Robust Perception and Detection Systems for Micro Aerial Vehicle based Intelligent Visual Inspection in Complex Environments

Leijian Yu

Department of Design, Manufacturing and Engineering Management
University of Strathclyde, Glasgow

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

12 December, 2023

Declaration

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Leijian Yu

Date: 07/12/2023

Acknowledgements

First and foremost, I would like to thank my supervisor Dr Erfu Yang for his kind support across all aspects of my PhD research and daily life during my whole PhD period. His rich knowledge base guided me to find my own research directions and finish this thesis. Moreover, I am deeply indebted to all the opportunities he gave me to transfer my knowledge from theory to the real world as well as broaden my horizons.

I would also like to thank Prof. Xiutian Yan for letting me access the necessary equipment and laboratory to conduct physical research experiments for my PhD research. His support gave me the chance to verify the approach I developed adequately.

I am grateful to the Strathclyde Net Zero Technology Centre(NZTC) robotics research team, especially Dr Gordon Dobie, Dr Charles MacLeod and Mr Beiya Yang. Their helpful suggestions and support assisted me to carry out my PhD work, especially for the experiment design, and they helped me to be a more effective researcher.

I also would like to thank Dr Cong Niu, Mr Dino Bertolaccini, Mr Mark Robertson, Mr Duncan Lindsay, Mrs Pamela Peacock and Mrs Gillian Eadie from the University of Strathclyde, Mrs Suzanne Buist and Mr Blair O'Connor

from NZTC and Prof. Peng Ren and Dr Cai Luo from the China University of Petroleum (East China) for the precious time they spent to support my research.

I would appreciate the endless love, support and encouragement I received during the period of perusing the PhD degree from my parents and friends. I would like to express my gratitude to Fang Liu for her support and encouragement. Moreover, I want to appreciate all the others who have spent time supporting me during my PhD. I cannot finish this research work without your kind help.

This work is funded by the China Scholarship Council and the International Fees Only Studentship from the University of Strathclyde (2018-2021), and it is also supported in part by the UK Net Zero Technology Centre (NZTC) under the LOCUST research project (2019-2021, Grant No.: AI-P-028). It is also supported partially by the Royal Society under the MOEA/D-PPR research project (2022-2024, Grant No.: IECNSFC211434). Without their support, I would not have had the chance to carry out this work.

Abstract

Structural failures caused by cracks or corrosion lead to catastrophic consequences in environmental, human and economic terms. Therefore, structural health assessments are essential for maintaining their structural integrity. Considering the small size and manoeuvrability of the Unmanned Aerial Vehicles (UAVs), the aerial inspection platform provides an efficient solution for inspecting high-risk sites such as drilling rigs and pressure vessels, which are traditionally inspected by experienced human engineers that mainly rely on their naked eyes. By deploying an autonomous aerial vision-based visual inspection system, the limitations of the human cost and safety factors of previously time-consuming tasks have the potential to be overcome. However, the maturity level of autonomous inspection UAVs still needs to be improved.

Motivated by the observations derived from improving the autonomous capability of aerial visual inspection, this thesis presents novel solutions to contribute to autonomous UAVs for asset visual inspection. First and foremost, the feasibility of using a UAV system with Visual Simultaneous Localisation and Mapping (VSLAM) for autonomous visual inspection in confined and low-illumination indoor environments is verified in the simulation environment for the first time. With image contrast-enhanced VSLAM, the UAV can track the planned tra-

jectory stably and record videos. Subsequently, corrosion detection and UAV localisation systems are further investigated to address the challenges that arise when implementing the UAV in complex environments and deploying algorithms on the UAV onboard platform.

In particular, to address the computational challenges of implementing a deep learning-based corrosion detector on UAV onboard platforms caused by the extensive usage of traditional convolutional layers, a solution with a lightweight model design is provided, achieving the first UAV onboard deep learning-based real-time corrosion detector. This advancement was achieved through lightweight convolution utilising Depthwise Separable convolution (DSconv), innovative feature extraction and fusion techniques leveraging the Convolutional Block Attention Module (CBAM) and the proposed improved Spatial Pyramid Pooling (SPP), refined detection strategies incorporating three-scale detection, and an optimised learning approach using the focal loss. The proposed lightweight but powerful corrosion detector is verified by leveraging the Nvidia Jetson TX2, and it achieves 20.18 Frames Per Second (FPS) and 84.96% mean Average Precision (mAP). The overall performance meets the requirements and outperforms other state-of-the-art detectors.

Then, the issue of the degraded performance of VSLAM-based UAV localisation systems in complex lighting and textureless environments is investigated. Initially, the inherent challenges faced by feature-based VSLAM in low-contrast environments, where extracting sufficient feature points is challenging, need to be addressed. To mitigate this issue, adaptive adjustments to the Features from Accelerated Segment Test (FAST) threshold and image enhancement from contrast and sharpening perspectives are proposed. These improvements are then

seamlessly integrated into monocular ORB-SLAM3, ensuring a stable and robust extraction of feature points. Compared with other advanced works, the developed VSLAM system achieves overall higher localisation accuracy and robustness in low-contrast environments while maintaining good performance in general environments.

To address the performance degradation or failure of VSLAM and Visual Odometry (VO) systems in environments with textureless and low-illumination conditions where sufficient feature points cannot be extracted, the deep learning-based feature point extraction method with a novel lightweight model has been investigated and incorporated into a VO system. Specifically, this model has been achieved by incorporating DSconv and Deformable Convolution (DFconv), whose kernel offsets are calculated through DSconv. Extensive experiments, including physical UAV flying tests, have been conducted to validate the feasibility and exceptional performance of the proposed method. Moreover, the developed model allows the UAV to localise itself and track the predefined trajectory in the textureless and challenging lighting environment, where both the other traditional and deep learning involved VO and VSLAM systems fail.

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

AGC Adaptive Gamma Correction

AGCWD Adaptive Gamma Correction with Weighting Distribution

AMCD Accurate UAV onboard Metallic Corrosion Detector

ANN Approximate Nearest Neighbour

AP Average Precision

APE Absolute Pose Error

ATE Absolute Trajectory Error

AUVs Autonomous Underwater Vehicles

CBAM Convolutional Block Attention Module

CDF Cumulative Distribution Function

CLAHE Contrast Limited Adaptive Histogram Equalisation

CNNs Convolutional Neural Networks

DRMS Dim-light Robust Monocular SLAM

DFconv DeFormable convolution

DoFs Degree of Freedoms

DSconv Depthwise Separable Convolution

DSM Direct Sparse Mapping

DVZ Deformable Virtual Zones

EKF Extended Kalman Filter

ELU Exponential Linear Unit

FAST Features from Accelerated Segment Test

Faster R-CNN Faster Region-based Convolutional Neural Network

FLOPs Floating Point Operations

FPS Frames Per Second

GLCM Grey-Level Co-occurrence Matrix

GPS Global Positioning System

GPU Graphics Processing Unit

HE Histogram Equalisation

HSV Hue-Saturation-Value

HT Hough Transform

ICL-NUIM Imperial College London and National University of Ireland Maynooth

IMUs Inertial Measurement Units

IoU Intersection-over-Union

LiDAR Light Detection and Ranging

mAP Mean Average Precision

MLE Mean Localisation Error

MS Matching Score

N/A Not Applicable

NDT Non-Destructive Testing

OIVIO Onboard Illumination Visual-Inertial Odometry

PDF Probability Density Function

PID Proportional–Integral–Derivative

PR Precision-Recall

Rep. Repeatability

RMS Root Mean Square

RNN Recurrent Neural Network

ROVs Remotely Operated underwater Vehicles

RPE Relative Position Error

SLAM Simultaneous Localisation and Mapping

SPP Spatial Pyramid Pooling

SSD Single Shot multibox Detector

SVM Support Vector Machine

Tanh Hyperbolic tangent

TP True Positive

TFLOPS Tera Floating Point Operations Per Seconds

UAVs Unmanned Aerial Vehicles

UGVs Unmanned Ground Vehicles

UUVs Unmanned Underwater Vehicles

VO Visual Odometry

VSLAM Visual Simultaneous Localisation and Mapping

Publications

Publications Arising from this Thesis

Articles

[1] **Yu, L.**, Yang, E., Luo, C., and Ren, P. (2021). AMCD: an accurate deep learning-based metallic corrosion detector for MAV-based real-time visual inspection. Journal of Ambient Intelligence and Humanized Computing, 1-12. https://doi.org/10.1007/s12652-021-03580-4.

[2] **Yu, L.**, Yang, E., and Yang, B. (2022). AFE-ORB-SLAM: robust monocular VSLAM based on adaptive FAST threshold and image enhancement for complex lighting environments. Journal of Intelligent and Robotic Systems, 105(2), 1-14. https://doi.org/10.1007/s10846-022-01645-w.

[3] **Yu.L.**, Yang, E., Yang, B., Fei, Z. and Niu, C. (2023). A robust learned feature-based visual odometry system for UAV pose estimation in challenging indoor environments. IEEE Transactions on Instrumentation and Measurement, 72, 1-11. https://doi: 10.1109/TIM.2023.3279458.

Conference Proceedings

[1] **Yu, L.**, Yang, E., Ren, P., Luo, C., Dobie, G., Gu, D., and Yan, X. (2019, September). Inspection robots in oil and gas industry: a review of current solu-

tions and future trends. In 2019 25th International Conference on Automation and Computing (ICAC) (pp. 1-6). IEEE. https://doi.org/10.23919/IConAC.2019.8895089.

[2] **Yu, L.**, Yang, E., Yang, B., Loeliger, A., and Fei, Z. (2021, July). Stereo vision-based autonomous navigation for oil and gas pressure vessel inspection using a low-cost UAV. In 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR) (pp. 1052-1057). IEEE. https://doi.org/10.1109/RCAR52367.2021.9517584.

Other Co-authored Publications:

Articles

[1] Luo, C., **Yu, L.**, Yang, E., Zhou, H., and Ren, P. (2019). A benchmark image dataset for industrial tools. Pattern Recognition Letters, 125, 341-348. https://doi.org/10.1016/j.patrec.2019.05.011.

[2] Luo, C., **Yu, L.**, Yan, J., Li, Z., Ren, P., Bai, X., ... and Liu, Y. (2021). Autonomous detection of damage to multiple steel surfaces from 360 panoramas using deep neural networks. Computer-Aided Civil and Infrastructure Engineering, 36(12), 1585-1599. https://doi.org/10.1111/mice.12686.

[3] Yang, B., Yang, E., **Yu, L.**, and Loeliger, A, (2021). High-Precision UWB-Based Localisation for UAV in Extremely Confined Environments. IEEE Sensors Journal, 22(1), 1020-1029. https://doi.org/10.1109/JSEN.2021.3130724.

[4] Yang, B., Yang, E., **Yu, L.**, and Niu, C. (2022). Adaptive Extended Kalman Filter Based Fusion Approach for High Precision UAV Positioning in Extremely Confined Environments. IEEE/ASME Transactions on Mechatronics. https://doi.org/10.1109/TMECH.2022.3203875.

Conference Proceedings

[1] Yang, B., Yang, E., and **Yu, L.** (2021, May). Vision and UWB-based anchor self-localisation system for UAV in GPS-denied environment. In Journal of Physics: Conference Series (Vol. 1922, No. 1, p. 012001). IOP Publishing. https://doi.org/10.1088/1742-6596/1922/1/012001.

[2] Fei, Z., Yang, E., Yang, B., and **Yu, L.** (2021). Image enhancement and corrosion detection for UAV visual inspection of pressure vessels. In Intelligent Life System Modelling, Image Processing and Analysis (pp. 145-154). Springer, Singapore. https://doi.org/10.1007/978-981-16-7207-1_15.

Chapter 1

Introduction

1.1 Research Background and Motivation

1.1.1 Robot-based Visual Inspection of Oil and Gas Facilities

Owing to the growth of the Industry 4.0 revolution, industries face growing demands for monitoring and maintaining the proper function of industrial facilities efficiently and effectively. Non-Destructive Testing (NDT) is the process of inspecting, testing or evaluating the properties of a component for industry without destroying the serviceability of the part. Inspections are a crucial part of the operations of many industrial sectors. The report presented by Mordor Intelligence [25] has indicated that the NDT market was valued at $16.72 billion in 2020 and is expected to reach a value of $24.64 billion by 2026, at a compound annual growth rate of 6.7% during the forecast period of 2021-2026. At the same time, the oil and gas sector still underpins modern society. According to the Energy Outlook 2022 released by the British Petroleum, the demand for oil and gas has increased above the pre-COVID-19 level [26]. Therefore, demand from

the oil and gas industries is fuelling the NDT market.

Visual inspection is currently the principal approach for maintenance and assessment of asset integrity [27]. Hence, the reliability and efficiency of inspection systems are vital to public safety and the economy. However, there are many facilities in oil and gas companies that own lots of oil derricks, pressure vessels and offshore rig infrastructures set in hazardous environments. These environments have risks of radiation, lack of oxygen, high temperatures and fire dangers, which put engineers at risk. In this case, eye-based visual inspection is inefficient, experience-dependent and can even be dangerous for experienced engineers [28]. Therefore, liberating human engineers from dangerous, expensive and time-consuming tasks becomes urgent.

To eliminate human participation, boost operational efficiency and improve safety, robotics are increasingly used to carry out inspection and maintenance activities on industrial properties [29]. The oil and gas business benefits from inspection robotics technology by being more productive, secure and dependable. According to [30], the inspection robot market was worth more than $1.68 billion in 2020, and it is projected to expand at a compound annual growth rate of 17.3 % from 2021 to 2029. Meanwhile, the oil and gas industry dominated the inspection robot market globally in 2020, accounting for more than 50% of the total market value.

Various robot-based platforms have been studied for efficiently assessing infrastructures, such as Unmanned Ground Vehicles (UGVs), Remotely Operated underwater Vehicles (ROVs), Autonomous Underwater Vehicles (AUVs) and Unmanned Aerial Vehicles (UAVs). To inspect assets in hazardous environments, UAVs have gained great interest due to their flexibility and manoeuvrability [31],

and they have been applied to lots of inspection tasks such as pipelines [32] and flare stacks [33]. A general UAV-based facility visual inspection process is indicated in Fig. 1.1. The visual inspection is carried out by experienced engineers who control the UAV remotely and identify defects from the captured data. The inspection efficiency and quality are influenced by the inspectors' skill and tiredness [34]. Therefore, this inspection procedure creates opportunities for furthering the advancement of autonomous UAV technologies and image analysis solutions to address challenges in efficient inspections [33], and the full automation of the visual data collection as well as its analysis is expected [35].

Figure 1.1: Scheme of UAV-based visual inspection

With the advancement of image processing techniques, a lot of image processing methods have been adopted for corrosion detection. However, corrosion does not have a unique shape, colour, or pattern that can be used to identify corrosion accurately [36]. Deep learning-based methods autonomously learn the relevant features, and some studies show that these methods outperform traditional corrosion detectors in terms of detection accuracy [37]. However, the high demand for computing resources has limited their applications on the UAV onboard platform [38]. Moreover, the requirement of real-time detection at a speed of at least 20 Frames Per Second (FPS) [39] is increasing in practical UAV visual

inspection due to the limited endurance time [40]. With real-time corrosion detection technology, end-to-end visual inspection practices become possible [41]. As a result, the labour intensity of corrosion detection can be further reduced, and the efficiency of visual inspection can be improved [31].

Figure 1.2: Example images of UAV first person view (adapted from [1])

To autonomously inspect facilities, the UAV must be aware of its location. Otherwise, the flying stability of the UAV is significantly degraded, and it cannot even track its trajectory for performing inspection tasks [42]. In outdoor inspection tasks, the UAV can rely on the Global Positioning System (GPS) for navigation to collect visual data. As UAVs continue to evolve, small UAVs have also started to be deployed into confined spaces to perform internal inspection tasks, such as inspecting the inside of pressure vessels and wind turbine blades [43]. These scenarios involve confined spaces that need to be visually inspected. Therefore, the UAVs used in these tasks must be small, which limits the UAV's payload capability [43]. As a result, the adoption of Visual Simultaneous Localisation and Mapping (VSLAM) or Visual Odometry (VO) in small UAVs is on the rise due to the use of lightweight cameras as sensors, with no prior knowledge required [44]. However, some indoor environments are enclosed spaces with no direct sunlight, leading to reduced visibility. To address this issue, the UAV needs to be equipped with artificial lighting resources. Specular reflections caused by coatings or walls can lead to glare, and the limited distance to the surface that

needs to be inspected constrains the texture information obtained by the camera, resulting in a lack of features. Some example images derived from the UAV visual inspection process [1] are shown in Fig. 1.2 to illustrate the challenging lighting and textureless conditions. These hazardous environments present challenges to the stability of the VSLAM and VO systems may result in system failures.

1.1.2 Research Aim and Objectives

Motivated by the opportunities and challenges mentioned above, the aim of this thesis is to investigate key technologies to support an aerial platform to autonomously conduct visual inspections of different assets, such as pressure vessels or drill platforms.

Specifically, the feasibility of deploying UAVs with VSLAM in confined and low-illumination environments to perform autonomous visual inspection tasks is investigated. Subsequently, this thesis presents a series of developments in corrosion detection and VO/VSLAM systems that address challenges derived from the existing literature. The corrosion detection algorithm aims to fill the gap of lacking an accurate real-time deep learning-based corrosion detector for the UAV onboard computer. The VO/VSLAM algorithms aim to provide position information for UAVs in complex environments and address the issue of sufficient feature points that cannot be extracted in low-contrast scenarios caused by unfavourable lighting conditions and textureless environments, which can lead to degraded performance or even failure of the VO and VSLAM systems. Therefore, the objectives of this thesis are:

1. Explore key knowledge of autonomous UAV systems for industrial facility visual inspection:
   (a) Survey the existing knowledge about autonomous UAV systems for industrial facility visual inspection.
   (b) Assess the current state of corrosion detectors.
   (c) Examine the present status of VO/VSLAM systems used in low-illumination and textureless environments.
2. Investigate the feasibility of deploying UAV with VSLAM in confined and low-illumination environments for autonomous visual inspection:
   (a) Develop a simulation environment to mimic the scenario of UAV-based autonomous visual inspection inside a pressure vessel.
   (b) Evaluate the performance of the VSLAM algorithms and the VSLAM-based autonomous UAV navigation system in the developed simulation environment.
3. Tackle the challenge of the high computing resource demand when deploying deep learning-based corrosion detection algorithms on UAV onboard platforms to achieve real-time and accurate corrosion detection:
   (a) Develop a deep learning model with lower computing complexity compared to other deep learning-based corrosion detectors in the literature.
   (b) Conduct experiments to validate the developed model.
4. Investigate and address the issue of insufficient feature points that can be extracted for feature-based VSLAM in environments with challenging illumination conditions:

   (a) Develop a strategy for VSLAM to extract reliable and robust feature points in environments with different illumination conditions.

   (b) Conduct experiments to verify the developed VSLAM method.

5. Study and address the gap in extracting sufficient feature points in textureless and low-illumination environments for UAV onboard feature-based VO/VSLAM.

   (a) Develop a lightweight deep learning-based feature extractor for UAV onboard VO systems:

   (b) Conduct experiments to validate the UAV onboard deep learning feature-based VO method.

6. Conclude research findings and provide suggestions for future work.

1.2 Research Methodology

1.2.1 Introduction

Research can be defined as a systematic process carried out with a specific objective, which aims to explore and discover new knowledge [45]. This definition suggests that research is founded on logical relationships rather than just beliefs. The research methodology can be defined as how the research is approached, taking into account philosophical and theoretical assumptions and their influence on the research [45]. However, some researchers use the term "methodology" synonymously with "methods" [46]. The definition of research methodology in [47] is "the entire framework or design of the research: the choice of paradigm, methods, and tools or technique", and it is used in this thesis. A research methodology

can be determined by the chosen techniques and the ways they are applied to the research process.

The choice of research methodology relies on the research subject, the researcher's interests, as well as the adopted philosophy [48]. This procedure is guided by the employed philosophies in order to carry out the research. The research philosophy and methodology determine the choice of research strategies and methods [48]. The relationship between philosophical worldviews, strategies and research methods is shown in Fig. 1.3. Research design connects research philosophies and the research strategy. However, philosophical worldviews determine the research strategy and the adopted research methods.

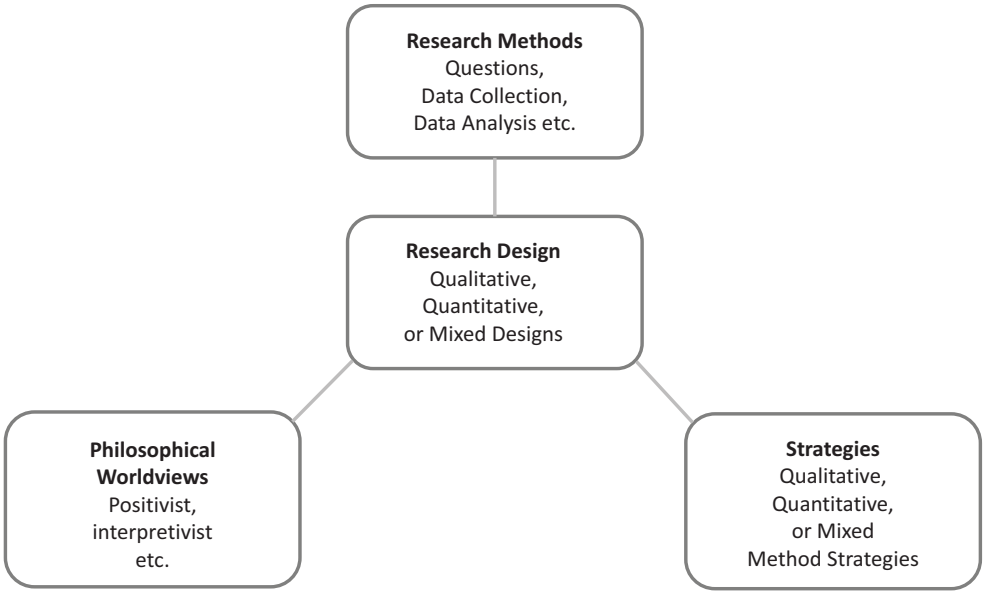


Figure 1.3: The relationship between philosophical worldviews, strategies and research methods (Adapted from [2]).

In the tradition of science, two primary research philosophies are acknowledged: positivism and interpretivism [49]. Positivists believe that reality is stable and can be observed and described objectively [50]. Positivism highlights measurement and rationality, asserting that knowledge is derived from objec-

tive and quantifiable observations of events or actions. Therefore, in positivist research, data collection and interpretation are carried out in an objective way. Positivism is closely associated with the utilisation of quantitative data collection techniques [51]. Interpretivism is founded on the belief that reality is subjective, multiple and shaped by social construction [52]. Therefore, the understanding of reality is limited to individual's personal experiences, which may vary from someone else's, and it is influenced by the individual's historical and social viewpoint. Interpretive approaches utilise questioning and observation to uncover or develop a comprehensive understanding of the investigated phenomenon. This is closely related to qualitative methods of collecting data [53].

This thesis aims to investigate the advanced localisation and corrosion detection solutions to assist the UAV to perform visual inspection tasks autonomously. Therefore, positivism is chosen as the research philosophy, and quantitative research is designed and conducted. Quantitative research is a method for testing objective theories by exploring the relationship among variables [54]. These variables are usually measurable, allowing for the analysis of numerical data through statistical methods. In this thesis, the data are collected and analysed through the following approaches:

The literature review forms the foundation of the research [45]. Therefore, a comprehensive literature review was conducted in Chapter 2 to assess the current state of relevant studies. The insights gained from this review were then utilised to identify research gaps and refine the scope of this thesis.

Simulation aims to analyse the behaviour of a complex system [55]. Simulation is employed in scenarios where analytical problem-solving is typically challenging and often involves random variables. The UAV autonomous navigation

system used in this thesis consists of different components, such as localisation, controllers, and so on. This work focused on investigating the performance of the VSLAM-based localisation module. Therefore, simulation was deployed to analyse the feasibility of the VSLAM-based autonomous aerial system (Chapter 3). However, creating a simulation that closely mimics real-world events can be challenging.

Laboratory experiments identify the precise relationships among a small number of variables in a designed and controlled laboratory environment [56]. The data is collected and analysed using quantitative analytical techniques with the aim of formulating generalisable statements that can be applied to real-world scenarios. The major weakness of laboratory experiments is the 'limited extent to which identified relationships exist in the real world due to oversimplification of the experimental situation and the isolation of such situations from most of the variables found in the real world' [49]. Therefore, data collected from real-world environments were used to test the performance of the developed algorithms, and laboratory experiments was employed in Chapters 4, 5 and 6.

1.2.2 Research process

The research can be conducted by the guidance of the research philosophy and methodology. The research process used in this thesis is shown in Fig. 1.4, which is adapted from [57]. There are a total of six steps included in the research process:

(1) Identifying the research problem;

(2) Comprehensive literature review including concepts, theories and previous

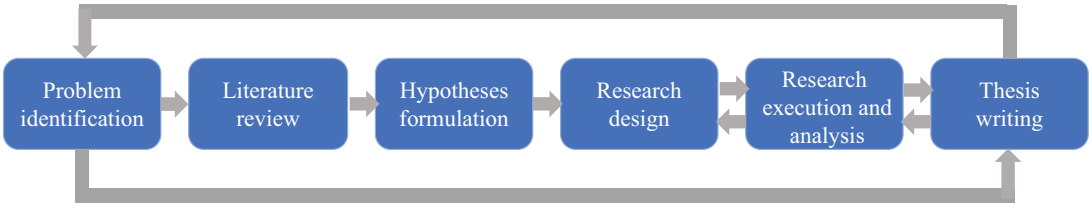


Figure 1.4: Research process (adapted from [57])

research findings;

(3) Developing the hypothesis;

(4) Conducting the research design;

(5) Carrying out the research and analysing the results;

(6) Preparing the thesis to report the results.

Instead of adhering to a rigidly specified schedule, these activities continually overlap. Moreover, the initial step dictates the achievement of the last step that has to be reached. Serious problems may occur and the study might not be finished if later procedures were not considered in the beginning phases.

1.2.3 Research Questions

Research questions are questions that the research project sets out to answer. Identifying proper research questions is an essential process of effective research. To conduct the research on autonomous corrosion detection and robust UAV localisation, the following research questions were identified for this thesis.

1. Is it feasible to deploy UAVs equipped with VSLAM technology into confined and low-illumination environments for autonomous visual inspection tasks?

2. How to tackle the challenge of high demand for computing resources in deploying the deep learning-based detector on the UAV onboard computer to achieve real-time and accurate metallic corrosion detection?

3. How to address the issue of poor performance in UAV onboard VO/VSLAM systems in complex environments, particularly in scenarios under low-light or overly bright conditions, as well as textureless conditions where a sufficient number of feature points cannot be extracted?

Each of these questions has been proven difficult to address when applied to relevant existing work in literature that provides an existing solution to the UAV inspection scene. By addressing these research questions during the development of solutions to corrosion detection and UAV localisation, the presented efficient corrosion detection algorithm and robust VO/VSLAM systems successfully achieve the aim of this research while extending the state-of-the-art UAV autonomous visual inspection scenario.

1.2.4 Literature Review

A summary of previously published works on a certain topic can be called a "literature review" [45]. It grasps the literature and demonstrates prior knowledge of the targeted research area. To address the research questions mentioned above, comprehensive literature reviews about sensors and robotic platforms for inspections, vision-based corrosion detection systems, UAV positioning techniques and principles of Convolutional Neural Networks (CNNs) are carried out and presented in Chapter 2.

1.2.5 Research Hypotheses

The research hypothesis is "a statement about the expected outcome of a study" [58]. It is a predictive statement about the expected study outcome. The research hypothesis for the feasibility of deploying autonomous UAV into the confined low illumination environment with VSLAM is that through the improvement of the feature extraction capability of VSLAM, the autonomous UAV can track the predefined trajectory to collect visual data. The research hypothesis for the corrosion detection task is that through the lightweight model design, the deep learning-based corrosion detector can accurately identify different categories of corrosion under different environments and obtain real-time results with UAV onboard platforms. The research hypothesis for the UAV localisation component is that by improving the feature extraction capability, the VO/VSLAM algorithms can deliver reliable localisation performance in complex environments, especially in challenging lighting environments and textureless scenarios.

1.2.6 Research Design

To test the above research hypothesis, research solutions are created and formulated. A customised simulation environment is developed to prove the effectiveness of the autonomous visual inspection with the improved VSLAM. Advanced deep learning-based techniques are adopted and optimised for the UAV onboard corrosion detector to accomplish accurate and efficient corrosion detection. To deploy the UAV into the challenging lighting environment, a novel VSLAM approach that owns the adaptive image enhancement and feature extraction capabilities has been developed. To further improve the robustness of the VO/VSLAM system in the textureless and challenging lighting environment, the deep learning-based

feature point extraction method has been optimised and implemented into the VO framework. Both public datasets and physical experiments are utilised to verify the proposed approaches.

1.2.7 Research Execution and Analysis

Both the software and hardware are designed and utilised to conduct the simulation and experiments to assess proposed solutions. Specifically, a customised simulation environment has been designed to verify the feasibility of the UAV with VSLAM for autonomous visual inspection in challenging indoor environment. A novel deep learning-based accurate UAV onboard corrosion detector, a improved traditional VSLAM for the low-contrast environment and a optimised deep learning feature-based VO systems have been developed. A quadrotor with onboard computing platforms is set up, and the public dataset and experimental settings are configured. The experimental results are collected and analysed to reveal their benefits, shortcomings and potential improvements.

1.2.8 Thesis Writing

To summarise the entire research and deliver the findings to more audiences, the research outcomes and contributions are described in-depth in this thesis and additional research publications.

1.3 Contributions

To accomplish the previously outlined objectives, the contributions of this thesis can be summarised as follows:

- The feasibility of deploying the UAV with VSLAM to achieve autonomous visual inspection in confined and low-illumination indoor environments has been proven through a customised simulation environment for the first time.
- The issue of high demand for computing resources when deploying deep learning-based corrosion detection on the UAV onboard computer caused by extensive usage of traditional convolution layers is addressed through lightweight model design. It is achieved through lightweight convolution utilising Depthwise Separable convolution (DSconv), innovative feature extraction and fusion techniques leveraging the Convolutional Block Attention Module (CBAM) and the proposed improved Spatial Pyramid Pooling (SPP), refined detection strategies incorporating three-scale detections, and an optimised learning approach using the focal loss.
- The challenge of extracting sufficient feature points in low-contrast environments for the VSLAM system is addressed through an image contrast based adaptive Features from Accelerated Segment Test (FAST) threshold and image contrast enhancement from the perspective of image contrast and sharpening.
- The challenge of extracting sufficient feature points in low-contrast and textureless environments for the UAV onboard VO/VSLAM system is addressed through the adoption of the deep learning-based feature point extraction method with the lightweight model. The advancement is achieved

by incorporating DSconv and the Deformable Convolution (DFconv), whose kernel offsets are calculated through DSconv, to extract feature points in the challenging environment.

1.4 Thesis Organisation

To present the preceding contributions, the rest of this thesis is divided into six chapters as follows:

Chapter 2 reviews different types of robot platforms for infrastructure inspection (objective 1(a)). Besides, the corrosion detection systems (objective 1(b)) and UAV localisation techniques (objective 1(c)) are also surveyed. The principles of CNN have also been introduced. The limitations of the current state-of-the-art approaches are identified, and the knowledge gaps are also discussed in this chapter.

Chapter 3 evaluates the feasibility of deploying the UAV in confined and low-illumination environments with VSLAM to perform the autonomous visual inspection task. Specifically, a simulation environment was developed to evaluate the feasibility of deploying the UAV with VSLAM to achieve autonomous visual inspection of the inside of the pressure vessel (objective 2(a)). Moreover, a basic vision-based UAV autonomous inspection scheme in the low-illumination environment is developed and verified (objective 2(b)). The overall structure involves four parts: the UAV platform to execute the flight command; the visual localisation module to obtain the UAV position in the dark environment; the waypoint controller to calculate the desired position and the position tracking controller to generate the control signal.

Chapter 4 focuses on addressing the problem of detecting corrosion accurately and efficiently with the UAV onboard computing resources. A novel deep learning-based accurate metallic corrosion detection algorithm has been developed, which is optimised for the UAV onboard platform to improve the inspection efficiency (objective 3(a)). Extensive experiments are carried out to verify the performance of the corrosion detector (objective 3(b)).

Chapter 5 presents a novel VSLAM system to localise the robot in the complex illumination environments. Efficient image enhancement and adaptive feature extraction threshold are presented and embedded into the VSLAM framework to overcome the problem caused by the unideal lighting conditions (objective 4(a)). The performance of the developed VSLAM has been validated through extensive experiments (objective 4(b)).

Chapter 6 describes a deep learning feature-based VO system to handle environments that are low-textured and challenging in lighting conditions (objective 5(a)). Experimental results show that the proposed system cannot only handle the challenging scenarios that traditional feature-based VO and VSLAM systems fail but can also be deployed on the UAV platform (objective 5(b)).

Chapter 7 concludes this thesis with a collective discussion of the research findings derived from each of the studies in relation to the overall research and provides suggestions for future research (objective (6)).

Chapter 2

Literature Review

This chapter provides a solid foundation for the research questions and scopes covered by this thesis. Thus, a basic review of the latest NDT techniques and robotic platforms for inspecting assets in oil and gas companies is presented. Inspection robots can make the inspection process more thorough and simpler to perform. In addition, inspection robots improve data organisation and help in lowering the cost of operations. Thus, deploying robotic systems to substitute human engineers for accessing hazardous environments is the future trend. Therefore, state-of-the-art robotic inspection solutions are reviewed. Based on vehicle type, the inspection robots in the oil and gas industry can be divided into the following categories: ROVs, AUVs, UGVs and UAVs. The ROVs and AUVs are both Unmanned Underwater Vehicles (UUVs), so they will be covered in the UUVs part. These robots have different mechanisms and structures for different inspection tasks. Some of them are focused on inspecting oil storage tanks, while some of them are designed for pipeline inspection. Nevertheless, most of them need experienced engineers to manipulate them to conduct the inspection process. Greater robot flexibility and autonomy levels can make the inspection more

intelligent and efficient.

To comprehensively understand inspection robots in the oil and gas industry, this chapter reviews the key technologies in different kinds of inspection robots, discussing the challenges and highlighting the trends in future research that will make the inspection process more efficient, intelligent and cost-effective. In particular, the key inspection solutions are surveyed and compared to identify their advantages and limitations. Moreover, key technologies, especially for the UAV deployed corrosion detection algorithms and localisation systems, which improve the inspection efficiency, are reviewed to summarise their limitations and research gaps.

2.1 Sensors and Robotic Platforms for Inspections

2.1.1 Inspection Methods and Sensors

For inspection purposes in the oil and gas industry, no single NDT solution works for all defect detection solutions [59]. Therefore, a number of NDT sensors with their appropriate inspection methods are used to detect corrosion and cracks. According to different types of sensors, the most commonly applied inspection technologies can be roughly divided into four classes, i.e., visual inspection, ultrasonic inspection, magnetic inspection and eddy current inspection.

Visual inspection is one of the most basic and common inspection means. In the very beginning, experienced engineers used their naked eyes to check the condition of assets [60]. But now, cameras have allowed the inspection robots to pursue a view of the structure. Then, the structure will be reviewed by an

experienced engineer remotely [61]. With the development of image processing techniques, the image analysis process can be finished in an autonomous manner [62]. Visual inspection is simple and one of the most straightforward inspection techniques to perform. However, it is only suitable for detecting surface damage, and the inspection quality is sensitive to illumination conditions [63].

Besides the visual inspection mentioned previously, the ultrasonic inspection is another primary way found in literature. Ultrasonic sensors can emit and receive ultrasound waves, which are propagated into the material. Cracks can be detected by measuring the time difference between the generated and reflected ultrasound [64]. There are many advantages to using ultrasonic sensors, such as high accuracy, high sensitivity and suitability for monitoring all kinds of materials [65]. However, it will not work when the defect lies along the line of the wave travelling [66].

Magnetic sensitive sensors work with ferrous material assets [67]. After applying a magnetic field to these facilities, most of the magnetic flux lines will go through these metal materials. If there is a defect, magnetic flux lines will be bent. Some of the magnetic flux lines will leak out. The magnet sensitive sensors can detect the magnetic leakage field. The detected signal can be analysed to reveal the changes in structure [68]. This method can realise high-speed inspection and defects on both the external surfaces and subsurface [69]. However, it can only work for ferromagnetic materials, and the sizes of defects detected are very limited [70].

Eddy current inspection is similar to magnet sensitive inspection in some ways. It uses eddy currents generated by coils. When there is a crack in the structure, the eddy current will be altered. At the same time, the impedance of

the coil will also be affected. Monitoring the change in impedance in the coils can tell the condition of the facility [71]. This method is sensitive to surface defects and can be used to inspect multilayer structures [72]. Nonetheless, it is very susceptible to magnetic permeability changes and cannot detect defects parallel to the surface [73].

Table 2.1: Comparison of different inspection technologies

Technology	Defect location	Advantages	Limitations
Visual inspection [60] [61] [62] [63]	Surface	Easy to use; good at finding surface blemishes	Influenced by the environment easily; subject to operator's skill
Ultrasonic testing [64] [65] [66]	Surface; internal	Sensitive to detects; fast	Critical requirement for surface condition; difficult to inspect complex structures
Magnetic testing [67] [68] [69] [70]	Surface; subsurface	Higher sensitivity than ultrasonic for ferromagnetic materials	Limited to ferromagnetic materials; difficult to measure the defect depth quantitatively
Eddy current testing [71] [72] [73]	Surface; Subsurface	Low requirement for the surface condition	Influenced by magnetic properties; orientation of probe during scanning can effect results
thermography testing [74] [75] [76] [77] [78]	Surface; Subsurface	Fast; large detection area	Influenced by reflectance of the surface

Infrared thermography testing uses thermal imaging cameras to visually represent surface temperatures of a facility to identify abnormal temperature patterns that indicate possible defects [74]. Thermographic measurements can be con-

ducted in either the active or passive mode. In passive mode, the camera records the existing thermal profile emitted from the surface. On the contrary, the active mode captures the surface temperature decay profile when subjected to an external thermal stimulus for a certain duration [75]. There are no standard baselines that can be used to identify abnormal temperature patterns. Therefore, defect detection with passive thermography is often considered a qualitative approach. In contrast, active thermography experiments are conducted under controlled conditions. The type and amount of stimulation are specified, which enables defect identification with quantitative analysis [76]. The infrared thermography testing can inspect a large area efficiently, and the detection results are intuitive [77]. Nevertheless, obtaining accurate results requires knowledge of the target's emissivity and reflective temperature [78].

