole image. From this disparity map, a depth map is then calculated, referring to Figure 2.2 the depth is simply the Z value that has been recovered. The per-pixel depth calculation can be seen in Equation 2.6

$$\begin{bmatrix} p_x^L \\ p_y^L \\ p_z^L \end{bmatrix} = \frac{b}{d} \cdot \begin{bmatrix} x_l^L \\ y_l^L \\ f_r \end{bmatrix} \tag{2.6}$$



(a) Left Image

(b) Right Image

(c) Disparity Map

Figure 2.3: A graphic of the disparity calculated per pixel across a pair of stereo images.

2.3 Visual Odometry

For a localisation system to be meaningful, a representation of the environment is required to tie each localisation to the real world. This can be achieved by implementing a system that estimates the motion of a robot given a sequence of stereo images. Such a system can create a transform between each image pair and thus place each image pair relative to the starting location using a sequence of transforms. The process of determining these transforms is called visual odometry. Visual odometry is a way of estimating both the translational and rotational motion of the robot due to the camera's extrinsic matrices. This paper makes use of the visual odometry pipeline proposed by Geiger *et al.* [24] due to the relatively high frame rate the motion estimation system can achieve as well as the ability to utilise high-resolution images. The visual

odometry pipeline is shown in Figure 2.4 and discussed in this section. The images are rectified and undistorted before being input into the visual odometry pipeline.

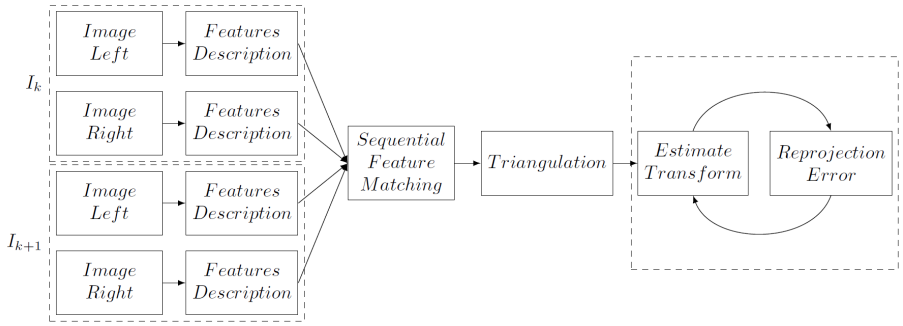


Figure 2.4: The visual odometry pipeline receives a processed left and right image in a pair and applies a feature descriptor. This information is then used in conjunction with the previous pair in a sequential matching process to determine the points of interest for the triangulation step, after which an iterative loop ensures an accurate motion estimate.

Feature Description

Referring to Figure 2.2, each image plane has a pixel p which corresponds to point P in the world. To determine the location of this pixel p, a feature dectector is used. There are a number of feature detectors such as SIFT [25]. SURF[26], FAST [27], ORB[28] and STAR[29]. Feature detectors identify regions of interest, such as corners, edges, ridges or blobs, within the image and the feature descriptor then describes these regions in order to be used in other processes, such as feature matching. Thus the feature dectector and descriptor together produce results for downstream in the process. The choice of feature detector is often related to the characteristics of the environment, and given the same image, there could be a different number of features depending on the descriptor used. Geiger *et al.* opted to utilise blobs and corners in their feature descriptor implementation.

Sequential Feature Matching

The transform estimation step of the pipeline requires features to be matched between both the left and right images as well as consecutive frames. Given the feature locations, Geiger *et al.* perform a sort of circular matching between the four frames. This is visually described in Figure 2.5 where I_k denotes the current frame pair and I_{k-1} denotes the previous pair.

This circular match only gets accepted if the features of the last image in the loop match with the first. When comparing the left and right images in a pair, a geometric constraint is applied to the matches. This geometric constraint utilises epipolar geometry which, referring to Figure 2.6 [30], consists of an epipolar plane formed by C_l, C_r and P, an epipolar line that is the intersection of this plane and the respective camera image planes and epipolar points that lie on the epipolar line, p_L and p_R are epipolar points. Geometric constraints applied between the

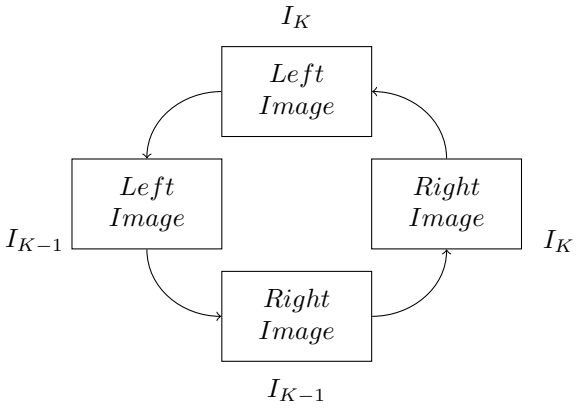


Figure 2.5: Circular feature matching implemented by Geiger *et al.* [24]

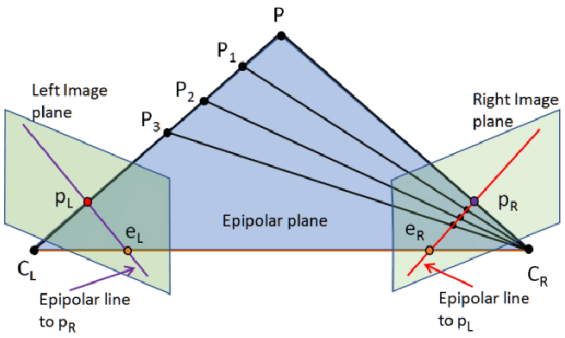


Figure 2.6: A graphical definition of Epipolar geometry by Chotrov *et al.*[30]

left and right images ensure that the matched features are on known epipolar lines. Matches that are supported by at least two neighbours are kept.

Triangulation

Triangulation of points in the pipeline, described in Section 2.2.2, has two functions. Firstly, during the matching process, triangulating the points to the left image in the stereo pair helps to ensure that only high-quality feature matches are presented to the motion estimation loop, ones that have an incosistancy in depth are rejectd as outliers. Secondly is the projection of points in 3D space to the previous left image. The features are grouped and spread uniformly across the image to maintain computational efficiency. This reduced number of features gets projected into 3D space with respect to the previous frame.

Motion Estimate

Motion estimation in the pipeline is an iterative phase, though, in practice, only a couple of iterations are required to meet the convergence threshold. This iteration seeks to reduce the reprojection error, ensuring the pipeline delivers an estimation of high confidence. This part of the pipeline also implements random sample consensus [RANSAC] [31], which is a process

of estimating a mathematical model given a collection of data to determine which observed points are relevant, inliers, and which points can be ignored, outliers. Calculation of inliers is important for systems introduced later in the thesis.

2.4 Generative Adversarial Networks

This work is concerned with augmenting a vison-based localisation framework with generated data in an attempt to improve localisation, one way of doing this is by using a generative adversarial network. Porav *et al.* in [3] have shown an improvement in localisation performance in the presence of large appearance change by utilising input images that have been altered to mimic that which has already been seen. Porav *et al.* altered these images by implementing a generative adversarial network [GAN], which is a generative framework in which two models are simultaneously trained in a manner similar to that of a minimax two-player game, where the advantage of one player is to the detriment of the other. The network consists of two components, a generator and a discriminator. It is the objective of the generator to capture the distribution of the source domain accurately and the objective of the discriminator to distinguish between instances created by the generator and instances that already existed in the target domain [32]. A disadvantage of a GAN is that improper synchronisation between the discriminator and the generator can lead to mode collapse, the effect that any image input into the model will return the same output and therefore does not sufficiently caputre the diversity of the training data. This can be caused by an imbalance in generator and discriminator during training, the limited capacity of the generator in terms of layers, a loss function that doesn't adequately penalise the generator or bias to certain modes in the training data. The solutions to the mode collpase are often specific to the dataset and the problem and therefore need to be investigated each time.

Image-to-Image translation

Following the development of the GAN [32], the pertinent application for the network with respect to this work is that of image-to-image translation. Image-to-image translation is a type of computer vision problem where the main objective is to learn a mapping from the source domain to a target domain in order to generate images that match the target domain. Previously this was conducted using paired datasets, such as the pix2pix framework [33], which has achieved some impressive results in a number of tasks. In the absence of paired datasets, however, say like time-wise recordings of an agricultural environment, one needs to implement a system that can be reliably trained using unpaired datasets. Building on the work of [33], Zhu *et al.* [16] have created a framework that does just that, boasting results that set the framework above its contemporaries. The cycleGAN network achieves these remarkable results by introducing a cycle consistency loss. Figure 2.7 describes this feature graphically.

Where a standard GAN uses a single generator and one or more discriminators, the cycleGAN

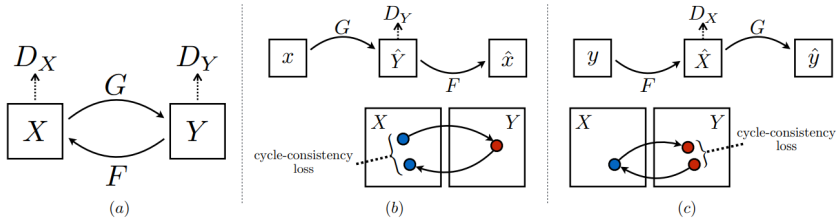


Figure 2.7: The cycle consistency loss is a comparison of real and recovered images and is specific to each of the two generators. Image credit: [16]

framework uses two generator discriminator pairs. Referring to Figure 2.7, the idea is that one GAN will learn the mapping from the source to the target domain and the other will learn a mapping from the target to the source domain. A cycle consistency loss is then determined when an image is mapped from X to Y and then back from Y to X, the recovered image is then compared to the original using an L1 loss function. The cycle consistency loss is defined using an L1 as follows:

$$L_{A\to B} = \|X_A - G_{A\to B}(X_A)\|_1$$

$$L_{B\to A} = \|X_B - G_{B\to A}(X_B)\|_1$$

Where $G_{A\to B}$ and $G_{B\to A}$ be the generators for domain A to B and domain B to A translations, respectively and X_A represents an image in domain A, where as X_B represents an image in domain B. The total cycle consistency loss is then the sum of these two functions:

$$L_{cyc} = L_{A\to B} + L_{B\to A}$$

This additional loss encourages the generator networks to produce translations that can be reversed without much degradation in information, effectively enforcing a cycle-consistency property in learned mappings. The loss function also acts to combat mode collapse and ensure that the resulting images are semantically meaninigful.

Chapter 3

Localiser

3.1 Introduction

The fundamental aspect of the localisation framework implemented in this work is that of visual place recognition (VPR). While visual pose estimation is a function of the visual odometry pipeline, this chapter addresses the VPR aspect of the system. The main objective of a visual place recognition system is, given an image, to accurately determine if it is a place that it has already seen [13]. Fundamentally a VPR system needs to have a map of known places and the ability to compare presented images to that map in order to determine and report the probability that it has seen that place before and where it is within the map. Considering that VPR is a visual-based localisation system, it has all the same drawbacks that other visual systems have such as viewpoint changes, lighting conditions or environmental changes. How a VPR system deals with these challenges will determine its efficacy. FABMAP2.0 [34] has proven to be a strong contender as the VPR system for the localisation framework, with proven success over significant distances [35]. OpenFABMAP [36] is an implementation of [34] created to be used with openCV and is the implementation used in this work. As inputs, openFABMAP requires a vocabulary which describes the environment, a Chow-Liu tree which describes the relationship between the words in the vocabulary and sample descriptors that are used to create the new place likelihoods given an input image. Each of these inputs is discussed in detail in this chapter.

3.2 Vocabulary

OpenFABMAP [36] makes use of a bag of words approach [37] to create what they call the vocabulary or codebook. The vocabulary is one of, if not the primary contributor to the performance of the system and therefore care needs to be taken when creating one. The generation of the vocabulary should be created with a selection of images which sufficiently capture the primary attributes of the environment while using a feature descriptor that is appropriate given

the environment and the constraints of time and compute resources. This process of creating a vocabulary is not limited to a specific feature descriptor and in fact, openFABMAP gives the option to use a number of widely used descriptors, however, the choice of the descriptor is fixed and new vocabularies will need to be trained should the choice of descriptor change. The clustering is conducted on the feature vectors over the entire training set. The feature vectors that correspond to a certain visual word are determined by using the Voronoi region around the cluster centres on the image. This clustering creates a description of the training set that is more discrete and allows the use of efficient retrieval techniques to be used in operation. Figure 3.1 is an example of the vocabulary created by Cummins *et al.* [14]



Figure 3.1: Example of a vocabulary word. The left side of the image details the word itself and the right side of the image shows the word in context. Image credit: Cummins *et al.*[14]

3.3 Chow-Liu Tree

Even though the clustering process used in vocabulary generation creates a discrete representation of the training data, large training sets can still be impractical to work with. OpenFABMAP addressed this by making use of another distribution that closely resembles the vocabulary, but that is far more practical to utilise, in the form of a Chow-Liu Tree [38]. The method developed by Chow *et al.* is able to approximate the primary distribution using a tree structure that not only reduces the Kullback-Leibler divergence [39], which is a measure of the difference between two statistical distributions where smaller values represent similar distributions, but also returns the optimal distribution of links between the visual words. This system innately defines vocabulary words with very little mutual information as independent and links are created between words with mutual information. Put another way, the tree structure naturally captures links between objects in the environment that are frequently seen together. This is a great benefit to appearance-based localisation as it increases the confidence that a particular word is correctly identified in the context of the scene.

3.4 Sample Descriptors

Sample descriptors are generated by using the same selection of images used to create the vocabulary and Chow-Liu tree. These images are then processed by using the created vocabulary and Chow-Liu tree. These sample descriptors are used to prime the localiser for use in the calculation to determine the probability that a place the robot is currently viewing is new or not during the execution phase. This probability calculation forms the basis of the failure criterion in which the localiser will return a flag to distinguish the view as a new place.

3.5 Implementation

The openFABMAP localiser has a clear series of steps that must be executed. These include the training, the initialisation and priming of the localiser and then the execution of the localiser. Figure 3.2 shows this graphically, and the following sections describe the steps and implementation in detail.