The summary of these methods is listed in Table 2.1.

2.1.2 Unmanned Ground Vehicles for Facility Inspection

According to the shape and function of the onshore facilities in the oil and gas industries, the equipment can be roughly divided into vertical structures and pipelines. The vertical structures contain drilling, production and storage assets such as flare stacks and tanks. Pipelines are primarily focused on transportation purposes. To inspect these kinds of facilities, UGVs are the most popular choice nowadays.

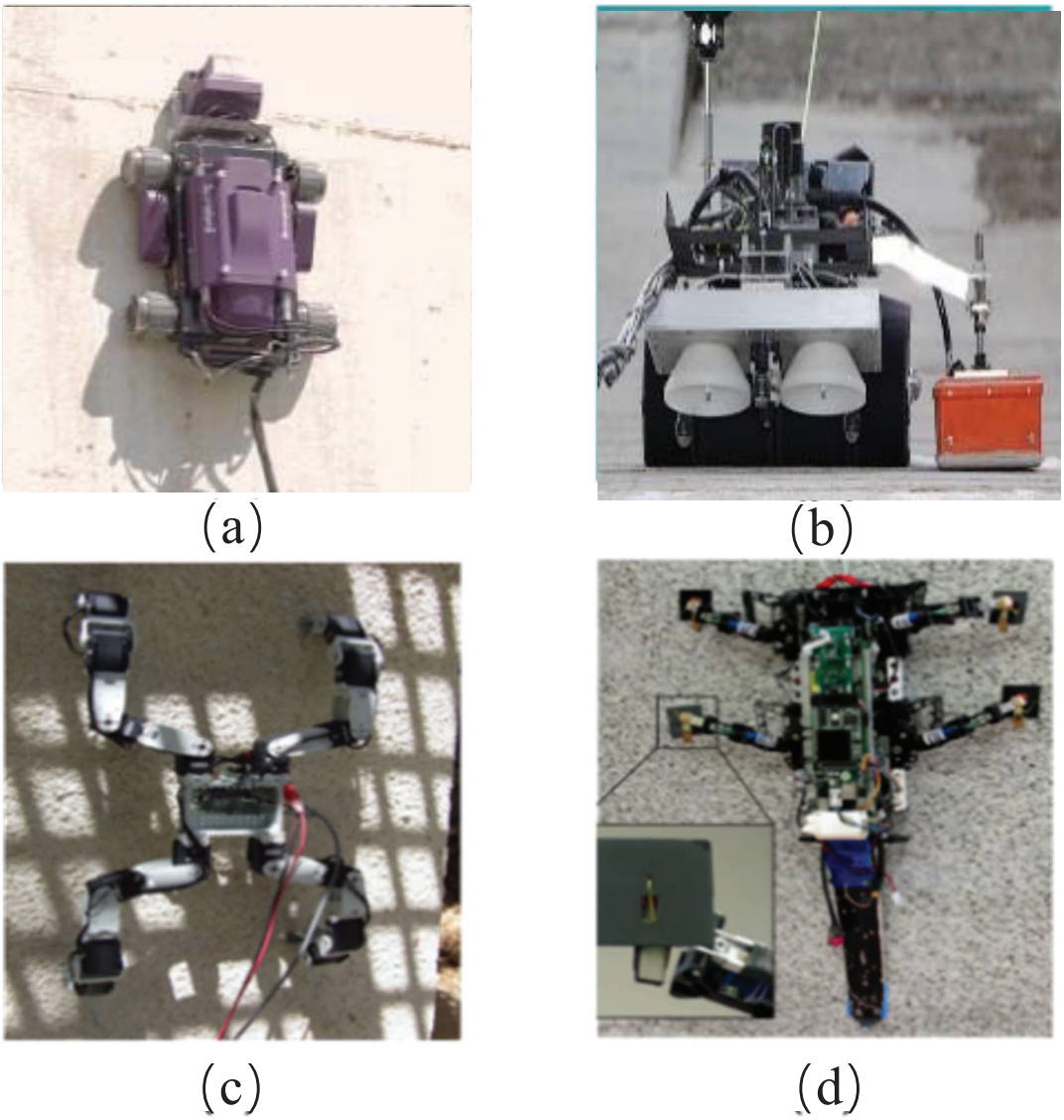


Figure 2.1: (a) A magnetic adhesion robot [3], (b) a pneumatic adhesion robot [4], (c) a cat insperied adhesive robot [5], (d) a gecko inspired robot [6].

2.1.2.1 Vertical Structure Inspection

For inspecting vertical structures, wall-climbing robots have gained significant interest. The climbing technologies are the main difference between these robots. At the same time, the most important task in the design and development of a climbing robot is to develop an appropriate mechanism to ensure that the robot adheres to different types of walls and surfaces reliably without sacrificing its mobility. According to the adhesion and locomotion principles, climbing methods can be cauterised as magnetic adhesion, pneumatic adhesion, and bio-inspired grasping grippers [79], as shown in Fig. 2.1.

Vertical structures in oil and gas companies are usually made of carbon steel. Since this kind of material is ferromagnetic, magnetic adhesion can be highly desirable in this kind of environment. There is a lot of work utilising magnetic adhesion to inspect these facilities. One method is to use permanent magnets. Such as [3], they selected high-strength permanent magnets as the adhesion mechanism, which can hold the robot firmly on the walls. Another advantage of these methods is that robots do not need extra power for the adhesion mechanism. In some circumstances, variable adhesion is required, and the speed of switching required is high. Thereby, the electromagnetic adhesion mechanism will become specifically applicable [80].

Pneumatic adhesion mechanisms are another widely used technology in vertical structure inspection robots. The attraction force between the robot and the wall is proportional to the pressure difference between the pressure chamber or suction cups and the atmosphere. Unlike magnetic adhesion methods, which can only work on ferromagnetic surfaces, the pneumatic adhesion mechanism is suitable for a wider range of materials. Using suction cups is a very popular

method. In [81], three suction cups, a supporting plate, a vacuum pump, and some accessories were used to compose the suction module. The rover developed in [4] was equipped with a vacuum adhesion mechanism as a fall protector while performing inspection tasks.

There is also a variety of work adopting biomimetic adhesion methods to realise excellent climbing robots. In [5], researchers developed a robot that consists of four legs with gripping devices made of 12 fishing hooks. This robot can imitate the movements of rock-climbers and the way cats hold surfaces when they climb in the vertical direction. A gecko-inspired adhesive method was proposed in [6]. Inspired by the gecko toes, it used a rigid tile supported by a compliant material and loaded by an inextensible tendon. This mechanism allowed the climbing part to make full contact with the surfaces.

2.1.2.2 Onshore Pipeline Inspection

For onshore pipeline inspection robots, the style of locomotion is a vital part, which can reflect the whole performance of the robots [82]. According to the difference in driving source and control ability of movement mechanism, robots can be sorted into the pig, wheeled, tracks, legged, inchworm, snake and screw types [83]. Fig. 2.2 gives some examples of these kinds of robots.

The pig type [7] itself is a simple device which collects the data from the pipeline. The pig carries out the inspection tasks along with the flow of oil or gas. What is more, the pig has no driving mechanism and is driven through the pipeline by oil or gas flow. The wheel type [8] uses wheels to touch the pipe wall. It can easily adapt to various pipelines with springs. The track type [9] is often treated as the alternative to the wheeled robot. The wheels are bounded

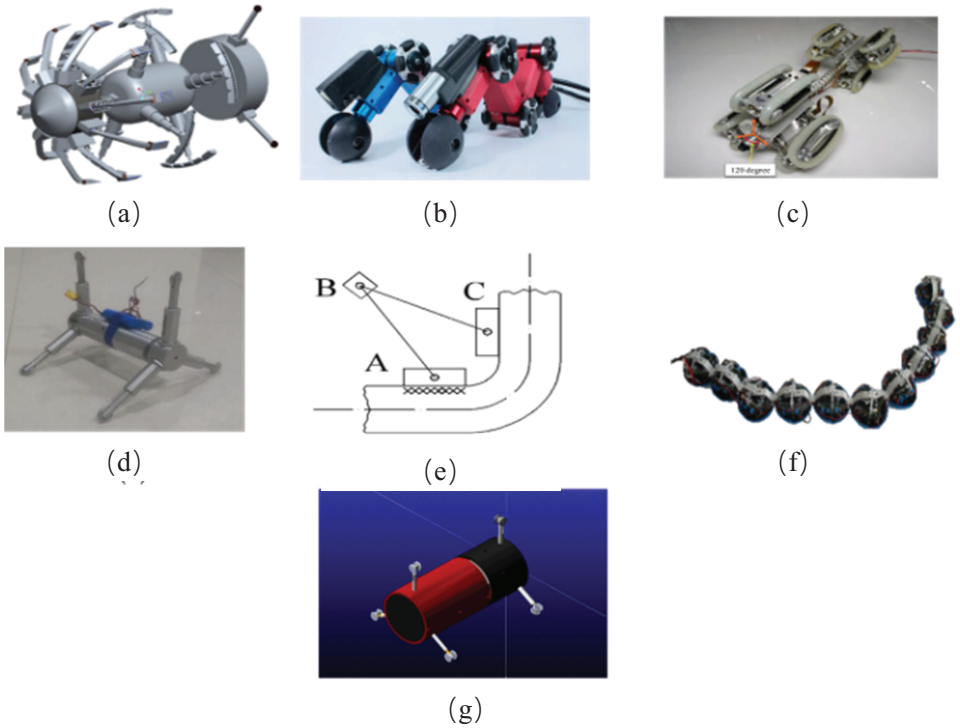


Figure 2.2: (a) The pig type [7], (b)the wheel type [8], (c) the track type [9], (d) the legged type [10], (e) the inchworm type [11], (f) the snake type [12], (g) the screw type [13].

by the belt, which can enlarge the surface contact area and reduce the chances of losing pipe wall contact. The legged type [10] uses legs to contact the pipe wall. This type of robot can produce highly sophisticated motions and is suitable for pipes with obstacles. The inchworm type [11] uses the traction generated by the large force applied to the front or back module. Compared with other types of inspection robots, it has an advantage in curved pips. Snake type [12] consists of several identical body segments with joints, which allows it to generate a wide range of different motions. The screw type [13] moves forward through the rotary motion, achieving good performance in vertical pipelines.

2.1.3 Unmanned Underwater Vehicles for Facility Inspection

Oil and gas companies have thousands of kilometres of pipelines and other assets in the sea to produce and transport their products. These undersea structures can easily develop corrosion and cracks since they are often made from metal materials. In order to prevent financial and environmental disasters caused by leaking products, these facilities need to be inspected frequently. ROVs and AUVs are effective and affordable platforms for performing underwater inspection tasks [84]. The inspection ROVs are operated by the surface operator. They are alternative vehicles to human workers in conditions that are too deep and too dangerous for human beings [85]. What is more, with the help of ROVs, inspection tasks can be performed in 24 hours and 7 days. Due to these advantages, the oil and gas companies have developed ROV technologies since the 1980s [29]. After that, a large number of advanced ROVs with a group of sensors were used to carry out inspection tasks, which varied in size and weight. These features of some ROVs used for visual inspections are listed in Table 2.2. These ROVs consist of

Table 2.2: Features of some ROVs used for visual inspection

ROVs	Max depth (m)	Weight (kg)	Forward speed (knots)	Size (mm)
VideoRay pro 4 [86]	305	38.5	4.2	375, 290, 220
Novaray model 2000 [87]	305	25.5	6	1020, 997, 229
Falcon [88]	300	60	3	1000, 500, 600
Constructor 220 HP [89]	3048	4500	3.1	3220, 1700, 2165
Mojave [90]	300	85	3.5	1000, 600, 500

a vehicle body, control cabin, umbilical, video cameras, handling system, launch system and power supplies. For the deep-sea inspection tasks, the main cost of using an ROV is caused by the human operator. This cost can be reduced if these

tasks can be performed autonomously. Based on the previous technologies and economic constraints, using the AUVs may be a replacement for the ROVs for inspection purposes, and currently, it has gained great research interest. Compared with ROVs, AUVs demand more accurate and efficient sensors, guidance systems and algorithms for the execution of this kind of mission [91]. Authors in [92] presented a Deformable Virtual Zones (DVZ) method that is a sensor-based control approach. It builds a model of a virtual zone around the robot. Once the obstacles are detected by proximity sensors, there will be deformation, and the control signal will be calculated by minimising the deformation in the DVZ. The FlatFish project [93] utilised the asset layout-based navigation methods. Since almost all of the underwater assets are connected together, they can be inspected by following pipelines and tie-backs to reach the different parts of the facilities. The sonar and camera are utilised to target and inspect the facilities. In [94], image-processing technologies are adopted for the AUV's navigation. At first, it detects the pipeline corners in the captured image. After that, the obstacles are identified by Hough Transform (HT), and velocity and angle values can be calculated according to this information. Finally, the AUV can move along the pipeline by itself and inspect these structures autonomously.

2.1.4 Unmanned Aerial Vehicles for Facility Inspection

UAVs equipped with relative sensors can work as an excellent alternative to traditional inspection techniques [95]. It can not only save time but also lower the cost. The North Sea E & P company conducted a survey and showed that using UAVs to inspect assets can be twenty times faster and half the cost of traditional inspection methods [96]. This kind of method has gained great interest

in inspecting assets in the oil and gas industry. Until now, all the commercial inspection UAVs are still manually controlled. During the inspection procedure, the UAV will be controlled by one experienced pilot, while the live video for inspection purposes will be monitored by another experienced inspection engineer. Famous oil and gas companies in the world, such as BP, Shell, Apache, BG Group and Statoil have cooperated with Cyberhawk, which is the world leader in the drone inspection domain, to inspect their facilities. Intel Falcon 8 Plus used by Cyberhawk has shown a reliable and efficient performance [97]. The Intel Falcon 8 Plus has a patented V-shaped design with eight rotors, which makes the UAV more stable and ensures an unobstructed data capture procedure. By carrying three redundant inertial measurement units (IMUs) with efficient data fusion technology, the flying system can perform reliable responsiveness and stability during flights. The inspection module consists of an RGB camera and a thermal camera, which helps the UAV navigate while capturing detailed data for orthography and 3D reconstruction that can be used for inspecting the assets and further analysis [98]. Thanks to the manoeuvrability and flexibility of the small UAV, UAV has also started to be developed for confined indoor inspection tasks to improve inspection efficiency and reduce cost [99]. The most notable indoor inspection UAV is Elios [100], the first collision-tolerant drone developed by Flyability. The diameter of the Elios is below 400mm. It is surrounded by a carbon fibre shell, which can protect the body from a collision. When colliding with the obstacles, the UAV will bounce off and roll along the surface to find the path. It has been widely utilised for indoor facility visual inspection by oil and gas companies.

For UAV inspection, one UAV needs two experienced human engineers, one in charge of operating the UAV and another responsible for identifying defects

from captured image sequences. If the inspection tasks can be performed autonomously, the inspection procedure can not only be speeded up but also improve cost efficiency by reducing engineering labour fees and minimising accidents due to human operation errors. The Petroleum Institute of Abu Dhabi has developed a UAV autonomous tracking and navigation controller for inspecting straight oil and gas pipelines [32]. Their autonomous procedure consists of four parts: Firstly, the gradient of Gaussian is used to extract the edge of the object in image sequences. Secondly, the HT is implied to identify the pipeline. After that, a Proportional–Integral–Derivative (PID) controller is designed to achieve angular and lateral correction, which ensures the position and orientation of the UAV are aligned with the pipeline. Finally, automated navigation along the pipeline is realised.

2.1.5 Key Findings

Through the comparison of different sensors and relevant techniques for inspection discussed above, each sensor has its own advantages and drawbacks. When choosing the proper sensor for inspection robots, the application scenario and constraints need to be taken into consideration. Visual inspection is one of the most basic and popular methods to determine the surface condition, and it can be performed with only one camera by the robotic platforms. As a result, it is chosen as the inspection technique to be discussed in this thesis.

In addition, three kinds of inspection robots in the oil and gas industries have been surveyed. Among these robots, autonomous UAVs are efficient alternatives to UGVs. Similarly, AUVs are more efficient than ROVs to perform repeated close-distance visual inspections [93]. The UAVs and AUVs perform the inspec-

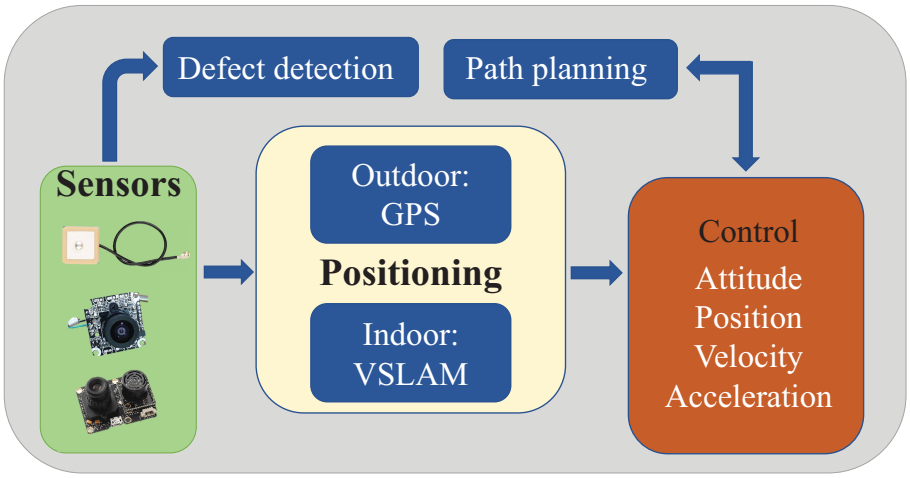


Figure 2.3: Framework of UAV-based autonomous visual inspection

tion tasks autonomously, which reduces costs for the oil and gas industries. What is more, with the development of indoor localisation, intelligent control strategies and path planning methods, the autonomous navigation performance of UAVs and AUVs can still be further improved. The autonomous inspection will also come true with the help of advanced computer vision algorithms. The survey shows that the maturity level of autonomous inspection UAVs and AUVs will be developed, considering the robustness and reliability. The autonomous UAVs and AUVs for inspection do not need human interaction, which can further reduce the operational cost and time-consuming [33]. In this thesis, the UAV will be utilised as the autonomous visual inspection platform.

The schematic of a UAV-based autonomous visual inspection system should consist of localisation, control, path planning and defect detection. This thesis will focus on the localisation and defect detection parts. Considering the limited payload capability of the UAV, the visual sensor used for inspection can also be applied for the localisation purpose. Therefore, the VSLAM systems, which do not require previous knowledge of the environment, have the potential to be

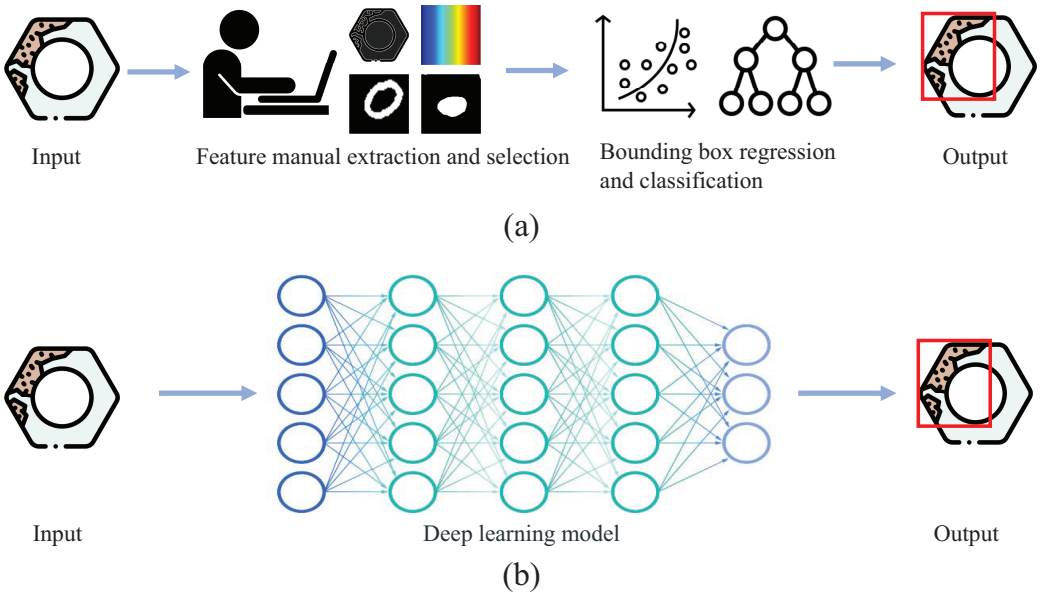


Figure 2.4: Different of the workflow of the traditional corrosion detector (a) and deep learning-based corrosion detection approaches (b).

deployed for UAV indoor autonomous visual inspection [101]. In conclusion, the autonomous visual inspection framework utilised in this thesis is depicted in Fig. 2.3. In particular, this thesis focuses on the process of using sensor data to locate the UAV and identify corrosion defects.

2.2 Vision-based Corrosion Detection Systems

Without the autonomous corrosion detection system, an additional expert engineer is required to monitor the picture sequences collected by the UAVs resulting in an expensive visual inspection procedure. Thus, to realise a fully autonomous inspection system to reduce the labour cost further, the ability to detect corrosion and cracks autonomously is essential. Owing to there is not so much research focused on developing corrosion detection systems for the UAV platform, both the pure image processing techniques focusing on corrosion detection and the

UAV-deployed defects detection algorithms are reviewed in this section.

To detect corrosion from captured images, both traditional approaches and deep learning-based algorithms can be utilised. The main difference between traditional corrosion detection algorithms and deep learning-based corrosion detectors is depicted in Fig. 2.4. For the traditional corrosion detectors, the features to determine the crack regions are defined and selected by experienced engineers. On the contrary, deep learning-based methods automate the learning of feature sets from the given input dataset. Thereby, based on the feature extraction and selection methods, corrosion detectors can be classified into traditional algorithms and deep learning-based methods. Additionally, the types of corrosion suitable for visual inspection are also introduced.

2.2.1 Types of Corrosion for Visual Inspection

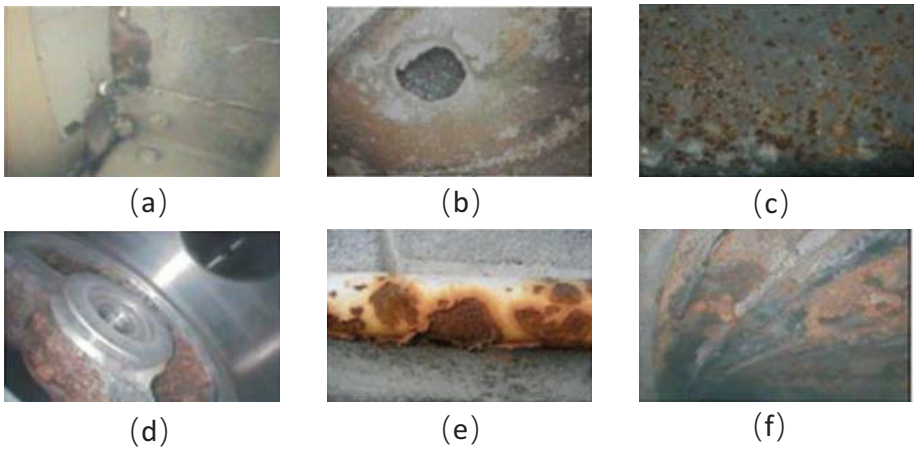


Figure 2.5: Corrosion images (adapted from [14]). (a) Galvanic, (b) crevice, (c) pitting, (d) dealloying, (e) exfoliation and (f) erosion.

Corrosion is commonly defined as the degradation of a material, typically a metal, or its properties due to a reaction with the surrounding environment [102]. In order to identify metallic corrosions through visual inspection, it is necessary

for corrosions to be observed on the surface. In addition, their sizes need to be large enough to be detected by ordinary visual sense. Based on the visual characteristics of corrosions, there are six types suitable for visual inspection, namely galvanic, crevice, pitting, dealloying, exfoliation, and erosion [14]. Fig. 2.5 shows the six types of corrosion.

Galvanic corrosion occurs on the metallic surface at discrete portions by an anodic portion and a cathodic portion [103]. Galvanic corrosion describes the corrosion of one metal over another when they are in electrical contact, particularly in the presence of a suitable electrolyte. Crevice corrosion arises near a crevice between two joining surfaces. Crevice corrosion results from differences in oxygen concentrations between the inside and outside of the crevice. [104]. Pitting corrosion happens within small areas on the metallic surface. These areas are covered with impurities or water and have a lower concentration of oxygen, causing them to function as the anode, while the surrounding areas serve as the cathode. As a result, the metal undergoes dissolution through the electrochemical mechanism [105]. Dealloying mainly occurs in alloy metals, and its mechanism is similar to difference of materials in the galvanic series. One of the alloying elements serves as the anode, while the others act as the cathode. When they are exposed to an electrolyte, an electrochemical reaction is initiated. As a consequence, the reactive components are lost, and the alloy metal preserves the corrosion-resistant elements in a porous state [106]. Exfoliation corrosion takes place when corrosion spreads along intergranular pathways parallel to the material surface. This action causes a wedging effect that separates metal layers [107]. Erosion results from the relative movement between metal surfaces and corrosive fluids. When the fluid contains solid particles that are harder than the affected metal surface, erosion occurs due to the combined effects of corrosion and abra-

sion. Conversely, when the fluid contains particles softer than the metal, erosion occurs as a result of corrosion and attrition [108].

2.2.2 Traditional Algorithms for Corrosion Detection

Colour, as one of the most basic and popular features, is widely used for computer vision tasks. A colour-based corrosion detector was proposed in [15], and a classifier was trained to classify corrosion by adopting a codeword dictionary consisting of the stacked histogram for red, green and blue colour channels. Utilising colour information for corrosion detection was further investigated in [109]. As Hue-Saturation-Value (HSV) values of corrosion areas are confined to the hue-saturation plane, they utilised a classifier that works over HSV space to recognise corrosion. Shapes and sizes of corrosion were applied to detect and classify the pitting corrosion in [16]. The texture analysis for corrosion detection was proposed in [17]. In their theory, based on image colour, Grey-Level Co-occurrence Matrix (GLCM) and grey-level run lengths, 78 features were extracted from the corrosion area. After that, a decision boundary for classifying corrosion images was constructed by the Support Vector Machine (SVM). In [110], the texture analysis was utilised for pitting corrosion detection. Statistical measurements of colour channels, GLCM and local binary patterns were computed to characterise the properties of the metal surface, and 93 texture features were obtained. The SVM was then employed to detect the pitting corrosion. However, these traditional approaches require previous knowledge about corrosion and its optimal features. Some features used for corrosion detection are shown in Fig. 2.6. Moreover, determining optimal features of corrosions is still challenging [111].

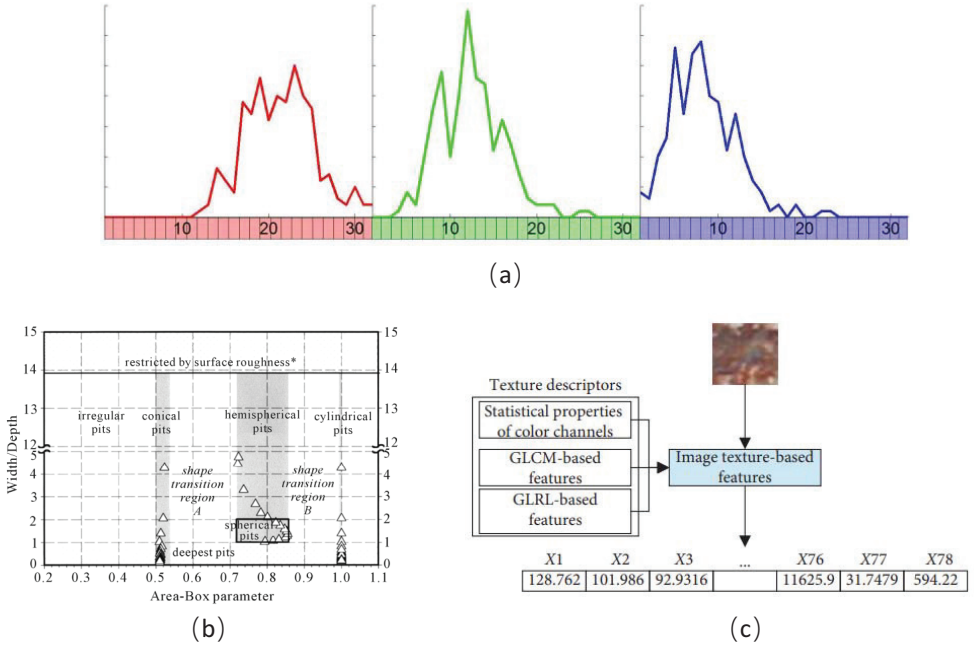


Figure 2.6: Features used for corrosion detection or classification. (a) Codeword consisting of red, green and blue stacked histograms [15]; (b) Diagram used for pit geometry classification [16]; (c) Feature extracted for corrosion identification [17].

2.2.3 Deep Learning-based Methods for Corrosion Detection

Deep learning-based methods, especially CNNs, have made many breakthroughs in computer vision tasks [112]. The CNNs can extract features autonomously for all kinds of objects, which is more accurate and robust than traditional hand-designed features [113], such as edges and shapes. It has led to significant improvements in many vision inspection tasks. Several studies on high-accuracy corrosion detection with CNNs have already been proposed.

Researchers finetuned a CNN network to classify and identify the corrosion position through the sliding window technique [37]. The results indicated that the use of a CNN was shown to outperform the previous state-of-the-art corrosion detection approach where wavelet features were used. Based on corrosion levels,

a custom designed CNN was utilised classify oil and gas pipeline images. Then, the recursive region-based method was proposed to locate corrosion regions. Experimental result showed that the classification accuracy of the customised CNN was 98.8% [36]. Du *et al.* [114] proposed a two-parallel CNN architecture to extract corrosion features, and these features are classified by the SVM. Compared with the traditional corrosion detection methods, this study fills the blank of detecting the corrosion degree of the grounding grid by the image method. Apart from the aforementioned approaches, there are also some other works that adopt CNN-based object detection approaches to locate corrosion directly. Faster Region-based Convolutional Neural Network (Faster R-CNN) [115] was trained by 1737 images to detect steel corrosion and bolt corrosion in [116]. It overcame the challenge of determining the window size when utilising traditional sliding window methods to localise the damage. Li *et al.* [117] modified the You Only Look Once (Yolo) [118] architecture to 27 convolutional layers to detect corrosion of flat steel. The defect detector reached a performance of 99% defect detection rate with a speed of 83 FPS. The rust was detected through the improved Single Shot multibox Detector (SSD) [119] in [120]. Specifically, the backbone of the SSD was modified and the attention mechanism was adopted. As a result, the detection accuracy of the proposed method reached 90.2%. Andersen *et al.* [121] added a regression head to Faster R-CNN to detect corroded areas of vessels, and mAP of the corrosion detection was 49.8%. Multiple structure defects was detected by the Faster R-CNN with the modified base network, and the detection result reached 93.11% mAP. The backbone of Faster R-CNN was improved to detect structure defects [122]. The DEA_RetinaNet [123], which adopted difference channel attention and adaptively spatial feature fusion to RetinaNet [124], was developed to identify steel surface defects, and it achieved 78.25% mAP. The

mAP of the defect defection results was 93.31%. Li *et al.* [125] used the Faster R-CNN to detect defects on aluminium surface, and the mAP of 79.49% was achieved.

2.2.4 Key Findings

Based on the visual characteristics of corrosion, there are six types suitable for visual inspection, namely galvanic, crevice, pitting, dealloying, exfoliation, and erosion. However, as shown in Table 2.3, different researchers reclassified the corrosion into different categories.

Then, corrosion detectors were proposed to detect corrosion in digital images. The traditional corrosion detectors aim to investigate the basic common features of corrosion areas, such as colour or texture information. With these features obtained, the whole image can be divided into small patches and classified as a corrosion area or a normal area. However, without prior knowledge about the corrosion area and the correct selection of features, the performance of these detectors degenerates significantly, and their performance is influenced by the environment dramatically [111].

Deep learning-based corrosion detectors learn the appropriate features autonomously. These studies have shown that deep learning-based corrosion detectors outperform traditional corrosion detectors towards detection accuracy [37] [114]. As a result, the inspection quality will improve. The summary of deep learning-based detectors for corrosion visual inspection is listed in Table 2.3. If the required information is not presented in the publication, it will be represented as Not Applicable (N/A). Tera Floating Point Operations Per Seconds (TFLOPS) for single-precision floating-point format data, which represents the

Table 2.3: Summary of deep learning-based corrosion detectors

Reference	Corrosion types	Model	Image size	Graphics card	TFLOPS	FPS	Accuracy
[36]	4 types of corrosion including, no corrosion, Low-level corrosion, etc.	Custom-CNN with recursive region-based method	256 × 256	Quadro M5000	4.252	More than 1	Accuracy of 98.8%
[37]	Corrosion	Corrosion7	128 × 128	GTX Titan X	6.691	30.30	Mean F1 score of 96.58%
[114]	4 types of corrosion including severe corrosion, mild corrosion, etc.	CNN with SVM	32 x 32	N/A	N/A	N/A	Accuracy of 89.37%
[116]	5 types of corrosion including medium steel, bolt corrosion, etc	Improved Faster R-CNN	500 × 375	GTX 1080	8.873	33.33	mAP of 87.8%
[117]	6 types of defects including scar, scratch, etc.	Improved Yolo	300 × 300	GTX 1080Ti	10.69	83	mAP of 97.55%
[120]	Rust	SSD_BotNet50	N/A	Tesla T4	8.14	4	Accuracy of 90.2%
[121]	Edge corrosion, welded seam corrosion	Improved Faster R-CNN	N/A	RTX 2080Ti	13.45	158.73	mAP of 49.8%
[122]	Steel corrosion, steel crack, loosened bolt	Improved Faster R-CNN	1280 × 720	GTX 1080	8.87	16.67	mAP of 93.11%
[123]	8 types of defects, including scratch, dirty crazing, etc.	DEA_RetinaNet	200 × 200	RTX 2060 SUPER	7.18	12.2	mAP of 79.11%
[125]	4 types of defects, including scratches, dirty spots, etc.	Faster R-CNN	N/A	GTX 1650	2.98	17	mAP of 79.49%

computing speed of the Graphics Processing Unit (GPU), is utilised to indicate the computing performance of the GPU.

Table 2.3 shows that there is a lack of consistency when it comes to reporting the performance of inspection systems. The corrosion has been classified to different types by different researchers, and the number as well as the quality of images are also varied in different studies. In addition, different researchers use different matrices to measure corrosion detectors' performance, such as accuracy, F1 score and mean Average Precision (mAP). In addition, there is no human inspector performance reported in the literature, especially for oil and gas facility visual inspection. The Faster R-CNN, SSD, Yolo and RetinaNet are popular baseline models used in the literature, and the mAP is the most popular evaluation matrix. Among these models, the mAP obtained by RetinaNet will be considered as the baseline in this thesis due to its high mAP in many studies [126] [127].

Moreover, existing work only focuses on improving detection accuracy without consideration of UAV onboard applications. Their networks contain a large number of standard convolutions, resulting in high computational complexity. As shown in Table 2.3, the high-end GPUs are required to obtain real-time corrosion detection (at least 20 FPS [39]). Nvidia Jetson TX2 [128], which is one of the most popular UAV onboard computers [129], only has a constrained performance of 0.67 TFLOPS. For this reason, these approaches cannot be applied to UAV onboard platforms directly.

2.3 UAV Positioning Techniques

Robots need to know where it is before they can perform tasks autonomously. Thus, autonomous localisation is the basic requirement for autonomous navigation [130]. However, in the asset inspection scenario, electromagnetically guided localisation systems such as the GPS [131] are extensively utilised in the outdoor environment. In indoor inspection scenarios, vision-based localisation and navigation systems have become popular these days, considering the relatively low requirements and easy deployment of cameras [132]. VSLAM algorithms process the images captured by the onboard camera to locate the robot, estimate its state, and simultaneously build a map of the surrounding environment. In addition, the VSLAM approaches do not require prior knowledge of the environment. These features make it suitable for visual guide systems [133]. Related work has proven the reality of leveraging VSLAM to develop robotic autonomous navigation systems. For instance, the graph-based VSLAM approach has been deployed in UAVs and AUVs to realise robot localisation in [134]. Thereby, in this part, the survey mainly focuses on introducing the GPS and VSLAM-related works.

2.3.1 Global Positioning System

GPS is the most common type of global navigation satellite system, and it consists of 26 satellites to provide 3D information for the receiver. The GPS system was developed by the Department of Defence in the USA. To provide the position for the UAV, GPS should be in contact with at least 4 satellites simultaneously [135]. It is one of the most popular localisation techniques for UAVs in outdoor

environments.

According to the report presented in [136], the global average position domain accuracy of the GPS is less than 1.665m in the horizontal direction and 3.603m in the vertical direction, which is not precise enough for UAV close inspection tasks. Therefore, the use of real-time kinematic devices or post-processing kinematic systems nowadays for UAV outdoor navigation is the most popular solution, owing to their accuracy, which ranges between 20 mm and 50 mm [137]. Although these filter-related technologies are one of the best options on the market for UAV position estimation in outdoor environments, they can be very expensive due to the amount of equipment needed.

Meanwhile, GPS signals are voluble due to the possibility of being blocked temporarily by buildings during outdoor applications [138]. Thus, lots of research adopts filtering techniques to improve the robustness of UAV localisation. In [139], an Extended Kalman Filter (EKF) was applied to estimate the location of those UAVs temporarily losing their GPS connection. The EKF could also be used to fuse information from other sensors, such as inertial navigation sensors, to compensate for the GPS system. According to the trajectory following results presented in [140], the error was around 2.5m. The state-dependent Riccati equation nonlinear filtering was proposed in [141]. It aimed to solve problems related to linearisation, which pose problems for EKF when fusing GPS and inertial data for UAV localisation.

However, in some inspection scenarios, the UAV needs to be deployed in indoor environments such as inside pressure vessels or tanks. Thus, the localisation methods for UAVs in GPS-denied environments should also be developed.

2.3.2 Vision-based Localisation Systems

To substitute the GPS system, a lot of localisation methods based on the Light Detection and Ranging (LiDAR) sensor, time-of-flight sensor and visual sensors have emerged. Due to the stable performance in varied lighting environments and precise results, there are some works that adopted the LiDAR to the UAV platforms [142] [143]. However, the high cost and massive weight still cannot be ignored for small UAV platforms. The ultrasonic and ultra-wideband sensors are light and cost-efficient for UAV localisation in confined spaces. Nevertheless, the requirement for deploying the auxiliary anchor nodes restricts their applications.

With the development of the Simultaneous Localisation and Mapping (SLAM) technology and the capabilities of texture obtained from cameras, the usage of cameras as the input sensors for SLAM is blossoming. Thus, in this section, the survey will mainly focus on the VSLAM technologies, especially VSLAM approaches for low-contrast or textureless environments. Specifically, the VSLAM systems are classified as traditional VSLAM systems and deep learning feature-based VSLAM approaches based on whether deep learning-related features have been utilised in the VSLAM system. The basic theory and related work will be presented in the following parts.

2.3.2.1 Theory of Visual Simultaneous Localisation and Mapping and ORB-SLAM3

VSLAM is an advanced technique for robot environment perception with camera information. The primary purposes of this approach are to address the issue of camera localisation and perceive the layout of surrounding objects [144]. Based

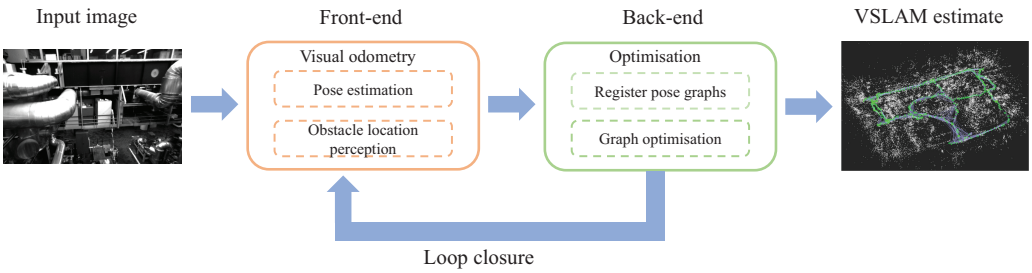


Figure 2.7: The standard architecture of VSLAM (adapted from [18])

on [18], the general schematic of the VSLAM is depicted in Fig. 2.7. Rather than being just one algorithm, VSLAM is more of a notion. The front end and the back end are the two primary parts of the VSLAM system. The VO, also known as the front end, calculates the mobility of the camera between frames and the location of landmarks. The back end serves as the optimisation procedure, and it optimises camera posture as determined by the visual odometer at various times. The loop closure process determines whether the robot has arrived at a previously visited place. Finally, according to the trajectory and surrounding objects captured by the camera, a map will be constructed to represent them. VSLAM is a subset of SLAM that incorporate the camera as the input sensor.

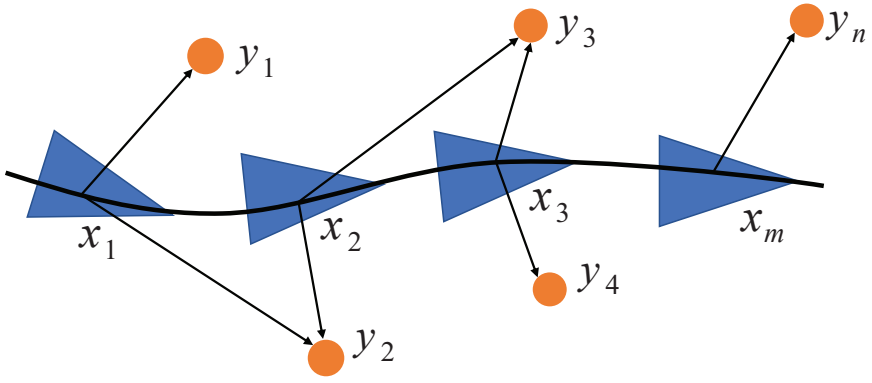


Figure 2.8: The pipeline of SLAM

According to [18], a common formulation of SLAM will be introduced in the following part. As demonstrated in Fig. 2.8, a robot moves in an unknown

environment. The location of the robot is represented by $\mathbf{X} = \mathbf{x}_k : k = 1, ..., m$, and the poses of landmarks observed by the sensor are $\mathbf{Y} = \mathbf{y}_i : i = 1, ..., n$. The pose of the robot at the moment k can be calculated through motion equation:

$$\mathbf{x}_k = f_{mot}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \tag{2.1}$$

where f_{mot} represents the motion model. $\mathbf{u}_k$ is the estimated sensor movement, and the measurement noise for the camera movement is indicated by $\mathbf{w}_k$. With the assumption that the landmark $\mathbf{y}_i$ can be observed at the moment k, the observation data $\mathbf{z}_{k,i}$, which describes the charismatics of landmarks, can be formulated through the observation equation.

$$\mathbf{z}_{k,i} = h_{obs}(\mathbf{y}_i, \mathbf{x}_k, \mathbf{v}_{k,i}) \tag{2.2}$$

where h_{obs} demonstrates the observation model, and $\mathbf{v}_{k,i}$ indicates the measurement noise for landmarks. The motion equation describes the relationship between sensor movement, while the observation model describes the relationship between the sensor and landmarks. Based on practical applications, the $\mathbf{v}_{k,i}$ and $\mathbf{w}_k$ can be modelled in a Gaussian, non-Gaussian or mixed distribution. The fundamental mathematical model of SLAM is made up of motion equation and observation equation. This state estimation problem can be addressed by using a filter or nonlinear optimisation techniques.

An example is provided to show what the format of f_{mot} and h_{obs} might be. Assuming a 2D scenario, a uniform motion model is used to estimate the movement of the sensor, and the sensor pose $\mathbf{x}_k$ consists only position $[x_x, x_y]_k^T$. The movement between timestamp $k-1$ and k is represented by $[\Delta x_x, \Delta x_y]_k^T$.

Then, the motion equation can be reformulated by:

$$\begin{bmatrix} x_x \\ x_y \end{bmatrix}_k = \begin{bmatrix} x_x \\ x_y \end{bmatrix}_{k-1} + \begin{bmatrix} \Delta x_x \\ \Delta x_y \end{bmatrix}_{k-1} + \mathbf{w}_k \tag{2.3}$$

At the same time, a landmark $\mathbf{y}_i$ at position $[y_x, y_y]_k^T$ is observed by the sensor. If the observation data $\mathbf{z}_{k,i}$ obtained from the sensor is relative distance of $\mathbf{x}_k$ and $\mathbf{y}_i$, and it is represented by $d_{k,i}$. Then the observation equation can be represented by:

$$d_{k,i} = \sqrt{(y_{x,i} - x_{x,k}))^2 + (y_{y,i} - x_{y,k})^2} + \mathbf{v}_{k,i} \tag{2.4}$$

Based on the basic theory mentioned above, lots of VSLAM systems have been developed. The ORB-SLAM3 [19] is claimed as the most robust VSLAM system in the literature. As shown in Fig. 2.9, it has three main threads, i.e., tracking, local mapping and loop and map merging. Besides, it utilises the multi-map technology named Atlas to improve the robustness of the system.

Tracking thread: The tracking thread processes each frame of the video. Firstly, it will initialise the system with the first frame that contains sufficient feature points. Then, the system starts tracking the pose of the current frame. When the tracking fails, the relocation function is activated to relocate the current frame among all the maps. If being relocated, the corresponding map becomes the active map. Otherwise, a new active map is created while all other maps are stored in the Atlas as non-active maps. The bundle adjustment is applied to process the active map points to minimise the reprojection error and optimise the pose of the current frame. If the current frame meets certain conditions, it will be selected as a keyframe.

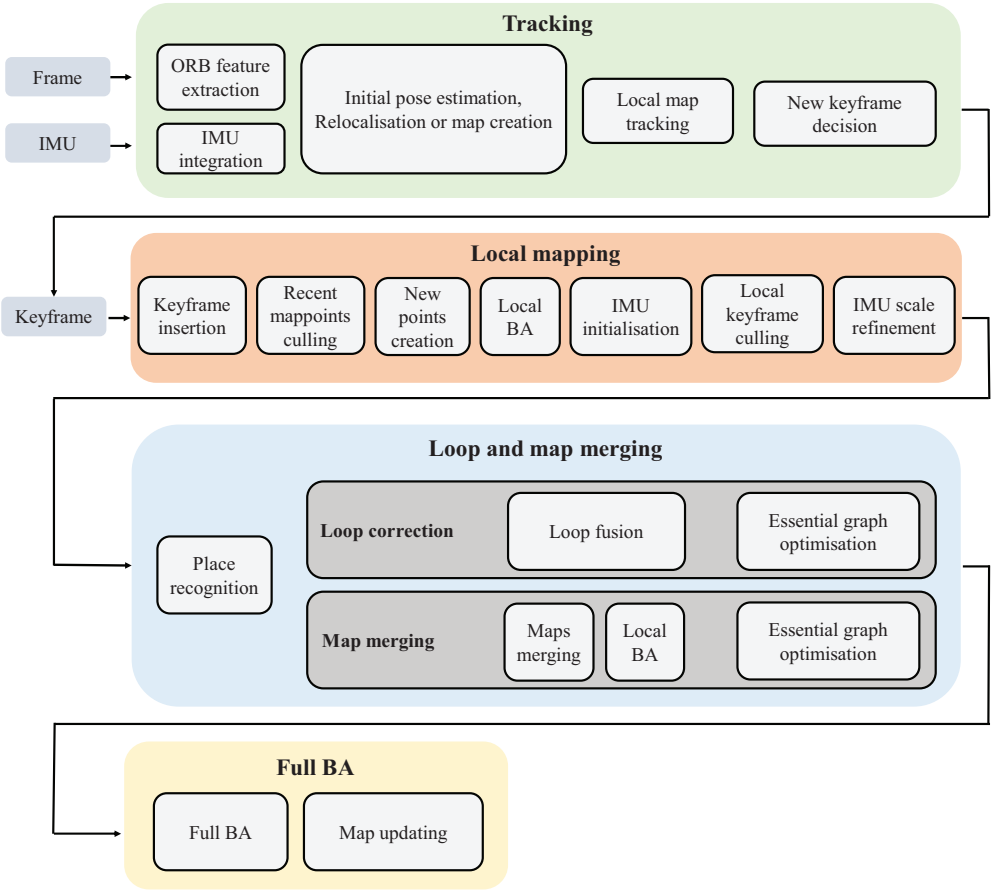


Figure 2.9: The structure of the ORB-SLAM3 (adapted from [19])

Local mapping thread: Keyframes generated by the tracking thread are sent to the local mapping thread. The newly added keyframe and corresponding map points are inserted into the active map. A local bundle adjustment is performed to optimise the poses of map points and keyframes. To maintain the size of the map, the redundant map points and keyframes will be deleted.

Loop and Map Merging Thread: The input of this thread is the refined keyframes by the local mapping thread. It detects the overlap scenes between the active map and the Atlas. If it exists in different maps, the active map and matching map will be merged as a new active map. Otherwise, the loop closure is

utilised in the active map. When the loop correction is finished, an independent thread executes a global bundle adjustment to reduce the accumulated drift error.

Atlas: The Atlas is a multi-map representation that contains all non-active maps in the ORB-SLAM3. The active map is utilised by the tracking thread to locate the incoming frames. All other maps are treated as disconnected maps and stored in the Atlas. When the system re-enters the mapped scene, the active map will be merged with the relative non-active map.

2.3.2.2 Traditional VSLAM Systems

Depending on whether image feature extraction and matching are needed, these methods can be grouped into two categories, i.e., feature-based methods and direct methods. In terms of direct VSLAM methods, some works have been done towards complex lighting environments. Extensive experiments have been conducted to verify direct VSLAM systems towards changing illumination environments in [145]. Experiments showed that most direct VSLAM systems failed due to abrupt illumination changes while the brightness constancy assumption was adopted. Sun *et al.* [146] combined the RGB channel linearly to compensate for the lighting changes. In [147], illumination changes were modelled for affine lighting correction. Thereby, the illumination invariance was handled for the direct VSLAM system. As these methods still rely on the brightness constancy assumption, direct VSLAM systems still cannot handle complex lighting environments.

In recent years, some advanced feature-based VSLAM systems have emerged, and they are suitable for embedded computing platforms due to the sparse points utilised for pose estimation [148]. To overcome the challenges caused by the

unideal lighting conditions, image enhancement has been widely utilised to handle the challenging lighting environment. The Histogram Equalisation (HE) was adopted for the HE-SLAM, which improved the contrast of captured low-contrast images in [149]. Compared to the ORB-SLAM2 [150], the HE-SLAM was more robust in a harsh environment. However, the HE is significantly affected by background noises. To improve the robustness of the HE-SLAM, Yang *et al.* [151] adopted the Contrast Limited Adaptive Histogram Equalisation (CLAHE) algorithm to the ORB-SLAM2 framework. The trajectory generated by this method was closer to the ground truth than that calculated by the HE-SLAM and ORB-SLAM2. The Dim-light Robust Monocular SLAM (DRMS) was proposed in [152], which utilised the linear transformation and CLAHE algorithms as the image preprocessing to enhance the brightness and contrast of input images. After that, the performance of the proposed VSLAM in the dim-light conditions was improved. However, the CLAHE algorithm calculates the neighbourhood histogram for each pixel and performs histogram equalisation processing for sub-regions of the image, which is computationally expensive [153]. Meanwhile, the aforementioned methods mainly focus on either global or local enhancement for all kinds of images, and they may not perform well for different types of images [154]. Moreover, the decreased sharpness of images owing to image transformations should also be taken into consideration [155]. However, even with the enhanced images, sufficient feature points still could not be extracted in some challenging environments. Thus, investigating the VSLAM combined with multiple features has gained research interest. Pumarola *et al.* [156] proposed the PL-SLAM that relied on line features as well as the ORB features. To this end, a more robust performance in environments with challenging illumination conditions was achieved. Huang *et al.* [157] processed ORB and Brisk feature points at the same time for a low-lighting envi-

ronment to improve the robustness of the VSLAM system. However, extracting multiple features requires extra computing resources. Thereby, their applications for mobile robots are restricted due to the robots' limited onboard computing capabilities. In addition, the application scenarios are still constrained by the rules governing feature design, and different types of features may exhibit inconsistencies or mismatches with each other in some scenarios, leading to degraded performance of the VSLAM system (as shown in Tables 5.1 and 5.2).

2.3.2.3 Deep Learning Feature-based VO and VSLAM Approaches

As there are not many works aiming to deploy deep learning feature-based VSLAM systems on UAVs, this part focuses on introducing general deep learning feature-based VO and VSLAM systems in this section. A deep neural network was adopted to increase representations of captured images for the VSLAM algorithm, which improved the robustness of the VSLAM system in high dynamic range environments [158]. DarkSLAM [159] leveraged EnlightenGAN [160] to enhance low-light images. With the enhanced image, robust performance was achieved in the low illumination environment. In [161], a framework consisting of CNN and Recurrent Neural Network (RNN) was developed to detect keypoints and their corresponding descriptors for pose estimation. The performance of the whole system was on par with the ORB-SLAM2. DF-SLAM [162] combined the FAST detector for keypoint detection and the TFeat network [163] for feature point description. The feature points and descriptors were then deployed into the ORB-SLAM2, and the DF-SLAM outperformed the ORB-SLAM in some image sequences. The HF-Net [164] was incorporated into the DXSLAM [165] to extract local and global features. Compared with traditional VSLAM systems,

the robustness of the DXSLAM in different environments was improved significantly. The SuperPointVO [166] adopted the framework of traditional VO systems but incorporated the self-supervised interest point detection and description method named SuperPoint [167] to replace the hand-engineered feature extraction, and it achieved similar performance to the advanced VO systems. The LIFT-SLAM [168] employed the Learned Invariant Feature Transform (LIFT) [169] to replace the ORB feature extraction module, and the robustness of the VSLAM system was enhanced. Compared with traditional methods, these systems leverage the stability and robustness of the deep learning-based feature description module in the VSLAM pipeline to obtain more accurate and robust localisation performance. However, deep learning-based methods need even more computing resources than traditional methods, and most of these works focus on improving localisation while ignoring efficiency. In other words, integrating deep learning techniques into the VSLAM pipeline in performance-constrained platforms, such as UAV onboard platforms, is still an open problem [170].

To this end, there are also some works starting to improve the efficiency of the deep learning feature-based VSLAM systems. Several simplified networks have been developed and integrated into the VSLAM pipeline. MobileNetV2 [171] was used as an encoder and trained using the knowledge distillation method in [172] to greatly reduce the model size and increase the running speed. A quantised local feature extraction module was described in [173]. A simplified network which contains four convolutional layers in the backbone network was proposed in [174]. Even so, most of these methods still need more execution time than traditional VSLAM algorithms. If deep learning feature-based algorithms can achieve similar or shorter runtime compared to traditional methods, they will have broader applications and the potential to be deployed on computing resource-constrained

platforms such as the UAV onboard platform. GCNv2 [175] predicted the keypoint and descriptor in a low-resolution feature map to improve the efficiency, and the simplified network named GCNv2-tiny based SLAM ran at 20Hz on the Nvidia Jetson TX2. Nevertheless, the GCNv2 predicts the projective geometry rather than generic feature matching. Thus, the generalisation capability was limited.

2.3.3 Key Findings

GPS is the most popular positioning system for UAV in outdoor environments and can provide global position information. There are lots of autonomous UAV navigation systems developed utilising the position information from the GPS. However, it can be interfered easily. Therefore, lots of works have adopted filter-based technologies to improve the accuracy and robustness of GPS systems. Even so, GPS is blocked in the indoor environments.

Considering the requirement for sensors, VO/VSLAM, which only relies on the camera, shows its advantages for UAV-based indoor applications. There is no doubt that the previously developed methods have made a significant improvement in the robustness and accuracy of VO/VSLAM systems. However, these approaches are still in the start-up stage. Compared with direct VO/VSLAM methods, feature-based VO/VSLAM systems are more robust. The quality of extracted feature points has a significant impact on VO/VSLAM localisation accuracy. Thus, lots of works have tried to improve the quality of extracted features. Moreover, thanks to the development of deep learning based feature extraction techniques, researchers stared to substituting traditional feature points with deep learning-based feature points.

Table 2.4: Summary of techniques adopted in VO/VSLAM to improve feature extraction capability

Methods		References	System	Advantages	Limitations
Traditional image enhancement	HE	[149]	Intel Core i7-8750H CPU and 8 GB RAM	Improved feature extraction performance, low computational complexity	Contrast under- and over-enhancement, limited performance for low texture environment
	HE	[157]	Intel Core i5-6300HQ CPU and 8GB RAM		
	CLAHE	[151]	N/A		
	Linear transformation and CLAHE	[152]	Intel Core i7-9700 CPU and 32GB RAM		
Feature point improvement	Combination of ORB and Brisk	[157]	Intel Core i5-6300HQ CPU and 8GB RAM	Reduced feature point loss, robustness enhancement	High computational demand, limited by feature design rules
	Point and line features	[156]	Intel core i7-4790 CPU and 8GB RAM		
Deep learning based image enhancement	CNN and RNN	[158]	Intel Core i7-4770K CPU with Nvidia GTX Titan	Improved adaptability and stability	Require a large volume of training data, high computational complexity and the demand for high-performance GPUs
	EnlightenGAN	[159]	N/A		
Deep learning based feature point improvement	Feature point detection and description with CNN and RNN	[161]	Intel Core i7-4790 CPU with Nvidia GTX 1080	Robust feature points identification and description, reduced tracking loss	
	Feature point description with TFeat network	[162]	Intel Core i5-4590 CPU with Nvidia TITAN X		
	Feature point detection and description with HF-Net	[165]	Alienware laptop equipped with Nvidia GTX 1070		
	Feature point detection with SuperPoint	[166]	Intel Core i7-8700K CPU with Nvidia GTX 1080ti		
	Feature point description with LIFT	[168]	N/A		
	Feature point detection and description with MobileNetV2 encoder and SuperPoint decoder	[172]	Intel Core i5-7300H CPU with Nvidia GTX 1050		
	Feature point detection and quantised description with CNN	[173]	Nvidia GTX 1080Ti		
	Feature point detection and description with CNN	[174]	Intel Core i5-4200H CPU and Nvidia GTX 950 M		
	Feature point detection and Binarised description with CNN and RNN	[175]	Intel Core i5-4200H CPU and Nvidia GTX 950 M, Nvidia Jetson TX2		

The summary of techniques used to improve the feature extraction capability of VO/VSLAM systems is listed in Table 2.4. If the required information is not presented in the publication, it will be represented as N/A. There is a lack of consistency when it comes to reporting the performance of different VO/VSLAM systems. Some researchers use the Root Mean Square (RMS) Absolute Trajectory Error (ATE) [159], Relative Position Error (RPE) [151], ATE [156] or Absolute Pose Error (APE) [157] to evaluate the performance of VO/VSLAM systems. Some works report the keyframe error [156], while some studies present the trajectory error [152]. Moreover, different datasets are used by different researchers. Some studies constructed the datasets themselves [158]. Even when the same dataset is used, different sequences are chosen by different researchers to test the VO/VSLAM systems [161] [174]. Furthermore, some researchers modified the sequences, such as selecting specific frames [175] or deleting frames from the datasets [147]. The VO/VSLAM may perform significantly differently in various scenarios. The VSLAM system was evaluated by 27 sequences in [173], with the minimum ATE being 9.22 mm and the maximum ATE reaching 513.91 mm. These challenges make it very difficult to compare localisation accuracy. Thus, the effectiveness of the VSLAM system in the inspection scenario needs to be verified. Based on the comparison experiments carried out by the authors themselves, deep learning feature-based VO/VSLAM systems outperform traditional feature-based VO/VSLAM methods in many scenarios. However, as shown in Table 2.4, desktop-level GPUs are required, and the high demand for computing resources still restricts their application on UAV platforms. Thus, the performances of ORB-SLAM3 [19], claimed to be as robust as the best traditional VSLAM systems available in the literature and significantly more accurate, are considered as the baseline in this thesis.

2.4 Principles of Convolution Neural Networks

The basic CNN model is comprised of the convolution layer, pooling layer and full connection layer [176]. As its name indicates, the primary component of the CNN architecture that accomplishes feature extraction is the convolutional layer. Combining linear and nonlinear processes, such as the convolution operation and the activation function, is common practice in feature extraction.

Convolution is known as a special kind of liner process that is utilised to extract features from input data. It transforms the input with the kernel. By multiplying and summing the elements between each element of the kernel and each point of the input, an output, also known as a feature map, is produced. To represent different facets of the input feature maps, an indefinite number of output feature maps are made using various kernels. Different kernels may be treated as different feature extractors in this way. Even though it is commonly referred to as convolution, the operation carried out on picture inputs using CNNs is more accurately described as cross-correlation, which is a slightly modified form of convolution in which one of the inputs is time-reversed [177]. According to [178], the format of the convolution can be represented by :

$$Y = W * X \tag{2.5}$$

where the W is the kernel weight, and it convolutes with the input X.

The convoluted value $y_{i,j}$ can be calculated through

$$y_{i,j} = \sum_{a=1}^{kh} \sum_{b=1}^{kw} w_{-a,-b} x_{i+a,j+b} \tag{2.6}$$

where height and width of the kernel are represented by kh and kw, respectively. $w_{a,b}$ indicates the value of element in the kernel weight at the position (a, b). Similarity, $x_{i,j}$ is the value of element in the input matrix at the position (i, j). Thanks to the convolution technique explained above, the centre of each kernel cannot overflow the outermost input tensor element. It shrinks the height and width of the output feature map in comparison to the input tensor. To solve this problem, padding, usually zero padding, is utilised. In this method, rows and columns of zeros are added to each side of the input tensor, allowing the kernel's centre to fit on the element's outermost point while maintaining the same in-plane dimension throughout the convolution operation. Zero padding is typically used in modern CNN designs to maintain in-plane dimensions so that more layers may be applied in the model. Without zero padding, the size of each subsequent feature map would decrease following the convolution process. A stride, which also describes the convolution procedure, is the distance between two successive kernel points. The most popular stride to use is 1. However, to produce downsampling of the feature maps, a stride greater than 1 is occasionally employed. A pooling procedure is an alternate method to achieve downsampling, which will be introduced later.

The results of a linear process, such as convolution, are then fed into a nonlinear activation function. Smooth nonlinear functions, such as the Sigmoid [179] or hyperbolic tangent (Tanh) function [180], were previously used because they are mathematical representations of biological neuron behaviour. Due to the simple deployment feature, Rectified Linear Unit (ReLU) [181] is the most popular nonlinear activation function utilised at the moment. The formulas of the Sigmoid, ReLU and Tanh are shown in Eq.(2.7), (2.8) and (2.9), respectively. Their curves

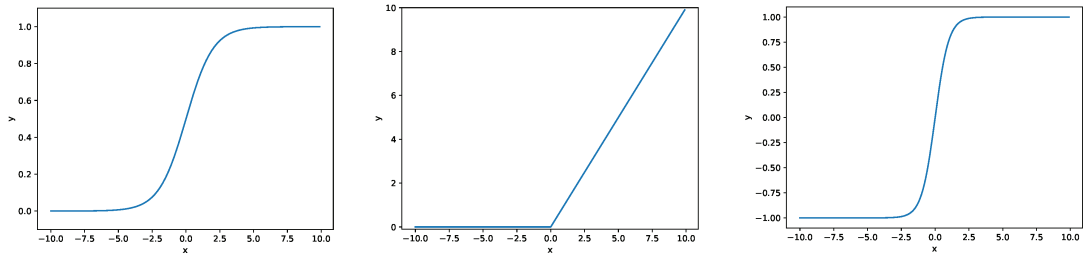


Figure 2.10: Curves of commonly utilised activation functions. (a) Sigmoid, (b) ReLU and (c) Tanh

are also illustrated in Fig. 2.10.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

$$f(x) = max(0, x) \tag{2.8}$$

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.9}$$

A pooling layer serves as a common downsampling method that reduces the in-plane dimensionality of the feature maps and adds translation invariance to minor shifts and distortions. To this end, the number of resulting learnable features is reduced. Unlike convolution layers, there is no learnable parameter in any of the pooling layers. Similar to convolution layers, hyperparameters in pooling layers include the filter size, stride, and padding.

Max Pooling (MaxPool), the most common type of pooling procedure, separates patches from the input feature maps. After that, the largest value in each patch is used as the output, while all other values are discarded. In practice, it is typical to utilise a MaxPool with a filter of size 2×2 with a stride of 2 in a CNN model. In this technique, the height and width of input feature maps are downsampled by a factor of 2. Instead, the depth dimension of feature maps

remains unchanged.

Another popular pooling operation is Average Pooling (AvgPool). The main difference between the MaxPool and AvgPool is described in Fig. 2.11. In contrast to the MaxPool, which outputs the max value of patches, the AvgPool averages all components of patches as an output to downsample the feature map. Meanwhile, the depth channel of feature maps is reserved.

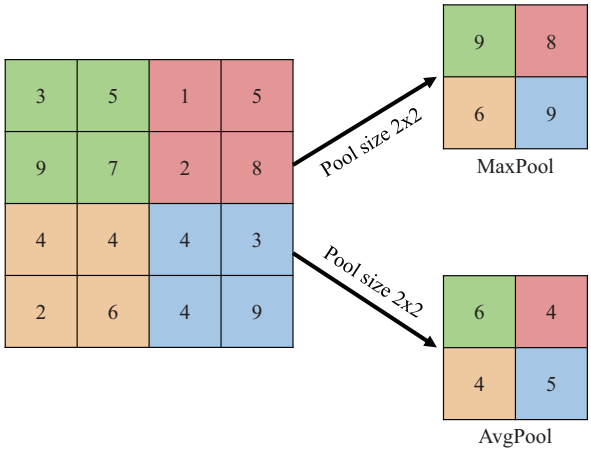


Figure 2.11: The difference of max pooling and average pooling

Fully connected layers, also referred to as dense layers, have a learnable weight linking each input to each output. The feature maps retrieved by the CNN model, which is comprised of convolution layers and pooling layers, are mapped by a subset of fully connected layers to produce the final results.

2.4.1 Depthwise Separable Convolution

The DSConv [182] can reduce CNN parameters significantly. Unlike the traditional convolution processes images from height, width and channel dimensions simultaneously, the DSConv divides the convolution process into depthwise convolution and pointwise convolution. Fig. 2.12 demonstrates the comparison of

the DSConv with standard convolution.

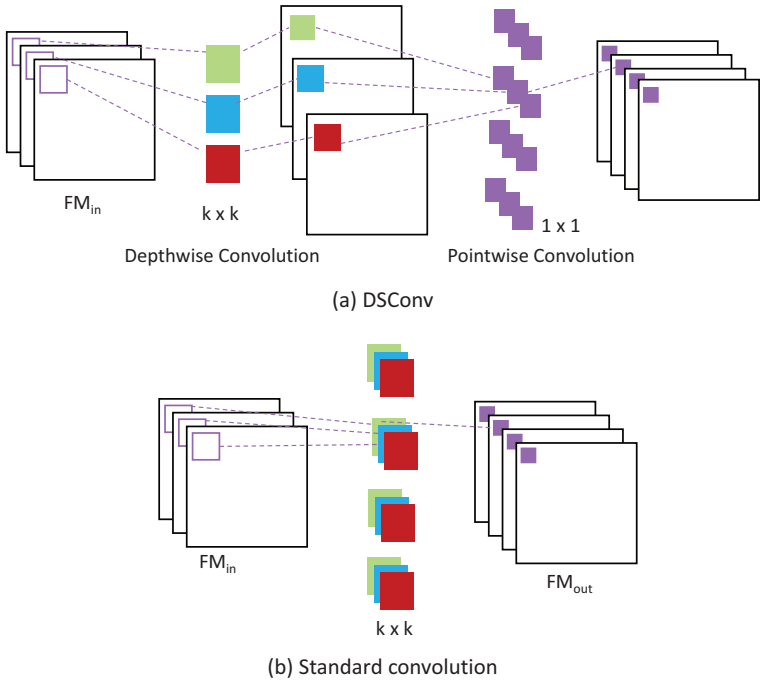


Figure 2.12: Comparison of the DSConv with standard convolution

The first step of the DSConv is a depthwise convolution. In this step, the number of filters is the same as that of input channels, which ensures that only one feature map is generated per input channel. The equation of the depthwise convolution is shown as follows:

$$DWConv(W_d, X)_{(i,j)} = \sum_{m,n}^{M,N} W_d(m,n) \cdot X_{(i+m,j+n)} \tag{2.10}$$

where W_d indicates the weight matrix of depthwise convolutional filters. X denotes input feature maps. (i, j) represents the coordinates of a point within the output feature maps. M and N are the height and width of input feature maps. Meanwhile, m and n represent the height and width of the convolutional filter, respectively.

The second step involves applying a set of 1×1 convolutional layers to fuse feature maps generated by the depthwise convolution. This is called the pointwise convolution. The pointwise convolution focuses on the combination of spatial features, which only changes the number of channels while keeping the width and height of feature maps. The formula of the pointwise convolution is written as:

$$PWConv(W_p, x)_{(i,j)} = \sum_{c}^{C} W_p(c) \cdot x_{(i,j,c)} \tag{2.11}$$

where W_p indicates the weight matrix of pointwise convolutional filters. x denotes the output feature maps generated by the depthwise convolution operations. C is the total number of channels of input feature maps. c represents the channels of convolution filters.

Overall, the whole process can be represented by:

$$\begin{aligned} &DSConv(W_d, W_p, X)_{(i,j)} = \\ &PWConv(W_p, DWConv(W_d, X)_{(i,j)})_{(i,j)} \end{aligned} \tag{2.12}$$

At the same time, the formula of the standard convolution with weight matrix W can be represented by:

$$Conv(W, X)_{(i,j)} = \sum_{m,n,c}^{M,N,C} W(m, n, c) \cdot X_{(i+m,j+n,c)} \tag{2.13}$$

The parameters of the DSConv are reduced significantly compared with the traditional convolution. There is an assumption that the number of output channels is o. According to Eq. (2.13), total parameters of the standard convolution are $m \times n \times c \times o$. While the DSConv is utilised to generate the same output feature maps, based on Eq. (2.12), total parameters are $m \times n \times c + c \times o$. A quanti-

fied comparison of parameters between the DSConv and standard convolution is presented:

$$\frac{m \times n \times c + c \times o}{m \times n \times c \times o} = \frac{1}{o} + \frac{1}{m \times n} \tag{2.14}$$

For example, input feature maps contain 3 channels. The convolutional kernel size is 3×3, and there are 4 sets of convolutional filters to output 4 feature maps. The standard convolution processes images from height, width and channel dimensions simultaneously, and the number of parameters is 108. Meanwhile, the same input feature maps are processed by the DSConv to output the same size and channels of feature maps. First, every single channel of the input feature map is processed by a 3×3 convolutional filter. Then, 4 sets of $1 \times 1 \times 3$ convolutional filters are utilised to process the generated feature map and output 4 feature maps. The number of total parameters of the DSConv is 39, which is almost 1/3 of the traditional convolution.

2.4.2 Key Findings

An introduction to the basic CNN is presented in this section. The basic CNN model consists of the convolution layer, pooling layer, and full connection layer. The CNN layer aims to extract features from the input data, the pooling layer acts as the downsampling method to reduce the dimension of the feature maps, and the full connection layer integrates the features and maps them to the final results. The CNN model is a multi-layer structure that extensively uses convolution operations [183]. The DSConv divides the convolution process into depthwise convolution and pointwise convolution, resulting in a significant reduction of CNN parameters. According to equations presented in the above section, parameters are used for computations with the input feature maps. Therefore,

through the reduction of CNN parameters, the computing complexity of CNN is reduced, which makes the CNN model have the potential to be deployed on the UAV onboard platform.

2.5 Summary

This chapter reviews different types of inspection methods, robotic platforms, UAV positioning systems, vision-based corrosion detectors, and the principles of CNNs. The challenges investigated in this thesis are derived from the following sections:

In Section 2.1, various inspection methods and robotic platforms are reviewed. The advantages and drawbacks of different inspection methods and robots are analysed. Visual inspection is the most basic means for assessing the surface conditions of the infrastructure, and the UAV has gained increasing interest due to its efficiency in performing inspection tasks. Therefore, utilising autonomous UAVs to perform visual inspection tasks has the potential to assess the infrastructure effectively and efficiently. Due to the flexibility and manoeuvrability of small UAVs, they can easily access confined indoor environments, such as pressure vessels. This feature allows them to efficiently perform indoor visual inspection tasks. However, there has been a lack of investigation of the autonomous UAVs for visual inspection in confined indoor environments due to the hazardous conditions and the limited payload capacity of small UAVs. Therefore, the feasibility of the autonomous UAV with VSLAM, which does not require extra sensors or prior knowledge of the environment, for autonomous visual inspection of a pressure vessel is explored in Chapter 3.

Section 2.2 introduces corrosion types and comprises corrosion detectors in the literature. The performance of traditional corrosion detectors depends on human-crafted features such as colour and texture information. Thereby, the selection of optimal features determines the overall performance of the corrosion detector, and the feature selection process relies on the experience of the engineer. Deep learning-based corrosion detectors do not need to set the optimal features, and they can learn the representative features from the training dataset autonomously. As a result, it can reduce reliance on human experts and achieve high-accuracy detection results. As summarised in Table 2.3, it is difficult to make a fair comparison of different corrosion detectors. Moreover, there is no human inspector performance reported in the literature. Considering the popular models used in the literature, the mAP of RetinaNet is considered as the baseline in Chapter 4. The most popular UAV onboard platform, the Nvidia Jetson TX2, has a performance capacity of only 0.67 TFLOPS. Consequently, there are still challenges in deploying existing deep learning-based corrosion detectors on UAV onboard platforms to achieve real-time (at least 20 FPS) corrosion identification with satisfying corrosion detection accuracy, primarily due to the demand for high-end GPUs.

Section 2.3 has been devoted to UAV positioning systems, especially GPS and VO/VSLAM systems. GPS technology has proven to be an extraordinarily valuable and versatile technology. It relies on satellites and is suitable for rough UAV outdoor localisation. However, when it comes to the small size of UAVs and indoor inspections, the accuracy of GPS is insufficient and even unavailable. The VO/VSLAM system does not rely on the satellite system and can be deployed in indoor environments. Considering the robustness and computing complexity of VO/VSLAM systems, feature-based VO/VSLAM systems are more feasible

for embedded platforms. Since UAVs need to be deployed in complex environments with varying illumination conditions where extracting sufficient feature points becomes challenging, incorporating image processing technology with the VO/VSLAM system appears to be a potential solution. However, some studies customised the dataset to show the high localisation accuracy of VO/VSLAM systems. Therefore, the issue of extracting sufficient feature points and deploying VO/VSLAM in complex lighting environments, such as dark or overly bright environments, still needs to be addressed. In addition to the complex lighting conditions, textureless environments also challenge the feature extraction capability of the VO/VSLAM system, leading to degraded localisation accuracy and robustness. Combining different types of features and even substituting the feature extraction methods with deep learning-based methods has gained a lot of interest. As shown in Table 2.4 and the summary presented in Chapter 2.3.3, adopting deep learning-based feature points has proven its effectiveness in handling textureless environments. However, there is still a gap in adopting deep learning-based feature points for UAV onboard platforms to cope with challenging environments due to the high demand for desktop-level GPUs. Because of the accurate and robust performance of ORB-SLAM3, it is chosen as the baseline VSLAM system in this thesis.