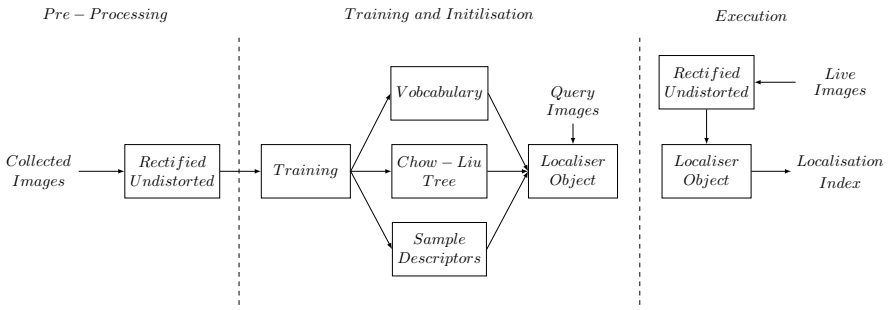


Figure 3.2: Implementation process of openFABMAP

Training

The training phase is responsible for the creation of the vocabulary, the Chow-Liu tree and the sample descriptors. These three objects are specific to the choice of environment that the robot will be operating in as well as the descriptor that is used. In practice, this occurs for each new environment seen by the robot. The vocabulary is a common source of performance issues in the implementation of the system, as the number of words in the vocabulary determines the specificity. There is a trade-off between the number of false positives and specificity, with a lower specificity producing more false positives but being able to adapt better to changing conditions. Previous implementations of the system give us an indication of the number of words that might be appropriate, such as 10000 words for a high-definition outdoor set [14] or 6000 words in the case of a standard-definition outdoor set [1]. Similarly to [14], we used the SURF [40] descriptor, as it satisfied the compute requirements and has proven to be popular in the computer vision community [41]–[48]. Glover *et al.* suggest that the more data present during the training phase, the better the vocabulary will be. In practice, the uniformity of

agricultural environments leads to a plateau of words in that more data does not necessarily yield a larger vocabulary. Figure 3.3 shows this effect, starting with an incredibly small training set of 250 random images, sampled from a data collection at Backsburg, described in Chapter 7.2.3, containing 18709 images, there is a steep climb in the number of vocabulary words most likely as the relatively low number of images are unique but quickly slows with a greater number of images as more images begin to show only small variations from each other. These small increases in vocabulary size come at the cost of an exponential increase in the time required to train.

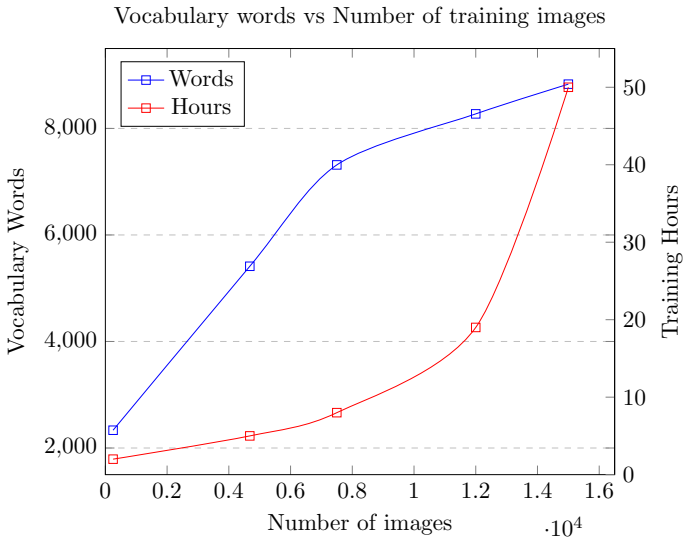


Figure 3.3: Displaying the relationship between the number of vocabulary words and the number of images in the training set.

Glover *et al.* also mentioned that viewing the same scene multiple times also biases the algorithm, in an agricultural environment where all the crop rows ideally look the same, this might be troublesome. Using Figure 3.3, a comparison of vocabulary sizes was conducted for one of the experimental environments. Like the results shown in Figure 3.3, the images were randomly sampled from a data collection at Backsburg containing 18709 images that have no overlap with the test region. Figure 3.4 shows the results of this test.

With reference to Figure 3.4, there is some correlation between the number of localisations and the size of the vocabulary however there is also a decline in the percentage of good localisations that are returned peaking at around 6000 words. Where a localisation is considered good when the returned image, having been processed by the visual odometry pipeline, returns a number of inliers that is above a predetermined threshold. Figure 3.5 shows this process.

Initialisation

In order to initialise the localisation object, outputs of the offline training phase are used as inputs. This localiser object is then primed with images that were recorded prior to this initialisation phase but separate from the data collection that was used to generate the vocabulary.

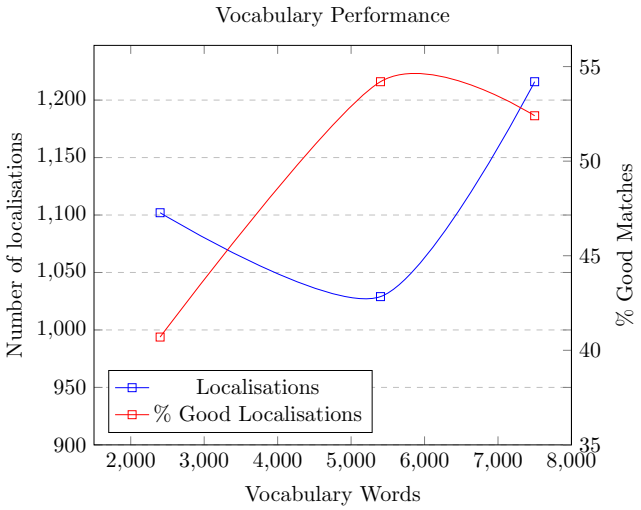


Figure 3.4: Displaying the performances of different vocabularies on the same localising object.

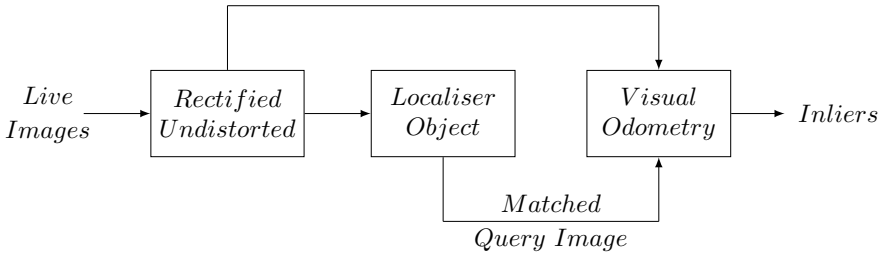


Figure 3.5: Inlier calculation pipeline

The localiser then assigns each image a sequential index. We term the sequential bank of indexed images the query images.

Execution

During the execution phase, the robot can either be live and moving through an environment providing a stream of images or this stream of images can be simulated using a previously recorded outing to the environment. Referring to Figure 3.2, the localiser compares the presented image to the bank of query images and generates a likelihood using the sample descriptors. This likelihood is a measure of similarity between live and query images such that areas previously visited return a high likelihood. The index of the query image with the highest likelihood is returned by the localiser. Contained within openFABMAP [36] is a failure criterion, in which should the system fail to localise completely and is in a new place, an easily identifiable flag is returned. These instances are easy to handle as this output of the system can confidently be discarded, another instance however is when a query image that is visually different from the current live image is returned as a successful localisation. It is these false positive localisations that need to be identified in order to produce a certain level of confidence in the images that are returned by the framework. Figure 3.6 shows some examples of false positive results and Figure 3.7 shows some examples of correct localisations.



(a)



(b)



(c)



(d)

Figure 3.6: A few examples of false positives returned by the localiser. In each instance, the inliers were larger than the threshold value.



(a)

(b)

(c)

Figure 3.7: A few examples of true positives returned by the localiser. In each instance, the inliers were larger than the threshold value.

Chapter 4

Experience

4.1 Introduction

The concept of an experience, with respect to appearance-based localisation, was introduced in the work of Churchill [2]. An experience, in the sense that it is used in this work, is primarily the means with which to integrate the localisers into the network and the manner with which to determine the efficacy of localisers given either generated or recorded images. It is on this premise that this chapter seeks to describe an experience which was developed from the work of Churchill *et al.* The core concept of the experience remains the same as the experiences described in [2] however, there are implementation differences which are discussed here in detail.

4.2 Background

In work done by Churchill *et al.* [2], an experience is simply the recording of all data associated with the visual odometry input and output for an environmental traversal. The work of Porav *et al.* [3] uses this same approach in the definition of an experience. This experiential data includes all the live images, their features as well as the metric transformation chain connecting each image with the previous and subsequent frames. Each time the robot is in an area previously visited, the system attempts to localise on the presented stream of images; to do this, each previously recorded experience gets assigned a localiser. As discussed in Chapter 3 the localiser attempts to determine the robot's location in the environment by matching the live image with the query images in the saved experience. In the system created by Churchill *et al.* [2], should n-number of localisers return a match, the system ceases to record new information under the assumption that the current number of experiences is sufficient to localise with a specified level of confidence. However, should at a future time the robot revisit this region and the localisers fail, the system will inherently record new information in that region, thus creating a new experience. Figure 4.1 details this principle of data collection for experiences.

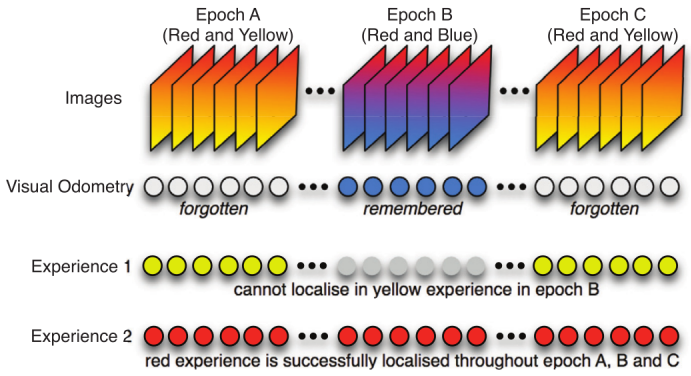


Figure 4.1: A graphic illustration of the data collection of experiences in the presence of others. Image credit: Churchill *et al.* [2]

This allows the system to naturally capture the regions of the environment that change both gradually and dramatically. In practice, this results in experiences that might not have information for a particular region of the workspace as prior experiences and their accompanying localisers were sufficient to localise. Churchill *et al.* overcomes the effects of large appearance change by capturing the change of the environment using frequent traversals and relying on localisation failures to determine when the environment is sufficiently different to warrant more data to be collected. This work has proven successful over three months with an accumulated distance of thirty-seven kilometres, demonstrating robust localisation in seasonal change and distinct areas.

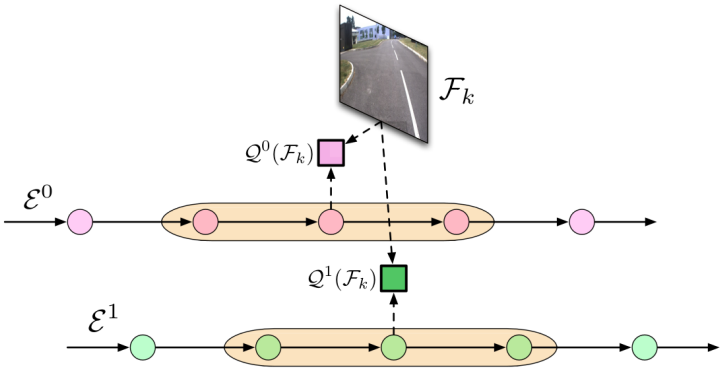


Figure 4.2: An example of the experience structure of Churchill. Image credit: Churchill [2]

Figure 4.2 describes the structure of an experience in the context of the work conducted by Churchill *et al.* [2] where Q is the assigned localiser, and the oval depicts the positions, or query images, within the experience that is compared for localisation. In a sense, a fence is placed around the last successful localisation, restricting the presented image's comparison to a select number of nodes.

4.3 Experience Generation

Similar to the work of Churchill *et al.* [2], the experiences in this work primarily consist of the data that has been collected during previous traversals to a workspace, different to Churchill *et al.* is that our experiences are constructed offline and are complete traversals of the workspace. Firstly, the inclusion of the entire traversal allows for a larger number of point that can be used for comparison with generated data. Should the traversals be incomplete, direct comparisons with generated cannot be made in regions that do not contain data and therefore will be limited to areas of high variablity as they will consist of the largest number of recorded data with respect to time, as per the implementation of Churchill *et al.* Secondly, as the generation of experiences occurs offline, all the data of a traversal is required to correctly reconstruct the transformation chain from the images using visual odometry. The process of experience generation can be broken into a few steps. Figure 4.3 shows the process from data collection to experience creation,

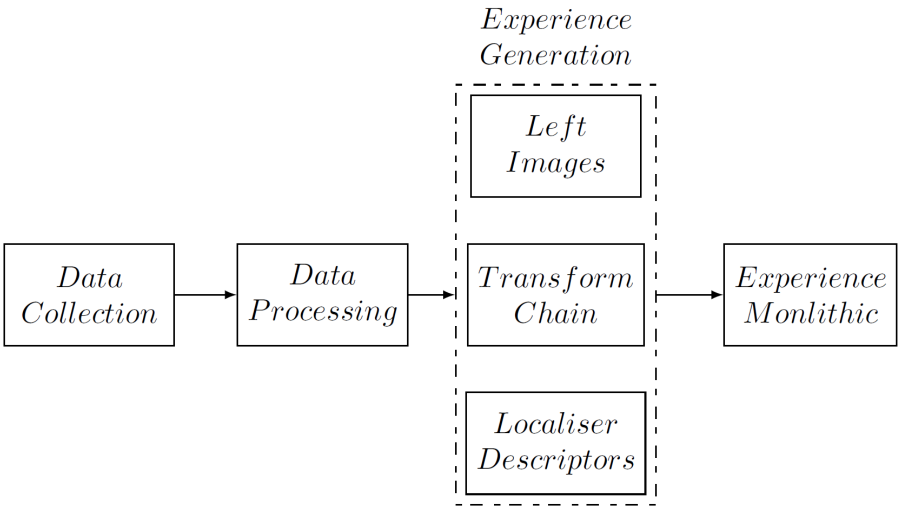


Figure 4.3: The experience generation workflow.

Data Collection and Processing

The data collection phase occurs during a human-piloted traversal of the environment. For the data input into the experience object, only the stereo images are recorded during the traversal. These images are then rectified and undistorted according to the process described in Chapter 2.2.1. These images are then used to generate the experience.

Experience Generation

The experience generation receives the stream of stereo image pairs and generates the visual odometry trajectory using the process described in Chapter 2.3. The left images are processed offline by the localiser using the trained vocabulary and Chow-Liu tree [38], as described in

Chapter 3, to generate the bag of words descriptors specific to that vocabulary and Chow-Liu tree pairing. These descriptors are used during the comparison process to determine the index the localiser returns as a successful localisation. In this manner, the experience monolithic, consists of images, bag of words descriptors and timestamps. Each image has an associated timestamp and bag of words descriptor, and the collection of these three items for each image is termed a node, graphical representation of how these nodes are used in practice can be seen in Figure 4.2 and Figure 4.4. The transformation chain created by the visual odometry pipeline and the series of nodes is stored separately.