Chapter 3

Simulation of UAV-based Autonomous Internal Visual Inspection of a Pressure Vessel

3.1 Introduction

From the introduction above, it is clear that visual inspection plays a vital role in maintaining the proper functions of industrial facilities. Besides, developing UAV-based autonomous visual inspection systems can improve the inspection efficiency significantly, and there are still research gaps in realising the expected functions. According to the findings in Section 2.1.5, the localisation, control, path planning and defect detection are the main components of the UAV-based autonomous visual inspection system. However, there is a lack of investigation into the feasibility of deploying the UAV with VSLAM to achieve autonomous indoor visual inspection. Thereby, this chapter develops a simulation environ-

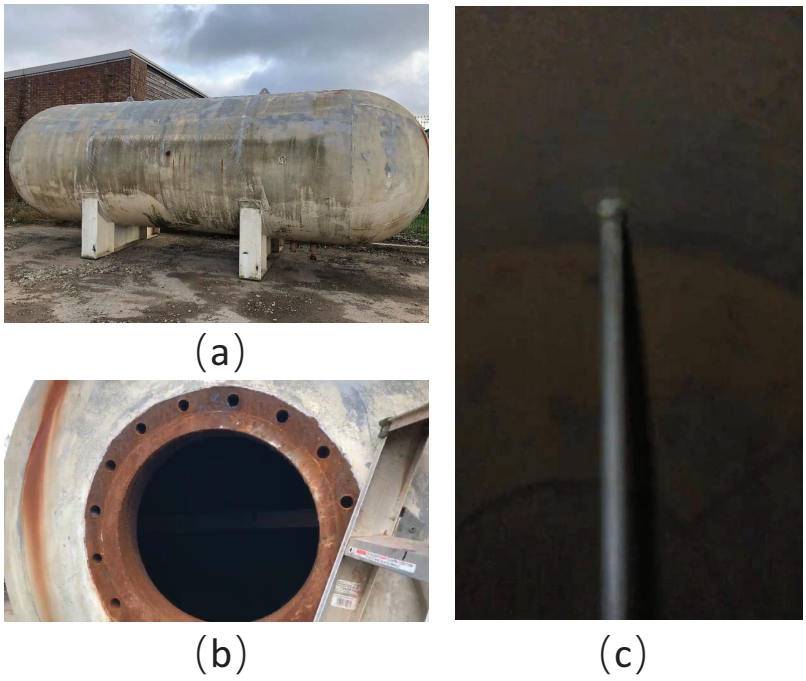


Figure 3.1: One pressure vessel. (a) The overview of the pressure vessel; (b) the entrance of the pressure vessel; (c) the inside of the pressure vessel.

ment to verify the autonomous navigation capability of the UAV equipped with VSLAM. The pressure vessel is a common container in the company, and it is a low-visibility and hard-to-access environment even for experienced engineers. Hence, it is chosen to verify the performance of the developed autonomous UAV system. As the UAV is particularly susceptible to damage in complex and hazardous environments, simulation experiments are used to explore and validate the generalised UAV-based visual inspection solution. The texture of the corrosion is difficult to reproduce in the simulation environments. The corrosion detection part is ignored in this simulation environment, and it will be introduced in Chapter 4. Thus, this chapter investigates the capability of UAV with VSLAM to autonomously record video of the inner surface of the pressure vessel in the simulation environment.

However, the interior of a pressure vessel is a GPS-denied and low-illumination environment (as shown in Fig. 3.1). Autonomous navigation for UAV-refined visual inspection inside the pressure vessel is challenging and has not been evaluated. Moreover, the ORB-SLAM3 fails to extract sufficient feature points to locate the UAV. Therefore, the contrast enhancement method is incorporated into ORB-SLAM3. With the improved ORB-SLAM3, the performance of a stereo vision-based autonomous navigation system for the automatic acquisition of images inside oil and gas pressure vessels with the UAV is evaluated. The main contribution is summarised as follows:

The feasibility of deploying the UAV with VSLAM to achieve autonomous visual inspection in confined and low-illumination indoor environments has been proven through a customised simulation environment for the first time.

The rest of the chapter is organised as follows. Details of the stereo vision-based autonomous navigation approach are given in Section 3.2. In Section 3.3, the experimental environment is demonstrated. Section 3.4 shows and analyses the performance of the developed approach. Finally, conclusion and future work are provided in Section 3.5.

3.2 Approach Description

The scheme of the stereo vision-based autonomous navigation approach used in this chapter is shown in Fig. 3.2. Based on the images captured by the UAV onboard stereo camera, the UAV will locate itself through the visual localisation technology. Then, compared with the planned trajectory, the next target position will be updated. Afterwards, the proper control signal will be generated to control

the UAV to move to the target position. The whole process will be repeated until the UAV finishes the whole trajectory.

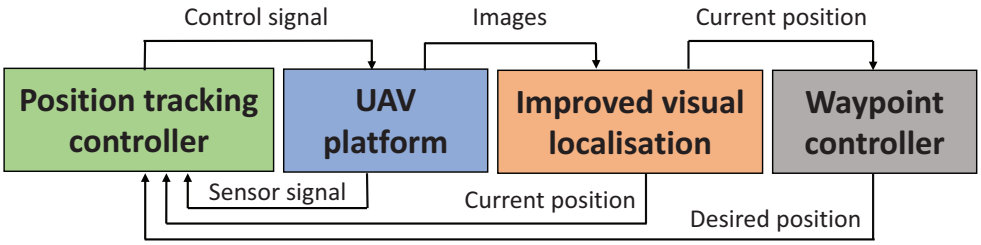


Figure 3.2: Scheme of the vision-based autonomous navigation approach.

3.2.1 Improved VSLAM Approach

To navigate successfully in a GPS-denied and low-illumination environment, the UAV must have self-localisation capability. However, the ORB-SLAM3 relies heavily on matching ORB feature point pairs. In the low-light environment, the number of stable ORB feature points drops significantly. Thus, the system fails to obtain enough input information, and posture cannot be calculated and corrected. Within the default settings in [19], the system tracks 500 ORB feature points to locate the position and map the surrounding environment. When ORB feature points are less than 500 in all frames, the pose estimation process cannot be performed, so the initialisation and tracking fail [155].

Image pre-processing technologies have been widely used to improve the performance of computer vision tasks, such as the image sharpening technique for facial emotion recognition [184] and speckle reduction for synthetic aperture radar (SAR) image recognition [185]. In the oil and gas pressure vessel inspection scenario, the light is insufficient for the ORB-SLAM3, and the image contrast enhancement technology improves the visual quality of dimmed images. Thus, the

image contrast enhancement method is adopted before the ORB feature point extraction process in the tracking thread. The image contrast enhancement technology makes the image contain more prominent textures, thereby increasing the number of stable ORB feature points. Eventually, the robustness and stability of the ORB-SLAM3 are improved.

3.2.1.1 Adaptive Image Enhancement

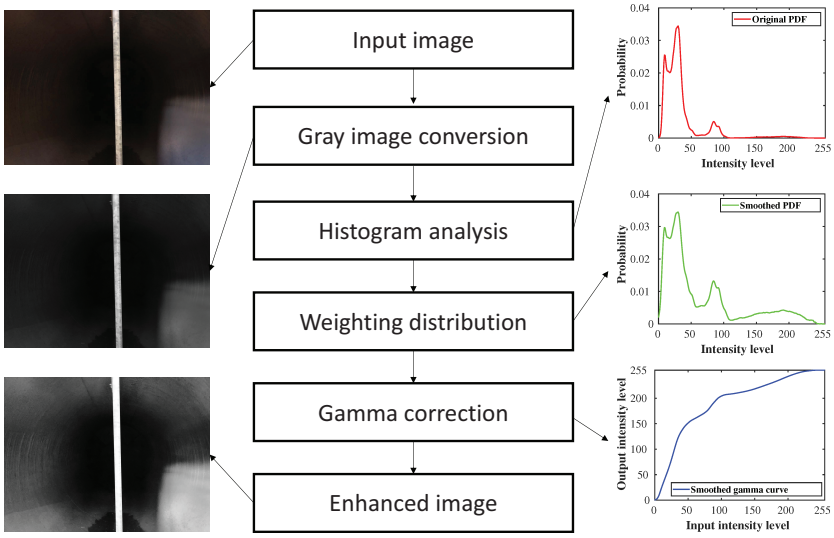


Figure 3.3: Diagram of the adaptive gamma correction with weighting distribution (adapted from [20]).

In the image contrast enhancement domain, considering the easy adjustment and efficient implementation capabilities, gamma correction [20] has been widely utilised. It enhances image contrast by directly modifying pixel values based on regulation. An Adaptive Gamma Correction algorithm with Weighting Distribution (AGCWD) [186] is adopted and improved to process frames before the ORB feature point extraction procedure. Fig. 3.3 shows the steps for enhancing the image at a high level. As ORB feature points are extracted from the grey image, the colour image needs to be transformed into the grey image first.

In the dimmed grey image, most pixels lie at the low-intensity level. After the weighting distribution and gamma correction, more pixels will be distributed in the high-intensity region, thereby improving image contrast.

Specifically, gamma correction techniques use the parameter γ to adjust the luminance of the image. The transform-based gamma correction is represented by

$$F(l) = l_{max}(\frac{l}{l_{max}})^{\gamma} \tag{3.1}$$

where l and l_{max} represent the intensity of each pixel and the maximum intensity in the input image, respectively. In Eq. (3.1), the highest intensity in the output image is restricted by the maximum intensity of the input image. To deploy it in a low-illumination environment, the l_{max} is set to 255, which is the highest intensity value in the grey image. In this work, the modified gamma correction can be defined as

$$F(l) = 255(\frac{l}{255})^{\gamma} \tag{3.2}$$

The Probability Density Function (PDF) can be approximated by

$$PDF(l) = \frac{n_l}{N} \tag{3.3}$$

where n_l represents the number of pixels that have intensity l. N denotes the total number of pixels in the image. Based on the PDF, the Cumulative Distribution Function (CDF) can be formulated as

$$CDF(l) = \sum_{k=0}^{l} PDF(k) \tag{3.4}$$

The adaptive gamma correction used in this work is written as

$$F(l) = 255(\frac{l}{255})^{(1-CDF(l))} \tag{3.5}$$

Additionally, the weighting distribution function is utilised to modify the statistical histogram with fewer adverse effects. The weighting distribution function is expressed as

$$PDF_{wd}(l) = PDF_{max}(\frac{PDF(l) - PDF_{min}}{PDF_{max} - PDF_{min}})^{\alpha} \tag{3.6}$$

where α indicates the adjusted parameter, and it is 0.5 in this work, which is the median value of the original limits [187]. PDF_{max} and PDF_{min} denote the maximum and minimum PDF of the statistical histogram, respectively. So, the modified CDF is defined as

$$CDF_{wd}(l) = \frac{\sum_{l=0}^{l} PDF_{wd}(l)}{\sum PDF_{wd}} \tag{3.7}$$

where the sum of PDF_{wd} is represented as follows

$$\sum PDF_{wd} = \sum_{l=0}^{l_{max}} PDF_{wd}(l) \tag{3.8}$$

Finally, the γ parameter can be calculated by

$$\gamma = 1 - CDF_{wd}(l) \tag{3.9}$$

The temporal technique [186] is applied to reduce the computational complexity. The information content contained in each frame is represented by the following entropy function:

$$H_{ic} = -\sum_{l=0}^{l_{max}} PDF(l) log(PDF(l)) \tag{3.10}$$

The differences between the information contents contained by two frames can be defined as

$$T_h = |H_{iccur} - H_{icpre}| \tag{3.11}$$

The first frame is stored and utilised to calculate the γ transformation curve. According to [186], the T_h is set to 0.05. When the T_h exceeds 0.05, the stored frame is updated with the current frame. At the same time, the γ transformation curve is modified. Otherwise, the existing γ transformation curve is applied directly to transform the intensity level of the incoming frame.

3.2.2 Position Tracking Controller

After the UAV is located by the improved ORB-SLAM3, the waypoint controller compares the current position with the planned inspection path to compute the desired position. Afterwards, a position tracking controller is needed to control the movement of the UAV.

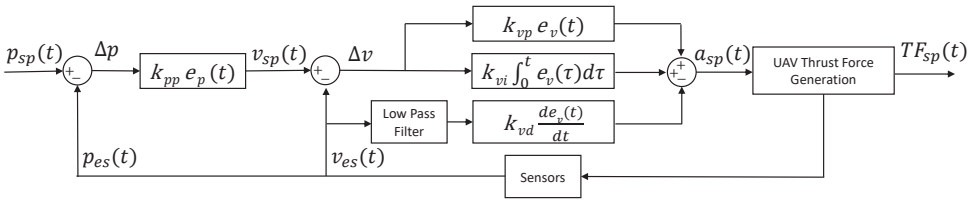


Figure 3.4: Scheme for position tracking controller.

In this section, a vision hybrid position tracking controller is developed from the PX4, the leading open-source autopilot stack for the UAV [188], to realise an accurate position control mechanism for the UAV. The position tracking controller is demonstrated in Fig. 3.4. Herein, based on the PID control law [189], a P loop for position error ($e_p(t)$) and a PID loop for velocity error ($e_v(t)$) are cascaded as the position tracking controller. The PID velocity control loop contains three

parameters, taken as constant k_{vp}, k_{vi} and k_{vd} which are responsible for adjusting the proportional, integral and differential units, respectively. Its continuous form can be given as

$$a_{sp}(t) = k_{vp}e_v(t) + k_{vi}\int_0^t e_v(\tau)d\tau + k_{vd}\frac{de_v(t)}{dt}e_v(t) \tag{3.12}$$

The UAV is controlled by a digital controller operating in a sampled-data feedback loop. Define

$$z_i(k) = z_i(k-1) + z(k) \tag{3.13}$$

$$z_d(k) = z(k) - z(k-1) \tag{3.14}$$

where $z(k)$ represents an error variable. $z_i(k)$ is the integrator state, and $z_d(k)$ indicates the differentiator state. Define the discrete-time PID velocity controller as

$$a_{sp}(k) = k_{vp}(k)z(k) + k_{vi}(k)z_i(k) + k_{vd}(k)z_d(k) \tag{3.15}$$

Eq. (3.15) can be rewritten as

$$a_{sp}(k) = \theta_v(k)\phi_v(k) \tag{3.16}$$

where

$$\theta_v(k) \triangleq \begin{bmatrix} k_{vp}(k) & k_{vi}(k) & k_{vd}(k) \end{bmatrix} \tag{3.17}$$

$$\phi_v(k) \triangleq \begin{bmatrix} z(k) \\ z_i(k) \\ z_d(k) \end{bmatrix} \tag{3.18}$$

In the same manner, the P position control can be represented as

$$v_{sp}(k) = k_{pp}(k) z_p(k) \tag{3.19}$$

where k_{pp} represents proportional parameter, and $z_p(k)$ is the position error.

3.3 Experimental Environment

Due to the high risk of property damage and battery constraints, extensive physical UAV flying tests are costly and time-consuming. As an alternative solution, simulation allows testing and validating the developed algorithm in "realistic" scenarios, which can avoid the potential risk in real flights. In the UAV research domain, the Robot Operating System (ROS) [190] is the most popular and convenient middleware suite. Moreover, it comes with the Gazebo simulator [191] that contains a physics engine to imitate the actual motions of the UAV in the customised environment. Thus, to closely mirror the performance of the quadrotor with the PX4 autopilot, the simulation environment developed in this chapter is based on the ROS-Gazebo-PX4 toolchain. Besides parameters introduced in the following part, default parameters are utilised. Gazebo 7.16.1 contains a simulated pressure vessel and a simulated UAV model. The UAV model is customised from the PX4 official iris model to add a stereo camera and spotlight. The stereo camera is fixed on the front centre of the UAV. The resolution of the camera is 752×480, with a baseline of 5cm. The focal lengths on the x and y axes are 376 and 376, respectively, and the aperture centres of the camera are 376 and 240, respectively. The framerate is set to 20Hz. The PX4 v1.8.0 firmware is used for dynamic simulation. The spotlight is fixed on the bottom centre. The PX4 com-

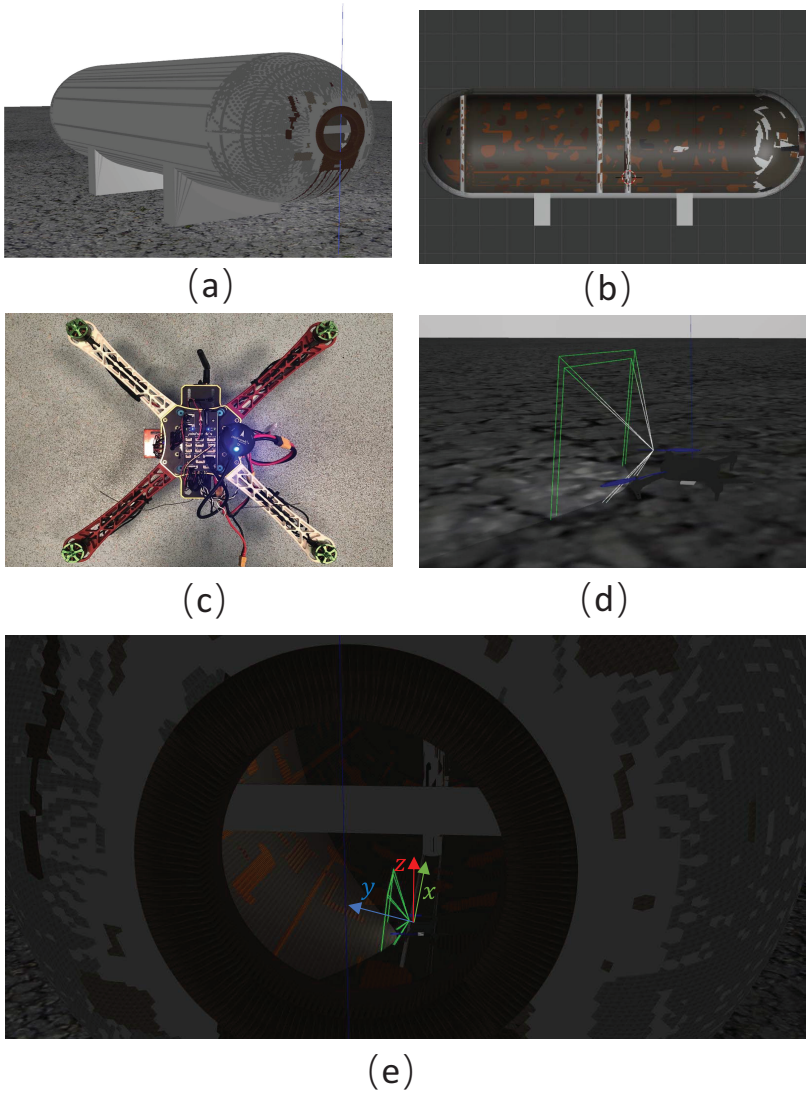


Figure 3.5: Physical and simulation components. (a) The simulated pressure vessel; (b) the section view of the simulated pressure vessel; (c) the physical UAV; (d) the simulated UAV equipped with a stereo camera and a spotlight source; (e) the developed simulation environment.

municates with the Gazebo to receive sensor data from the simulated world and send the motor commands back. Meanwhile, all the components are coordinated through ROS Kinetic.

Specifically, based on the pressure vessel shown in Fig. 3.1, a pressure vessel model is constructed (as shown in Fig. 3.5(a)). Its dimensions are 7m x 2.5m x 2.5m. As demonstrated in Fig. 3.5(b), it contains shells, several horizontal bars and vertical pipes. To simulate the customised quadrotor shown in Fig. 3.5(c), a simulated UAV is developed from the PX4 flight control stack, and it is equipped with a stereo camera and a spotlight. Due to the horizontal bar at the entrance of the pressure vessel, the UAV is placed inside the pressure vessel. The overall layout of the simulation environment is illustrated in Fig. 3.5(e). Within this scenario, a world coordinate system is established. The x-axis is supposed to be the depth direction of the pressure vessel, the y-axis is parallel to the width direction, and the z-axis denotes the altitude.

3.4 Results

3.4.1 Comparison of Feature Point Extraction and Matching

In this experiment, the light intensity of the spotlight keeps the same during the whole inspection procedure, and the first frame captured by the stereo camera within the developed simulation environment is selected. The results of image contrast enhancement are illustrated in Fig. 3.6. The contrast of the original images is low, and the limited textures can be observed from the image. Within the enhanced images, more visual textures are revealed.

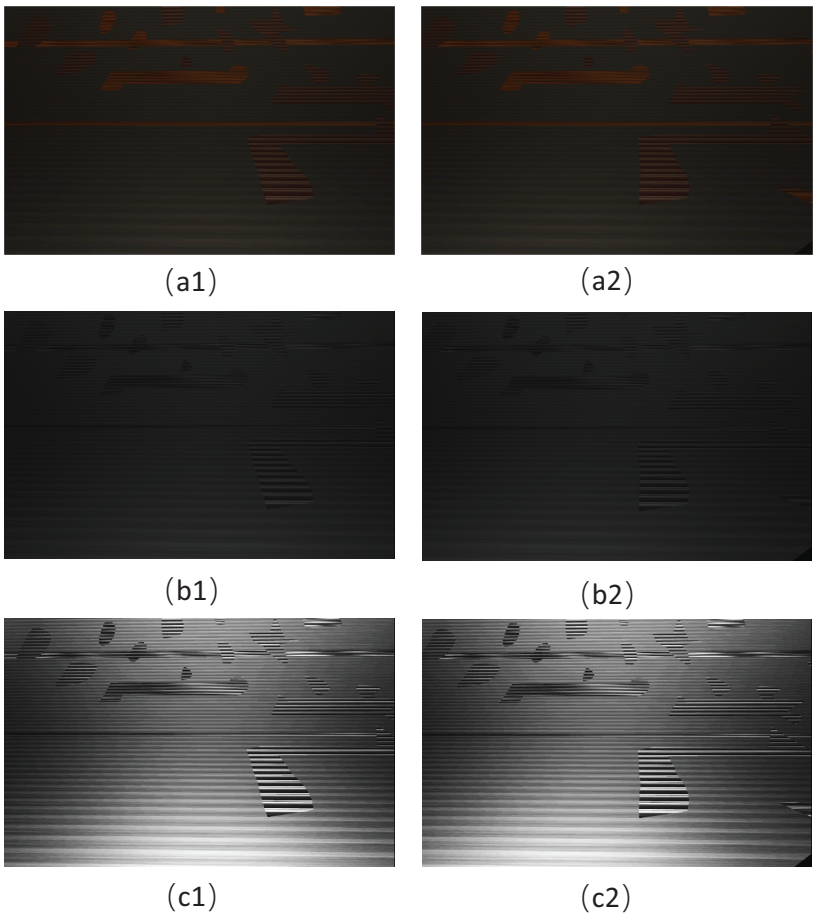


Figure 3.6: Image contrast enhancement. (a1) and (a2) are a pair of images captured by the on-board stereo camera; (b1) and (b2) are the original gray images transformed from (a1) and (a2); (c1) and (c2) are the enhanced gray images.

The ORB feature point detection and matching processes are compared using the ORB-SLAM3 with the improved ORB-SLAM3, and all the parameters are adopted from [19] for a fair comparison. Matches after selection through the Sum of Absolute Difference (SAD) [192] are supposed to be good matches. The visual results are shown in Fig. 3.7, while the statistical results are demonstrated in Table 3.1. The ORB-SLAM3 fails to extract 500 ORB feature points from the first frame to initialise the system. Compared to the ORB-SLAM3, more than 5 times ORB feature points can be extracted by the improved ORB-SLAM3.

Meanwhile, the number of good matching points realised by the improved ORB-SLAM3 increases by 600%.

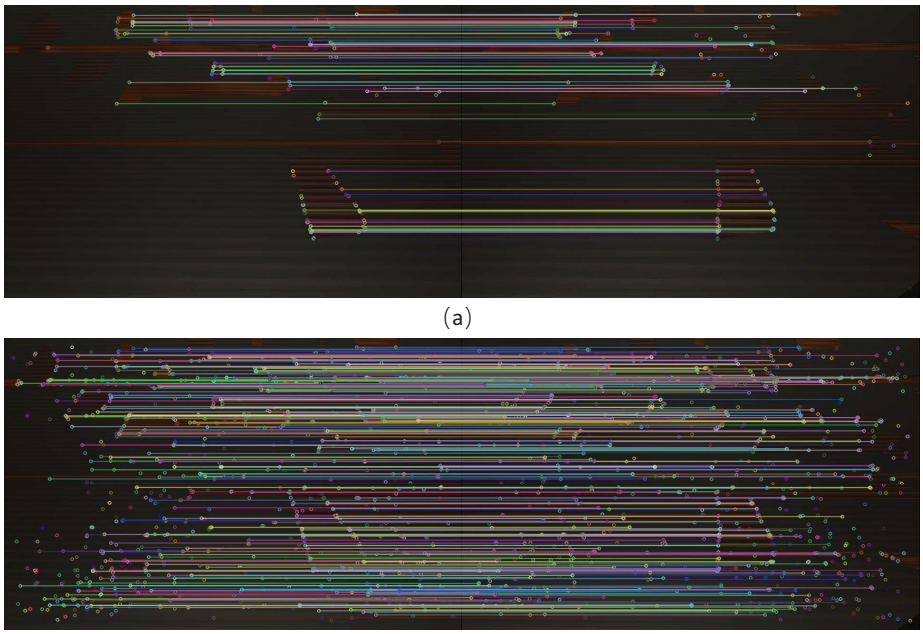


Figure 3.7: Feature points extraction and matching. (a) Feature points extraction and matching based on original images; (b) feature points extraction and matching based on enhanced images.

Table 3.1: Comparison of feature point extraction and matching in the first frame

Method	ORB feature points		Good matches	System initialisation
	Left image	*Right image*		
ORB-SLAM3	200	208	103	Fail
Improved ORB-SLAM3	1051	1092	746	Success

The results illustrate that with the image contrast enhancement method, enough ORB feature points can be extracted to initialise the system. What is more, more effective matching points for the subsequent processes such as tracking, mapping and loop detection are achieved to improve the stability and

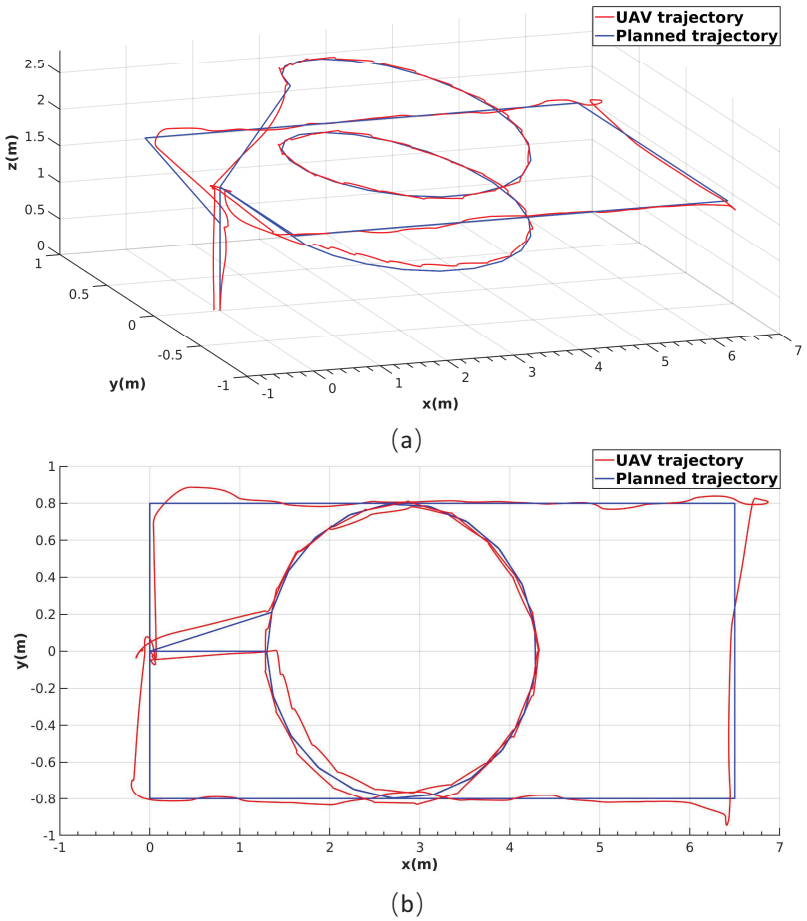


Figure 3.8: Trajectory following results. (a) The overview of the UAV 3D trajectory; (b) the top view of the UAV trajectory.

robustness of the ORB-SLAM3. More details are presented in the following section.

3.4.2 Trajectory Tracking Performance Evaluation

The task of the aerial vehicle is to follow a pre-defined 3D trajectory to record videos of the pressure vessel. Fig. 3.8 demonstrates the 3D trajectory of the UAV and the pre-defined inspection plan. Parts of the inspection process are visualised in Fig. 3.9. The square trajectory is designed to inspect the shells of

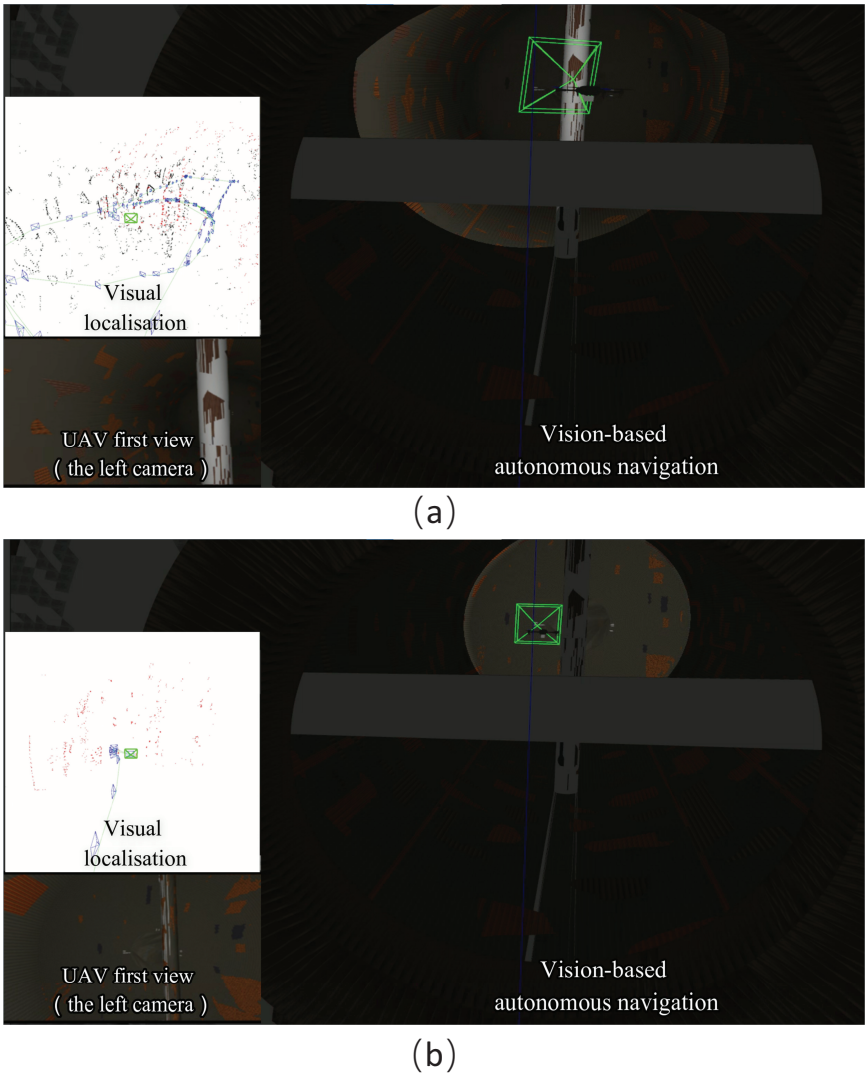


Figure 3.9: Demonstration of the visual inspection process. (a) Inspection of the vessel shell, (b) inspection of pipeline.

the pressure vessel. What is more, the pipelines are inspected by tracking the helical path. The results indicate that keeping a constant distance to the shell in directions y and z is not an issue for the whole system. The parameters of the position tracking controller are adjusted for translation and shared with rotation, which negatively affects the tracking accuracy, especially in direction x due to the UAV heading in the y-axis positive direction at the beginning. The overall

results validate that the UAV can locate and navigate itself stably in a pressure vessel for the visual inspection application. A supplementary video[1] is provided to show more detailed results.

3.5 Summary

In this chapter, the simulation has been carried out to verify the feasibility of deploying the UAV with VSLAM to achieve autonomous visual inspection of the pressure vessel. To address the issue of ORB-SLAM3 failure due to insufficient feature points extracted from the low illumination environment, the image contrast enhancement technique using adaptive gamma correction with weighting distribution was adopted. Then, a P-PID controller was deployed for position tracking. The autonomous navigation system of the UAV with VSLAM was verified in a deeply customised ROS-PX4-Gazebo simulation environment. The results showed that the improved ORB-SLAM3 could achieve more ORB feature points and matching points than the ORB-SLAM3 in the low-lighting environment, which addressed the challenge of feature extraction in the low-illumination environment. Moreover, with the improved VSLAM system, the UAV could track the planned trajectory stably to take images of the inner surfaces and structures of the pressure vessel. Thus, the feasibility of deploying the UAV with VSLAM to achieve autonomous visual inspection was proven for the first time.

However, environmental noises also have a significant impact on the stability of the UAV system, and they cannot be fully tested in simulation environments. Specifically, in the simulation environment, the texture of the vessel cannot be fully replicated, and the light distribution is uniform. Additionally, the images

[1] https://youtu.be/p1zKOHhxKfI

captured by the UAV are clear, which means no blurred images are taken, and the influence of light reflections is not evaluated. To address these issues, the VSLAM algorithm used in this chapter will be further improved in Chapters 5 and 6 and validated through real-world scenarios. In addition, the corrosion detection module that is ignored in this chapter will be introduced in Chapter 4.

Chapter 4

AMCD: Accurate UAV onboard Metallic Corrosion Detector

4.1 Introduction

To develop UAV-based autonomous visual inspection systems, creating high-accuracy corrosion detectors based on advanced computer vision techniques will be the primary concern [37]. Moreover, the requirement of real-time detection at a speed of at least 20 FPS [39] is increasing in practical UAV visual inspection due to the limited endurance time [40]. Hence, this chapter addresses the ignored corrosion detection module in the depicted scheme in Chapter 3.

Over the last couple of years, a variety of algorithms for corrosion detection have been proposed. Among them, texture and colour analysis by a filter-based approach or a statistical model have gained great interest. The colour wavelet filter bank is one of the most popular techniques for detecting corrosion through filtering texture and colour features. However, when the optimal features are not

identified, the detection accuracy will decrease heavily [193].

Recently, CNNs have been proven to surpass humans in the ImageNet classification task [194]. According to the investigation, an increasing number of researchers have adopted CNNs to assist their research, such as morbidity identification [195], Synthetic Aperture Radar (SAR) image classification [196], vehicle detection [197], wind turbine blade structural state evaluation [198] and bridge crack detection [199]. These results suggest that CNNs could also be utilised to achieve high-accuracy corrosion detection. Unlike previous approaches, CNNs do not need prior-designed low-level features, which are not robust enough for computer vision tasks. For CNN-based computer vision tasks, features are determined inherently by CNNs and the training dataset. The results in [200] indicated that CNNs are robust enough to detect or classify objects with different scales, orientations and illuminations. Thus, an opportunity has emerged for CNN-based detectors to achieve much more accurate corrosion detection than traditional approaches.

Although several existing works have shown accurate corrosion detection with CNNs, the high demand for desktop-level CPUs still poses a challenge in adopting these methods onto the UAV onboard computer (Nvidia Jetson TX2). As discussed in Sections 2.2.3 and 2.2.4, these studies focus on image processing without consideration of the limitations of the UAV platform. The high-end GPUs are necessary to achieve real-time corrosion detection. However, the Nvidia Jetson TX2 only has a performance capacity of 0.67 TFLOPS. Consequently, there are still challenges in deploying existing deep learning-based corrosion detectors on UAV onboard platforms to achieve real-time corrosion identification. Since there is no standard baseline for detection accuracy in the literature, the mAP obtained

by RetinaNet [124] will be considered as the baseline due to its high mAP in the literature [126] [127]. In this chapter, an accurate deep learning-based corrosion detector is proposed for real-time implementation on UAV onboard platforms. The main contributions of this chapter are summarised as follows:

The issue of high demand for computing resources when deploying deep learning-based corrosion detection on the UAV onboard computer caused by extensive usage of traditional convolution layers is addressed through lightweight model design. It is achieved through lightweight convolution utilising DSconv, innovative feature extraction and fusion techniques leveraging the CBAM and the proposed improved SPP, refined detection strategies incorporating three-scale detections, and an optimised learning approach using the focal loss.

The rest of this chapter is organised as follows. Details of the proposed detector are given in Section 4.2. Section 4.3 shows the experimental environment and results. Section 4.4 concludes the whole chapter.

4.2 Proposed Efficient Corrosion Detection Algorithm

4.2.1 Framework of Corrosion Detection

Based on the introduction above, there are lots of excellent object detectors emerging. The Yolov3-tiny is an object detector, which has been proven to be fast on embedded platforms [127]. Fig. 4.1 shows the network structure of the Yolov3-tiny. There are seven convolution layers and six MaxPool layers for extracting image features. Two-scale detection is utilised to detect different-sized targets. The detection process of the Yolov3-tiny is described as follows:

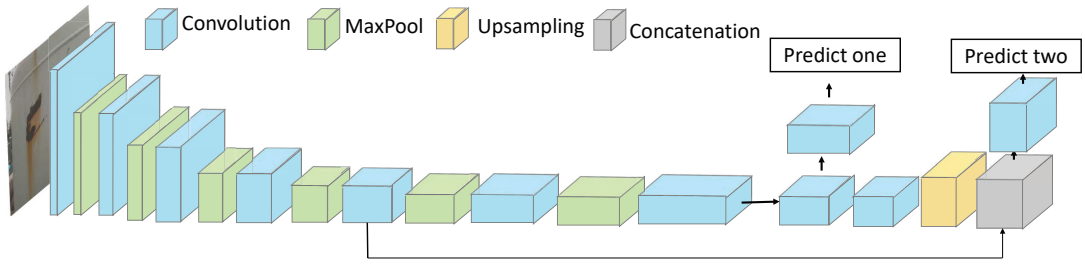


Figure 4.1: Framework of the Yolov3-tiny

Step 1 Load the input image and resize the image to a size of 416*416

step 2 Extract features with convolutional and MaxPool layers

Step 3 Produce feature maps of size 13*13 on a small scale

Step 4 Upsample small-scale feature maps to size 26*26 and connect them to the same size feature maps generated by the feature extraction network

Step 4 Produce feature maps of size 26*26 on a large scale.

Step 5 Divide the input image into 13*13 and 26*26 grids for two-scale object detection. Based on the predefined anchors, the grid will be responsible for predicting the object when the centre of the object lies in the grid.

Step 6 Output the two-scale prediction results

Step 7 Fuse different scale prediction results and acquire accurate bounding boxes

Since the simple and shallow network is designed as the backbone, the detection accuracy of the Yolov3-tiny is not high enough [201]. According to the initial test, the mAP for the corrosion detection is 79.02% (as shown in Table 4.2). Therefore, the Yolov3-tiny cannot be utilised directly due to the fact that its accuracy cannot

meet the requirement, which is the mAP of 83.5% achieved by the baseline model. Moreover, the Yolov3-tiny deploys many convolution layers with 512 and 1024 convolution filters, which leads to a large number of parameters contained by the network. Finally, the model size is 34.7 MB, and it requires around 70% GPU resources on the Nvidia Jetson TX2 [202].