4.4 Experience Structure using Keyframes

The structure for the experiences in this work is the same as that of Churchill *et al.* in that each node is connected to the previous and the subsequent node using transformations. These transformations have been decoupled from the nodes but are linked through timestamps. This decoupling allows for sub-sampling techniques to improve the localisation performance further. The sub-sampling technique implemented in this paper uses user-defined thresholds in the distance travelled and yaw angle observed and can be fine-tuned for localiser performance, however there are other implementations of keyframing techniques for improving visual place recognition [49], [50]. These thresholds are applied along the transformation chain, producing a sub-sampled experience monolithic. Thresholds ensure that regions that undergo large changes in the robot's pose will have more images stored, and straight sections will have fewer. As the primary means of localising in this work is appearance-based, the process of sub-sampling has the benefit of reducing the likelihood of false positive results or "jumping" that could occur in the presence of two or more query images that are highly similar in appearance, as these retained images have a lower level of visual similarity. Using these sub-sampling techniques, the experience object reports localisations less frequently but more accurately. The images retained using this sub-sampling method are termed keyframes in this work, and the intermediate frames between selected keyframes are discarded. Due to the decoupling of the visual odometry trajectory and the experience monolithic, this sub-sampling does not affect the trajectory of the experience. Thus, the entire transformation chain is retained. This is an important aspect of the decoupling as it allows for navigational functions such as teach and repeat to be performed using this localiser. Figure 4.4 shows the effect of reducing the nodes on the transformations.

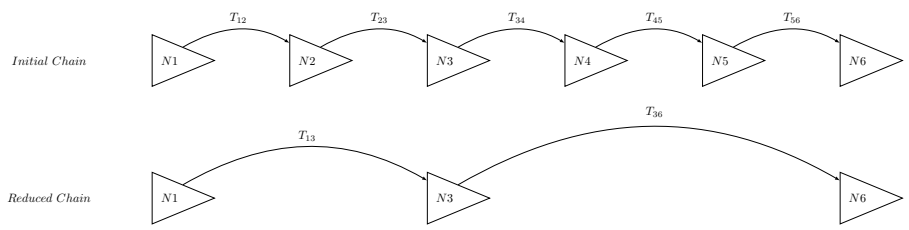


Figure 4.4: The process of reducing the nodes retains the transformation information between nodes. Where N depicts a node and where $N1$ through $N3$ might be a corner, and $N3$ through $N6$ would represent a straight section in the trajectory.

Chapter 5

Experience with Style Transfer

5.1 Introduction

Throughout the course of the thesis, there has been mention that previous solutions require a number of traversals to the environment to capture the appearance change of the environment and therefore increase the instances in which the robot can successfully localise. This however can be an issue in environments where data can only be sparsely collected. This chapter details a potential method of mitigating the requirement of frequent data collection while still attempting to capture the appearance change in the environment.

5.2 Background

There has been success using style transfer techniques on the input images of an experience-based localisation system as demonstrated by Porav *et al.* [3], in which they chose to alter the live images in order to successfully localise to experiences that are significantly different to the live stream. Building on the work of Zhu *et al.* [16], already discussed in Chapter 2.4, and using SURF feature descriptors, the Porav *et al.* [3] trained a cycle-consistency GAN to generate these synthetic images. They trained the network in two phases, the first phase using an unaligned dataset making use of the cycle consistency loss that is indicative of cycleGAN [16] as well as introducing a SURF feature descriptor-based loss function to ensure retention of the feature matching properties of the live image. In the second phase of the training, selected a small number of well-matched images by using a 6 degree of freedom ground truth, which is ultimately the pose and position of the robot at each instance, in effect choosing to select images that have highly similar viewpoints. They then applied an L1 loss to the feature descriptor and detector outputs. Porav *et al.* found a significant improvement in the number of matches, which is indicative of image similarity, present between the new synthetic image and the experience representing the adverse condition. This higher number of matches resulted in a vast improvement in the localisation performance and drastically reduced the lost distances.

Figure 5.1 shows the increased number of matches between images.

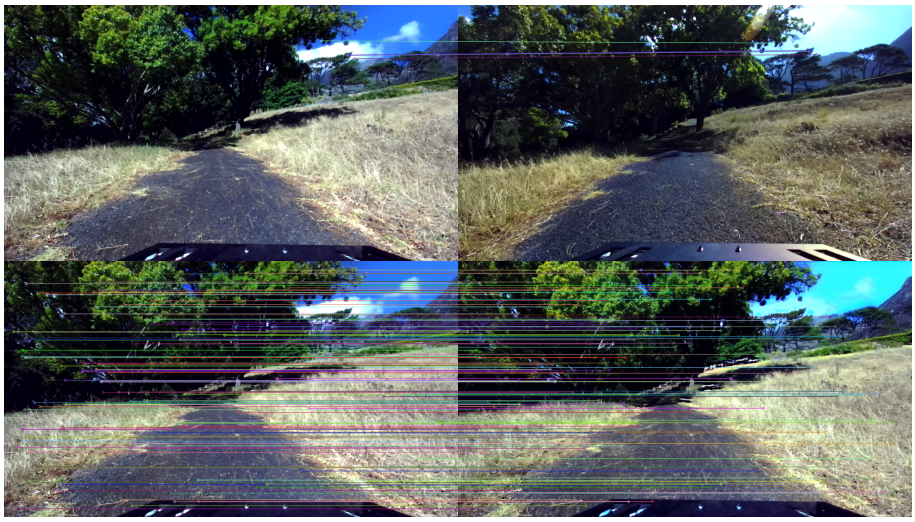


Figure 5.1: A comparison of the improvement in matching performance experienced where the coloured lines represent matches between the images.

However, the drawback of this approach is that there is a requirement to have already experienced the condition in which you are trying to localise and have a model already trained to do the image-to-image translation. Interestingly there is work done that perhaps could solve this issue in the form of an adaption of cycleGAN called "domain FLOW" [DLOW] [18]. DLOW is an adaption of the work done by Almahairi *et al.* in the creation of Augmented cycleGAN [augGAN] [17]. Augmented cycleGAN was created to meet some of the restrictions of cycleGAN [16]. The main limitation of cycleGAN is that although it can be trained on unpaired datasets, it can only learn one-to-one mappings, and these limitations are more pronounced in instances where the source and target domain are significantly different.

Almahairi *et al.* propose that by adding and additional variable to the latent space of the network during the training phase, additional variations of the images from both the source and the target domain can be explored by the network which addresses the limitation of the one-to-one and can now learn a many-to-many model. With the inclusion of this variable in cycleGAN the Almahairi *et al.* have been able to create many-to-many mappings after marginalising with respect to the latent variables. Similar to the work conducted in [17], Gong *et al.* introduce the concept of a "domainness" variable. This variable is used to regulate the style output of the images while training and in the case of the multiple output domains, the domainness variable is a vector containing variables describing the relationships between the source and all the target domains. The advantage of DLOW in comparison to augGAN is that due to the influence of the domainness variable during the training process, DLOW can produce intermediate frames between the source and the target domains, a graphic of which can be seen in Figure 5.2

In the case of a single target domain, this is a percentage translation between the two domains but in the case of multiple output domains, the output can be a mixture of all the target domains and is determined by the domainness vector input during image generation. This creation of intermediate domains allows the possibility of, given two extreme points such as day and night,

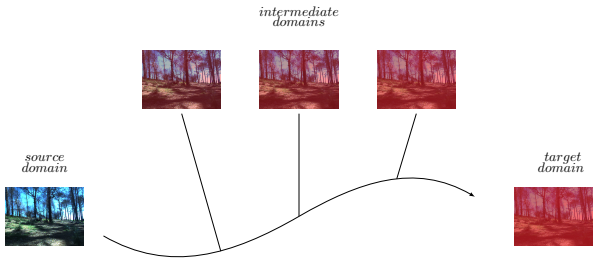


Figure 5.2: Representation of DLOW moving an image from the source to the target domain. Image adapted from [18]

an experience that matches mid-afternoon could be synthesised, thus reducing the requirement for frequent traversals to the environment.

5.3 Domain Flow [DLOW]

Although the primary version of the network is the instance in which there is a one-to-many mapping, Gong *et al.* has made available a version that learns a one-to-one mapping with the ability to create intermediate domains. Due to the complex nature of the one-to-many instance, it is convenient to discuss the network in the context of the one-to-one instance. The explicit purpose of the DLOW network is not to simply map the images from the source with a specific probablity distribution to the target domain with specific image characteristics and probability distribution but rather move the images from one domain to the next. This process creates a series of intermediate images with intermediate probability distributions that gradually change from the source to the target domain, a graphic of which is seen in Figure 5.2

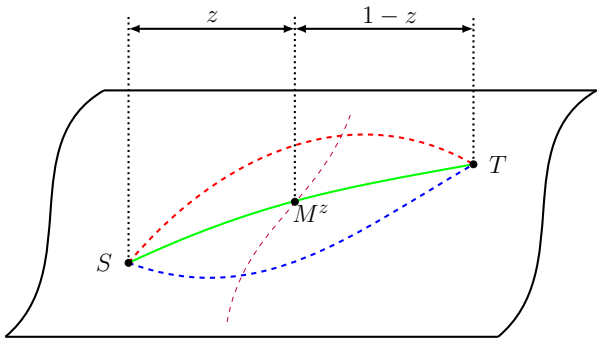


Figure 5.3: Statistical representation of the process of moving an image from the source to the target domain. Image adapted from [18]

Statistically, the path shown in Figure 5.3 is just one series of distributions connecting the source and target domains however, DLOW attempts to return the shortest geodesic path between the two domains. Expanding further, given an image M^z where $z \in [0, 1]$ and is a continuous domainness variable modelling the progression from source domain S to target domain T. In the event that $z = 0$, the intermediate domain distribution of M^z should be identical to that of S and when $z = 1$ the intermediate frame should be identical to that of T. This can be

achieved as the model makes use of the same cycle consistency loss that was introduced by cycleGAN [16]. Due to the nature of the generator moving the frame between the source and target domains, two discriminators are introduced to determine the efficacy of this process. One discriminator, D_s, acts to distinguish M^z from S and the other, D_t to distinguish M^z from T.

Figure 5.4 shows the structure of the model, which will assist during the discussion about the loss functions implemented.

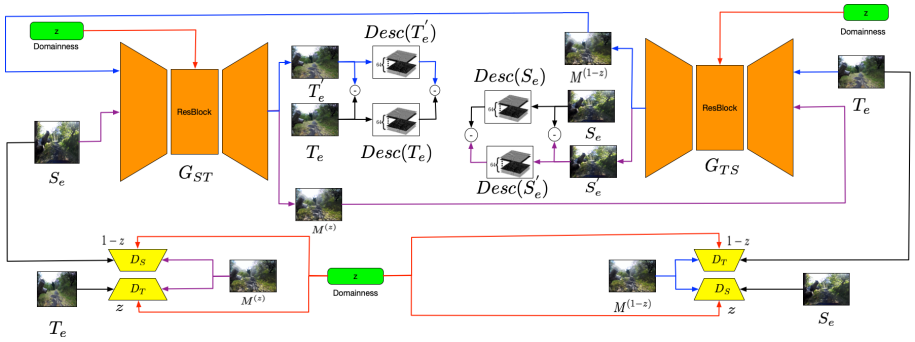


Figure 5.4: Graphic depiction of the DLOW network with a single source and target domain. Additional loss functions are also included. Image adapted from [18]

The adversarial loss functions for D_s and D_t are presented in Equation 5.1 and Equation 5.2 respectively.

$$L_{Sadv} = (D_S(G_{TS}(T_e)) - 1)^2 \tag{5.1}$$

$$L_{Tadv} = (D_T(G_{ST}(S_e)) - 1)^2 \tag{5.2}$$

The cycle consistency loss presented in [16] is enforced in order to maintain the semantics of the input image in the translated image but is altered to accommodate the additional discriminator, the new version of which can be seen in Equation 5.3.

$$\begin{aligned} L_{cyc} =& [||G_{ST}(G_{TS}(T_e, z), z) - T_e||_1] \\ &+ [||G_{TS}(G_{ST}(S_e, z), z) - S_e||_1] \end{aligned} \tag{5.3}$$

Due to the nature of image translation and the model learned by Generative Adversarial Networks [32], descriptor information that is required for the proper functioning of appearance-based systems can be lost. This loss in descriptor fidelity greatly reduces the performance of these systems when utilised with respect to generated images. In order to mitigate this effect, taking inspiration from the work conducted by Porav *et al.* [3], a descriptor-based loss function was created. As mentioned previously, this work primarily makes use of SURF descriptors [40] and therefore the SURF descriptor is a core component of the loss function developed. The first phase of the loss function calculates keypoints for both of the input images, S_e and T_e.

This set of keypoints is then applied to the respective input image as well as the associated generated image, in the case of S_e the generated image is represented by $S_s^{'}$ which is generated to be in the style of T_e. Using the same set of keypoints for both S_e and $S_s^{'}$ and T_e and $T_s^{'}$ respectively, the descriptor is calculated. An L1 loss is then applied to descriptors from the input image,, $Desc(S_e)$ and the generated image, $Desc(S_e^{'})$ as depicted in Equation 5.4 for the forward direction and Equation 5.5 for the backwards direction.

$$L_{Sdesc} = \sum_{i=1}^{n} |Desc(S_e) - Desc(S_e^{'})| \tag{5.4}$$

$$L_{Tdesc} = \sum_{i=1}^{n} |Desc(T_e) - Desc(T_e^{'})| \tag{5.5}$$

In the original input image, the keypoints represented areas of interest with respect to the SURF descriptor, applying these keypoints to the generated image and calculating the descriptors will be a measure of the degradation of the descriptors themselves during the translation process. So now while this L1 loss function is able to deal with the quality of the descriptors, it is not able to deal with the number of keypoints as the keypoints in both instances are the same ones. The number of keypoints is indicative of the overall quality of the feature-matching ability of the image, plainly speaking, the more keypoints present in an image the higher the probability that a sufficient number will match to allow the visual odometry and localisation pipelines to function appropriately. Therefore the second phase of the loss function is purely the ability of the input images to match with the generated image.

To achieve this, once again the SURF features and descriptors are found for both the input and the generated images, but this time no restriction is placed on the keypoints. These sets of features and descriptors are then matched using a brute force matcher utilising K nearest neighbours (k-NN) [51] which returns two matches per keypoint. In order to determine the quality of the matches returned, the Lowes ratio test [25] is applied to the results. The ratio test can be seen in Equation 5.6 where c is the threshold constant, A_i is the first match and B_i is the second match for a particular feature and n is the total number of matches between the images.

$$M_{Good} = \sum_{i=1}^{n} A_i < c(B_i) \tag{5.6}$$

The premise of Equation 5.6 is that given a match to a feature, a good match will be considerably closer to the point of interest than the next best match. The test is conducted by taking the first distance and evaluating it against the distance of the second-best match multiplied by a constant value between zero and one. This constant value allows the test to be as relaxed or strict as the application requires. For the loss function, the difference between the total number of matches and the number of good matches is calculated as seen in Equation 5.7, where M_{Good}

is the number of good matches passing the ratio test with a threshold value of 0.5. Maximising the number of good matches will reduce D_{diff} as M_{Good} approaches M.

$$D_{diff} = M - M_{Good} \tag{5.7}$$

The product of L1 loss function developed in the first phase and this difference is shown in Equation 5.8 and inherently penalises the generation of images with very poor matches. Due to the importance of the maintaining desciptor information therefore improving the matching quality of the generated images, it was found that simply adding the two terms together did not penalise the network adequately, the multiplication of the terms however proved more effective.

$$L_{Sdesc} = D_{diff} * L_{SDesc} \tag{5.8}$$

The complete generator objective $Lgen$ is shown in Equation 5.9 where each lambda term is a hyperparameter that weights the influence of each loss component.

$$L_{gen} = \lambda_{cyc} * L_{cyc} + \lambda_{desc} * L_{desc} + \lambda_{adv} * L_{adv} \tag{5.9}$$

5.4 Training Tests

Tests were conducted on the network in an attempt to characterise the outputs of the network and inform a training regime. Initially, the training attempts included the entire image as input and randomly cropped them to 80% of the original size. There were some initial successes however it became apparent that there is a switching phenomenon that occurs around a domainess value of 0.5. Figure 5.5 shows a significant drop in the average number of matches in this region of domainess values and is more pronounced with training time.

Figure 5.6 shows an extreme example of this attribute of the network. It is this attribute, along with the recommendations of Porav *et al.* [3] pertaining to the inclusion of a loss function as well as a fine-tuning stage with carefully selected images that training regime was developed. Moving from using the entire image to large images that are cropped considerably during an initial training phase and fine-tuning should result in a far better output from the network, as currently extreme conditions like that of Figure 5.6 result in DLOW placing the incorrect features in places where there is limited information in the source domain. Figure 5.7 shows a specific example of this. Figure 5.8 shows the results of a network trained using the loss function as well as the method proposed by Porav *et al.* [3].

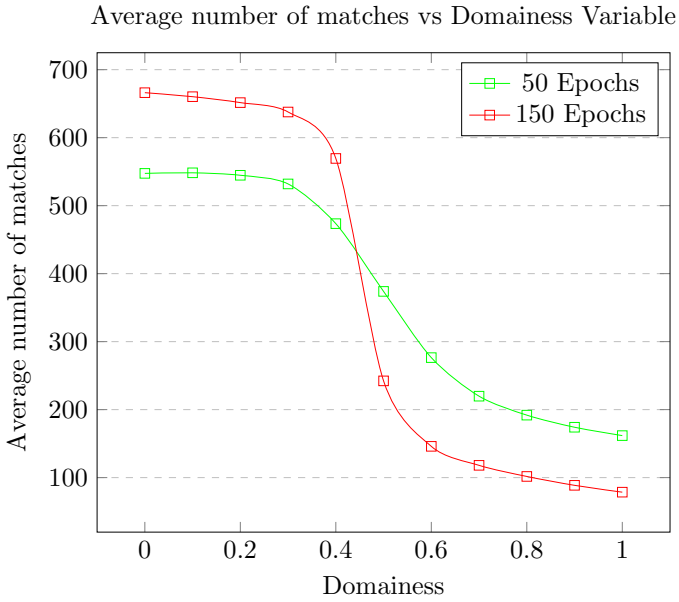


Figure 5.5: A comparison of the effect of training epochs on the number of average good matches

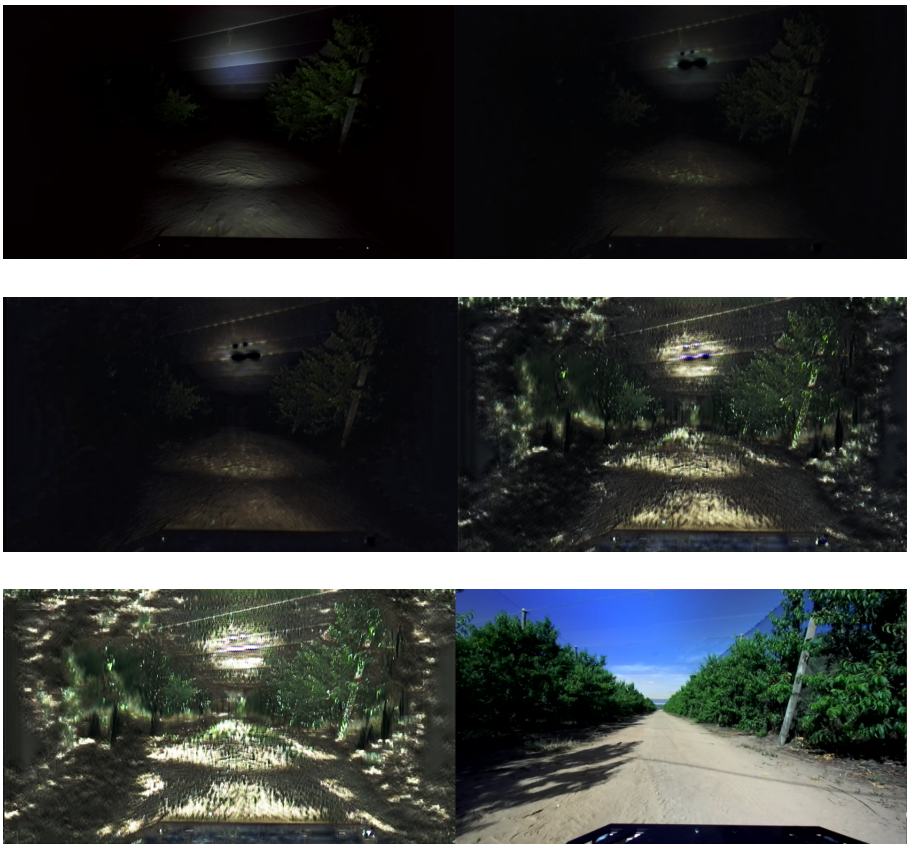


Figure 5.6: An extreme example of the extreme domain shift that occurs at a domainess value of approximately 0.5



(a) Real A **(b)** Fake B

Figure 5.7: An example of artefacts introduced by DLOW



(a) Real A **(b)** Domainess: 0.25



(c) Domainess: 0.5 **(d)** Domainess: 0.75



(e) Domainess: 1.0 **(f)** Real B

Figure 5.8: Output of the network trained in two phases, an initial general train and the second a fine-tuning phase conducted on a few select image pairs.

5.5 Training Regime

The training regime of the networks occurred in two phases similar to the work of Porav *et al.* [3]. The first phase was trained on an unaligned set of 1500 - 2000 random images from each representative condition. The images were resized from 1280x720 pixels to 720x720 pixels and then randomly cropped to 256x256 for training. The training was performed on an NVIDIA A100 GPU with a batch size of 2 over 110 epochs using the Adam optimiser [52] with a fixed learning rate of 0.0002. The second phase of the training used a small aligned set of data, approximately 10%. This small set of aligned data was determined similarly to the method used by Porav *et al.*, using the sequence of transforms developed by the visual odometry pipeline for the parent experience. The fine-tuning phase used a fixed learning rate of 0.0002 for the first 100 epochs, followed by a linearly decaying learning rate for the remaining 100 epochs.

5.6 Experience Generation

Referring back to Chapter 4.3, the content of an experience is primarily the images that were recorded during an outing to the environment and the output of an offline process acting on that data. In this chapter we have been discussing a framework with which we might be able to begin to synthesise or "hallucinate" experiences, this is a core functionality of this work and could mitigate the requirement of frequently collected data. In order to create an experience, a sequence of left and right images is required to generate the visual odometry trajectory, however, in the event that we are synthesising an experience from an existing one, the visual odometry has already been calculated and decoupled from the experience. Therefore only the left images need to be generated to create a new experience provided the timestamp information is retained to link the generated images to the relevant descriptors and transforms. The process of creating a generated Experience A' is shown in Figure 5.9

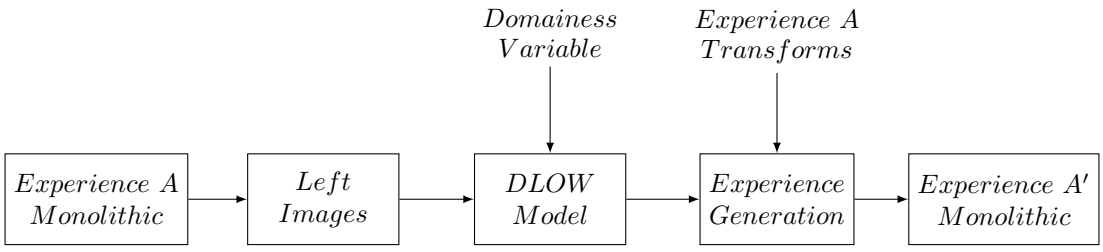


Figure 5.9: The process of creating a generated experience from an existing one.

Chapter 6

Experience Based Network

6.1 Introduction

This chapter details the experience-based network in its entirety, its structure and the process of determining which experience has returned the best localisation. Referring again to the work of Churchill *et al.* [2] this chapter will first discuss their implementation and then move on to discussing the version presented in this work.

6.2 Background

The system developed by Churchill *et al.* [2] was primarily used in long-term navigation. This was achieved, as previously mentioned by assigning each experience a localiser. The system would deem to have successfully localised if n-number of localisers return an adequate match. In their work, successful localisation across multiple experiences would trigger the creation of links or edges between the experiences such that the nodes in different experiences are linked together. The primary function of these links is to create an experience graph such that lost localisers assigned to experiences can be reinitialised should a linked node in another experience localise successfully. In doing so, the system inherently can navigate for longer distances as there is a collaborative effort between the experiences and their associated localisers. This is of particular importance as experiences, as they have defined them, might not contain the entire traversal and therefore at the start of a pass could fail to localise. Thus in areas of high variability in appearance, a larger number of localisers will come online and in areas of more subtle differences, localisers would effectively become useless. It is this property of the network that has allowed long-term navigation in their research environment. Figure 6.1 shows a graphical explanation of the experience graph.

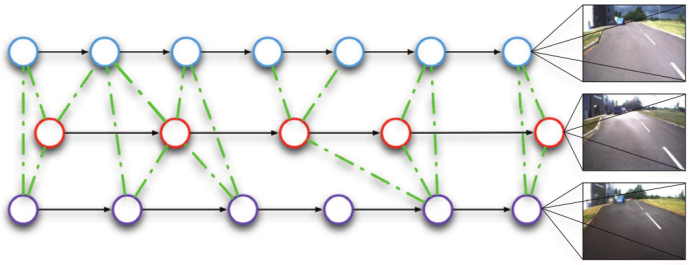


Figure 6.1: Graphic depiction of an experience-based network as developed by Churchill *et al.* The green dashed lines show the links between nodes. Note the different numbers of nodes in each experience instance. Image credit: Churchill *et al.* [2]

6.3 Experience Based Network

The difference in this works when compared to that of Churchill *et al.*, is in the implementation of the system and the creation of experiences. The difference in experience creation, where the experience is created offline using the entire traversal, has already been discussed in Chapter 4 so here we will discuss the implementation difference. In the work of Churchill *et al.*, the experience-based network is deeply integrated into the navigation system, the output of which is successful localisation and navigation. The output of our system is that of an image, which is fed into the downstream task of navigation to determine the point along a repeat trajectory during a repeat phase. It is, therefore, imperative that a single, accurately localised image is returned. To this end, if multiple localisers return localisations, a choice must be made. In the work of Churchill *et al.* [2] the creation of edges was primarily used to assist additional localisers that were lost, acting as re-initialisation points for them.

In the case of our implementation, there is less of a requirement for these edge links between experiences as each experience acts independently and the localiser has access to the full collection of nodes within the experience.I am primarily interested in the performance of the generated expereinces on the premise that generated data can be used in lieu of recorded experiences. It is for this reason that I have opted to not create edges between experiences as the localiser ought to rely wholely on the quality of the generated image, and not reducing the number of nodes that are available to the localiser enables a lost localiser to recover without intervention. Reducing the number of accessible nodes to the localiser would require the creation of experience edges. There could be an instance where, due to the nature of an agricultural environment and the similarities inherent in the crop structure, it could return a false positive that passes all the tests but is located at a different section of the trajectory.

We tuned our system to reduce the likelihood of false positives by using the method of subsampling, in which the associated trajectory is used to determine area of significant or minimal pose change and images are kept accordingly. It is assumed that if the robot has undergone a significant pose change the visual appaerance will also have undergone a significant change

and increasing the visual difference in images presented to the localiser increases the likelihood that a returned image will be correct. The trajectory thresholds can be fine tuned for each environment and are discussed in Chapter 4.3

6.4 Implementation and Performance Metrics

The network is constructed to have a central control hub to which all experience-localiser pairs present their results given a live image. Figure 6.2 describes this arrangement.