To address these problems in the Yolov3-tiny for corrosion detection, this chapter proposes a novel metallic corrosion detector. Inspired by the Yolov3-tiny (as shown in Fig. 4.1), the overall schematic architecture is presented in Fig. 4.2. The backbone is responsible for extracting features from images, and the detection part will output the position and category of the corrosion. A brand-new lightweight network has been designed as the AMCD focuses on achieving accurate corrosion detection on embedded platforms. What is more, the DSConv is adopted to reduce parameters. To enhance the feature extraction capability of the shallow network, the CBAM [21], three-scale prediction, improved SPP and focal loss [124] are also utilised. Finally, considering the limited computing resources and the target to detect corrosion at a speed of 20 FPS, the designed backbone contains 1 traditional convolution layer, 7 DSConv layers, 1 SPP layer and 4 CBAM modules. Details of the AMCD will be explained in the following parts.

4.2.1.1 Depthwise Separable Convolution

The DSconv is adopted to reduce model parameters to fit embedded computing platforms. The detailed introduction to DSconv can be found in Section 2.4.1.

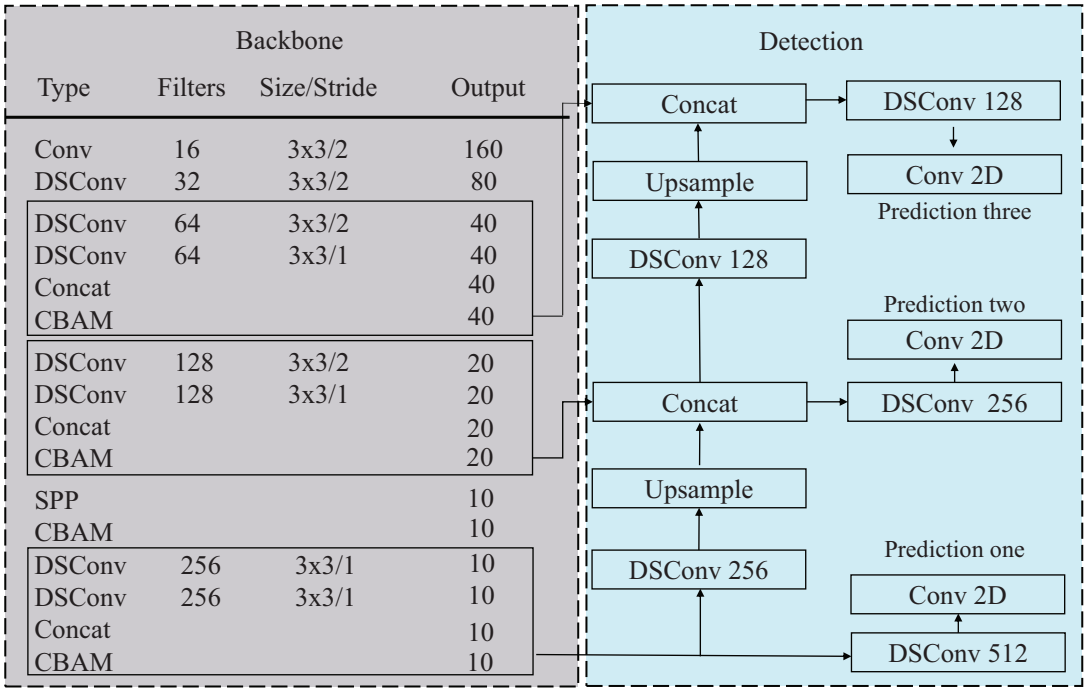


Figure 4.2: Structure of the AMCD

4.2.2 Attention Mechanism

Inspired by human visual attention mechanisms, CNNs can employ attention mechanisms to select optimal information from the training dataset. The attention module selects the most representative area in the image and allows the network to focus on it. Thus, more critical features can be extracted, and the detection accuracy will be improved. The attention mechanism has proven its effectiveness in many tasks, such as river detection [203], outdoor illumination estimation [204] and SAR image recognition [205].

The CBAM [21] outputs refined feature maps by channel and spatial attention sequentially. The overview diagram of the CBAM is shown in Fig. 4.3. In general, the channel attention module focuses on figuring out optimal feature maps between different channels of feature maps. The spatial attention module aims

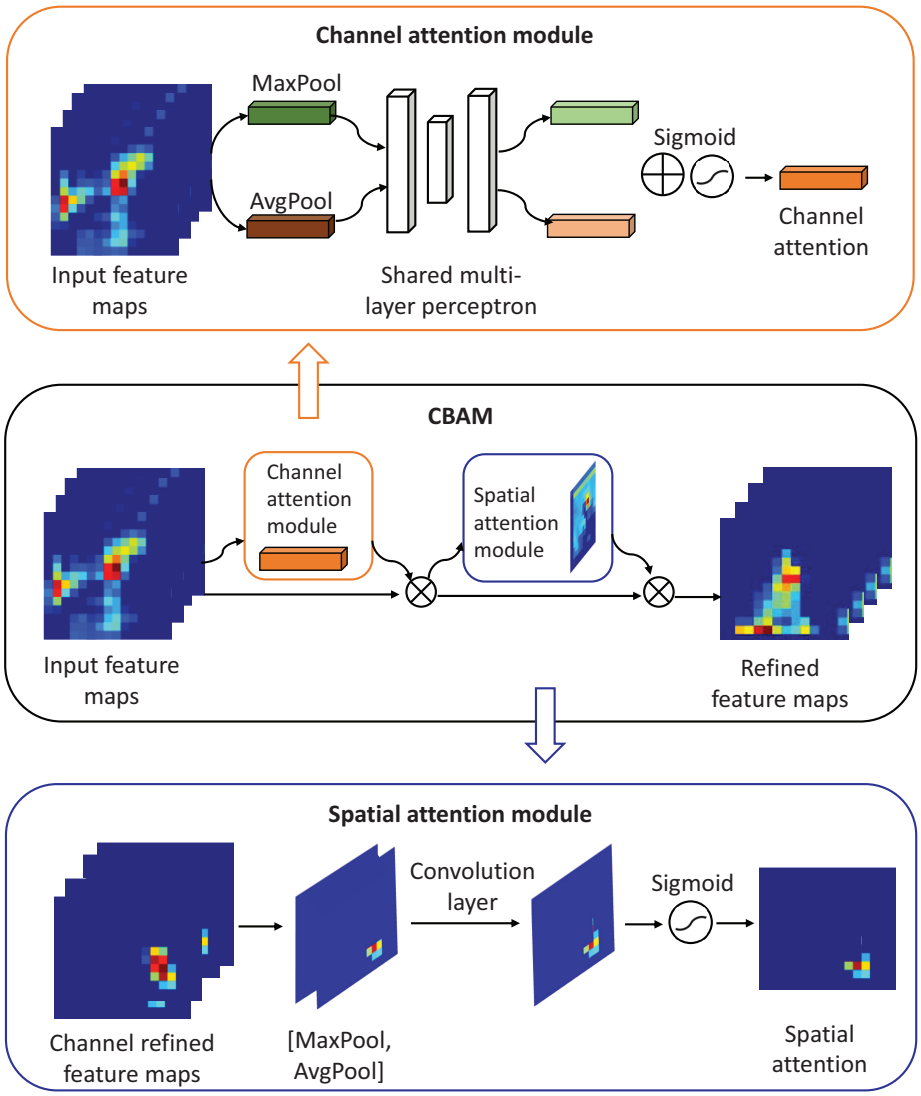


Figure 4.3: Diagram of the CBAM (adapted from [21])

to output a spatial attention map based on local information. The MaxPool and AvgPool operations are utilised to construct feature map statistics. The MaxPool could return the significant features of the target. At the same time, the AvgPool provides global statistics on feature maps. With the usage of MaxPool and AvgPool operations, the representation of features extracted by CNNs is improved. Channel attention focuses on global information, whereas spatial attention is em-

ployed locally. Therefore, the CBAM can extract comprehensive salient features to improve the performance of corrosion detection.

4.2.3 Improved Spatial Pyramid Pooling

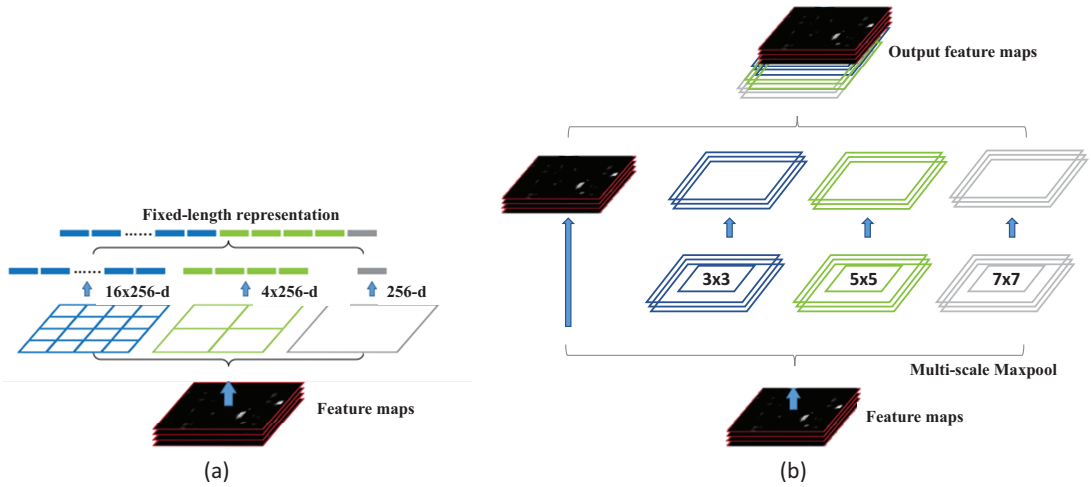


Figure 4.4: Structures of SPP. (a) The traditional SPP (adapted from [22]), (b) the improved SPP

The architectures of the SPP are shown in Fig. 4.4. Different from the traditional SPP proposed by [22], the improved SPP does not resize feature maps into feature vectors. Instead, the improved SPP outputs feature maps. Based on the size of input feature maps, MaxPool layers with kernel sizes of 3×3, 5×5 and 7×7 are utilised to pool feature maps. The stride of each pooling layer is 1, and padding is adopted to make sure the size of generated feature maps is the same as that of input feature maps. After the concatenation, there are 1024 feature maps generated by the improved SPP, which extracts and fuses local region features.

4.2.4 Loss Function

The Yolov3-tiny uses anchors to generate candidate object locations from the whole image. The number of potential bounding boxes containing objects is much less than those only containing background. What is more, negative samples contribute no useful learning signal and cause biased learning. Finally, it will lead to a degenerate detector, which cannot detect the corrosion correctly. To overcome this limitation, the focal loss is introduced into the AMCD, which gives a high loss value to an object. This makes the detector concentrate on object areas and is sensitive to the target. The formula for the focal loss is:

$$F_{fl} = \alpha_{loss} \times (1 - p_{cb})^{\lambda_{loss}} \tag{4.1}$$

The α_{loss} is the hyperparameter which down-weights the loss contributed by background. p_{cb} indicates the confidence of whether the candidate bounding box contains the object. λ_{loss} represents the exponential scaling factor which down-weights the loss generated by easy examples and makes the CNNs focus on difficult examples. In the AMCD, the α_{loss} and λ_{loss} are empirically set to 0.5 and 2, respectively.

According to Eq.(4.1), the loss function of the AMCD can be defined as:

$$
\begin{aligned}
Loss = & \\
& \sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}[(x_i-\widehat{x}_i)^2+(y_i-\widehat{y}_i)^2] \\
& +\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}(2-w_i\times h_i)[(w_i-\widehat{w}_i)^2+(h_i-\widehat{h}_i)^2] \\
& +F_{fl}\sum_{i=0}^{S^2}\sum_{j=0}^{A} 1_{ij}^{obj}[\widehat{C}_i log(C_i)+(1-\widehat{C}_i)log(1-C_i)] \\
& +F_{fl}\sum_{i=0}^{S^2}\sum_{j=0}^{A} 1_{ij}^{noobj}[\widehat{C}_i log(C_i)+(1-\widehat{C}_i)log(1-C_i)] \\
& +\sum_{i=0}^{S^2} 1_{ij}^{obj}\sum_{c\in classes}[\widehat{p}_i(c)log(p_i(c))+(1-\widehat{p}_i(c))log(1-p_i(c))]
\end{aligned}
\tag{4.2}
$$

where S^2 denotes the number of grid cells. B is the number of bounding boxes predicted by each cell. 1_{ij}^{obj} indicates that the jth bounding box predicted in cell i contains an object. At the same time, 1_{ij}^{noobj} refers to the predicted bounding box containing only the background. x and y are the centre coordinates of the bounding box. w and h represent the dimensions of the bounding box. The variables with$\widehat{}$indicate they are predicted values. Otherwise, they are groundtruth. C denotes the confidence of whether the bounding box contains an object or just a pure background. The prediction of classes is represented by $pi(c)$. Notably, F_{fl}, which represents the focal loss, is adopted to address the class imbalance problem.

4.3 Experiments

4.3.1 Dataset

To construct a dataset to train and verify the AMCD, 5625 images were captured by a DJI Phantom 4. Images are taken from different facilities, such as pressure vessels and oil wells, at a distance of 1 m to 10 m under different angles and illumination conditions to ensure their diversity. Based on the visual appearance, the corrosion types are classified as bar corrosion, nubby corrosion, fastener corrosion and exfoliation. If the aspect ratio of a damaged area is less than 1:2, this region will be treated as nubby corrosion. Otherwise, the damaged region will be considered to be bar corrosion. Bolt and nut corrosion are treated as fastener corrosion, and exfoliation corrosion includes cracked coatings. To annotate captured images with different corrosion types, labelImg (https://github.com/HumanSignal/labelImg) is utilised to put bounding boxes on images by human experts. Each bounding box contains the upper-left corner position, width and height of the box. Therefore, the format of the bounding box is (x, y, w, h). There is a total of 27039 corrosion areas labelled in 5625 images. Several annotated images are shown in Fig. 4.5. Bounding boxes with different colours represent different kinds of corrosion.

To generate training and test sets, labelled images are randomly divided by the contained corrosion. The training and validation datasets contain 4500 images. Another 1125 images are utilised to test the proposed detector.

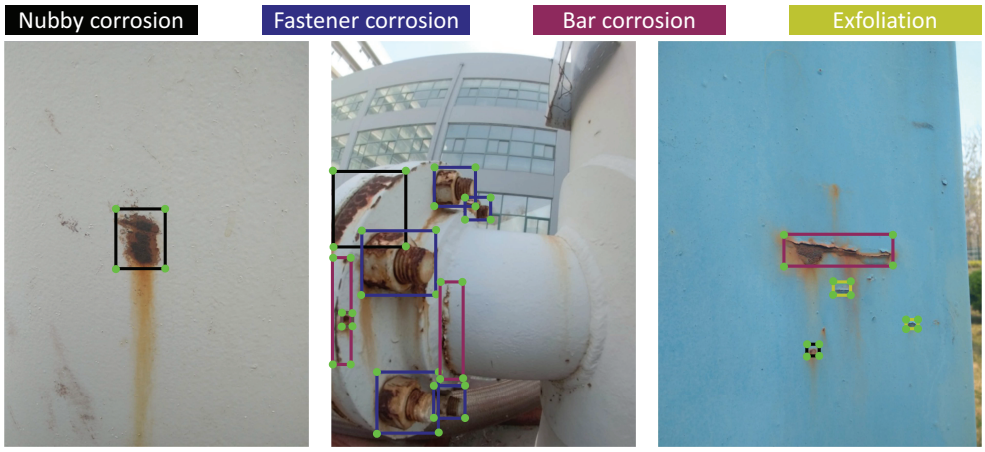


Figure 4.5: Labelled images

4.3.2 Experimental Setup

All training processes are conducted by applying Tensorflow 1.15 and CUDA 10.0 on a computer with an Intel Core i7-8750 CPU, 12 GB Random-access Memory (RAM) and 6 GB GDDR5 memory Nvidia GTX1060 GPU. To evaluate the performance of the AMCD on UAV onboard platform, testing processes are made on the Nvidia Jetson TX2. It is equipped with a hexa-core CPU and a Nvidia Pascal-family GPU with 256 CUDA cores. It loads with 8GB of memory and 59.7GB/s of memory bandwidth.

Transfer learning has the ability to transfer knowledge from a related task that has already been learned to a new domain. A lot of works have proven that transfer learning is an optimisation technique which saves training time and gets better test performance. Instead of using randomly initialised weights of CNNs, layers of the proposed model are initialised by the weight trained on PASCAL VOC2007 [206] and PASCAL VOC2012 [207] datasets. These are two widely used computer vision datasets and contain 20 different object classes for object recognition and detection. The VOC2007 dataset primarily contains 9,963

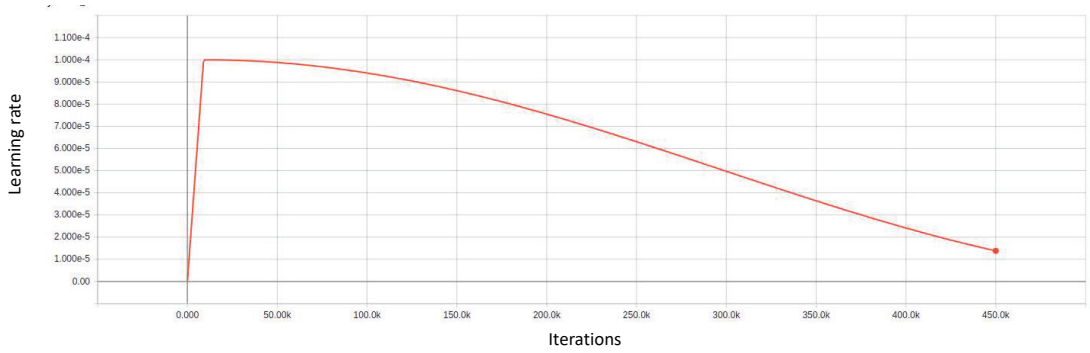


Figure 4.6: Learning rate curve during the training procedure

labelled images, while the VOC2012 dataset consists of 11,530 labelled images, providing a diverse set of images and annotations that can be used for training and evaluating machine learning models. The anchor sizes are clustered by K-means [208] as [(14,15), (18,21), (31,17), (25,26), (20,38), (36,35), (30,78), (63,48), (93,118)].

The proposed network is trained using the stochastic gradient descent algorithm [209], and the rand seed is set to 0. To make the training process stable and efficient, warmup [210] and cosine learning rate decay [211] are utilised. Fig. 4.6 depicts the variation of the learning rate during the training stage. The x axis represents iterations, and the learning rate is updated every iteration. In the warmup stage, the learning rate is increased linearly over the warm-up period until it hits the nominal rate, which reduces the primacy effect of the early training examples. Then, the learning rate begins to decay using the cosine function to train the model. The momentum parameter is 0.9995. The batch size is assigned as 2. The model is trained for 450000 iterations, and the model with the lowest loss values is chosen for testing. The loss curve during the training process can be seen in Fig. 4.7. A decreasing loss curve and small fluctuations in later iterations indicate the loss function is optimised and converged.

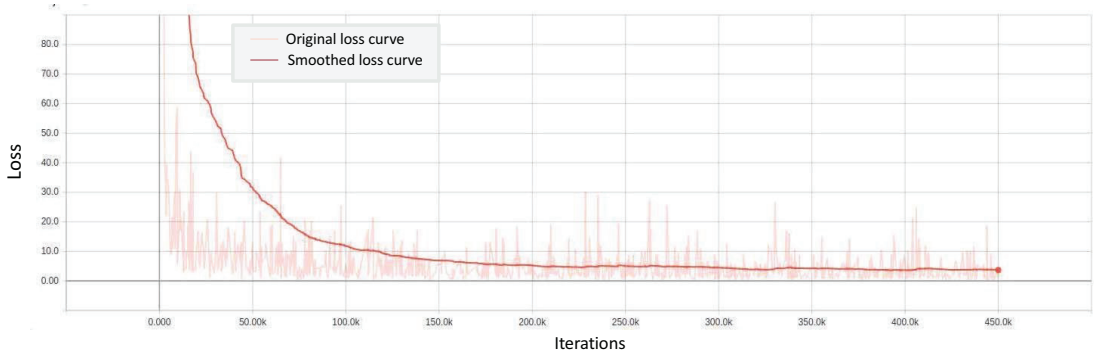


Figure 4.7: Loss decline curve of the AMCD

4.3.3 UAV platform-based Evaluation

4.3.3.1 Evaluation Metrics

Precision and recall concepts [212] are widely utilised to evaluate the performance of object detection approaches. Precision denotes the number of True Positive (TP) results divided by all positive detection results. The Recall is defined as the percentage of the TP in all correct detection results. The area under the precision and recall curve is called Average Precision (AP). The AP indicates the ability of the detector to locate objects and classify them into a single class. In general, the higher AP for a category of objects, the better performance of the detector in identifying them. The mAP represents the performance of the detector across all classes and can be defined by the average value of APs for all classes.

In reality, predicted results cannot match groundtruth perfectly. Thus, the Intersection-over-Union (IoU) metric is adopted to represent the overlap of the predicted bounding box with the groundtruth box. This allows predicted results can partially overlap with groundtruth. While the overlap area between suspicious corrosion and groundtruth exceeds the IoU threshold, the prediction result is classified as positive. Otherwise, the detection result is categorised as negative.

In this study, the IoU value is 0.5.

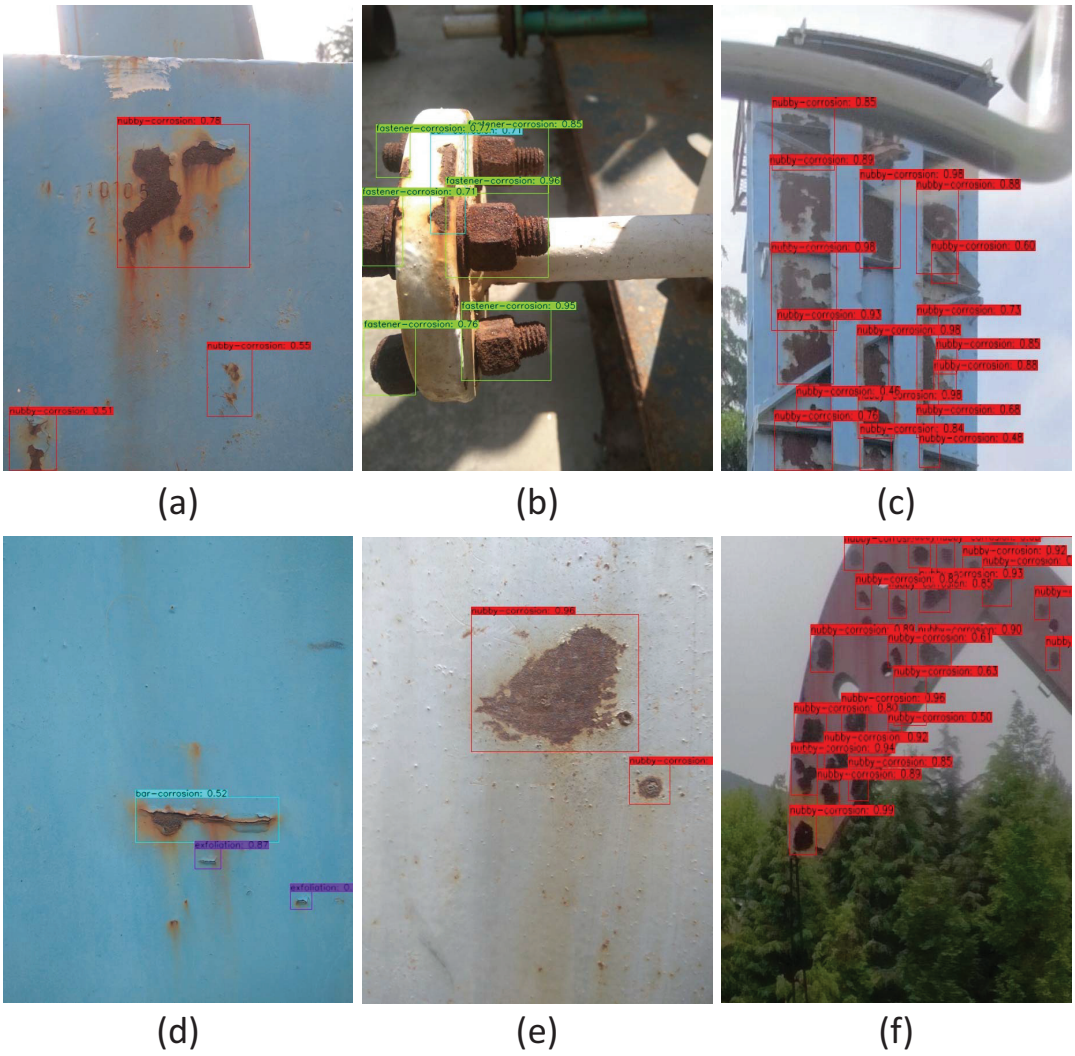


Figure 4.8: Some detection results

4.3.3.2 Performance of the AMCD

The trained model is used to identify different kinds of corrosion, and some recognition results are shown in Fig. 4.8. As images are taken under different angles and illumination conditions, their backgrounds are cluttered. Four kinds of corrosion can be detected correctly. What is more, when an image contains multiple

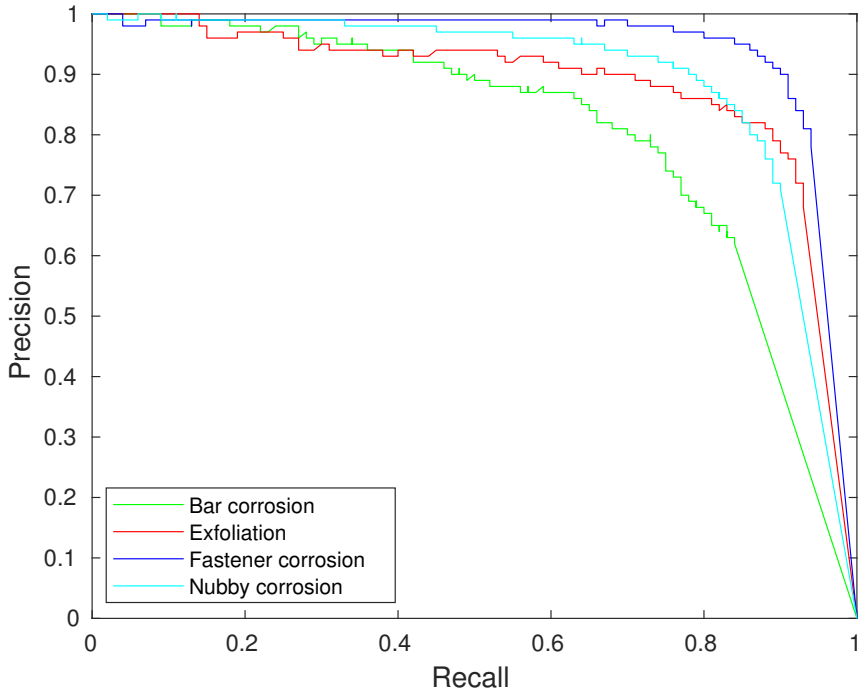


Figure 4.9: PR curves of four kinds of corrosions

types of corrosion, all of the corrosion can be identified. As shown in Fig. 4.8 (b), the fastener corrosion and bar corrosion are detected correctly, even though some shadows exist in the image. In Fig. 4.8 (f), small holes in the structure are very similar to the nubby corrosion. The AMCD can still locate corrosion areas precisely.

Precision-Recall (PR) curves and APs for four kinds of corrosion are demonstrated in Fig. 4.9 and Table 4.1, respectively. Based on the unique features of fastener corrosion, the detection results show high accuracy for this kind of corrosion. As its shape distinguishes bar corrosion from nubby corrosion, the features extracted from them are similar. The number of nubby corrosions in the training dataset is far greater than that of bar corrosion. Thus, bar corrosion can be easily misunderstood as nubby corrosion, leading to a relatively low detection

Table 4.1: Detection results of four kinds of corrosions

	Number of corrosions	Detection accuracy(%)	AP(%)
Nubby corrosion	3559	89.83	85.81
Bar corrosion	501	84.23	75.78
Exfoliation	290	93.45	86.36
Fastener corrosion	985	93.81	91.88

accuracy for bar corrosion. Some misdetection results are shown in Fig. 4.10. In addition, the AMCD still has limitations in detecting small corrosions. This is due to the lack of sufficient features extracted by the AMCD that can be used to distinguish the corrosion and its type. Besides its unique features, the exfoliation contains parts of the features of nubby corrosion or bar corrosion. Because the samples of the exfoliation are the fewest, its detection accuracy is between that of bar corrosion and nubby corrosion.

Figure 4.10: Demonstration of misdetected corrosions. (a) Some bar corrosions are mislabelled as nubby corrosions, and some corrosion are not detected. (b) Small nubby corrosions are not all detected.

Table 4.2: Comparison of corrosion detection performance

Detector	Backbone	Input dimension	mAP(%)
Yolov2-tiny	Yolov2-tiny	320	71.87
Yolov3-tiny	Yolov3-tiny	320	79.02
Yolov4-tiny	Yolov4-tiny	320	82.1
SSD	VGG16	300	81.2
RetinaNet	Resnet50	320	83.5
AMCD (ours)	AMCD	320	84.96

4.3.3.3 Comparison with Latest Detection Methods

In this section, the proposed network is compared with some state-of-the-art detectors. As the AMCD towards detecting corrosion with limited computing resources, the Yolov2-tiny [213], Yolov3-tiny, Yolov4-tiny [214], SSD [119] and RetinaNet [124] are selected for the comparison. To make a fair comparison, the input dimension of compared detectors is resized to a similar scale, and all compared detectors are trained with default parameters provided by the authors. As shown in Table 4.2, the RetinaNet achieves 83.5% mAP, which is the baseline in this chapter. The proposed detector achieves 84.96% mAP, which is the best among these algorithms, and it meets the requirement. When taking the model size and detection speed into consideration, details are shown in Fig. 4.12. Due to the shallow network architecture and DSConv being utilised in the AMCD, the detection speed reaches 20.18 FPS on average. It is almost 4 times faster than the RetinaNet, and it meets the requirement for real-time corrosion detection. With the adoption of the DSConv, the model size is reduced significantly, and it is only 6.1MB. This suggests that the AMCD could perform corrosion detection efficiently, which is essential for UAV onboard visual inspection applications.

Fig. 4.11 shows that the proposed method can achieve optimal corrosion detection results compared with other state-of-the-art algorithms. Other detectors

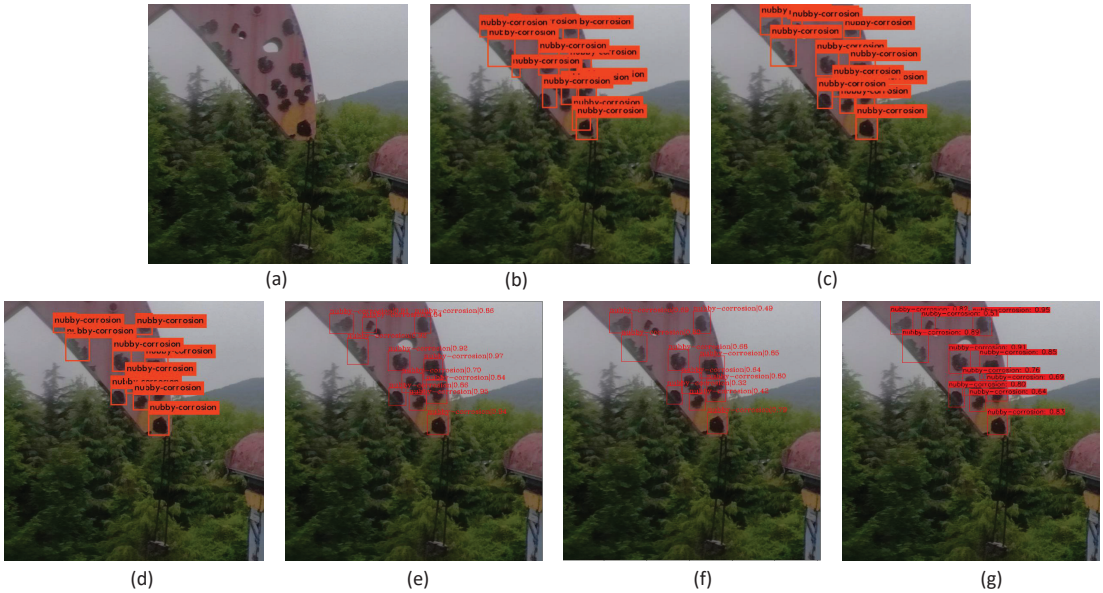


Figure 4.11: Original image (a) and detection results produced by the Yolov2-tiny (b), Yolov3-tiny (c), Yolov4-tiny (d), SSD (e), RetinaNet (f) and AMCD (g).

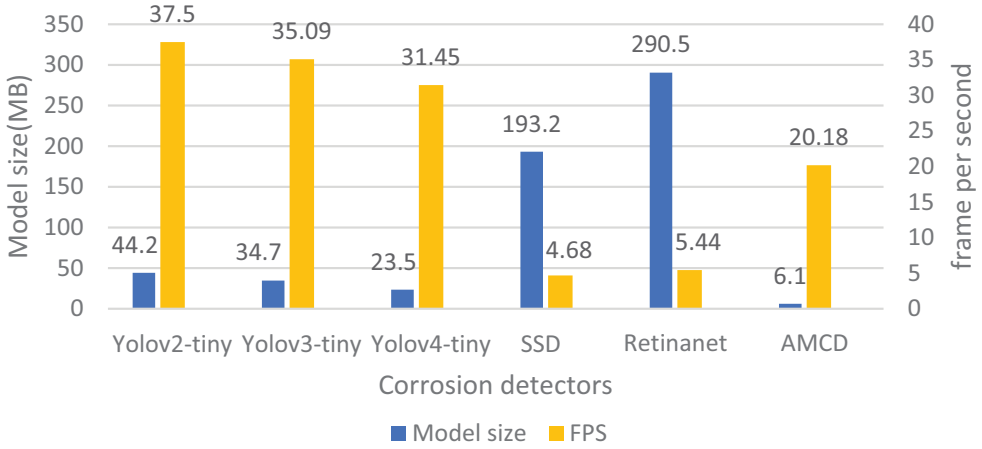


Figure 4.12: Comparison of the Yolov2-tiny, Yolov3-tiny, Yolov4-tiny, SSD, RetinaNet and AMCD in terms of model sizes and FPS

are limited by the tight layout and size of corrosion. The Yolov2-tiny is struggling to generate accurate bounding boxes for corrosion areas. The Yolov3-tiny, Yolov4-tiny and RetinaNet cannot detect small-size corrosion. The SSD and AMCD identify corrosion correctly in this image.

4.4 Summary

To the best of my knowledge, the first deep learning-based UAV onboard real-time corrosion detector was presented in this chapter. The computational challenges of implementing a deep learning-based corrosion detector on UAV onboard platforms due to the extensive usage of traditional convolution layers were addressed through the design of the lightweight corrosion detector. It was achieved through the lightweight convolution utilising DSconv, innovative feature extraction and fusion techniques leveraging the CBAM and the improved SPP, refined detection strategies incorporating three-scale detection, and an optimised learning approach using the focal loss.

Detailed experiment setup and execution processes were described in Section 4.3. 5625 images captured by the UAV were labelled with nubby corrosion, bar corrosion, fastener corrosion and exfoliation to train the proposed corrosion detector. The proposed approach achieves excellent performance in detecting and recognising different categories of corrosion. Experimental results proved that the proposed detector obtains satisfactory corrosion detection results, which is able to achieve 84.96% mAP for corrosion identification in the complex environment and get real-time performance (20.18 FPS) with an off-the-shelf UAV commercial onboard processing platform. Both the detection accuracy and efficiency met the requirements, which were 83.5% mAP and 20 FPS, respectively.

Chapter 5

Robust Feature-based Monocular VSLAM for Challenging Lighting Environments

5.1 Introduction

As discussed in Chapters 1 and 2, SLAM, which estimates the position of the robot while reconstructing the surrounding environment simultaneously in the unknown environments, has vital theoretical significance and application value. Moreover, it is the core technology of autonomous robots in the unknown environment [215]. VSLAM researches are blooming due to their convenience and relatively low requirements for sensors. VSLAM only relies on the camera, which obtains plenty of texture information and has been widely deployed on robotic platforms [216]. The accuracy and robustness of VSLAM are vital for autonomous navigation, especially in the complex lighting environment. For this reason, the

AGCWD is introduced into the stereo version of ORB-SLAM3 in Chapter 3 to realise robust localisation in a simulated pressure vessel. However, this approach can still not achieve robust localisation performance in complex environments, and processing stereo images needs lots of computing resources. On the contrary, monocular VSLAM systems only rely on the lightweight camera, and their simple calibration features make them particularly attractive for many robotic applications [217]. Thereby, in the following 2 chapters, two improved monocular VSLAM approaches will be introduced.

Depending on the image matching methods, monocular VSLAM systems can be divided into the featured-based method and the direct method [218]. The former extracts feature points from images and finds their corresponding based on geometric constraints, while the latter finds the corresponding of different frames based on their pixel intensities directly. The monocular VSLAM has been investigated from different perspectives, and lots of cutting-edge VSLAM methods such as the ORB-SLAM3 [219] and Direct Sparse Mapping (DSM) [220] have been developed. However, most advanced VSLAM systems are evaluated in well-lighted environments without considering challenging lighting conditions, such as dark, over-bright or dynamic illumination conditions. Visual blur or feature changes because of different illumination conditions that occur in these complex lighting environments. Therefore, the feature matching or frame-to-frame matching process is significantly affected by the changes in illumination conditions. As a result, monocular VSLAM systems may fail in these environments. Thus, developing a robust monocular VSLAM system for the challenging scenario with complex light has significant research and application value.