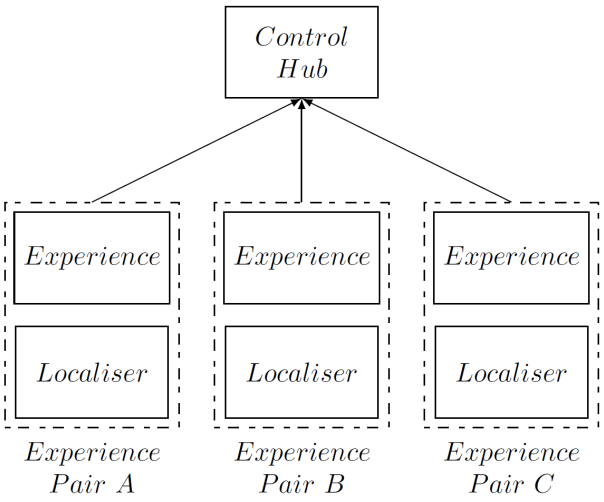


Figure 6.2: A graphic representation of the control hub and the experience pairs

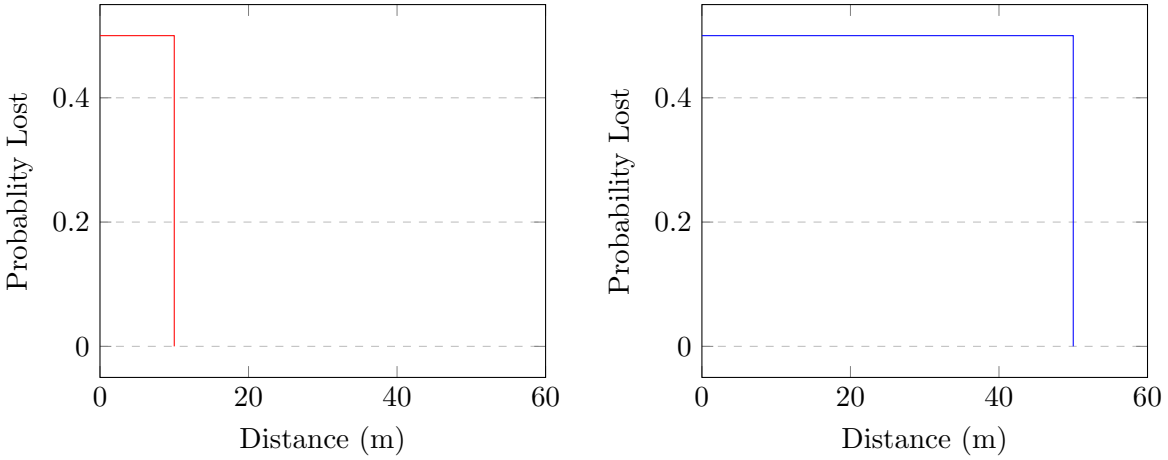
It is then the task of the central control hub to determine which of the experience results should be presented as the system's output. The function of the experience-localiser pair and the control hub is asynchronous to allow the addition and subtraction of experience-localiser pairs seamlessly and compensate for any processing time delays that could occur during operation.

6.4.1 Experience-Localiser Pair

Presented with a live image, the experience-localiser pair makes a comparison between each of the query images and the live image to determine the location of the robot, as discussed in Chapter 3.5. If the localisation is unsuccessful, the experience returns the failure to localise flag, indicating a new place is being visited and then repeats the process on the new live image. If the localisation is successful, the experience then interrogates the output of the localiser to ensure that it is not a false positive. Recall from Chapter 3.5 that the measure of a good localisation is the number of inliers that are present between the current live pair and the localised image. This number of inliers is due to the RANSAC [31] process that occurs in the visual odometry pipeline where a higher number of inliers is directly proportional to the similarity in the images leading to a robust transformation estimate, see Chapter 2.3 for more details. Therefore an additional function of the visual odometry pipeline is to distinguish true and false positive localisations. If

the image passes this visual odometry test, the image and the number of inliers are presented to the control hub.

To determine the efficacy of the localiser and experience pair, a performance metric is required. One such metric has already been introduced in Chapter 3.5, in the form of the number of inliers present in a match. Another metric used in this paper is a probability plot. The probability plot measures the probability that a robot will be lost up to a certain distance. Referring to Figure 6.3 to demonstrate the metric, both robots travelled a total distance of 100m however, one was lost five times for 10m each time, and the other only got lost once for a duration of 50m. We can then say that the localisation performance of the second robot is considerably worse due to the reliance on the visual odometry system.



(a) Five 10m failues

(b) One 50m failure

Figure 6.3: These plots show the probability that a robot will be lost for a given distance. In both cases the robot was lost for about 50% of the distance however the first robot was only lost for 10m per occasion.

6.4.2 Control Hub Implementation

It is the responsibility of the control hub to determine the best result and then present that as the output of the system. It does this simply by keeping a record of the current number of inliers that each experience-localiser pair has returned and presenting the result associated with the highest inlier value. In practice, false positives that do meet the inlier threshold do so normally with a low number of inliers and thus are disregarded in the presence of good localisations from the other experiences.

Chapter 7

Evaluation

7.1 Introduction

To test the localiser three distinct regions were used, namely Old Zoo, Lindenhof and Backsburg. Localisation experiments were conducted in all three environments with the majority occurring at the Old Zoo. This was primarily due to the accessibility of the area as well as the large variation in appearance due to flora in the environment. This chapter introduces the environments and research platform before discussing the experiments and their associated results.

7.2 Environments

7.2.1 Old Zoo

Old Zoo is an abandoned zoo located nearby the University of Cape Town's upper campus. The environment primarily consists of semi-tarred walkways, unmaintained flora as well as abandoned buildings. A key point of interest is an abandoned lion cage at the rear of the area. The vocabulary was created using images taken over a number of traversals and at numerous places located around the area. Figure 7.1 shows a few photos of the location.

The area was selected due to the variety of flora, with open sections and wooded areas along the path. The route, depicted in Figure 7.2, was chosen specifically for the variation in observed flora and containing points of simple and distinct turns. Figure 7.2 shows the path overlaid on a satellite image.



Figure 7.1: Photos of the Old Zoo environment



Figure 7.2: A satellite image of the path utilised in the Old Zoo environment

7.2.2 Lindenhof

The second environment of interest is a nectarine orchard located at a farm called Lindenhof near Paarl, Western Cape. The primary attribute of the environment is long and dense rows of leafy nectarine trees, boarded by wide farm roads, and the entire orchard was covered with a mesh netting. Figure 7.3 shows a few photos of the location.



Figure 7.3: Photos of the Lindenhof environment

The test location was selected due to the relatively neat nature of the rows. They had been mowed a few days prior, resulting in minimal vegetation between the rows of trees, such as grass, providing a good test case of the robot. Figure 7.4 details the routes used for the experiments and the vocabulary generation. In line with the recommendations in Chapter 3.5 the routes used to collect vocabulary images for each of the environments are considerably longer to improve the likelihood that the images used for training are diverse.



Figure 7.4: A satellite image of the paths utilised in the Lindenhof farm environment. Where green is the evaluation path and yellow is the vocab path. Markers indicate the start point.

7.2.3 Backsburg

The third environment utilised was a blueberry crop located at a farm called Backsburg near Paarl, Western Cape. The primary attribute of the environment is long and wide neat rows of low blueberry bushes with a ground covering. A farm road divides the blueberry crop into two distinct sections, and the entire blueberry crop is covered with a mesh netting. Compared to the first agricultural environment, the area is far more open. Figure 7.5 is a few photos of the location.

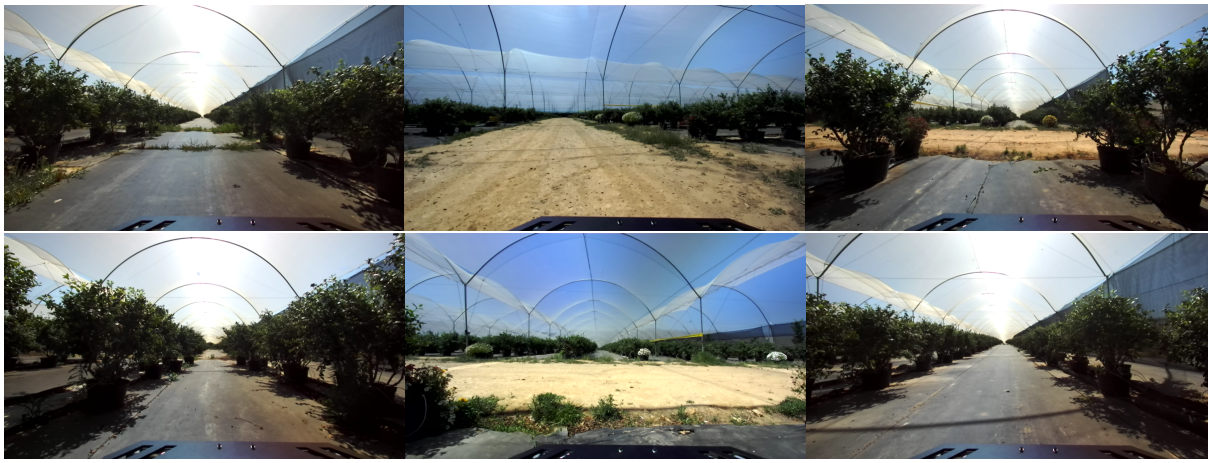


Figure 7.5: Photos of the Backenburg Farm environment

Figure 7.6 details the route used for experimentation as well as the route used to generate the vocabulary for the environment.



Figure 7.6: A satellite image of the paths utilised in the Backsburg farm environment. Where green is the evaluation path and yellow is the vocab path. Markers indicate the start point.

7.3 Research Platform

The platform this thesis uses is a Clearpath Husky, developed and manufactured by Canadian company Clearpath Robotics. The platform is a Robot Operation System [ROS] based robust outdoor four-wheeled robot. The robot uses skid steering and has a maximum speed of 1.0 m/s. Throughout this work, various additions were made to the robot, resulting in the final configuration described here. The primary sensor used is a StereoLabs ZED2i stereo camera which has a baseline of 120mm. Headlights were installed to capture early morning traversals. Figure 7.7 depicts the platform configuration.



(a) Husky **(b)** Husky with lights

Figure 7.7: Photos of the Clearpath Husky at the Old Zoo

7.4 Experiments

A number of data collections were conducted at the various locations. Localisation tests were conducted at Lindenhof and Backsburg to determine the performance of our localiser, for the different types of crops. The old zoo was primarily used for testing the efficacy of the experience-based network in environments of large visual change due to the variation in flora and the mixture of open and shaded areas. It was also used to measure the performance of the network in the presence of generated experiences.

7.4.1 Network Testing

Preliminary experiments were conducted without any image generation to determine the system's efficacy and lead to the discovery of inefficiencies and an openFABAP [36] implementation error. Inefficiencies were addressed with the introduction of a serialised data structure and the adoption of the robot operating system (ROS) framework, resulting in a network that not only is modular but has a localisation loop rate of 3-5Hz which is sufficient for a robot with a max speed of 1.0 m/s as the distance between localisations is less than the length of the robot.

7.4.2 Style Transfer

This section is primarily concerned with evaluating the ability of the DLOW network [18] to transfer the style of images from the source to the target domain by means of intermediate domain images. This evaluation was conducted in both the Lindenhof and the Old Zoo environments.

Old Zoo

Three models were trained using the Old Zoo dataset, one capturing the change indicative of the morning with the source domain at 04:30 and the target domain at 10:00, a second model capturing the change indicative of the afternoon with a source at 10:00 and a target at 16:00 and a final model attempting to capture the change of the entire day with a source at 04:30 and 16:00. Figure 7.8 is a comparison of the same viewpoint at the three different times.



(a) 04:30 **(b)** 10:00 **(c)** 16:00

Figure 7.8: A visual depiction of the three timeslots used as domains to train the Old Zoo models

The output for the models capturing the change for the morning, afternoon and entire day are shown in Figure 7.9 to Figure 7.14. The results of the models concerned with shifting domains across smaller time-gaps such as 10:00 to 16:00 yield better results, as can be seen in Figure 7.9, 7.10, 7.11 and Figure 7.12 where the domain shift is less pronounced.



(a) 04:30 **(b)** Domainess: 0.33 **(c)** Domainess: 0.66 **(d)** 10:00 **(e)** Domainess: 1.5

Figure 7.9: Output of the model trained to move images from source 04:30 to target 10:00



(a) 10:00 **(b)** Domainess: 0.33 **(c)** Domainess: 0.66 **(d)** 04:30 **(e)** Domainess: 1.5

Figure 7.10: Output of the model trained to move images from source 10:00 to target 04:30



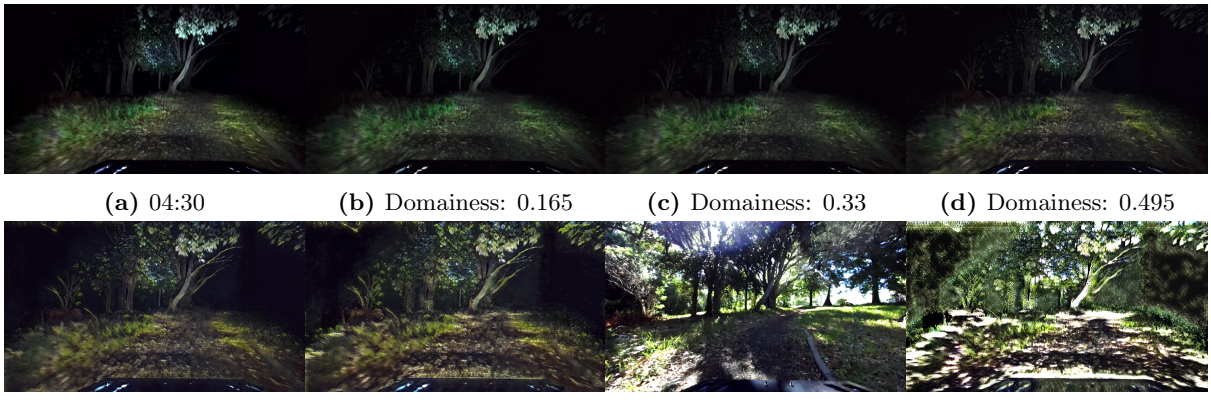
(a) 10:00 **(b)** Domainess: 0.33 **(c)** Domainess: 0.66 **(d)** 16:00 **(e)** Domainess: 1.5

Figure 7.11: Output of the model trained to move images from source 10:00 to target 16:00



(a) 16:00 **(b)** Domainess: 0.33 **(c)** Domainess: 0.66 **(d)** 10:00 **(e)** Domainess: 1.5

Figure 7.12: Output of the model trained to move images from source 16:00 to target 10:00

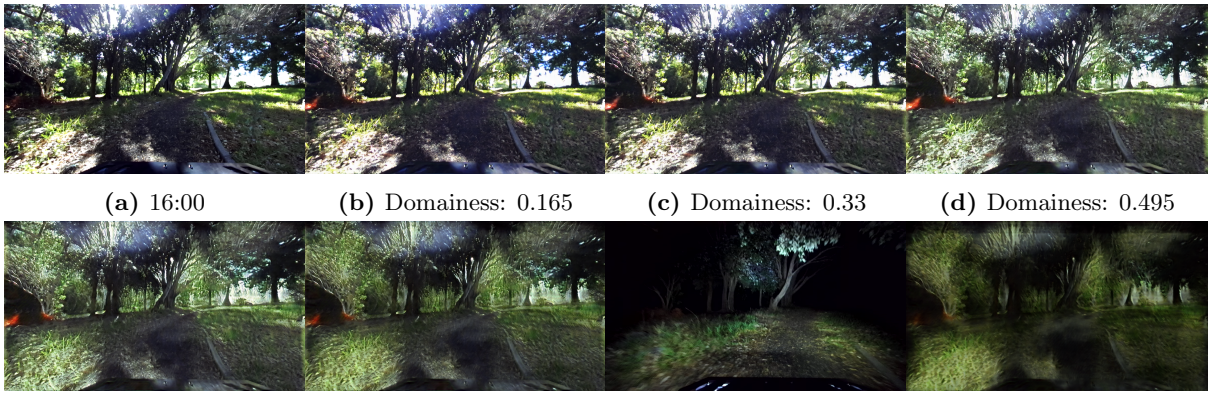


(a) 04:30 **(b)** Domainess: 0.165 **(c)** Domainess: 0.33 **(d)** Domainess: 0.495

(e) Domainess: 0.66 **(f)** Domainess: 0.825 **(g)** 16:00 **(h)** Domainess: 1.5

Figure 7.13: Output of the model trained to move images from source 04:30 to target 16:00

Referring to Figure 7.13, there is a clear progression from 16:00 to 04:30 but a severe degradation in the image quality with a distinct jump between domainess values of 0.495 and 0.66. Referring



(a) 16:00 **(b)** Domainess: 0.165 **(c)** Domainess: 0.33 **(d)** Domainess: 0.495

(e) Domainess: 0.66 **(f)** Domainess: 0.825 **(g)** 04:30 **(h)** Domainess: 1.5

Figure 7.14: Output of the model trained to move images from source 16:00 to target 04:30

to Figure 7.14, similar behaviour is witnessed with many artefacts being introduced after a domainess of 0.495. Although the network was trained specifically to avoid instances of this nature, it stands to reason that the current state of the model is inadequate to capture the change appropriately, or it is a function of the low light performance of the cameras used in this study resulting in images with less data available in low light conditions.

Lindenhof

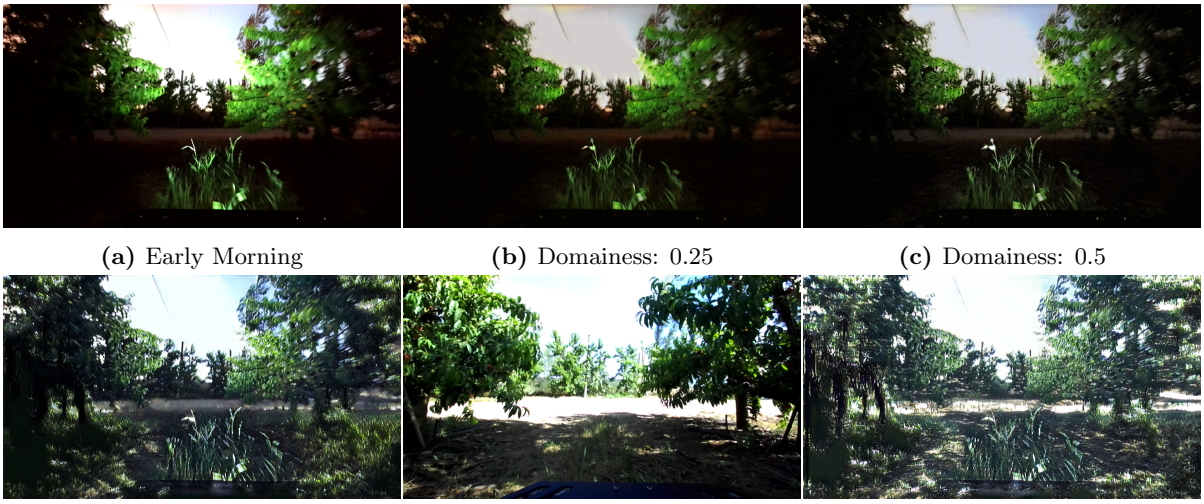
A single model was trained using data collected in an early morning session, and during the day, Figure 7.15 is a comparison of the same viewpoint at two different times. The output of the respective models are presented in Figure 7.16 and Figure 7.17. The phenomenon that occurs at a domainess value around 0.5 can also be clearly seen here.



(a) Early Morning **(b)** Daytime

Figure 7.15: A visual depiction of the two domains used to train the Lidenhof model

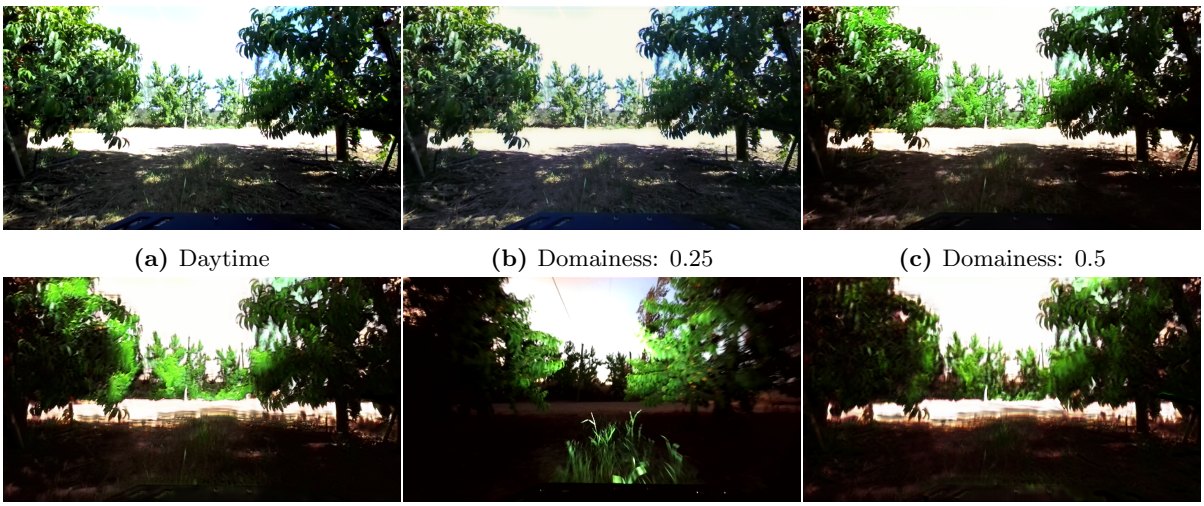
Again the domain shift jump can be seen in Figure 7.16 and the network performs better when converting from the high information domain of Daytime to Early Morning, see Figure 7.17



(a) Early Morning **(b)** Domainess: 0.25 **(c)** Domainess: 0.5

(d) Domainess: 0.75 **(e)** Daytime **(f)** Domainess: 1.5

Figure 7.16: Output of the model trained to move images from source 10:00 to target 16:00



(a) Daytime **(b)** Domainess: 0.25 **(c)** Domainess: 0.5

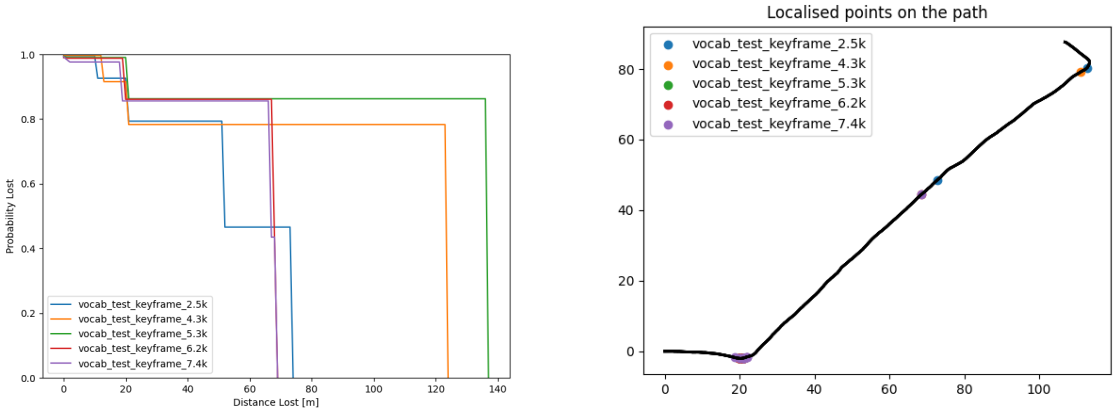
(d) Domainess: 0.75 **(e)** Early Morning **(f)** Domainess: 1.5

Figure 7.17: Output of the model trained to move images from source 10:00 to target 16:00

7.4.3 Localiser Testing in Agricultural Regions

Localisation tests were conducted at Lindenhof and Backsburg due to the disparate nature of the two regions. To test the efficacy of the localiser in each of these regions, a number of vocabularies were produced. As the vocabulary is a principal factor in the system's performance, vocabularies of various sizes were created and compared with each other. Concerning the Lindenhof environment, the experiences were generated from passes of the route shown in Figure 7.4 and sub-sampled using the keyframe generator with thresholds of 2.0 m set for distance travelled and 0.2536 radians for yaw angle. Figure 7.18 shows the results of the localisation test conducted at Lindenhof.

The Lindenhof route used to evaluate is the second half of the route shown in Figure 7.4.



(a) Probability **(b)** Localisation Points

Figure 7.18: Results of the localisation test conducted on the Lindenhof dataset

The route is 157.8m and consists of road sections on either side, but most of the path is the nectarine row itself. Referring to Figure 7.18, the performance leaves much to be desired. There is clearly a cluster of localisations on each road portion but a failure to localise for practically the entire nectarine row with only a small number of localisations in the centre. Interestingly the vocabularies of 2300 words, 6200 words and 7400 words performed the best, but there will still be a heavy reliance on the visual odometry system to estimate position along the row. Figure 7.19 shows the system's output in the road section, while Figure 7.20 describes the output in the nectarine row.



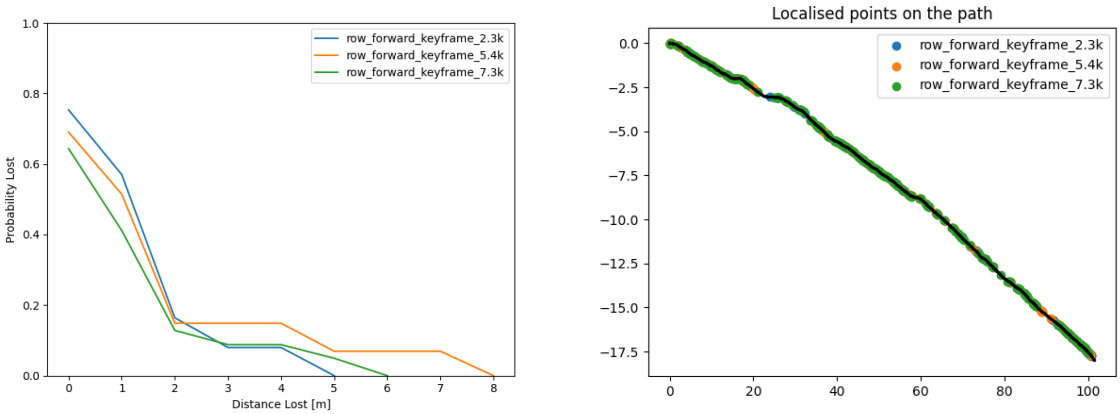
Figure 7.19: Example of localisation output of each vocabulary for road section, where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the size of vocab used.

The same experiment was conducted on the Backsburg dataset, using the same method for generating and sub-sampling the experiences. The route used at Backsburg is made up entirely of a row of blueberries, with only a few frames of the road at the end of the route. Figure 7.21 shows the result of the localisation test conducted at Backsburg.

Figure 7.21 shows consistent localisation across all vocabularies sizes along the 104.7m route, with vocabularies consisting of 2400 words and 7300 words performing the best, only getting lost for a maximum of 5m and 6m, respectively. Figure 7.22 shows the system's output and can begin to indicate the reason for the extreme difference in localisation performance between the



Figure 7.20: Example of localisation output of each vocabulary for nectarine row section, where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the size of vocab used.



(a) Probability **(b)** Localisation Points

Figure 7.21: Results of the localisation test conducted on the Backsburg dataset

two regions.



Figure 7.22: Example of localisation output of each vocabulary, where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the size of vocab used.

Comparing Figures 7.20 and 7.22, there is a vast difference in the visual input of the system. When comparing Figure 7.19, where the localiser at Lindenhof performed well with Figure 7.22, we can see visual similarities in the lighting and visual input into the system. Suggesting that there is an environmental condition in which the system performs optimally.

7.4.4 Localiser Testing with Style Transfer

Localisation tests concerning the time of day were conducted at Old Zoo. 11 hours of data were collected from 04:30 to 16:00, with traversals being conducted at 2-hour intervals for the route shown in Figure 7.2. At each instance, two traversals were conducted, one for evaluation and one for experience generation. Experiences were generated using a 6000 word vocabulary.

For three of the time collections, the test data collected was compared to its sibling collection and all other times recorded, with the number of good localisations logged. The experiences were generated and sub-sampled using the keyframe generator with thresholds of 2.0 m set for distance travelled and 0.2536 radians for yaw angle. These results are compared in Figure 7.23

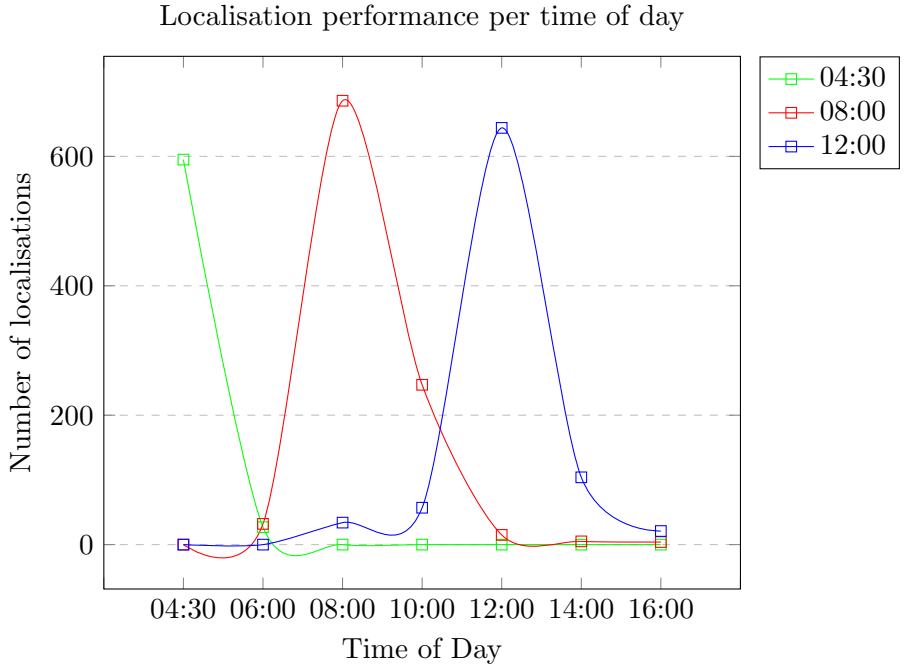


Figure 7.23: The effect of appearance change due to time of day on the performance of the localiser in the Old Zoo.