Towards this end, this chapter presents a robust monocular VSLAM system named AFE-ORB-SLAM through adopting the proposed adaptive FAST threshold and image enhancement techniques. In the proposed AFE-ORB-SLAM, the ORB-SLAM3 is chosen as the framework due to its excellent performance in well-lit environments. Unlike the VSLAM utilised in Chapter 2, which mainly focuses on the applications in the dimmed environment, the AFE-ORB-SLAM aims to achieve robust localisation in more challenging lighting environments. In order to handle the poor performance of the ORB-SLAM3 in challenging lighting environments, the truncated Adaptive Gamma Correction (AGC) is enhanced and combined with the unsharp masking method in the AFE-ORB-SLAM. Meanwhile, the proposed system is improved by the proposed efficient and adaptive FAST threshold method. The main contribution of this chapter is surmised as follows:

To address the issue of the degraded performance of feature-based VSLAM in an unideal lighting environment caused by sufficient feature points that cannot be extracted, an image contrast based adaptive FAST threshold and image contrast enhancement from the perspective of image contrast and sharpening are developed.

The rest of the chapter is structured as follows: Section 5.2 provides the framework of the AFE-ORB-SLAM. Section 5.3 introduces the details of improved image enhancement and feature extraction. Experimental results and analysis are given in Section 5.4. Section 5.5 concludes the whole work.

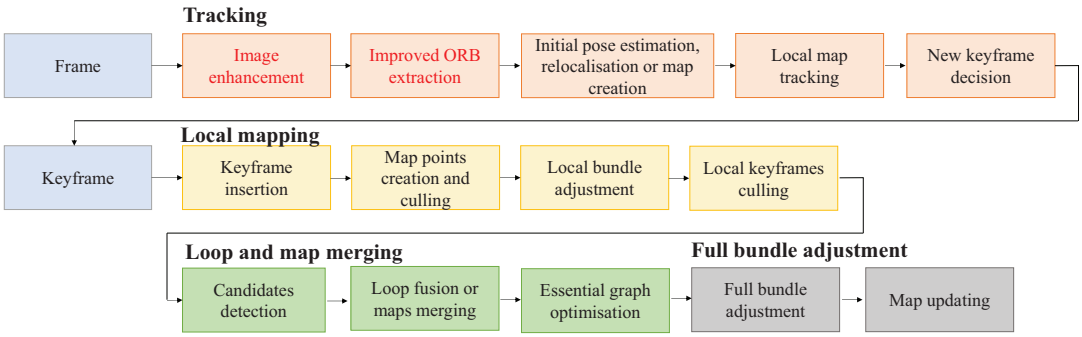


Figure 5.1: Structure of the AFE-ORB-SLAM

5.2 Structure of the AFE-ORB-SLAM

This chapter proposes the AFE-ORB-SLAM based on the ORB-SLAM3 framework for complex lighting environments. The overall schematic architecture is presented in Fig. 5.1. The two blocks with words in red are the main novel works proposed in this work. Three parallel threads (tracking, local mapping and loop and map merging) are utilised by the ORB-SLAM3. Besides, all the generated maps are managed by Atlas, which is a novel multiple-map system. A detailed introduction to the ORB-SLAM3 can be found in Section 3.2.2.

Although the ORB-SLAM3 with the monocular sensor achieves impressive performance in well-lit environments, its accuracy and robustness still suffer a lot in complex lighting environments. In these environments, the performances of feature extraction and matching drop significantly. When there are not enough matched ORB feature points obtained from the surrounding environment, the pose estimation process cannot be implemented, even leading to initialisation and tracking failures [155]. For the reasons mentioned above, it is crucial to incorporate the algorithm that can handle the variations of illumination into the ORB-SLAM3. In this chapter, the image enhancement technology and ORB

feature extraction are improved and deployed in the tracking thread to solve this problem. After the image enhancement, more distinct texture information is revealed. Besides, more stable ORB feature points could be obtained with the adaptive threshold of FAST feature extraction in complex lighting environments. Eventually, the accuracy and robustness of the ORB-SLAM3 are enhanced in complex lighting environments.

5.3 Robust VSLAM based on Image Enhancement and Adaptive FAST Threshold

5.3.1 Image Enhancement

The texture information is decreased in the dimmed or over-bright image. Thus, the captured images suffer from poor contrast. Contrast enhancement algorithms improve the visibility of objects in the dimmed or bright area by directly modifying pixel values based on the proper regulation [221]. Gamma correction [20] has gained lots of interest due to its easy adjustment and efficient implementation. The AGCWD [186] behaves well to enhance the images captured in the low-lighting environment. As the AGCWD focuses on improving the contrast of dimmed images, some detail loss occurs in the bright area. Inspired by Cao *et al.*'s work [23], truncated Cumulative Distribution Function (CDF)-based AGCWD (IAGCWD) is improved and adopted to process both dimmed and over-bright images. Thereby, the local over-enhancement can be reduced. To compensate for decreased sharpness because of image transmission and transformation, details and contours of the image are enhanced through unsharp masking technology. Eventually, with the combination of image contrast enhancement and image

sharpening adjustment technologies, texture information, especially for contours contained in the image, will be more prominent.

The overall structures of the IAGCWD and the proposed image enhancement method are shown in Fig. 5.2. The proposed image enhancement method consists of the contrast adjustment module and the sharpening adjustment module.

5.3.1.1 Image Contrast Enhancement

The standard deviation of the image intensity denotes the average contrast of the image [222], and it can be used to divide one image as the low contrast image and the high contrast image. The standard deviation of the image intensity is represented by λ. In this work, the following equation is derived to classify images:

$$I(x,y) = \begin{Bmatrix} I_{low}(x,y) & \lambda \leq 0.25 \\ I_{high}(x,y) & otherwise \end{Bmatrix} \tag{5.1}$$

Similar to Chapter 2, the PDF can be calculated by

$$PDF(l) = \frac{n_l}{N} \tag{5.2}$$

where l is the pixel intensity in the (x,y) position. n_l represents the number of pixels with intensity l, and N indicates the total pixels contained in the image. After the histogram distribution is smoothed by the weighting distribution function [187], the weighting distributed PDF can be formulated as

$$PDF_{wd}(l) = PDF_{max}(\frac{PDF(l) - PDF_{min}}{PDF_{max} - PDF_{min}})^{\alpha} \tag{5.3}$$

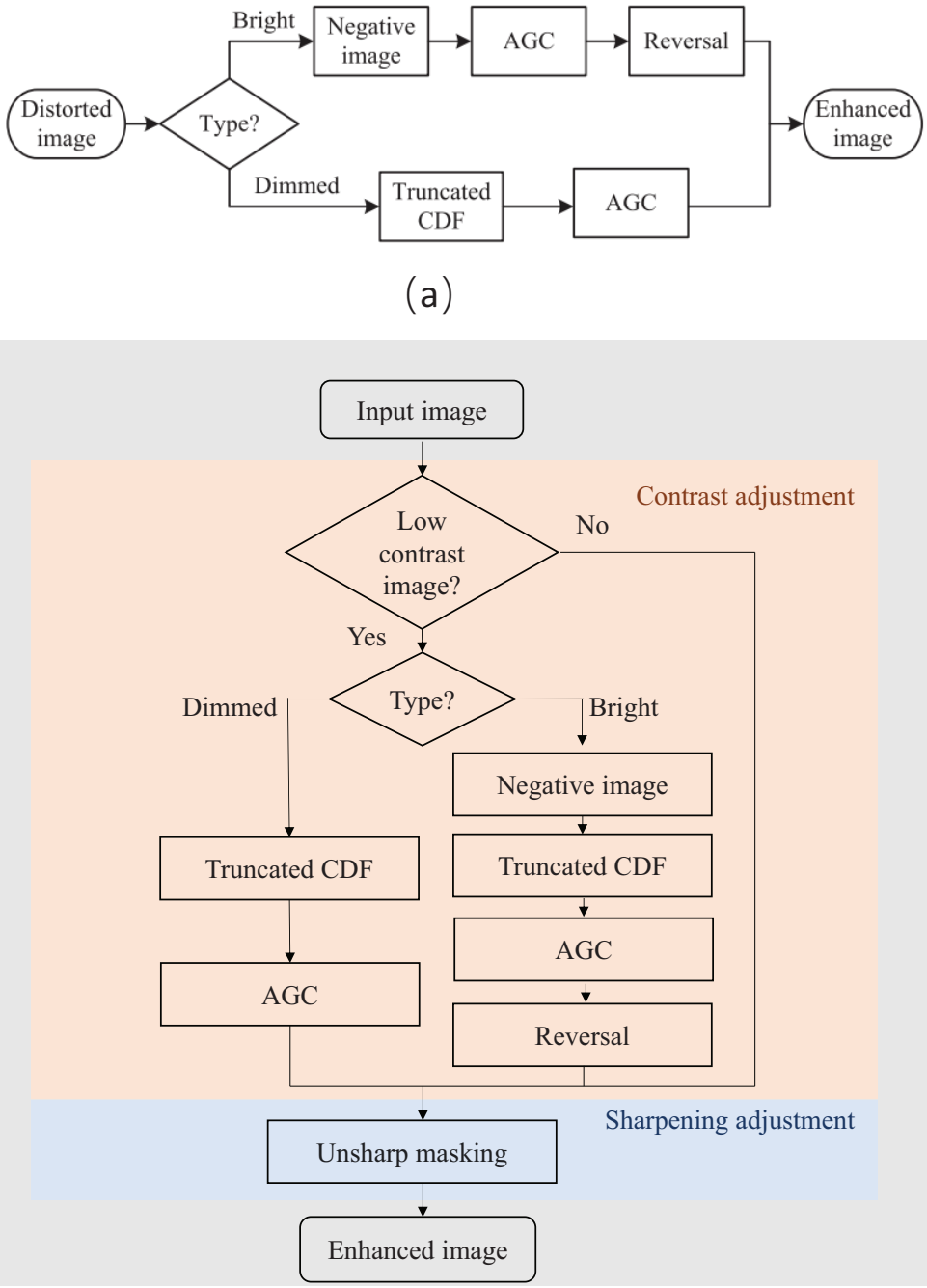


Figure 5.2: Structure of image enhancement. (a) IAGCWD [23], (b) the proposed image enhancement method.

where α is used to adjust the smooth level. The sum of $PDF_{wd}(l)$ can be calculated trough

$$SPDF_{wd} = \sum_{l=0}^{l_{max}} PDF_{wd}(l) \tag{5.4}$$

Thereby, the CDF can be formulated as

$$CDF_{wd}(l) = \frac{\sum_{l=0}^{l_{max}} PDF_{wd}(l)}{SPDF_{wd}} \tag{5.5}$$

Eventually, the parameter γ can be obtained through

$$\gamma = 1 - CDF_{wd}(l) \tag{5.6}$$

To improve the performance of the image enhancement, the dimmed and bright images should be processed differently. Thus, based on the average pixel intensity m_I, t is calculated to represent the overall brightness of the image. In this work, t is obtained through:

$$t = \frac{m_I - 128}{128} \tag{5.7}$$

Finally, the image is divided into the bright and dark sub-classes based on the value of t.

$$I(x,y) = \begin{Bmatrix} I_{bright}(x,y)\, t \geq 0 \\ I_{dark}(x,y)\ \ t < 0 \end{Bmatrix} \tag{5.8}$$

Following the image classification, the contrast of dimmed and bright images will be restored separately. The bright region in the dimmed image will be degraded due to an overly low gamma value. To this end, a truncated CDF [23] is utilised.

$$\gamma'_{wd} = max(\tau, \gamma) \tag{5.9}$$

τ is the threshold used for CDF truncation. It makes sure that bright regions are not adjusted by a low gamma value. From plentiful experimental observations, it is set to 0.3 in this work. Thereby, the detailed contour information in the bright area could be reserved. With the adoption of CDF truncation, the dimmed pixels will be processed by a small gamma value, while the restricted adjustment is applied to bright pixels. Thereby, in this work, the pixel intensity could be transformed with the following equation:

$$I_{ce}(l) = 255(\frac{l}{255})^{\gamma'_{wd}} \tag{5.10}$$

Specifically, the process to enhance the contrast of the dimmed image is introduced in Algorithm 1.

Algorithm 1 Contrast enhancement for the dimmed image

Step1: Calculate the $P(l)$ of the input image I.
Step2: Compute $PDF_{wd}(l)$ with the weighting distribution function
Step3: Obtain $CDF_{wd}(l)$ according to Eq.(5.5).
Step4: Calculate and choose proper γ'_{wd}
Step5: Output the contrast enhanced image I_{ce}

A large number of pixels in the dimmed or overly bright images have similar intensities. Over-bright images have high pixel intensities, and their negative images contain an enormous number of pixels with low-intensity values. Thus, the negative image of the over-bright image can be treated as a dimmed image, and it is formed by [23]:

$$I' = 255 - I(x, y) \tag{5.11}$$

Then, Algorithm 1 can be utilised directly to enhance I'. After that, the final contrast enhanced image I_{ce} could be obtained through the reverse of the enhanced negative image.

Finally, in this work, the contrast enhancement mask $T_{mask}(x, y)$ can be obtained as:

$$T_{mask}(x, y) = I_{ce}(x, y) - I(x, y) \tag{5.12}$$

5.3.1.2 Sharpening Adjustment

Image sharpening enhancement highlights the contour and makes the textures of the image clear. Unsharp masking [223] is a typical image sharpening technique. This technique utilises a low-pass filter to get a blurred image. Based on that, a mask is created and combined with the original image to make the texture of the image clear. Specifically, according to [223], the process of unsharp masking can be realised through the following steps:

The input image is processed by one low-pass filter

$$f(x, y) = I(x, y) * h_f(m, n) \tag{5.13}$$

where $*$ denotes the convolution operator, and $h_f(m, n)$ is a low-pass filter

Unsharp mask $g_{mask}(x, y)$ can be calculated through

$$g_{mask}(x, y) = I(x, y) - f(x, y) \tag{5.14}$$

The sharpened image can be obtained through

$$g_{sa}(x, y) = I(x, y) + k_{sha} \cdot g_{mask}(x, y) \tag{5.15}$$

where k_{sha} represents the sharpening level. For the unsharp masking technique, k_{sha} is set to 1. In this work, a Gaussian low-pass filter is used, which could be

represented by

$$G_0(x,y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{5.16}$$

in which σ is the standard deviation of the normal distribution.

Finally, in this work, the enhanced image can be represented by

$$I_{enhanced} = \begin{cases} I(x,y) + \alpha_{sha} \cdot g_{mask}(x,y) + \beta_{con} \cdot T_{mask}(x,y) & \lambda \leq 0.25 \\ I(x,y) + \alpha_{sha} \cdot g_{mask}(x,y) & otherwise \end{cases} \tag{5.17}$$

α_{sha} and β_{con} are two adjustable parameters, which controls the level of image image sharpening adjustment and contrast enhancement. As the unsharp masking technique is utilised in this work, α_{sha} should be set to 1.0. β_{con} is obtained by carrying out extensive experiments under different scenarios, and it is set to 0.3 in this work. Users can adjust them to achieve a more preferable result in a specific environment.

5.3.2 Adaptive FAST Threshold for Feature Extraction

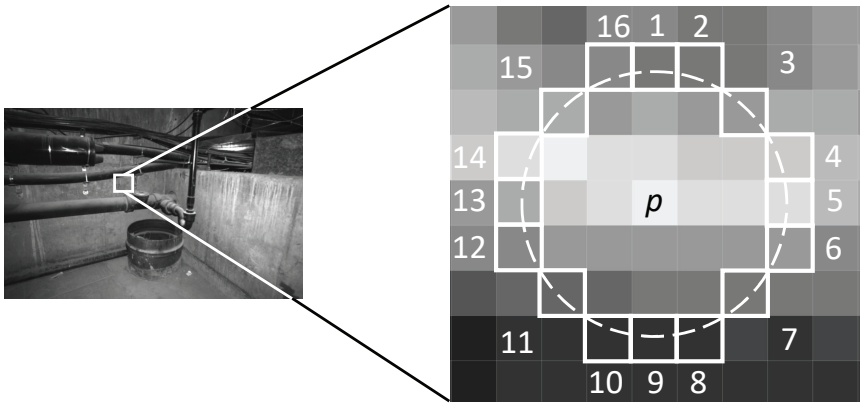


Figure 5.3: FAST keypoint extraction

The ORB feature is developed from the FAST keypoint and Binary Robust Independent Elementary Features (BRIEF) descriptor [224]. If the pixel intensity significantly differs from that of surrounding pixels, this pixel will be treated as a keypoint. To detect whether a pixel p is a FAST keypoint, the pixel intensity l_p will be compared with that of 16 pixels on a circle with a radius of 3 pixels (as shown in Fig. 5.3). A threshold θ is set manually to distinguish the current and surrounding 16 pixels. If there are over 12 contiguous pixels brighter than $l_p + \theta$ or darker than $l_p - \theta$, the current pixel will be considered as a FAST keypoint. To improve the detection efficiency, the differences between the current pixel and pixels on the circle with numbers 1, 5, 9 and 13 will be detected first. Wherein at least 3 points meet the condition that the pixel intensity difference is larger than θ or smaller than $-\theta$, the remaining 12 pixels on the circle will be detected. Otherwise, the pixel p will be discarded. Then, the scale, orientation invariance and BRIEF descriptor will be calculated by following the approach in [225].

Through the analysis above, the threshold θ is vital for the feature extraction process. Thereby, the performance of the whole VSLAM system will be improved through a proper θ value. However, a fixed θ cannot be adjusted to different illumination conditions. Thus, the feature extraction is degraded in different environments. To overcome this problem, an adaptive FAST threshold calculation method is proposed and adopted to the AFE-ORB-SLAM. Considering the computing efficiency, the λ used for the image enhancement is utilised to control the value of θ. Following the feature extraction process utilised in the ORB-SLAM3, two adaptive threshold values are set. The values of θ are set to 20 and 7 by the ORB-SLAM3. In this work, if enough feature points can be extracted from the

environment, a relatively large θ is used to obtain more reliable feature points.

$$\theta = \omega \cdot \lambda + 20 \tag{5.18}$$

If the number of extracted feature points is not enough in a quite low contrast image, a relatively small θ will be set.

$$\theta = \frac{\omega \cdot \lambda}{2} \tag{5.19}$$

ω is a parameter to control the threshold for ORB feature points extraction. In our work, as the texture information is enriched, ω is set to 128, which is the median value of the pixel intensity. To a specific scenario, users can adjust ω accordingly to obtain the best result.

5.4 Experiments

5.4.1 Experimental Environment

To verify the performance of the proposed AFE-ORB-SLAM, a laptop with Ubuntu 16.04 is used. The processor is Intel Core i7-8750H and the program uses C++ 17 compilation. Besides, the laptop is equipped with 12GB RAM. The Imperial College London and National University of Ireland Maynooth (ICL-NUIM) dataset with simulated lighting changes [145], Onboard Illumination Visual-Inertial Odometry (OIVIO) dataset [226] and the European Robotics Challenge (EuRoC) dataset [227] are utilised to verify the localisation accuracy and illumination robustness of the proposed AFE-ORB-SLAM.

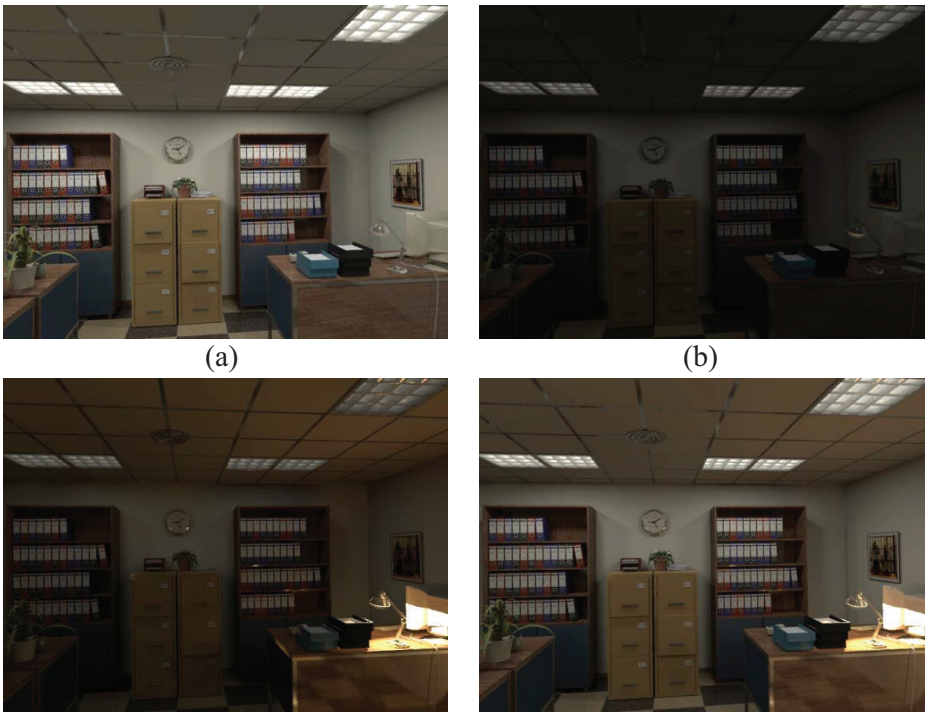


Figure 5.4: Example images in the ICL-MUIM dataset with simulated lighting changes [145]. (a) Original illumination, (b) global illumination, (c) local illumination, (d) local and global illumination.

The ICL-NUIM dataset with simulated lighting changes is a synthetic dataset, and the camera position is available as the ground truth. It contains image sequences under different illumination conditions. Thus, it is suitable for testing the performance of VSLAM systems under different lighting conditions. The office room sequences with static, local variation, global variation and local and global variation lighting conditions are used in this work. Some sample images are shown in Fig. 5.4.

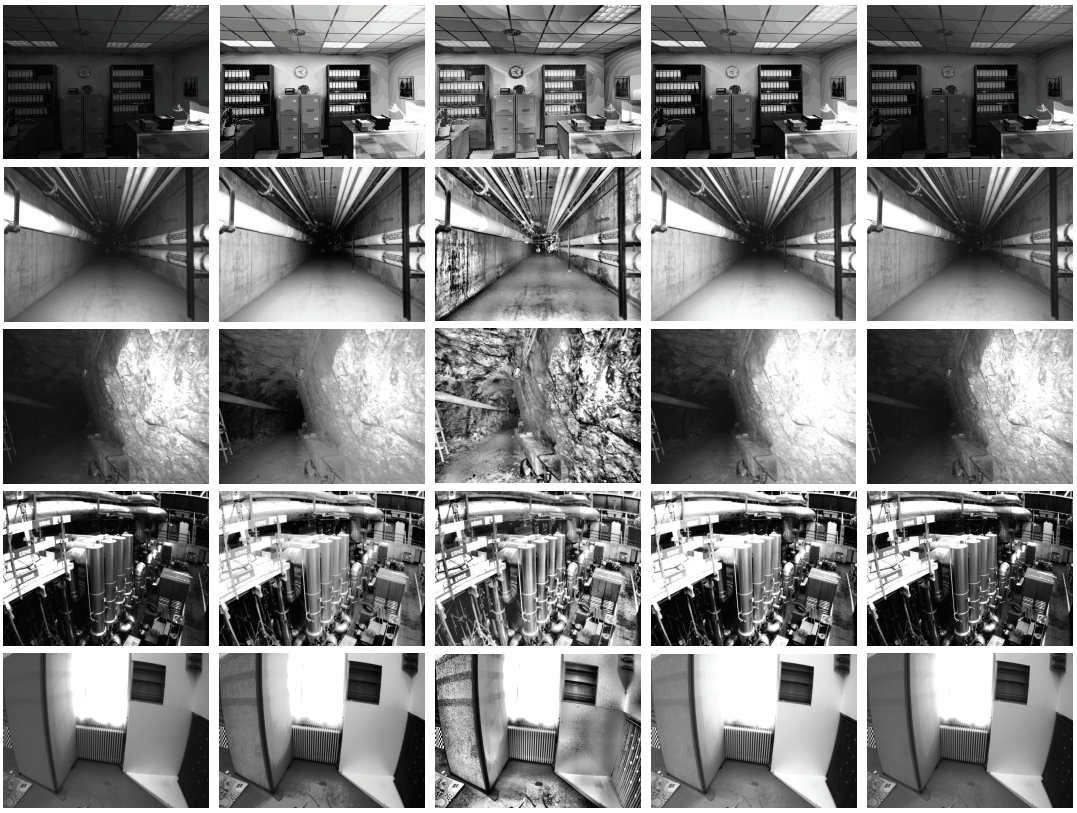
The OIVIO dataset contains 9 image sequences captured by the Clearpath Husky UGV in weakly lighted environments, such as mines, tunnels and other dark environments. There are 3 scenarios named "MINE GROUND-VEHICLE 1", "MINE GROUND-VEHICLE 2" and "TUNNEL GROUND-VEHICLE 1"

have the ground truth generated by the Leica TCRP1203 R300, and these sequences are utilised in our work to verify the performance of VSLAM systems. What is more, an onboard light of approximately 1350, 4500, or 9000 lumens is utilised to illuminate each scene. Some example images are shown in Fig. 5.5.

Figure 5.5: Example images in the OIVIO dateset [226]

Figure 5.6: Example images in the EuRoC dateset [227]

The EuRoC dataset contains 11 sequences collected by the AscTec "Firefly" hex-rotor helicopter. Among them, 5 sequences are recorded in a large machine hall with ground truth provided by a Leica Multistation. The other 6 sequences are recorded in a small Vicon room with ground truth provided by the motion capture system. To complete the V103 sequence, the ORB-SLAM3 relies on the multi-map system significantly, and the ORB-SLAM3 cannot complete the V203 sequence. Tracking lost will lead to unpredictable threats to robot platforms. Thus, in this work, the other 9 sequences are chosen to validate VSLAM methods to simulate their performances on a robot platform. Some example images are shown in Fig. 5.6



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

Figure 5.7: Results of image enhancement. (a) Original images; Enhanced images by HE (b), CLAHE (c), IAGCWD (d) and the proposed method (e).

If the trajectory has a loop, the motion trajectory generated by ORB-SLAM3 and other ORB-SLAM based VSLAM algorithms will be optimised by g2o [228].

5.4.2 Verification of Image Enhancement

To compare performances of different image contrast enhancement methods, the HE and CLAHE that are utilised in VSLAM systems, and the original IAGCWD are chosen. Fig. 5.7 demonstrates the results of different image enhancement algorithms. Fig. 5.7 (a) indicates the original images selected from different scenarios. As shown in Fig. 5.7 (b) and (c), some high contrast images are

achieved. However, if there are some noises contained in the images, the noises will also be significantly amplified. Fig. 5.7 (d) shows the results achieved by the IAGCWD, and it incurs over-enhancement in some bright regions. Fig. 5.7 (e) proves that the contrast and visibility of the texture information contained in images are enhanced by the proposed method.

5.4.3 Evaluation on the ICL-NUIM dataset with simulated lighting changes

To verify a VSLAM system, ATE [229] is a common practice. The ATE represents the difference between the ground truth and the path estimated by the VSLAM system. To verify the pose estimation performance of the AFE-ORB-SLAM, the PL-SLAM, DSM and ORB-SLAM3 with default parameters are selected for comparison. The median value of the localisation results for each method from 10 times running are presented.

Table 5.1: Performance comparison on the ICL-NUIM dataset with simulated lighting changes for the mean ATE (m) and RMS ATE (m). The best results are highlighted in a bold font.

	DSM		PL-SLAM		ORB-SLAM3		AFE-ORB-SLAM	
ICL-NUIM benchmark	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE
Syn1	**0.0028**	**0.0030**	0.0271	0.0311	0.0684	0.0732	0.0340	0.0375
Syn1-local	0.8524	0.8937	0.0385	0.0429	0.2214	0.2952	**0.0376**	**0.0407**
Syn1-global	**0.0031**	**0.0035**	-	-	0.1543	0.2065	0.0329	0.0365
Syn1-local-global	**0.0112**	**0.0119**	0.2158	0.3132	0.1309	0.1684	0.0299	0.0333
Syn1-average	0.2174	0.2280	0.0938*	0.1291*	0.1438	0.1858	**0.0336**	**0.0370**
Syn2	0.5426	0.5997	0.0964	0.1093	0.0848	0.1273	**0.0717**	**0.0817**
Syn2-local	0.5438	0.5920	**0.0841**	**0.1001**	0.0758	0.0951	0.0889	0.1160
Syn2-global	0.5266	0.5722	0.0935	0.1046	0.1103	0.1458	**0.0775**	**0.0839**
Syn2-local-global	0.5198	0.5664	0.0858	0.0992	0.0858	0.0982	**0.0670**	**0.0764**
Syn2-average	0.5332	0.5826	0.0899	0.1033	0.0892	0.1166	**0.0763**	**0.0895**

Table 5.1 indicates the mean ATE and RMS ATE of keyframe trajectories. If the VSLAM system cannot complete all the sequences, the results will be marked by *. With the feature points extracted with the proposed adaptive threshold from the enhanced image, the AFE-ORB-SLAM outperforms the original ORB-SLAM3 in all video sequences. Fig. 5.8 visualises the trajectory of the ORB-SLAM3 and AFE-ORB-SLAM in the office scenario with local and global variations in lighting conditions. A large offset occurs on the initialising stage of the ORB-SLAM3, while the AFE-ORB-SLAM has a relatively smaller error compared to the ground truth, and the proposed method outperforms the ORB-SLAM3 in all coordinate directions.

For the DSM, even though it could achieve the best localisation accuracy in several sequences contained in the Syn1 scenario, it is still vulnerable to different illumination conditions. Moreover, the developed method shows the smallest error considering the average performance of the same sequence under different illumination conditions. The effectiveness of the proposed image enhancement method and ORB feature points with the adaptive threshold is verified. The overall results prove the robustness and effectiveness of the AFE-ORB-SLAM in environments with different illumination conditions. This allows the autonomous system to work robustly in environments with different lighting conditions.

5.4.4 Evaluation on the OIVIO dataset

To further evaluate the performance of the AFE-ORB-SLAM, apart from the VSLAM methods utilised in Section 5.3, VSLAM systems improved by the image contrast enhancement methods are also utilised for comparison. The HE-SLAM and CLAHE-SLAM represent the monocular version of [149] and [151], respec-

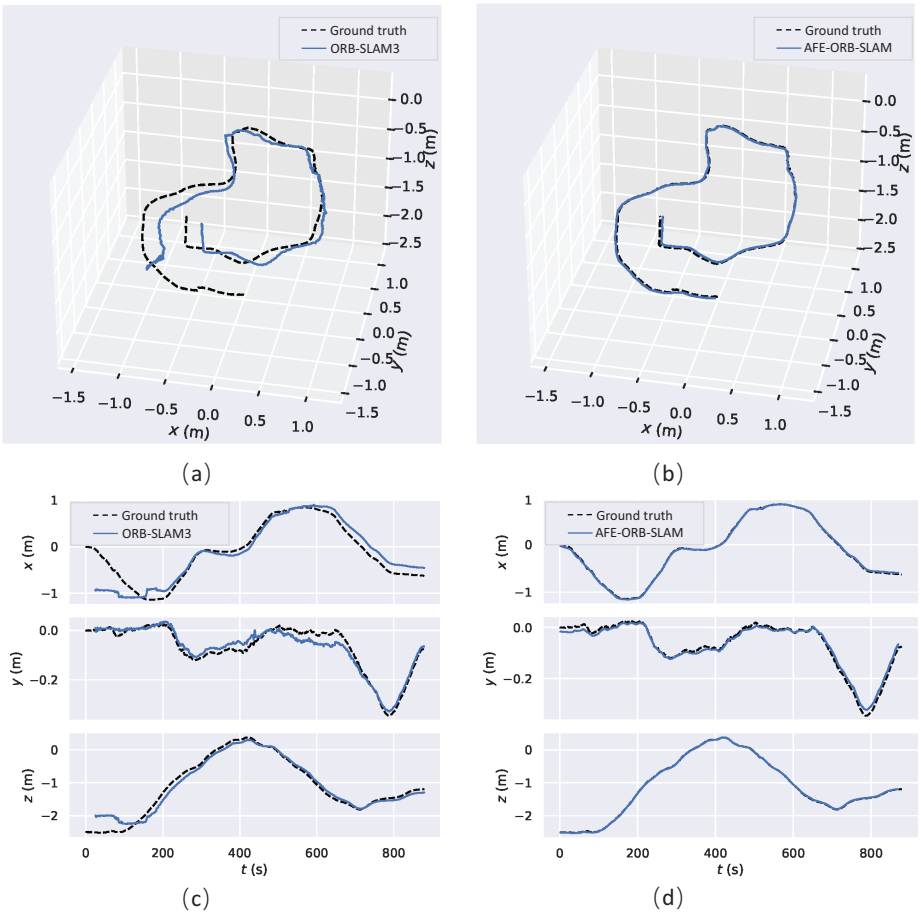


Figure 5.8: Visualised trajectory estimation of the ORB-SLAM3 and AFE-ORB-SLAM. (a) and (b) are the overview of the whole trajectory. (c) and (d) are the detailed camera position in x, y and z directions.

tively. The IAGC-SLAM indicates the ORB-SLAM3 with the IAGCWD as the pre-processing technique. Meanwhile, the effect of the proposed image contrast enhancement method and the adaptive FAST threshold for ORB feature extraction are analysed separately. If only the proposed image contrast enhancement method is adopted to the ORB-SLAM3, the VSLAM system is named the IE-SLAM. The TH-SLAM represents the ORB-SLAM3 improved by the adaptive FAST threshold for ORB feature extraction. Finally, the DSM, PL-SLAM, ORB-SLAM3, HE-SLAM, CLAHE-SLAM, IAGC-SLAM, IE-SLAM and TH-SLAM are

selected to compare with the proposed AFE-ORB-SLAM.

Table 5.2: Performance comparison on the OIVIO dataset for the RMS ATE (m). The best results are highlighted in a bold font.

OIVIO benchmark	DSM	PL-SLAM	ORB-SLAM3	HE-SLAM	CLAHE-SLAM	IAGC-SLAM	IE-SLAM	TH-SLAM	AFE-ORB-SLAM
MN_015_GV_01	5.5781	0.1858	0.1780	0.2848	0.2033	0.1746	0.1528	0.1461	**0.1265**
MN_050_GV_01	0.9177	0.2707	0.2218	0.2128	0.1965	0.2465	0.1810	0.1873	**0.1431**
MN_100_GV_01	0.6433	0.2494	0.1714	0.1765	0.1467	0.1799	0.1779	0.1578	**0.1254**
MN_015_GV_02	3.3304	0.2150	0.1186	0.1285	0.1014	0.1527	0.0937	0.1041	**0.0855**
MN_050_GV_02	0.8907	0.1977	0.1040	0.1203	0.1256	0.1587	0.0921	0.0969	**0.0891**
MN_100_GV_02	0.4740	0.1428	0.0964	0.1298	0.0886	0.1315	0.0936	0.0928	**0.0854**
TN_015_GV_01	0.8458	-	0.3231	1.1108	0.4548	0.4131	0.2751	0.2382	**0.1728**
TN_050_GV_01	1.0153	0.5171	0.2693	0.5378	0.2591	0.4320	0.2271	0.3393	**0.1569**
TN_100_GV_01	0.4948	0.3394	0.2551	0.2425	0.1608	0.2039	0.1649	0.3241	**0.1511**
Average	1.5767	0.2647*	0.1931	0.3281	0.1930	0.2325	0.1620	0.1874	**0.1262**

To simulate the performance of different VSLAM systems on robot platforms, the full trajectories generated by VSLAM systems are used to calculate the RMS ATE. As the DSM and PL-SLAM only output the keyframe trajectory, the keyframe trajectory is still utilised in this section. The median RMS ATE of 10 executions is provided in Table 5.2. If the VSLAM system cannot complete all sequences, the results are marked by *. Compared with the ICL-NUIM dataset with simulated lighting changes, sequences in the OIVIO dataset have long trajectories, and there is no loop closure during the whole process. What is more, the texture information is not as rich as that of the ICL-NUIM dataset with simulated lighting changes, especially for the TUNNEL scenario. The RMS ATE obtained in this dataset is larger than the ICL-NUIM dataset with simulated lighting changes.

The performance of DSM is significantly influenced by the illumination conditions, and it achieves the worst performance in almost all sequences. The PL-SLAM fails in the TN_015_GV_01 sequence due to the weak visual connectiv-

ity. The HE-SLAM, CLAHE-SLAM and IAGC-SLAM achieve higher localisation accuracy than the ORB-SLAM3 in several sequences. However, the noises contained in the images are also enhanced, and the over-enhancement exists in some regions. The average accuracy of the ORB-SLAM3 outperforms that of the HE-SLAM and IAGC-SLAM, while the CLAHE-SLAM and ORB-SLAM3 have similar average accuracy. The IE-SLAM and TH-SLAM obtain better results than the ORB-SLAM3 in most sequences. However, due to the less texture information contained in the TUNNEL scenario, the TH-SLAM performs worse than the ORB-SLAM3. Apparently, with the proposed image enhancement method and ORB feature points extracted through the adaptive threshold, the AFE-ORB-SLAM achieves the best localisation performance in all sequences. Moreover, these sequences are collected by the UGV in mines and tunnels, which further attests that the AFE-ORB-SLAM is robust to different illumination conditions and can achieve accurate localisation performance in real-world scenarios.

The visualised localisation results are exhibited in Fig. 5.9 for the TUNNEL scenario. It shows that the low visibility of the environment has a great impact on the DSM. The PL-SLAM and ORB-SLAM3 rely on feature matching against neighbouring frames. When not enough reliable matched feature pairs are obtained, significant performance degradation can be observed. Considering the HE and CLAHE algorithms cannot handle different images properly, their performances are also influenced by different illumination conditions. The proposed AFE-ORB-SLAM could localise the robot accurately under different illumination conditions.

The time usage of the DSM, PL-SLAM, ORB-SLAM3, HE-SLAM, CLAHE-SLAM and AFE-ORB-SLAM for the scenario under different illumination con-

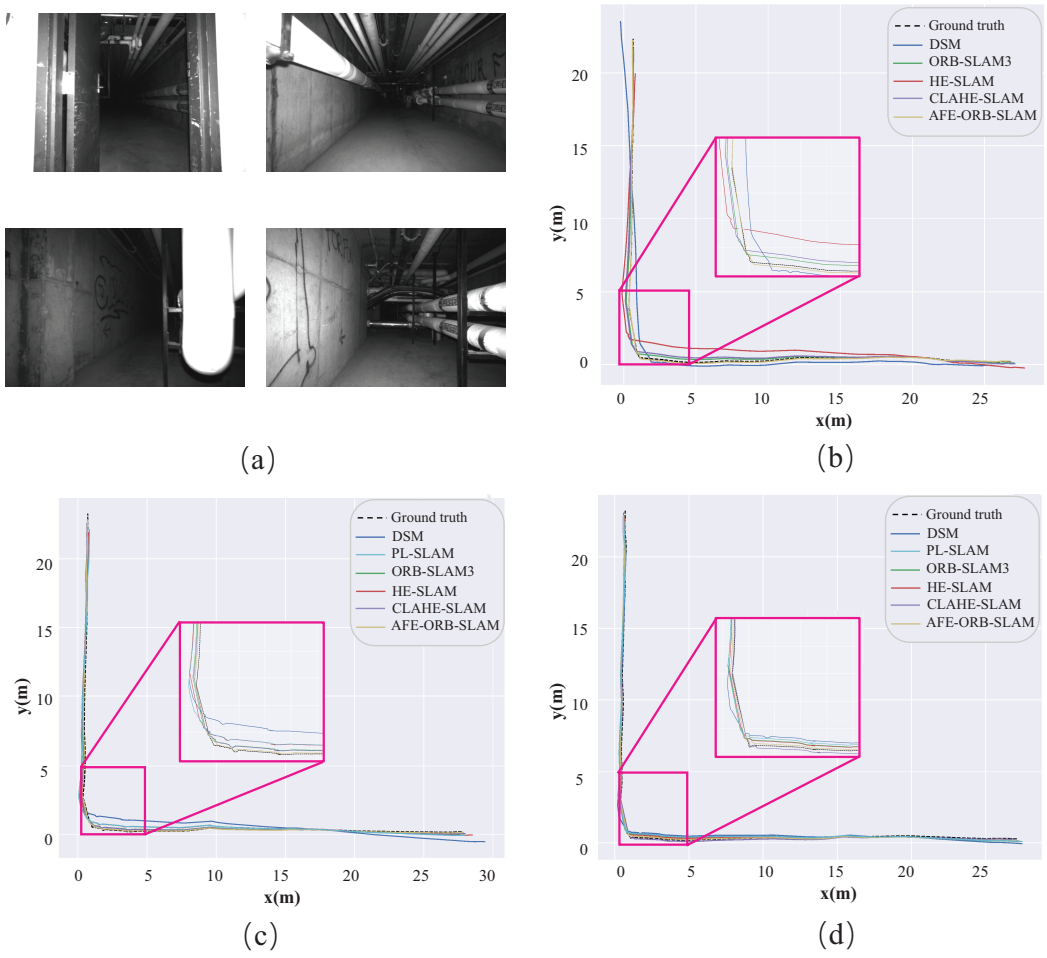


Figure 5.9: Trajectory comparison on the TUNNEL sequences. (a) are some sample images. (b), (c) and (d) indicate the localisation performance with the light of 1350, 4500, or 9000 lumens, respectively.

ditions are averaged and shown Fig. 5.10. The results further confirm that the AFE-ORB-SLAM is able to deal with the challenging scenes that provide less visual information effectively and efficiently. The average accuracy is improved by 34.65% with the comparison to the ORB-SLAM3. In contrast, the average processing time is only increased by 3.38%.

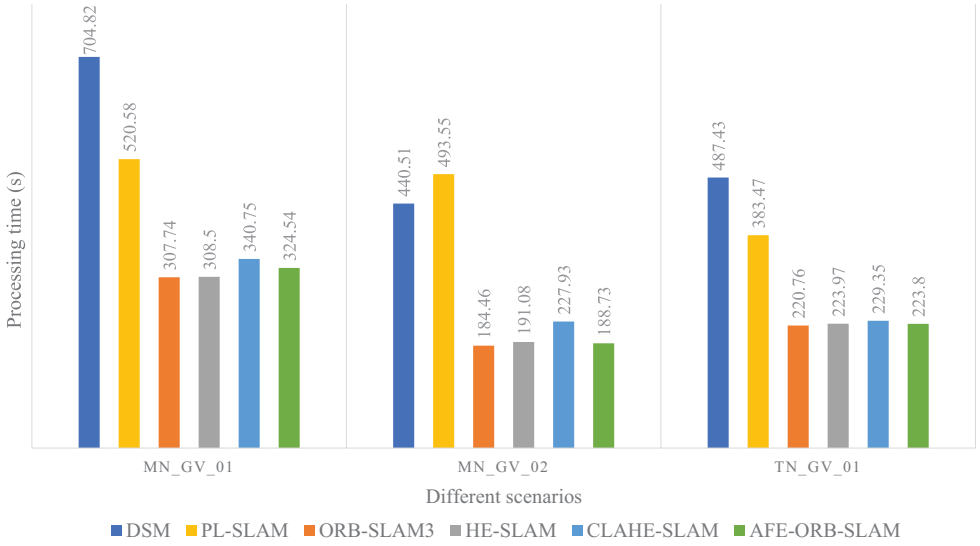


Figure 5.10: Comparison of time usage for different VSLAM systems

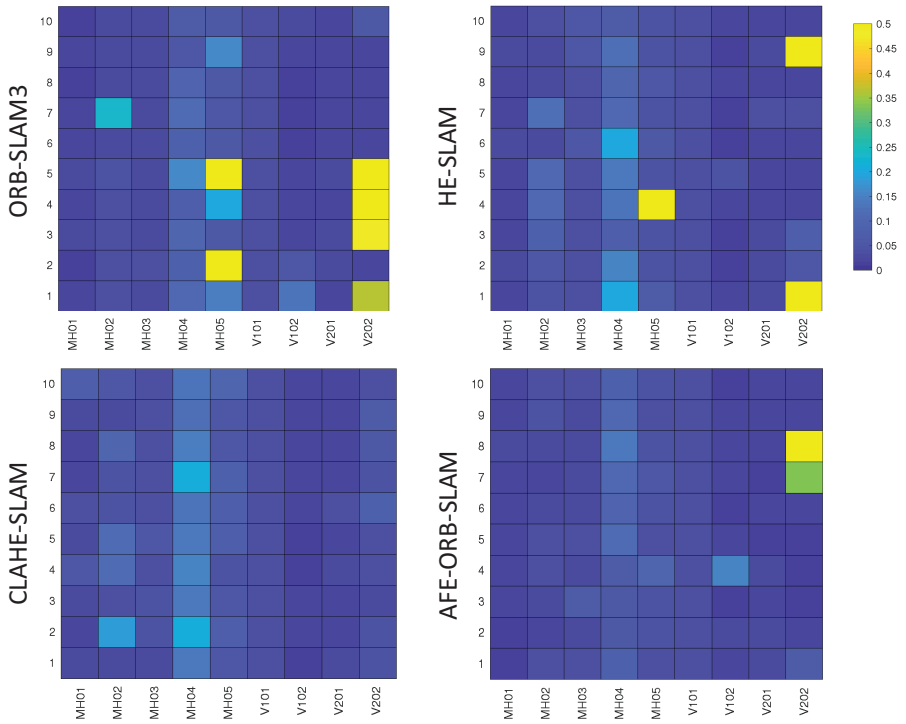


Figure 5.11: Precision comparison of different VSLAM methods

Table 5.3: Performance comparison on the EuRoC dataset for the RMS ATE (m). The best results are highlighted in a bold font.

EuRoC benchmark	DSO [230]	SVO [231]	DSM [220]	ORB-SLAM3	HE-SLAM	CLAHE-SLAM	AFE-ORB-SLAM
MH01	0.046	0.100	0.039	**0.017**	0.022	0.030	0.018
MH02	0.046	0.120	0.036	**0.032**	0.047	0.047	**0.032**
MH03	0.172	0.410	0.055	**0.028**	0.036	0.037	**0.028**
MH04	3.810	0.430	**0.057**	0.088	0.125	0.139	0.087
MH05	0.110	0.300	0.067	0.103	0.045	0.061	**0.041**
V101	0.089	0.070	0.095	**0.033**	**0.033**	**0.033**	**0.033**
V102	0.107	0.210	0.059	0.018	**0.016**	**0.016**	**0.016**
V201	0.044	0.110	0.056	**0.022**	0.023	**0.022**	0.023
V202	0.132	0.110	0.057	0.037	0.027	0.040	**0.017**
Average	0.506	0.207	0.058	0.042	0.042	0.047	**0.033**

5.4.5 Evaluation on the EuRoC dataset

The AFE-ORB-SLAM is further validated on the EuRoC dataset. Comparisons of the AFE-ORB-SLAM against the DSO [230], SVO [231], DSM, ORB-SLAM3, HE-SLAM and CLAHE-SLAM are presented in this section. The results published in [230] for the DSO, in [231] for the SVO and in [220] for the DSM are utilised. For other VSLAM systems, the median RMS ATE of 10 executions for the full trajectories is obtained. The results are shown in Table 5.3. As the most images in the EuRoC dataset contains rich texture information with regular lighting conditions, the ORB-SLAM3 achieves similar localisation accuracy with HE-SLAM and CLAHE-SLAM. Owing to the MH05 sequence containing images collected at night, the localisation performance improvement by applying the image contrast enhancement could be observed. The AFE-ORB-SLAM achieves a higher localisation accuracy in the MH05, V102 and V202 sequences than the ORB-SLAM3. Moreover, the proposed AFE-ORB-SLAM still achieves the best

average accuracy on selected scenarios.

To verify the robustness of different VSLAM systems, the results with 10 times execution are presented in Fig. 5.11. Different colour squares represent the RMS ATE obtained in each of the 10 executions. The results demonstrate that the precision of the ORB-SLAM3 could be improved by adopting proper image contrast enhancement method, and the CLAHE-SLAM achieves the best results. The proposed AFE-ORB-SLAM outperforms the ORB-SLAM3 in terms of not only the accuracy but also the precision.

5.5 Summary

Based on the VSLAM presented in Chapter 3, a robust monocular VSLAM named AFE-ORB-SLAM was proposed to locate the robot in complex lighting environments. The main goal of this work was to extract more reliable feature points from the images captured in challenging lighting conditions for VSLAM approaches. Thereby, the IAGCWD was adopted to improve the contrast of dimmed images. For over-bright images, their negative images were utilised and processed by the IAGCWD. After reversing the enhanced negative images, the final enhanced over-bright images were obtained. In addition to the contrast enhancement, the sharpening adjustment was also achieved through the implantation of the unsharp masking. Finally, the image was enhanced by both the contrast adjustment and the sharpening adjustment. Besides image processing, the way of feature extraction was also improved to extract more robust feature points. Specifically, the ORB feature extraction was enhanced by adopting the adaptive FAST threshold. When the contrast of the image was low, a relatively small threshold for ORB feature extraction would be utilised. On the contrary, a relatively large FAST

threshold was utilised when enough feature points could be obtained in a high-contrast image. As shown in Fig. 5.1, both the proposed image enhancement and adaptive FAST threshold were embedded into the ORB-SLAM3 to improve the localisation performance of the ORB-SLAM3 in complex lighting environments.

Extensive experiments were carried out in Section 5.4 to verify the performance of the AFE-ORB-SLAM. First of all, the effect of the image enhancement method was validated, and the results showed that the texture information contained in images was enhanced by the proposed method. Then, the ICL-NUIM dataset with simulated lighting changes and the OIVIO dataset were utilised to test the localisation accuracy of the AFE-ORB-SLAM. Experiments demonstrated that the AFE-ORB-SLAM was capable of achieving accurate and robust localisation performance in environments where the images were captured under different illumination conditions, even with less visual information. Finally, the AFE-ORB-SLAM was validated by the EuRoC dataset, where the lighting conditions were not challenging. Results showed that the AFE-ORB-SLAM preserved the excellent performance of the ORB-SLAM3 in the well-lit environments. All of these results proved the effectiveness of the adoption of the proposed image enhancement method and ORB feature points with adaptive threshold into The VSLAM system.

Thereby, the issue of the degraded performance of ORB-SLAM3 caused by the poor feature extraction capability in unideal lighting environments is addressed through the adoption of the proposed image enhancement method, which enhances the images from contrast and sharpening perspectives with the ORB feature points extracted with the image contrast-based adaptive threshold calculation.

Chapter 6

Deep Learning Feature-based Monocular VO for Challenging Environments

6.1 Introduction

Chapters 3 and 5 have introduced two improved VSLAM systems for challenging lighting environments. The image enhancement techniques were adopted to enrich the texture information. What is more, the adaptive FAST threshold is utilised in Chapter 5 to extract more robust feature points. Both algorithms are traditional VSLAM algorithms, and they apply hand-engineered features that represent feature regions in the image to calculate the pose of the camera and map points of the surrounding environment. Chapters 3 and 5 aim to extract more ORB feature points from the challenging illumination environments for the VSLAM system. However, ORB points are not robust enough for different envi-

ronments. Therefore, in this chapter, the robustness of the VO/VSLAM will be improved in another aspect: the types of the feature points.

Different kinds of feature points have already been adopted into VO/VSLAM systems. The parallel tracking and mapping processes of the PTAM [232] are based on FAST [233] feature extraction. After that, FAST feature extraction is improved by combing with the BRIEF and adopted in the most famous ORB-SLAM series [225] [150] [219]. These VSLAM systems are effective in general environments. However, when they are deployed into the complex environments, such as the unideal light conditions and textureless environments, their performance is degraded significantly, and they may even be unable to localise the camera.

Recently, as deep learning has defined the state-of-the-art in many research areas [234] [235], adopting deep learning-based features into VO/VSLAM systems has gained increasing interest from researchers. A complete survey can be found in Section 2.3. Compared with traditional features, features extracted by the CNN are more robust, which results in an improvement in the pose estimation accuracy. However, a powerful GPU is required to deploy deep learning-based methods. To some extent, efficiency is sacrificed to improve robustness and accuracy. To this end, the balance of accuracy and efficiency should be taken into consideration while deploying the deep learning-based features into the VO/VSLAM systems, especially for the UAV platforms that have limited payload capability. In this chapter, a deep learning feature-based VO system for UAV onboard platforms based on an efficient feature extraction network is proposed. Specifically, an efficient CNN model is proposed for keypoint detection. The designed network deals with constraints on computation as the UAV has limited resources. The

main contributions of this chapter can be summarised as follows:

To address the challenge of extracting sufficient feature points in low-contrast and textureless environments for the UAV onboard VO system, a lightweight CNN model is designed for feature extraction. Specifically, DSconv is employed to reduce the computing complexity of CNN, and the DFconv with the kernel offset calculated through the DSconv is utilised to extract feature points.

The rest of this chapter is organised as follows: Section 6.2 introduces the VO system based on the efficient feature extraction network. in Section 6.3, the experimental results and analysis are provided. The conclusion is given in Section 6.4.

6.2 Robust VO based on Deep Local Features

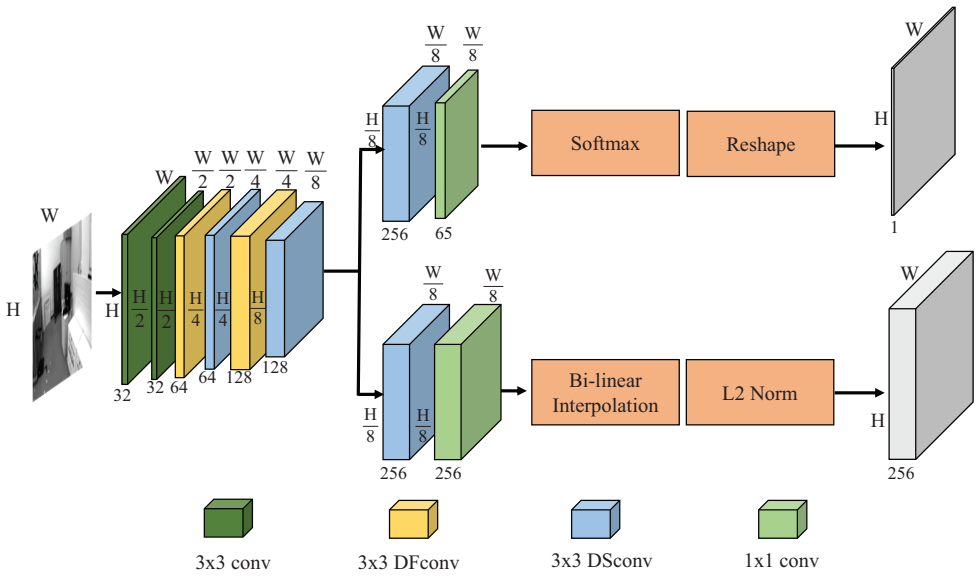


Figure 6.1: The structure of the network for feature detection and description

Considering the deep learning feature-based VO/VSLAM needs to focus on deployment on UAV platforms, the efficiency and accuracy of the feature extraction module should be balanced. Inspired by the SuperPoint, which is trained by the self-supervision method and achieves the desirable homography estimation results, a novel efficient CNN model for feature extraction is proposed. The DSconv [182] is adopted to reduce parameters, and it is also adopted to the DFconv [236] to improve the feature extraction capability. The proposed feature detection and description module is shown in Fig. 6.1. It shows that the model includes a shared backbone network, followed by two sub-modules for feature point detection and description. The input image is processed by traditional convolutional layers, DSconv layers and DFconv layers. The details are introduced as follows.

6.2.1 Deep Learning-based Feature Extraction and Description

6.2.1.1 Depthwise Separable Convolution

The DSconv is adopted to reduce model parameters to fit embedded computing platforms. The detailed introduction to DSconv can be found in Section 4.3.2. Therefore, This chapter will mainly introduce the improved DFconv.

6.2.1.2 Deformable Convolution

The DSconv scarifies the accuracy to reduce the computational cost, and traditional CNN is difficult to accommodate geometric variations properly owing to the fixed geometric structures. Compared with the traditional convolution, the DFconv layer adds a 2D offset to the sampling grid, which enables the free

form deformation of the convolutional kernel. As shown in Fig. 6.2, the DFconv consists of two feature preprocessing channels. The upper channel learns the sampling locations for the convolutional kernel. To reduce parameters and improve the robustness of the DSconv, the traditional convolution is replaced by the DSconv to calculate the 2D offset matrix. Then, the convolutional operation is performed between the input data and the deformed convolutional kernel accordingly. Thus, the DFconv can extract features from non-uniform shapes effectively.

In a 3×3 convolutional kernel with dilation 1, the convolutional grid R can be formalised as

$$R = \begin{Bmatrix} (-1,-1) & (-1,0) & (-1,1) \\ (0,-1) & (0,0) & (0,1) \\ (1,-1) & (1,0) & (1,1) \end{Bmatrix} \tag{6.1}$$

The output feature map on location p_0 can be obtained through:

$$y(p_0) = \sum_{p_p \in R} w(p_p) \cdot x_{ifm}(p_0 + p_p) \tag{6.2}$$

where w indicates the convolutional weights. x_{ifm} represents the input feature map. p_p means the position in R. In the DFconv, offsets $\left\{\Delta p_p | p = 1, 2, ..., P\right\}$ is added to R, and offset locations $p_p + \Delta p_p$ allows the convolutional kernel to form an irregular shape. Thereby, the DFconv is formulated as:

$$y(p_0) = \sum_{p_p \in R} w(p_p) \cdot x_{ifm}(p_0 + p_p + \Delta p_p) \tag{6.3}$$

As offset Δp_p learned by the DSconv is usually fractional, bilinear interpolation is implemented to revise the offset as an integer.

$$x_{ifm}(p) = \sum_{q} B(q,p) \cdot x_{ifm}(q) \tag{6.4}$$

where B denotes the bilinear interpolation kernel. p and q represent the fractional and integral locations, respectively.

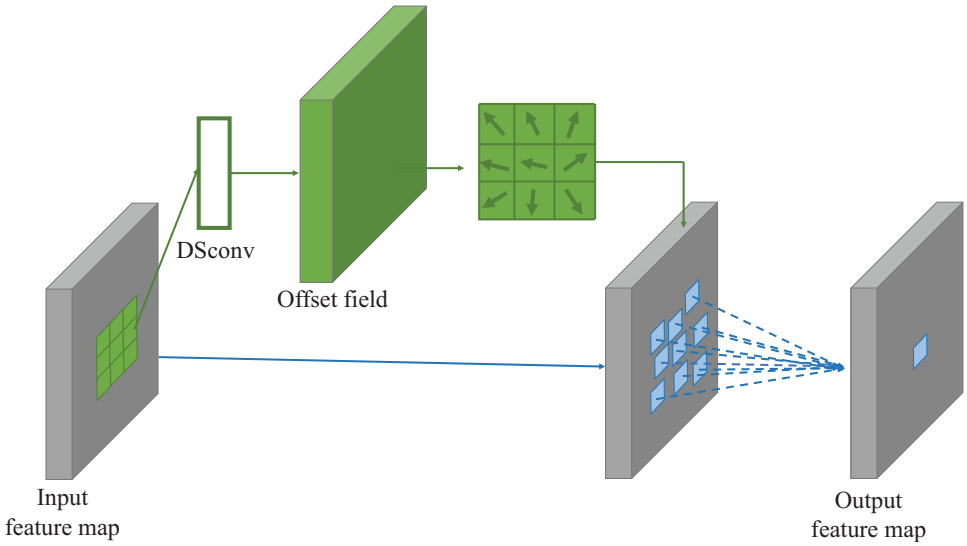


Figure 6.2: Deformable convolution

6.2.1.3 Activation Functions

In this work, the Exponential Linear Unit (ELU) [237] is chosen as the activation function. The expression for the ELU is:

$$f(x) = \begin{cases} x & x \geq 0 \\ \alpha_{elu}(e^x - 1) & x < 0 \end{cases} \tag{6.5}$$

The hyperparameter α_{elu} controls saturate for negative inputs. Unlike the ReLU only has positive values, the negative values of ELU push the mean unit activations closer to zero, which accelerates the training speed and improves the stability of the training process.

6.2.1.4 Training Process

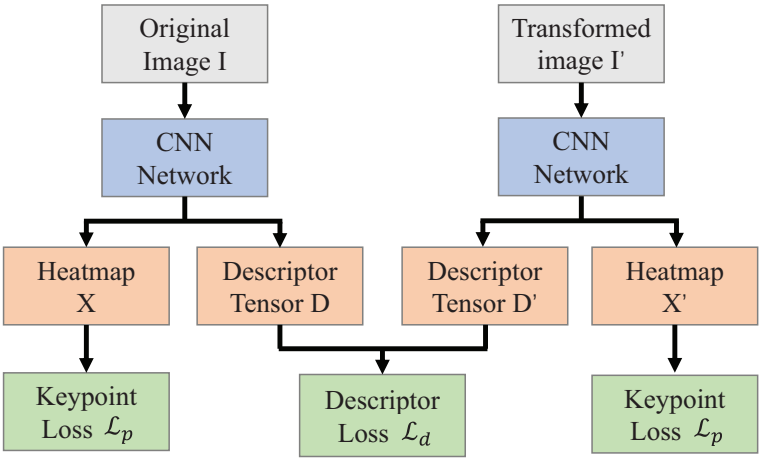


Figure 6.3: Training process

To avoid the data labelling process, which is time-consuming and laborious, the self-training strategy is adopted. The training process is illustrated in Fig. 6.3. Firstly, the model provided in SuperPoint [167] is utilised to generate pseudo ground truth for unlabelled images. To improve the robustness of the model, the Homographic Adaptation [167] is employed to enlarge the dataset. The augmentation process can be represented by

$$\hat{A}(X; f_{ipa}) = \frac{1}{N_H} \sum_{i=1}^{N_H} H_i^{-1}(f_{ipa}(H_i(X))) \tag{6.6}$$

where N_H is the number of the generated homography matrix. H and H^{-1} are the randomly generated homography matrix and the corresponding reverse. f_{ipa}

represents the interest point adaption function.

The feature point loss function $\mathcal{L}_p$ is a cross-entropy loss, which can be expressed as:

$$\mathcal{L}_p(X_o, L_o) = -\frac{1}{H_d W_d} \sum_{i=1,j=1}^{H_d,W_d} (l_{ij} log(x_{ij}) + (1 - l_{ij}) log(1 - x_{ij})) \tag{6.7}$$

where $H_d = H/8$ and $W_d = W/8$. X_o is the heatmap generated by the keypoint detector branch. L_o indicates the ground truth for keypoints. To improve the training efficiency, M_p positive pairs and M_n negative pairs of descriptor cells are sampled from $(H_d W_d)^2$ of positive and negative pairs to train the model. Variables with $'$ indicate that these variables are extracted from the transformed image. The encoded descriptor tensor D is generated from the original image. u_{ij} represent the centre of the descriptor vector d_{ij}. The correspondence of the descriptor pair $\left\{d_{ij}, d'_{i'j'}\right\}$ can be calculated through:

$$c_{iji'j'} = \begin{cases} 1, ||H(u_{ij}) - u'_{i'j'}||_2 \leqslant 8 \\ 0, \qquad otherwise \end{cases} \tag{6.8}$$

With the positive margin m_p and negative margin m_n, a hinge loss for descriptor loss can be defined as:

$$\begin{aligned} \mathcal{L}_d(D, D'; H) = & \frac{1}{(H_d W_d)^2} \sum_{i=1,j=1}^{H_d,W_d} \sum_{i'=1,j'=1}^{H_d,W_d} (\lambda c_{iji'j'} max(0, m_p - d_{ij}^T d'_{i'j'}) + \\ & (1 - c_{iji'j'}) max(0, d_{ij}^T d'_{i'j'} - m_p)) \end{aligned} \tag{6.9}$$

Finally, the training loss could be represented by

$$\mathcal{L}_{joint} = \mathcal{L}_p(X_o, L_o) + \mathcal{L}_p(X'_o, L'_o) + \mathcal{L}_d(D, D'; H) \tag{6.10}$$

6.2.2 SLAM Implementation

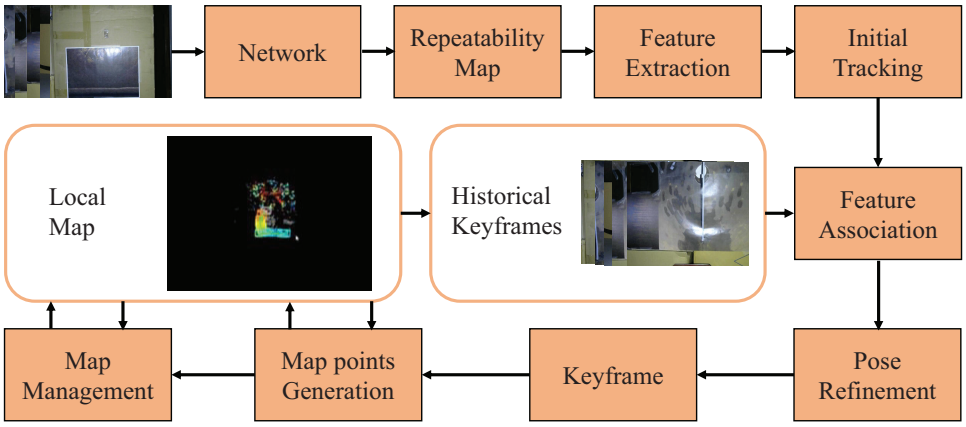


Figure 6.4: The scheme of the VO system (adapted from [24])

The proposed network is incorporated into the SP-ORB-SLAM [24], which leverages learned repeatability and description. The SuperPoint is substituted by the proposed network to improve the efficiency while preserving the accuracy of the entire VO system. Fig. 6.4 shows the overall structure of the VO method. The heatmap predicted by the keypoint detection branch is supposed to be the repeatability map, and it is sampled as 2D grids. After that, sparse features could be extracted. Furthermore, the camera pose is estimated directly based on the repeatability map and refined through the association of the sparse feature points with historical keyframes. Finally, the mapping module manages the sparse map.

The repeatable features can be identified across different images, and their locations are considered the local peaks of the repeatability map. The track-on-repeatability approach is adopted to localise the camera directly and coarsely. After that, the landmarks are associated with local feature points, and the camera pose is optimised by minimising the reprojection error. When a new keyframe is determined, the feature points observed by previous keyframes are mapped as the

map points. In the association step, the Approximate Nearest Neighbour (ANN) search and epipolar line search are adopted to associate feature points. Finally, the map maintenance process culls redundant keyframes and deletes outliers.

6.3 Experiments and analysis

To verify the performance of the proposed system, experiments including feature point extraction, localisation performance analysis and UAV flying tests are carried out. A laptop is utilised for training the model, which is equipped with the Intel Core i7-7700HQ CPU, 16 GB of memory and an external GPU enclosure connected via Thunderbolt 3. Specifically, the crate is equipped with an Nvidia Titan RTX GPU. The MS-COCO 2014 dataset [238], which consists of 80000 images, is utilised for training the proposed model, and images are resized to 240×320. The training is done using PyTorch 1.9.1. The model is optimised by the adam solver [239] with a learning rate of 0.0001. To improve the generalisation performance, data augmentation techniques such as random contrast and motion blur are adopted to enlarge the training dataset. The total training integration is 200000.

6.3.1 Datasets

HPatches [240] is a novel benchmark for local feature descriptor evaluation that contains 116 image scenes. There are 57 scenes that show large photometric changes, and the other 59 sequences have large viewpoint changes. Each sequence consists of 6 images and 5 ground-truth homographies between the first image and the others.

Similar to Chapter 5, 9 sequences are chosen from the EuRoC dataset to test the proposed method. More descriptions of these sequences can be found in Section 5.4.1

Unlike Chapter 5, which uses office room sequences with static, local variation, global variation and local and global variation lighting conditions in the ICL-NUIM dataset with simulated lighting changes, the scenes with flash lighting conditions are also utilised in this chapter.

Table 6.1: Feature extraction comparison

Feature extraction methods	Detector metric		Descriptor metric		Homography estimation		
	Rep.	MLE	MAP	MS	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$
ORB	0.64	1.03	0.51	0.18	0.14	0.40	0.49
SIFT	0.51	1.16	0.70	0.27	0.63	0.76	0.79
SuperPoint	0.61	1.14	0.81	0.55	0.44	0.77	0.83
GCNv2	0.64	1.14	0.78	0.44	0.45	0.73	0.81
deepFEPE	0.63	1.07	0.78	0.42	0.46	0.75	0.81
Proposed method	0.63	1.20	0.75	0.41	0.38	0.70	0.78

6.3.2 Evaluation on Feature Point Extraction and Description

To evaluate the feature point detection and matching ability of the proposed model, detector metrics including the repeatability (Rep.) and Mean Localisation Error (MLE), descriptor metrics consisting of mAP and Matching Score (MS) and the homography estimation metrics with different thresholds are measured on the HPatches dataset. The proposed model is compared with the deep learning-based feature extraction algorithms including SuperPoint [167], deepFEPE [240], GCNv2 [175], as well as traditional feature extraction methods such as ORB [241] and SIFT [242] with implemented by OpenCV. The results are presented in Table

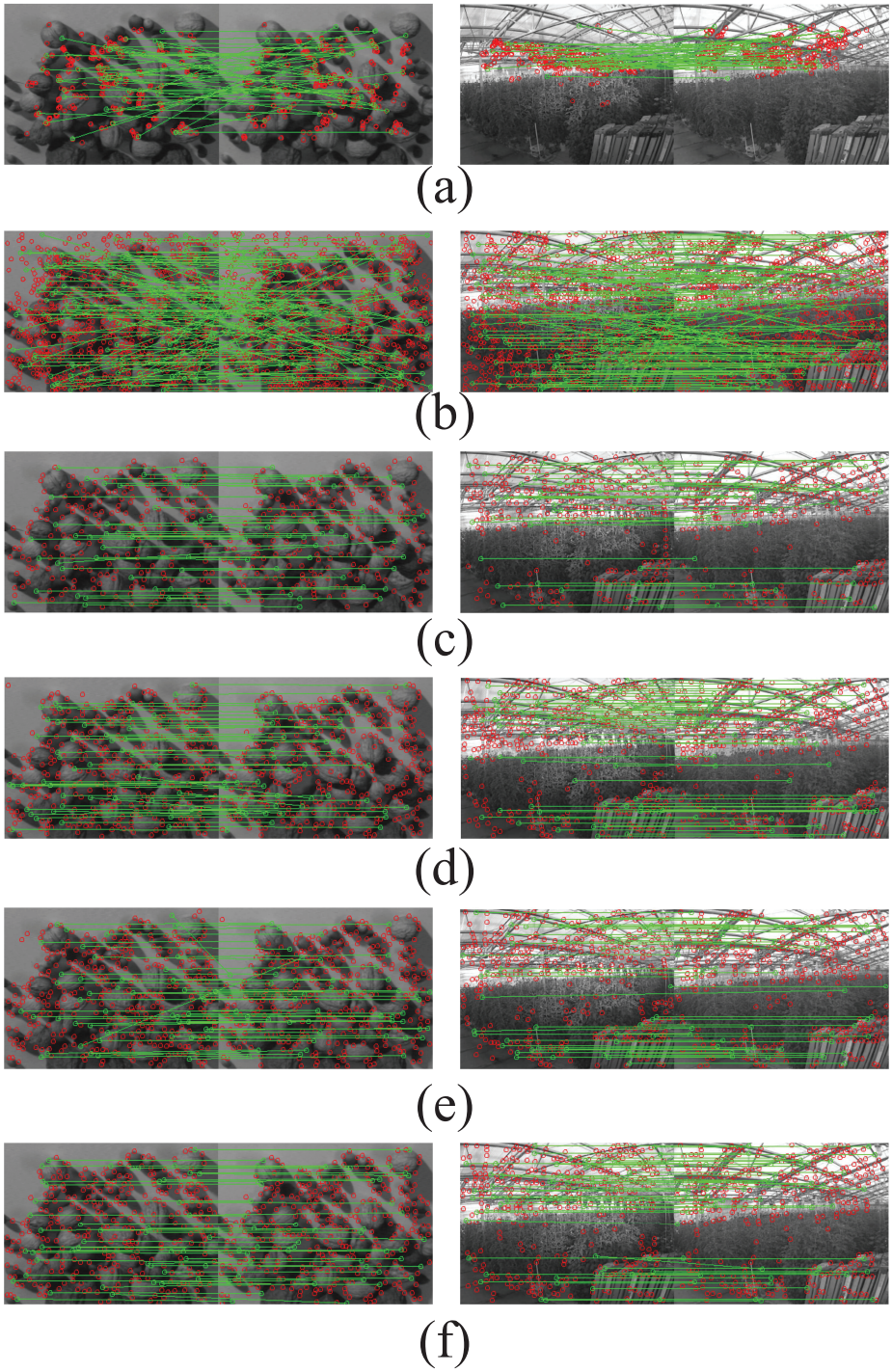


Figure 6.5: Feature extraction and matching: (a) ORB, (b) SIFT, (c) SuperPoint, (d) deepFEPE, (e) GCNv2, (f) the proposed model.

6.1. The visualised feature extraction and matching results are shown in Fig. 6.5. The red circles are feature points, and the green lines indicate the matches of feature points. The least number of feature points were extracted by the ORB, and a lot of mismatching exists. However, the ORB achieves the highest Rep., but scores for descriptor-focused metrics are the lowest. Therefore, it cannot perform well in the homography estimation task. The SIFT detects feature points with sub-pixel accuracy. Thereby, the greatest number of feature points are extracted, and the best performance in homography estimation with the $\epsilon = 1$ is achieved. Compared to traditional feature extraction methods, the learned descriptors outperform artificially designed representations. In the homography estimation, SuperPoint achieves the best score with a tolerance threshold of 3 and 5. Our method is slightly less accurate than other learned features, but still better than the ORB. When taking the model size and Floating Point Operations (FLOPs) into consideration, the comparison of learned features is shown in Fig. 6.6. Owing to the proposed efficient network structure, it can be seen that the proposed method contains fewer parameters. At the same time, the FLOPs is reduced significantly compared to other methods, which improves the model efficiency. Thus, it bridges the gap of high demand for GPU resources when deploying deep learning feature-based VO/VSLAM onto UAV onboard platforms.

6.3.3 Evaluation on Trajectory Estimation

To verify the localisation accuracy of the VO/VSLAM system, the ATE [229] is estimated for comparison. The ATE represents the absolute distance between the true trajectory and the calculated path. The comparisons of the proposed

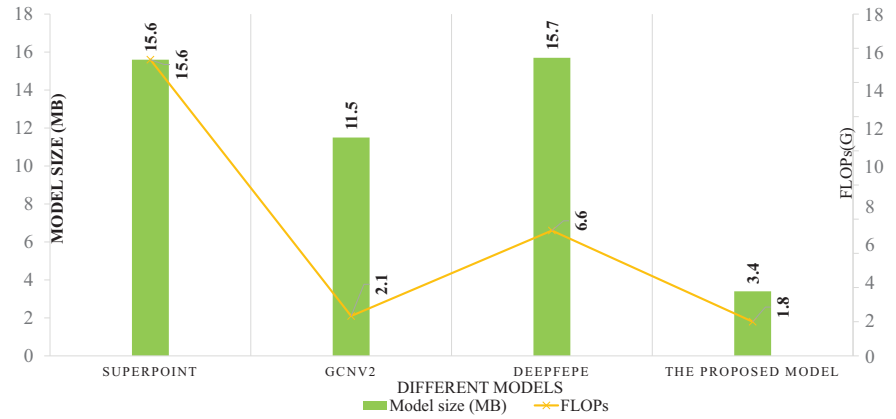


Figure 6.6: Model size and FLOPs comparison of different feature extraction methods

algorithm against the SP-ORB-SLAM, orb-slam3 and GCNv2-slam are carried out on the EuRoC dataset and the ICL-NUIM dataset with simulated lighting changes. As this work focus on the usage of the UAV onboard platform, no loop closure and relocalisation are allowed. Once they are activated, the result will be treated as a failure.

6.3.3.1 Evaluation on the EuRoC Dataset

The evaluation on the EuRoC dataset is carried out on a laptop with an Intel Core i7-8750H CPU, 20GB memory and a GeForce GTX 1050 Ti 4GB graphics card. The median value of the localisation results for each method from 10 runs is presented in table 6.2. Due to the GCNv2-SLAM struggling to keep the scale consistent for the monocular sequences, it fails in all scenarios. The ORB-SLAM3 achieves the best localisation accuracy in several sequences. However, it cannot finish the sequences V102 and V202 due to the fast motion and relative textureless environment. The SP-ORB-SLAM achieves the best average localisation accuracy. The proposed method finishes all sequences and obtains the sub-optimal average localisation accuracy.

Table 6.2: Performance comparison on the EuRoC dataset for the mean ATE (m) and RMS ATE (m). The best results are highlighted in a bold font.