Large peaks correspond to the instances where the test data was compared to the experience data collected at the same timeslot, and are practically near identical images hence a large number of localisations. The same experiment was then conducted using 08:00 as the test stream and generated images for the experiences. The plot for 08:00 is repeated, and the results of the generated experiences are plotted as scatter points on the graph in Figure 7.24

There is a strong correlation between the number of good localisations obtained from the experience and its parent data collection, in the sense that an experience made to mimic 14:00 from data collected at 10:00 outperforms the data collected at 14:00 in terms of good localisations. This indicates the loss function successfully retains the descriptor information from the parent data and transfers it to the generated experiences. This is further confirmed by looking at DLOW 16:00 and DLOW 04:30 in Figure 7.24. In each case, the parent experiences failed to localise, and thus the generated ones failed as well.

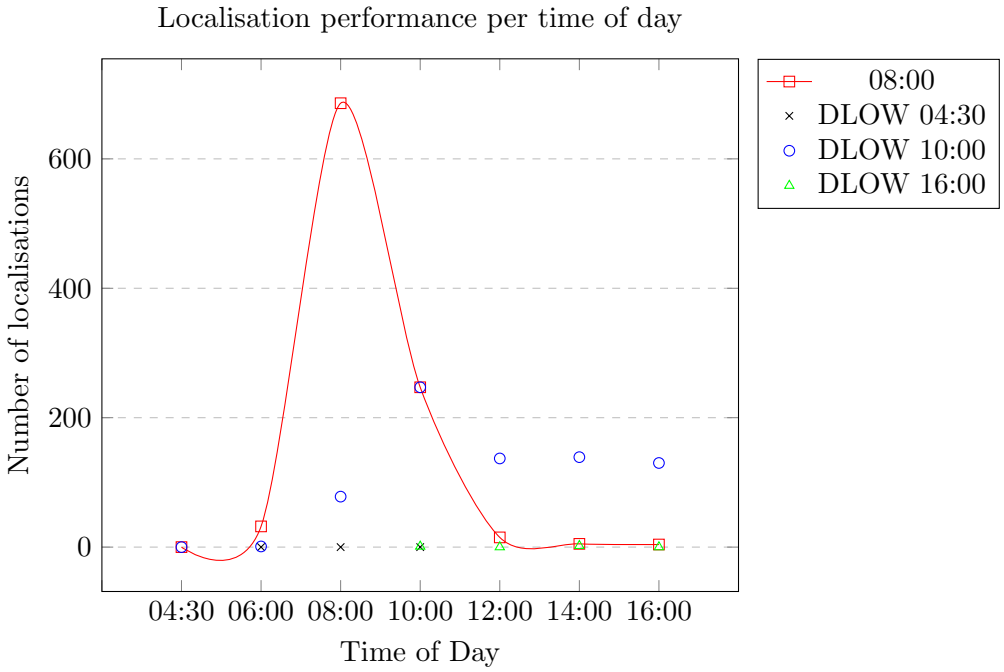


Figure 7.24: The effect of appearance change due to time of day on the performance of the localiser at Old Zoo in the presence of generated experiences. Where DLOW 10:00 are images created from the collection at 10:00 and transferred to either 16:00 or 04:30. DLOW 16:00 are images created from the collection at 16:00 being transferred to 10:00 and DLOW 04:30 are images created from the collection at 04:30 also being transferred to 10:00

Incorporating the generated and recorded experiences into two experiments where the collection at 08:00 was used as the test stream. The first experiment incorporated all the experiences generated from the 10:00 data collection, and the second compares the afternoon data collections and the generated experiences. Figure 7.25 shows a sample output from the localiser from the first experiment, followed by the probability plot in Figure 7.26.



Figure 7.25: A still of the localisation output for the experiment comparing all the experiences generated from the 10:00 data collection.

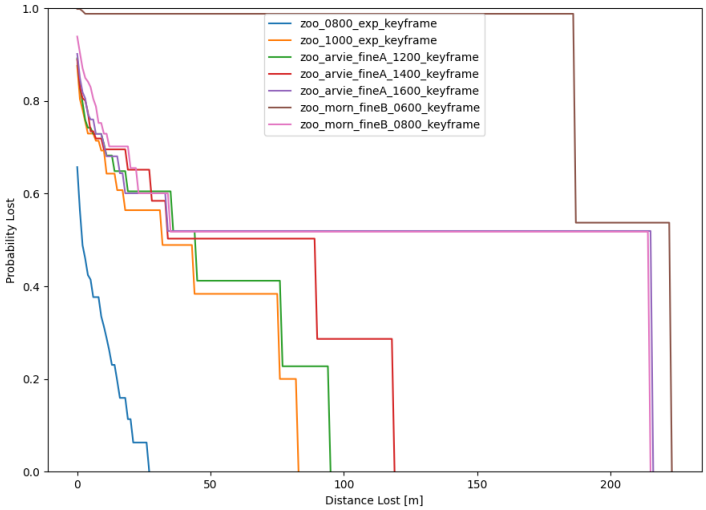


Figure 7.26: The lost probability of all the experiences generated from the data collection at 10:00.

Experiences generated from the model capturing the afternoon appearance change outperform the experiences generated using the model trained to capture the change with respect to the morning. Now concerning the second experiment, Figure 7.28 shows the probability that each of the localisers is lost for a certain distance, and Figure 7.27 shows the points along the trajectory that were successfully localised to.

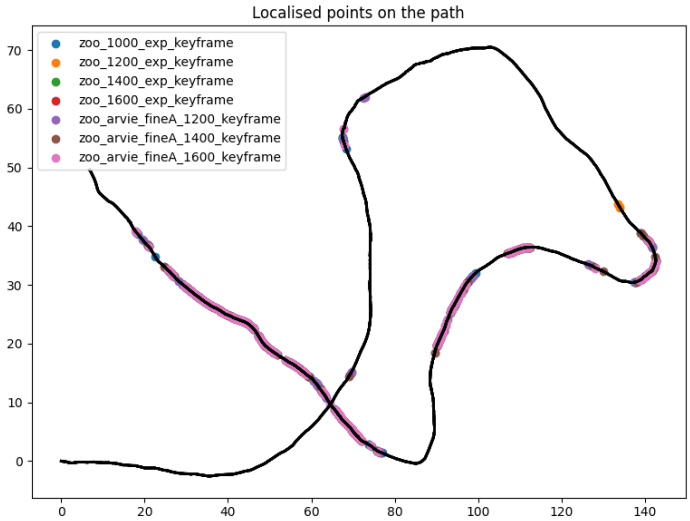


Figure 7.27: The regions along the trajectory that were successfully localised to. The overlapping of points suggest a high number of successful localisations in those regions.

There are instances where the generated experiences localised before the parent experience, as evidenced by Figure 7.27, effectively reducing the lost distance in each instance, and referring

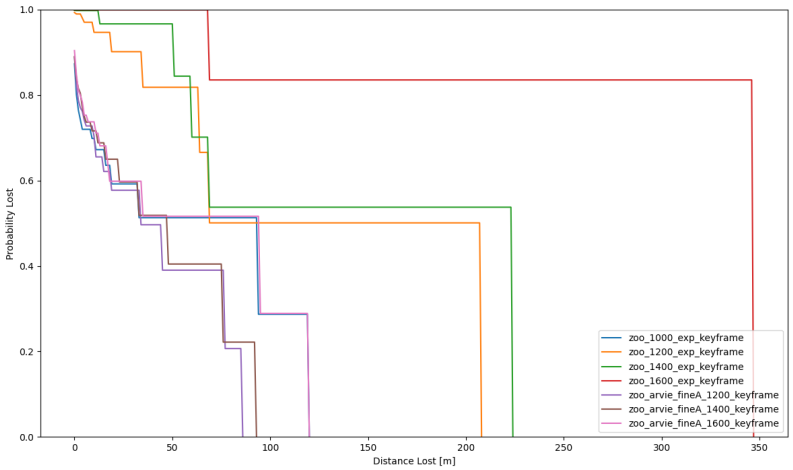


Figure 7.28: The probability plot of the individual experiences where data collected at 08:00 was used as the live stream.

to Figure 7.28, we can clearly see that each generated experience outperforms its mimicked experience another indication that the descriptor loss function performed correctly. However, the system's true strength is shown in Figure 7.29 when the collection of generated experiences outperforms the parent experience.

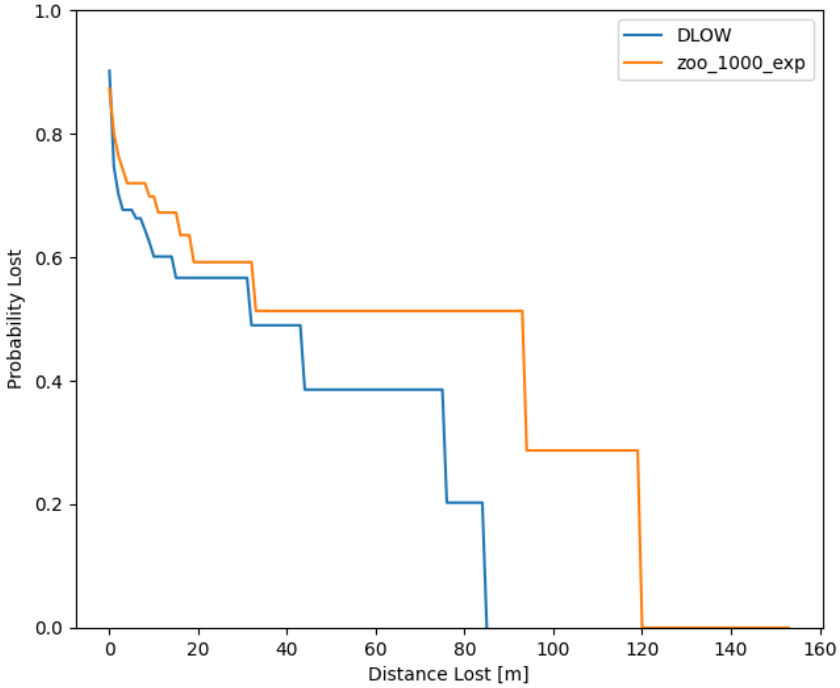


Figure 7.29: The probability plot of the collection of DLOW experiences vs the parent experience

Figure 7.29 indicates that the images created by the DLOW network are different enough such

that in instances where the parent experience fails, the generated ones succeed. If this were not the case the probability plots would be far similar and in fact, referring to Figure 7.28 we can see this condition when comparing *zoo_arvie_1600_keyframe* and *zoo_1000_keyframe* lines in the plot, as the performance of *zoo_arvie_1600_keyframe* is purely due to the descriptors retained from the original experience. An instance where the parent experience fails to localise but the generated ones do can be seen in Figure 7.30

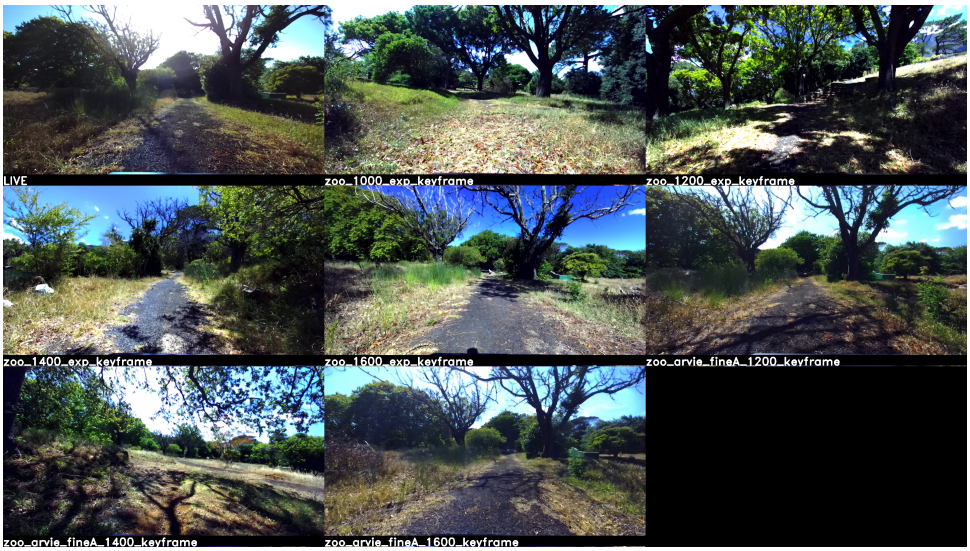


Figure 7.30: Example of localisation output where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the time of day mimicked and names containing *fineA* are generated.

This result is corroborated by an experiment performed on a smaller section near the lions cage at the Old Zoo earlier in the year. The experiences were generated from data collection at 06:00 and 14:00. Figures 7.31 and Figure 7.32 show the results.



Figure 7.31: Example of localisation output of the previous experiment, where the tile labelled "Live" is the image presented to the localisers and the subsequent outputs are labelled corresponding to the time of day mimicked. Times in grey are recorded and times in orange are generated.

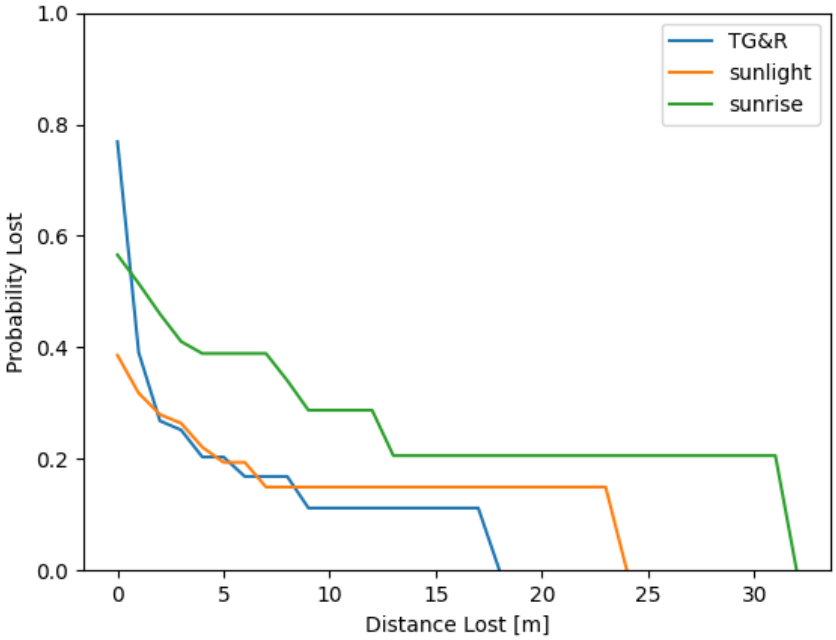


Figure 7.32: The lost probability of the experiment. Where DLOW represents the experiences as a collection, sunrise is the performance of the parent data collection and sunny is the performance of the collection closest in appearance to Live.

Chapter 8

Conclusions

This thesis set out to determine if it is possible to augment appearance-based localisation systems utilising experiences for outdoor environments that undergo large changes in appearance and whether we would be able to capture this change using sparsely collected data.

Chapter 3 introduced the core system utilised in this work and gave insights into the components with which the performance is closely tied and informed the choice of using a 6000-word vocabulary.

Chapter 4 introduced the experience with which each localiser was paired and the choice to use sub-sampled experiences to improve performance. Chapter 5 took this a step further by generating experiences from collected data, it is here where the limitations of the DLOW [18] network were first encountered and attempted to be mitigated by adjusting the training regime after some initial tests.

Chapter 7 presents the environments in which the testing occurred, each testing an aspect of the system to accurately determine the efficacy. The results at Lindenhof suggest that the cameras themselves, outside of the use of generated experiences, might be a cause for concern, failing to localise for practically the entire length of the dense, dark and and leafy nectarine row, see Figure 7.18 and comparatively performing excellently in the Backsburg environment with it large open rows of blueberries, Figure 7.21, which in contrast is open and well lit.

At the Old Zoo, we address the question proposed earlier: Can this system improve localisation under large appearance change? We were primarily concerned with the change indicative of the time of day and, to this end, captured the extremes attempting to use them to generate experience that can be used to localise during the day. In this task, the system faltered, with the network developed in Chapter 5 failing to produce images of the required quality in these extreme cases. However, the system improved localisation for more moderate changes in appearance, as evidenced by the generated experiences outperforming the respective collected experiences in two experiments.

These generated experiences were created from a data collection that performed well in its

own right, the data collection at 10:00, and suggests that the loss function developed to retain the matching performance of the parent experience functions correctly, successfully creating experiences that localise more often than the experience it was set to mimic, Figure 7.29. In the presence of moderate changes, DLOW can successfully alter images, albeit marginally, such that the collection of generated experiences outperforms the parent experience, which is closer to how the network is implemented in practice, as the experience-localiser pairs are not utilised in isolation but in the network.

Future Work

The improvement in the localisation performance of the appearance-based localisation system utilising a mixture of generated and recorded experiences for an albeit narrow change in appearance opens new opportunities for future work to be conducted

Agricultural dataset creation

There are excellent datasets created for urban and structured environments, such as KITTI Dataset [53], Newer College Dataset [54] and Oxford RobotCar Dataset [55]. However, datasets pertaining to agricultural regions are lacking, and the few available such as FieldSAFE [56] contain limited data and datasets like Chebrolu *et al.* are great for plant identification, but not in row navigation tasks. A dataset containing a suite of high-resolution sensors that captures the change in the environment due to farming activities, lighting and seasons would be beneficial for future research in agricultural environments.

Expanding application window for generated experiences

The system proposed in this work was limited by the output of the GAN which in turn was limited by the low-light performance of the camera. The creation of a dataset can improve the data input, but the performance gains seen from including generated experiences warrant research into the generation of images for localisation. It might be an interesting endeavour to investigate the use of lighting invariant images as training images for the model, perhaps that can further expand the range?

Replacing DLOW backbone

DLOW is a GAN model created in 2018/2019 and since then there have been some advances in other models. Replacing DLOW with other more modern models, or intoducing additional components to bring it more in line with current trends will be beneficial.

Bibliography

[1] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth, "Fab-map + ratslam: Appearance-based slam for multiple times of day," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 3507–3512. DOI: 10.1109/ROBOT.2010.5509547.

[2] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, Sep. 2013. DOI: 10.1177/0278364913499193. [Online]. Available: https://doi.org/10.1177%5C%2F0278364913499193.

[3] H. Porav, W. Maddern, and P. Newman, *Adversarial training for adverse conditions: Robust metric localisation using appearance transfer*, 2018. DOI: 10.48550/ARXIV.1803.03341. [Online]. Available: https://arxiv.org/abs/1803.03341.

[4] E. Brosh, M. Friedmann, I. Kadar, *et al.*, *Accurate visual localization for automotive applications*, 2019. DOI: 10.48550/ARXIV.1905.03706. [Online]. Available: https://arxiv.org/abs/1905.03706.

[5] H. Taira, M. Okutomi, T. Sattler, *et al.*, *Inloc: Indoor visual localization with dense matching and view synthesis*, 2018. DOI: 10.48550/ARXIV.1803.10368. [Online]. Available: https://arxiv.org/abs/1803.10368.

[6] G. Adorni, S. Cagnoni, S. Enderle, *et al.*, "Vision-based localization for mobile robots," *Robotics and Autonomous Systems*, vol. 36, no. 2, pp. 103–119, 2001, Viewing, Sensing, Coordination and Learning in EuroRoboCup 2000, ISSN: 0921-8890. DOI: https://doi.org/10.1016/S0921-8890(01)00138-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889001001385.

[7] D. Yuen and B. MacDonald, "Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 217–226, 2005. DOI: 10.1109/TRO.2004.835452.

[8] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization by combining an image-retrieval system with monte carlo localization," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 208–216, 2005. DOI: 10.1109/TRO.2004.835453.

[9] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021. DOI: 10.1109/

tro.2021.3075644. [Online]. Available: https://doi.org/10.1109%5C%2Ftro.2021.3075644.

[10] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 1643–1649. DOI: 10.1109/ICRA.2012.6224623.

[11] J. J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*, 2014.

[12] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, "Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," *Robotics and Autonomous Systems*, vol. 117, Apr. 2019. DOI: 10.1016/j.robot.2019.03.012.

[13] S. Lowry, N. Sünderhauf, P. Newman, *et al.*, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016. DOI: 10.1109/TRO.2015.2496823.

[14] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008. DOI: 10.1177/0278364908090961. eprint: https://doi.org/10.1177/0278364908090961. [Online]. Available: https://doi.org/10.1177/0278364908090961.

[15] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, *Netvlad: Cnn architecture for weakly supervised place recognition*, 2016. arXiv: 1511.07247 [cs.CV].

[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, 2017. DOI: 10.48550/ARXIV.1703.10593. [Online]. Available: https://arxiv.org/abs/1703.10593.

[17] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville, *Augmented cyclegan: Learning many-to-many mappings from unpaired data*, 2018. DOI: 10.48550/ARXIV.1802.10151. [Online]. Available: https://arxiv.org/abs/1802.10151.

[18] R. Gong, W. Li, Y. Chen, and L. Van Gool, *Dlow: Domain flow for adaptation and generalization*, 2018. DOI: 10.48550/ARXIV.1812.05418. [Online]. Available: https://arxiv.org/abs/1812.05418.

[19] "How farm automation is revolutionizing agricultural industry — world economic forum." (NaN), [Online]. Available: https://www.weforum.org/agenda/2023/07/farm-automation-technology-revolutionizing-agriculture/ (visited on 08/26/2023).

[20] "The future of agriculture companies is automated — bcg." (NaN), [Online]. Available: https://www.bcg.com/publications/2020/benefits-of-automation-in-the-agriculture-industry (visited on 08/26/2023).

[21] "Smart farming: How automation is transforming agriculture — eastern peak." (NaN), [Online]. Available: https://easternpeak.com/blog/smart-farming-how-automation-is-transforming-agriculture/ (visited on 08/26/2023).

[22] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. Cambridge University Press, ISBN: 0521540518, 2004.

[23] C. Hahne, A. Aggoun, V. Velisavljevic, S. Fiebig, and M. Pesch, "Baseline and triangulation geometry in a standard plenoptic camera," *International Journal of Computer Vision*, vol. 126, no. 1, pp. 21–35, Aug. 2017. DOI: 10.1007/s11263-017-1036-4. [Online]. Available: https://doi.org/10.1007%5C%2Fs11263-017-1036-4.

[24] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[25] G. LoweDavid, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.

[26] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," vol. 3951, Jul. 2006, pp. 404–417, ISBN: 978-3-540-33832-1. DOI: 10.1007/11744023_32.

[27] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," vol. 3951, Jul. 2006, ISBN: 978-3-540-33832-1. DOI: 10.1007/11744023_34.

[28] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," Nov. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[29] M. Agrawal, K. Konolige, and M. Blas, "Censure: Center surround extremas for realtime feature detection and matching," vol. 5305, Oct. 2008, pp. 102–115, ISBN: 978-3-540-88692-1. DOI: 10.1007/978-3-540-88693-8_8.

[30] D. Chotrov, Z. Uzunova, Y. Yordanov, and S. Maleshkov, "Mixed-reality spatial configuration with a zed mini stereoscopic camera," Nov. 2018.

[31] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, ISSN: 0001-0782. DOI: 10.1145/358669.358692. [Online]. Available: https://doi.org/10.1145/358669.358692.

[32] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, *Generative adversarial networks*, 2014. DOI: 10.48550/ARXIV.1406.2661. [Online]. Available: https://arxiv.org/abs/1406.2661.

[33] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976. DOI: 10.1109/CVPR.2017.632.

[34] M. J. Cummins and P. Newman, "Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model," in *International Conference on Machine Learning*, 2010.

[35] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, Nov. 2010. DOI: 10.1177/0278364910385483. [Online]. Available: https://doi.org/10.1177%5C%2F0278364910385483.

[36] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth, "OpenFABMAP: An open source toolbox for appearance-based loop closure detection," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, May 2012. DOI: 10.1109/icra.2012.6224843. [Online]. Available: https://doi.org/10.1109%5C%2Ficra.2012.6224843.

[37] Sivic and Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, 1470–1477 vol.2. DOI: 10.1109/ICCV.2003.1238663.

[38] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968. DOI: 10.1109/TIT.1968.1054142.

[39] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951, ISSN: 00034851. [Online]. Available: http://www.jstor.org/stable/2236703 (visited on 09/04/2023).

[40] A. C. Murillo, J. J. Guerrero, and C. Sagues, "Surf features for efficient robot localization with omnidirectional images," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3901–3907. DOI: 10.1109/ROBOT.2007.364077.

[41] A. Zarringhalam, S. S. Ghidary, and A. M. Khorasani, *Semi-supervised vector-quantization in visual slam using hgcn*, 2022. DOI: 10.48550/ARXIV.2207.06738. [Online]. Available: https://arxiv.org/abs/2207.06738.

[42] W. Zhao, H. Zhao, Z. Cao, and L. Pang, *Composite fixed-length ordered features for palmprint template protection with diminished performance loss*, 2022. DOI: 10.48550/ARXIV.2211.04884. [Online]. Available: https://arxiv.org/abs/2211.04884.

[43] S. Huang, C. Cai, F. Zhao, D. He, and Y. Zhang, "An efficient wood image retrieval using surf descriptor," in *2009 International Conference on Test and Measurement*, vol. 2, 2009, pp. 55–58. DOI: 10.1109/ICTM.2009.5413007.

[44] T. Hong-Phuoc and L. Guan, *A scale and rotational invariant key-point detector based on sparse coding*, 2020. DOI: 10.48550/ARXIV.2010.06264. [Online]. Available: https://arxiv.org/abs/2010.06264.

[45] E. S. K. Nasab, R. Almasi, B. Shoushtarian, E. Golkar, and H. Rabbani, *Automatic detection of microaneurysms in oct images using bag of features*, 2022. DOI: 10.48550/ARXIV.2205.04695. [Online]. Available: https://arxiv.org/abs/2205.04695.

[46] A. Fontes and J. E. B. Maia, *Visual odometry for rgb-d cameras*, 2022. DOI: 10.48550/ARXIV.2203.15119. [Online]. Available: https://arxiv.org/abs/2203.15119.

[47] S. Bilik and K. Horak, *Sift and surf based feature extraction for the anomaly detection*, 2022. DOI: 10.48550/ARXIV.2203.13068. [Online]. Available: https://arxiv.org/abs/2203.13068.

[48] M. Griguletskii, M. Chekanov, and O. Shipitko, *Tomoslam: Factor graph optimization for rotation angle refinement in microtomography*, 2021. DOI: 10.48550/ARXIV.2111.05562. [Online]. Available: https://arxiv.org/abs/2111.05562.

[49] Z. Li, L. Zhou, and Z. Lin, "Robust visual place recognition method for robot facing drastic illumination changes," *Journal of Physics: Conference Series*, vol. 2209, no. 1, p. 012 001, Feb. 2022. DOI: 10.1088/1742-6596/2209/1/012001. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2209/1/012001.

[50] N. J. B. Dias, G. T. Laureano, and R. M. Da Costa, "Keyframe selection for visual localization and mapping tasks: A systematic literature review," *Robotics*, vol. 12, no. 3, 2023, ISSN: 2218-6581. DOI: 10.3390/robotics12030088. [Online]. Available: https://www.mdpi.com/2218-6581/12/3/88.

[51] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *International Statistical Review / Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989, ISSN: 03067734, 17515823. [Online]. Available: http://www.jstor.org/stable/1403797 (visited on 09/09/2023).

[52] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014. DOI: 10.48550/ARXIV.1412.6980. [Online]. Available: https://arxiv.org/abs/1412.6980.

[53] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[54] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld lidar, inertial and vision with ground truth," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4353–4360. DOI: 10.1109/IROS45743.2020.9340849.

[55] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. DOI: 10.1177/0278364916679498. eprint: http://ijr.sagepub.com/content/early/2016/11/28/0278364916679498.full.pdf+html. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498.

[56] M. Kragh, P. Christiansen, M. Laursen, *et al.*, "FieldSAFE: Dataset for obstacle detection in agriculture," *Sensors*, vol. 17, no. 11, p. 2579, Nov. 2017. DOI: 10.3390/s17112579. [Online]. Available: https://doi.org/10.3390%5C%2Fs17112579.

[57] N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, and C. Stachniss, "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1045–1052, 2017. DOI: 10.1177/0278364917720510. eprint: https://doi.org/10.1177/0278364917720510. [Online]. Available: https://doi.org/10.1177/0278364917720510.