	ORB-SLAM3		SP-ORB-SLAM		GCNv2-SLAM		The proposed method	
EuRoC benchmark	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE
MH01	0.020	0.022	**0.011**	**0.012**	-	-	0.014	0.017
MH02	0.016	0.018	**0.011**	**0.012**	-	-	0.012	0.014
MH03	0.027	0.031	**0.023**	0.027	-	-	**0.023**	**0.026**
MH04	**0.074**	**0.081**	0.091	0.102	-	-	0.087	0.098
MH05	**0.036**	**0.041**	0.042	0.047	-	-	0.050	0.058
V101	**0.031**	**0.033**	0.032	0.034	-	-	0.035	0.038
V102	-	-	0.229	0.248	-	-	**0.201**	**0.215**
V201	0.026	**0.023**	**0.022**	0.024	-	-	0.027	0.036
V202	**-**	**-**	**0.035**	**0.052**	-	-	0.104	0.127
Average	**0.033***	**0.036***	0.055	0.062	-	-	0.061	0.070

6.3.3.2 Evaluation on the ICL-NUIM Dataset with Simulated Lighting Changes

Table 6.3: Performance comparison on the ICL-NUIM dataset with simulated lighting changes for the mean ATE (m) and RMS ATE (m). The best results are highlighted in a bold font.

	ORB-SLAM3		SP-ORB-SLAM		GCNv2-SLAM		The proposed method	
ICL-NUIM benchmark	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE
Syn1	0.335*	0.779*	0.053	0.062	-	-	**0.041**	**0.046**
Syn1-local	0.479*	0.545*	**0.040**	**0.046**	-	-	0.043	**0.046**
Syn1-global	0.128*	0.148*	0.059	0.066	-	-	**0.048**	**0.054**
Syn1-local-global	0.271*	0.330*	**0.028**	**0.035**	-	-	**0.028**	0.042
Syn1-flash	-	-	0.128*	0.141*	-	-	**0.102**	**0.112**
Syn1-average	0.303*	0.451*	0.062	0.070	-	-	**0.052**	**0.060**

To obtain the results of the deep learning-based VO/VSLAM methods, a powerful GPU is preferred. Because of the payload and power constraints on the UAV onboard platform, the high-power GPU is unavailable. Thus, a small-sized Nvidia

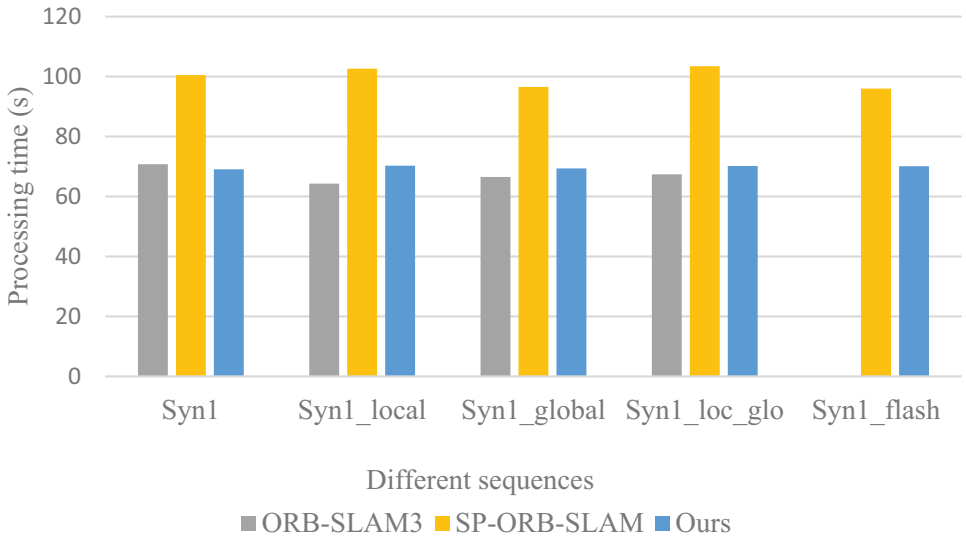


Figure 6.7: Time usage comparison

Jetson TX2, which is the most popular onboard platform for UAV, is utilised to verify the proposed methods. Considering the limited computing resources of Jetson TX2, all images in the ICL-NUIM dataset with simulated lighting changes are resized to 320×240. Similarly, the median localisation accuracy of successful trajectory estimation among 10 times of execution is presented in Table 6.3. Unlike the results on the powerful laptop, the proposed method obtains the best performance on the Jetson TX2. Due to less texture information being observed in the scenario with the flashlight, the ORB-SLAM3 fails to initialise and track the pose of the camera. The performance of the SP-ORB-SLAM is restricted due to the limited computing resources. Only the proposed algorithm could finish all of the sequences in 10 times of execution, and the robustness of the proposed method is presented in Fig. 6.8. Different colour squares represent the RMS ATE obtained in each of the 10 executions. It shows that the proposed algorithm could achieve robust localisation performance with a maximum deviation of 0.3m. Comparisons for time usage are depicted in Fig. 6.7. The ORB-SLAM3

is the most efficient method. However, it cannot achieve robust and accurate localisation performance in this scenario. Compared to the SP-ORB-SLAM, the efficiency of the proposed method is improved by 30.03% with the total power consumption decreased by 37.31%.

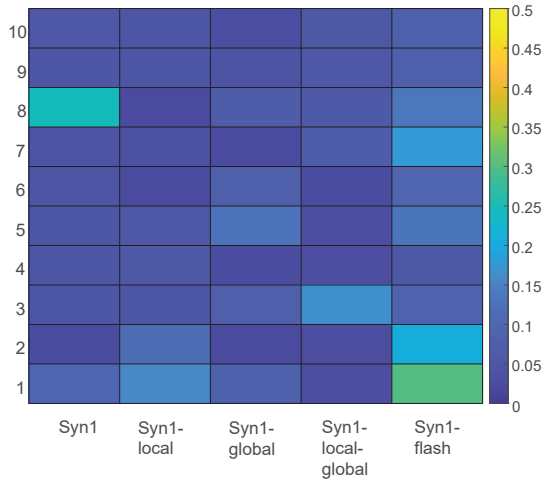


Figure 6.8: Robustness evaluation of the proposed method

6.3.3.3 Evaluation on the Real-world Sequence

To further verify the effectiveness of the proposed method, an analysis of the performances of different VO/VSLAM methods in a real-world scenario is provided. Fig. 6.9 (a) and (b) show the top view and front view of the quadrotor to capture image sequences, and the overall environment is shown in Fig. 6.9 (c). The quadrotor is based on the PX4 autopilot, and the Nvidia Jetson TX2 is used as the onboard computer. The detailed hardware configurations can be found in Appendix A. Some boxes and images captured in a pressure vessel are utilised to set up the environment. To make the environment more challenging, the lights are turned off. An OLIGHT Baton 3 is adopted to provide light for the environment. The Logitech C270 is used as the image input sensor. Considering

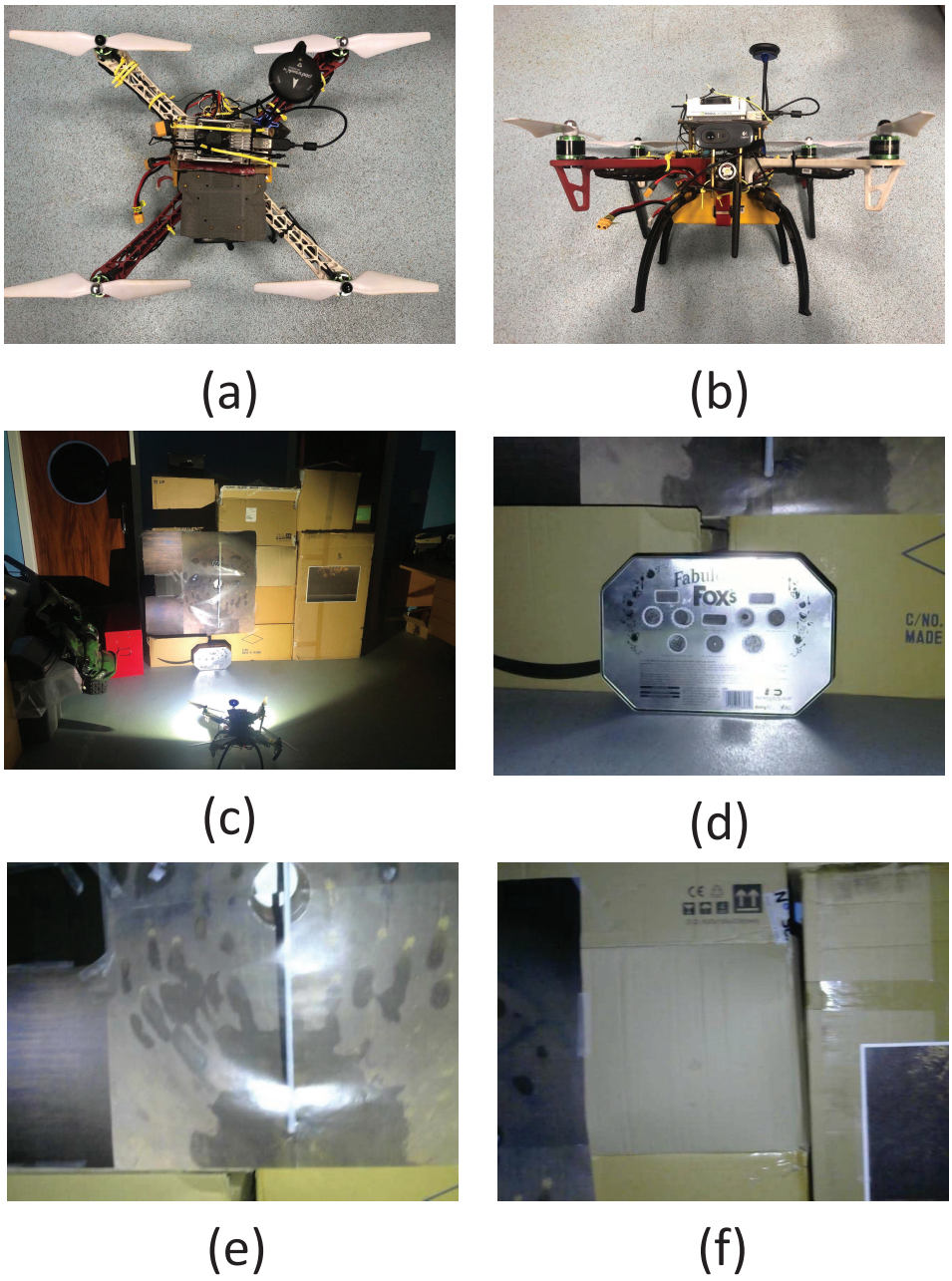


Figure 6.9: Experimental environment setup. (a) The top view of the quadrotor, (b) the front view of the quadrotor, (c) the experimental environment, (d) a cookie box for VO initialisation, (e) and (f) two sample images captured by the onboard camera.

the limited computing resources of the Nvidia Jetson TX2, the quadrotor is held in hand to record the image sequence. Some recorded images are shown in Fig.

6.9 (d), (e) and (f). The cookie box (shown in Fig. 6.9 (d)) contains rich texture information, and it is used to initialise the SLAM system. The image shown in Fig. 6.9 (e) contains motion blur and reflections. Fig. 6.9 (f) indicates the textureless environment. The Jetson TX2 is used to verify different VO/VSLAM methods. GCNv2 still cannot track all the image frames in this image sequence. The ORB-SLAM3 cannot keep the scale during the whole process, especially in the textureless region. The SP-ORB-SLAM fails to initialise the system. The proposed system could estimate the trajectory, and the trajectory can be seen in Fig. 6.10.

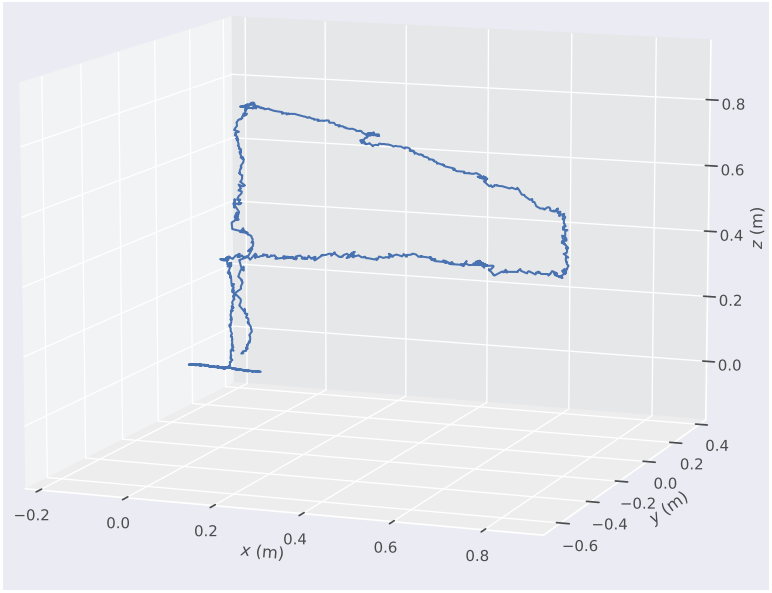


Figure 6.10: VO trajectory estimation

6.3.4 UAV Flying Tests

Finally, the flying test with the quadrotor and the environment introduced in Section 6.3.3 is carried out. Besides the ROS, the UDP is adopted to transmit position information through the VO system to the ROS topic. More details

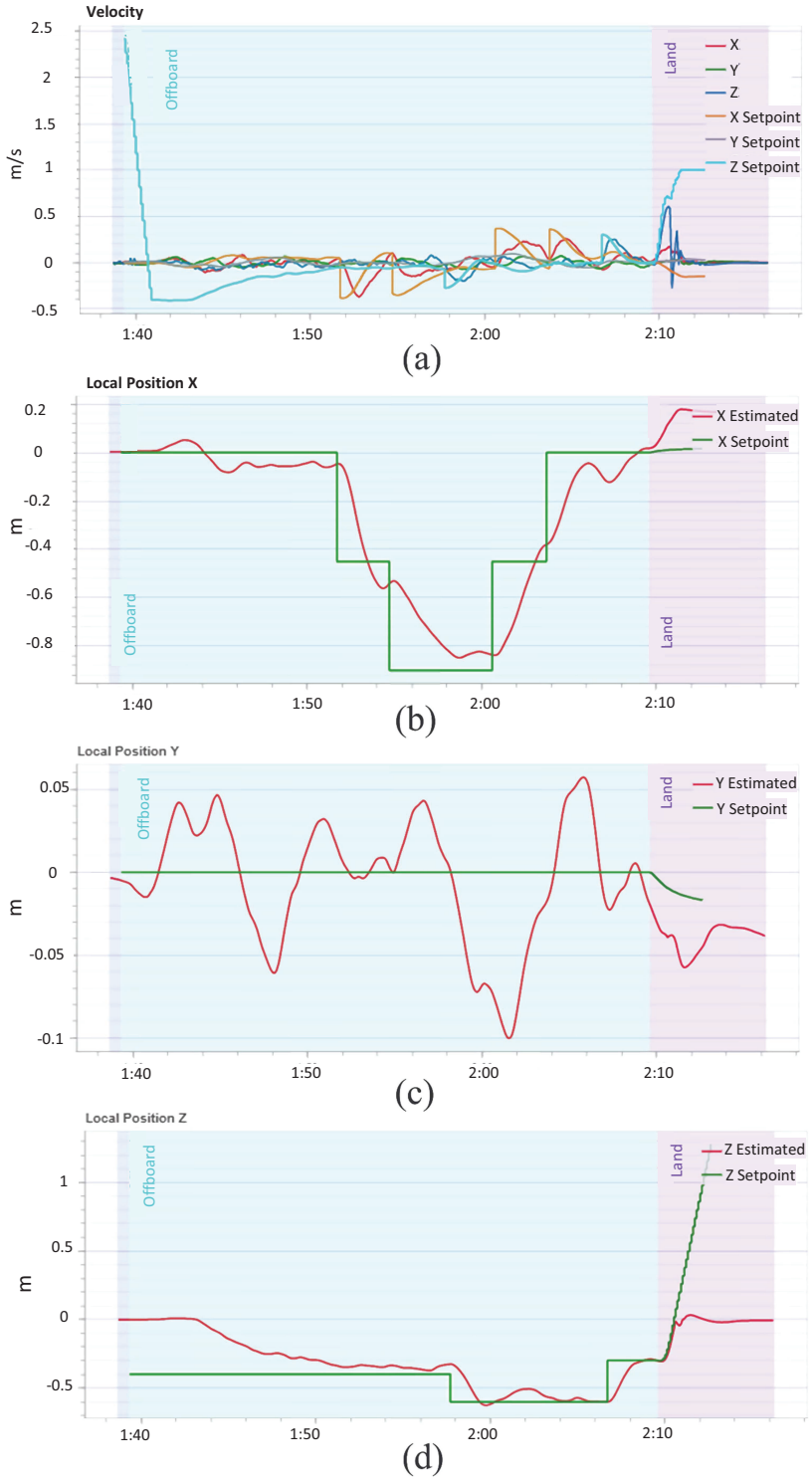


Figure 6.11: UAV velocity and position estimation

can be found in Appendix B. Through the conclusion obtained in the prevision, only the proposed methods can provide position information for the quadrotor in this challenging environment. Thus, only the proposed method is verified in this flying test. The scale of the monocular VO is obtained from the PX4 autopilot. During the experiments, the UAV is hold in hand to initialise the VO first. After that the UAV is capable of following the desired path in this low-illumination and textureless environment. More detailed results can be found at the supplementary video[1]. Fig. 6.11 displays the velocity and position estimation results over time in the trajectory following process. In summary, the experimental results further attested that the proposed method is capable of locating the UAV in a challenging indoor environment.

6.4 Summary

In this chapter, a more robust VO approach than the AFE-ORB-SLAM introduced in Chapter 5 was proposed. To extract robust feature points from the textureless environments with challenging lighting conditions, the deep learning-based feature extraction method was adopted and optimised. As this work focused on the UAV-based autonomous visual inspection system, the DSconv is implemented as the main convolution operation to reduce parameters. To improve the feature extraction capabilities, the improved DFconv is utilised. In particular, the DSconv instead of the traditional convolution is implemented to calculate the offset for the DFconv layers. In addition, the ELU is utilised as the activation function to keep the training process stable with a small number of calculations. After that, as described in Section 6.2.1.4, the whole network

[1] https://youtu.be/rV-iYm6m_Co

was trained in a self-supervision manner. Finally, the proposed network was embedded into the SP-ORB-SLAM to provide efficient but robust feature points.

Numerous experiments were conducted to validate the performances of the proposed feature extractor network and the improved SP-ORB-SLAM. Although the developed feature extraction network could not achieve the best performance as shown in Table 6.5. The proposed network still outperforms ORB in homography estimation. Compared to other learned methods, the model size and FLOPs were reduced significantly.

For the improved VO system, the performances of the localisation were tested and compared with other state-of-the-art VO/VSLAM systems. Results indicated that the improved VO system was able to be deployed in challenging environments where the ORB-SLAM3 failed, and the processing time was similar to the ORB-SLAM3. When the computing resources were restricted, such as on the Nvidia Jetson TX2, the proposed approach obtained the best results in most sequences. Furthermore, it still handled the scenario where the ORB-SLAM3 failed. As shown in Fig. 6.10 and Fig. 6.11, the UAV could localise itself and track the predefined trajectory in the textureless and challenging lighting environment, where both the traditional and deep learning-involved VO/VSLAM failed.

In summary, through the adoption of deep learning-based feature point extraction with a lightweight model, the VO/VSLAM could extract sufficient and robust feature points and locate the UAV with the UAV onboard computing resources in the low-illumination and textureless environment. Thereby, the challenge of extracting sufficient feature points in low-contrast and textureless environments for the UAV onboard VO/VSLAM system was addressed.

Chapter 7

Conclusion and Future Work

The research presented in this thesis aims to investigate how the task of UAV-based autonomous visual inspection in complex environments can be better supported. Specifically, the contributions to knowledge are:

- The feasibility of deploying the UAV with VSLAM to achieve autonomous visual inspection in confined and low-illumination indoor environments has been proven through a customised simulation environment for the first time.
- The issue of high demand for computing resources when deploying deep learning-based corrosion detection on the UAV onboard computer caused by extensive usage of traditional convolution layers is addressed through lightweight model design. It is achieved through lightweight convolution utilising DSconv, innovative feature extraction and fusion techniques leveraging the CBAM and the proposed improved SPP, refined detection strategies incorporating three-scale detections, and an optimised learning approach using the focal loss.

- The challenge of extracting sufficient feature points in low-contrast environments for the VSLAM system is addressed through image contrast-based adaptive FAST threshold and image contrast enhancement from the perspectives of image contrast and sharpening.
- The challenge of extracting sufficient feature points in low-contrast and textureless environments for the UAV onboard VO/VSLAM system is addressed through the adoption of the deep learning-based feature point extraction method with the lightweight model. The advancement is achieved by incorporating DSconv and the DFconv, whose kernel offsets are calculated through DSconv, to extract feature points in the challenging environment.

In this chapter, the research approach utilised in this thesis is summarised in Section 7.1. Detailed work and contributions are discussed in Section 7.2. The limitations of this work and future work are presented in Section 7.3.

7.1 Research Approach

As discussed in Section 1.2, positivism was adopted as the worldview for this research to answer the research question. Thus, the research methodology utilised in this research was the quantitative method, and the data were collected and analysed through the literature review, simulation and laboratory experiments.

7.2 Summary of the Thesis

7.2.1 Review of Related Work

Existing research works towards inspection methods and robotic platforms were presented in Chapter 2 (*objectives 1(a)-(c)*). The review of these works indicates that adopting the autonomous UAV for visual inspection tasks was able to reduce labour costs and improve efficiency. To realise an autonomous UAV system, the robust localisation, controller, path planning and corrosion detection systems served as the main components. However, there is no end-to-end autonomous UAV visual inspection system, primarily because of the challenges posed by confined and low-illumination environments. Additionally, the UAVs utilised in such settings have limited payload capacities. Thus, the first research question was derived: *Is it feasible to deploy UAVs equipped with VSLAM technology into confined and low-illumination environments for autonomous visual inspection tasks?*

The challenges and research gaps in the specific modules should be further identified. In this thesis, the research domains focused on accurate corrosion detection and robust UAV localisation in complex environments. Then, a comprehensive review of corrosion detection methods, including the traditional corrosion detector and deep learning-inspired corrosion detection methods, was presented. Towards detection accuracy, deep learning-based corrosion detectors outperform traditional corrosion detection methods due to their advanced feature extraction capability. However, these models cannot be deployed on the UAV platform due to the high demand for computing resources. Thus, the second research question was derived: *How to tackle the challenge of high demand for computing resources*

in deploying the deep learning-based detector on the UAV onboard computer to achieve real-time and accurate metallic corrosion detection?

After that, a comprehensive literature review of feature-based VO/VSLAM algorithms in textureless and challenging lighting environments was carried out. The literature review indicated that adopting image processing technology into the VO/VSLAM framework is a solution to improve the quality of feature points used by VO/VSLAM systems. However, some studies customised the dataset to show the high localisation accuracy of the VO/VSLAM systems. Thus, the generalisation of the image enhancement enhanced VO/VSLAM system in complex lighting environments, such as dark or over-bright environments, still needs to be addressed. Combining different kinds of features, especially by substituting feature extraction methods with deep learning-based methods, has proven the capability to improve the accuracy and robustness of VO/VSLAM in complex environments. However, there is still a gap in adopting deep learning-based feature points to UAV onboard platforms to cope with challenging environments, due to the high demand for desktop-level GPUs. Thereby, the third research question is derived: *How to address the issue of poor performance in UAV onboard VO/VSLAM systems in complex environments, particularly in scenarios under low-light or overly bright conditions, as well as textureless conditions where a sufficient number of feature points cannot be extracted?*

To answer the above research question, research objectives were identified in Section 1.1.2.

7.2.2 Summary of Conducted Studies

Chapter 3 aimed to answer research question 1. To verify the feasibility of the VSLAM-based autonomous UAV visual inspection system for pressure vessel inspection, a deeply customised ROS-PX4-Gazebo simulation environment, which contains a pressure vessel and a quadrotor, was developed to mimic the practical UAV-based pressure vessel inspection scenario (*objective 2(a)*). The ORB-SLAM3 could not initialise due to insufficient feature points that could be extracted in the low-illumination environment. Through the adoption of the adaptive gamma correction algorithm with weighting distribution as the preprocessing process to handle low-contrast images, sufficient feature points could be obtained. Thereby, the improved ORB-SLAM3 could be used to locate the position of the UAV. A trajectory, which consists of a square path and a helical path, was designed to inspect the inside of the pressure vessel. The pose calculated by the improved ORB-SLAM3 was compared with the planned inspection trajectory to determine the next target point. Then, a P-PID controller was adopted to control the UAV to track targeted waypoints. Experimental results showed the VSLAM-enabled autonomous UAV system could be utilised for autonomous indoor visual inspection tasks (*objective 2(b)*).

Chapter 4 focused on answering research question 2. To address the issue of high demand for computing resources when deploying deep learning-based corrosion detection on the UAV onboard computer caused by extensive usage of traditional convolution layers, the DSconv layers were employed as the main layers. Moreover, the SPP has been improved to output fused feature maps that contain more useful information for corrosion detection. At the same time, the CBAM attention module, focal loss and three-level target detection were also adopted to

achieve the required corrosion detection accuracy (*objective 3(a)*). Experimental results showed that the proposed corrosion detector achieves 84.96% mAP for multiple corrosion detection in complex environments, while the required mAP is 83.5%. In addition, it achieved 20.18 FPS on the Nvidia Jetson TX2, which meets the required 20 FPS (*objective 3(b)*).

The research question 3 was answered in Chapters 5 and 6. Chapter 5 investigated the traditional VSLAM system to address the issue of poor feature point extraction performance caused by different illumination environments. Considering the excellent performance of ORB-SLAM3, it was chosen as the framework used in this chapter. The truncated AGC was improved by the combination of unsharp masking to enhance the image in both contrast and sharpness. Then, the improved image enhancement approach was utilised for the pre-processing of the ORB-SLAM3. To further improve the performance of the developed VSLAM system, the image contrast-based adaptive FAST threshold was also proposed and adopted to the ORB-SLAM3 to extract more robust feature points (*objective 4(a)*). The ICL-NUIM dataset with simulated lighting changes, the OIVIO dataset and the EuRoC dataset were utilised to verify the performance of the improved VSLAM system, and experimental results indicated that the improved VSLAM system outperformed other cutting-edge monocular VSLAM methods in most scenarios regarding localisation accuracy (*objective 4(b)*). However, it still failed in textureless and low-light environments.

Thus, the issue of extracting sufficient feature points in low-contrast and textureless environments for the UAV onboard VO/VSLAM system was addressed in Chapter 6. Specifically, the deep learning-based feature extractor with a lightweight model has been developed and adopted to the VO system. Finally, a

robust deep learning-based monocular VO system has been presented to achieve robust localisation performance in textureless and low-light environments. Similar to the efficient and accurate corrosion detector, the DSconv was still chosen as the main convolutional operation. Meanwhile, DSconv was also applied to the deformable convolution to calculate the offsets, and the improved deformable convolution was utilised to extract more representative feature points in complex environments (*objective 5(a)*). With the adoption of the proposed deep learning-based feature extraction network, the performance of the VO system has been significantly improved, and it has handled scenarios where other VO/VSLAM systems failed. Extensive experiments, including the public datasets (EuRoC and ICL-NUIM datasets with simulated lighting changes), the recorded real-world sequences and the flying test, were carried out to assess the performance of the proposed method. Experimental results confirmed that the proposed monocular VO approach outperformed other traditional and deep learning feature-based VO/VSLAM systems on the Nvidia Jetson TX2, which was utilised as the UAV onboard controller in this thesis. Moreover, due to the improvement in computing efficiency, the proposed VO system allowed the UAV to track the planned trajectory in textureless and low-illumination environments where both the traditional and deep learning feature-based VO and VSLAM systems fail (*objective 5(b)*).

7.3 Future Work

Due to the limited time, there are several limitations to the current work (*objective (6)*). The key techniques developed in this thesis are verified by the datasets and laboratory-based mock environments. Even though the datasets are collected in the real world, the noise and vibration caused by the UAV utilised in this

project are not taken into consideration. The laboratory-based experimental environment is a controlled environment, and it is not as challenging as the real inspection scenario. Thereby, to further improve the robustness and efficiency of the UAV-based autonomous visual inspection system, the following work can be carried out:

Lots of the hyperparameters used in CNN training are set empirically or derived from the literature. Hyperparameter optimisation techniques can be employed to identify the optimal parameters for the developed models and achieve their optimal performance. Furthermore, the training stability of the CNN model should also be investigated to ensure robust training processes with minimal noise.

The accuracy and robustness of VO/VSLAM systems need to be further improved. The developed VO approach does not allow the fast movement of the UAV. The novel model compression techniques that compress deep learning models for efficient deployment without scarifying too much predictive performance could be adopted. The common practices for model compression are pruning, quantisation and knowledge distillation. Besides improving the performance of the pure VO/VSLAM algorithm, the sensor fusion methods could also be taken into consideration. Considering the difficulty of sensor deployment in challenging environments and the limited payload capability of the UAV, the IMU would be the ideal sensor to be merged with the VO/VSLAM. When the VO/VSLAM fails due to the fast changes of the visual feature, the UAV could also localise itself through the IMU. Once the VO/VSLAM works again, it will take over the localisation process to reduce cumulative errors.

The UAV tracks the predefined path to carry out inspection tasks. Hence, coverage path planning could also be considered. The coverage path planning not only guarantees the full coverage inspection of the facilities but also assists in the dense 3D model reconstruction of the facilities. In addition, the coverage path planer can be treated as the global planner to work with a local planner. Finally, the UAV will have the capability to avoid dynamic obstacles that can further improve the safety of the inspection procedure.

For the UAV-based close visual inspection tasks, a robust or intelligent controller to navigate the UAV in complex environments efficiently is essential. Therefore, investigations into the flight controller would make the inspection process safer and more efficient.

Bibliography

[1] Halo Robotics, "Inspeksi drone di dalam gas pressure vessel dengan flyability elios 2," https://www.youtube.com/watch?v=B9VVsbv6dWE, 2019.

[2] C. K. Tembo and A. Akintola, "A retrospection of methodological pluralism in the journal of financial management of property and construction (2005-2020)," *Journal of Financial Management of Property and Construction*, vol. 27, no. 3, pp. 348–364, 2022.

[3] Eddyfi Technologies, "Scorpion2 battery-powered, robotic crawler," https://www.eddyfi.com/en/product/scorpion-2, 2022.

[4] InnoTecUK, "Icm rover advanced vacuum adhesion inspection crawler," http://www.alldesignonline.com/innotecuknew/wp-content/uploads/2018/05/InnoTecUK-ICM-Rover-Overview-Brochure-LR.pdf, 2022.

[5] A. Sintov, T. Avramovich, and A. Shapiro, "Design and motion planning of an autonomous climbing robot with claws," *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 1008–1019, 2011.

[6] E. W. Hawkes, E. V. Eason, A. T. Asbeck, and M. R. Cutkosky, "The gecko's toe: Scaling directional adhesives for climbing applications,"

IEEE/ASME transactions on mechatronics, vol. 18, no. 2, pp. 518–526, 2012.

[7] A. A. Mazreah, F. B. I. Alnaimi, and K. S. M. Sahari, "Novel design for pig to eliminate the effect of hydraulic transients in oil and gas pipelines," *Journal of Petroleum Science and Engineering*, vol. 156, pp. 250–257, 2017.

[8] A. Kakogawa and S. Ma, "Design of a multilink-articulated wheeled pipeline inspection robot using only passive elastic joints," *Advanced Robotics*, vol. 32, no. 1, pp. 37–50, 2018.

[9] Y.-S. Kwon and B.-J. Yi, "Design and motion planning of a two-module collaborative indoor pipeline inspection robot," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 681–696, 2012.

[10] A. A. Gargade and S. S. Ohol, "Development of actively steerable in-pipe inspection robot for various sizes," in *Proceedings of the Advances in Robotics*, 2017, pp. 1–5.

[11] C. Nițu, B. Gramescu, A. S. Hashim, and M. Avram, "Inchworm locomotion of an external pipe inspection and monitoring robot," in *International Conference on Innovation, Engineering and Entrepreneurship*. Springer, 2018, pp. 464–470.

[12] F. Trebuňa, I. Virgala, M. Pástor, T. Lipták, and L. Miková, "An inspection of pipe by snake robot," *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, p. 1729881416663668, 2016.

[13] H. Tourajizadeh, M. Rezaei, and A. Sedigh, "Optimal control of screw in-pipe inspection robot with controllable pitch rate," *Journal of Intelligent & Robotic Systems*, vol. 90, no. 3, pp. 269–286, 2018.

[14] S. A. Idris, F. A. Jafar, and N. Abdullah, "Study on corrosion features analysis for visual inspection & monitoring system: A ndt technique," *Proceedings of Mechanical Engineering Research Day*, vol. 2015, pp. 111–112, 2015.

[15] F. Bonnín-Pascual and A. Ortiz, "Detection of cracks and corrosion for automated vessels visual inspection." in *CCIA*, 2010, pp. 111–120.

[16] M. C. Pereira, J. W. Silva, H. A. Acciari, E. N. Codaro, and L. R. Hein, "Morphology characterization and kinetics evaluation of pitting corrosion of commercially pure aluminium by digital image analysis," 2012.

[17] N.-D. Hoang and V.-D. Tran, "Image processing-based detection of pipe corrosion using texture analysis and metaheuristic-optimized machine learning approach," *Computational Intelligence and Neuroscience*, vol. 2019, 2019.

[18] X. Gao and T. Zhang, *Introduction to visual SLAM: from theory to practice*. Springer Nature, 2021.

[19] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "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, 2021.

[20] P. Charles *et al.*, "Digital video and hdtv algorithms and interfaces," *Morgan Kaufmann Publishers, San Francisco*, vol. 260, p. 630, 2003.

[21] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[23] G. Cao, L. Huang, H. Tian, X. Huang, Y. Wang, and R. Zhi, "Contrast enhancement of brightness-distorted images by improved adaptive gamma correction," *Computers & Electrical Engineering*, vol. 66, pp. 569–582, 2018.

[24] H. Huang, H. Ye, Y. Sun, and M. Liu, "Monocular visual odometry using learned repeatability and description," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8913–8919.

[25] Mordor Intelligence, "Non-destructive testing (ndt) market - growth, trends, covid-19 impact, and forecasts (2022 - 2027)," https://www.mordorintelligence.com/industry-reports/global-non-destructive-testing-market-industry#, 2021.

[26] bp, "Energy outlook 2022," https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/energy-outlook/bp-energy-outlook-2022.pdf, 2022.

[27] A. N. Moosavi, "Corrosion in onshore production and transmission sectors—current knowledge and challenges," *Trends in Oil and Gas Corrosion Research and Technologies*, pp. 95–109, 2017.

[28] A. Alharam, E. Almansoori, W. Elmadeny, and H. Alnoiami, "Real time ai-based pipeline inspection using drone for oil and gas industries in bahrain," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*. IEEE, 2020, pp. 1–5.

[29] A. Shukla and H. Karki, "Application of robotics in offshore oil and gas industry—a review part ii," *Robotics and Autonomous Systems*, vol. 75, pp. 508–524, 2016.

[30] Research and Markets, "Inspection robots market size, market share, application analysis, regional outlook, growth trends, key players, competitive strategies and forecasts, 2021 to 2029," https://www.researchandmarkets.com/reports/5552843/inspection-robots-market-size-market-share#tag-pos-1, 2021.

[31] Q. Chen, X. Wen, S. Lu, and D. Sun, "Corrosion detection for large steel structure base on uav integrated with image processing system," in *IOP Conference Series: Materials Science and Engineering*, vol. 608, no. 1. IOP Publishing, 2019, p. 012020.

[32] A. Shukla, H. Xiaoqian, and H. Karki, "Autonomous tracking and navigation controller for an unmanned aerial vehicle based on visual data for inspection of oil and gas pipelines," in *2016 16th international conference on control, automation and systems (ICCAS)*. IEEE, 2016, pp. 194–200.

[33] V. Sudevan, A. Shukla, and H. Karki, "Inspection of vertical structures in oil and gas industry: A review of current scenario and future trends," in *Research and Development Petroleum Conference and Exhibition 2018*, vol. 2018, no. 1. European Association of Geoscientists & Engineers, 2018, pp. 65–68.

[34] C. Ma, J. Zou, D. Lei, X. Ye, S. Zang, and J. Ma, "Crack inspection of wooden poles based on unmanned aerial vehicles," in *2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*. IEEE, 2022, pp. 290–294.

[35] Survey report, "2023 state of visual inspection," https://percepto.co/wp-content/uploads/2023/03/2023-State-of-Visual-Inspection-Market-Survey.pdf, 2023.

[36] B. T. Bastian, N. Jaspreeth, S. K. Ranjith, and C. Jiji, "Visual inspection and characterization of external corrosion in pipelines using deep neural network," *NDT & E International*, vol. 107, p. 102134, 2019.

[37] D. J. Atha and M. R. Jahanshahi, "Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection," *Structural Health Monitoring*, vol. 17, no. 5, pp. 1110–1128, 2018.

[38] A. Albanese, M. Nardello, and D. Brunelli, "Low-power deep learning edge computing platform for resource constrained lightweight compact uavs," *Sustainable Computing: Informatics and Systems*, vol. 34, p. 100725, 2022.

[39] S. Jiang, Y. Wu, and J. Zhang, "Bridge coating inspection based on two-stage automatic method and collision-tolerant unmanned aerial system," *Automation in Construction*, vol. 146, p. 104685, 2023.

[40] S. Jiang, Y. Cheng, and J. Zhang, "Vision-guided unmanned aerial system for rapid multiple-type damage detection and localization," *Structural Health Monitoring*, vol. 22, no. 1, pp. 319–337, 2023.

[41] S. Dorafshan and M. Maguire, "Bridge inspection: Human performance, unmanned aerial systems and automation," *Journal of Civil Structural Health Monitoring*, vol. 8, pp. 443–476, 2018.

[42] Y. Sun, W. Wang, L. Mottola, R. Wang, and Y. He, "Aim: Acoustic inertial measurement for indoor drone localization and tracking," in *Proceedings of*

the 20th ACM Conference on Embedded Networked Sensor Systems, 2022, pp. 476–488.

[43] A. Kulsinskas, P. Durdevic, and D. Ortiz-Arroyo, "Internal wind turbine blade inspections using uavs: Analysis and design issues," *Energies*, vol. 14, no. 2, p. 294, 2021.

[44] W. G. Aguilar, G. A. Rodríguez, L. Álvarez, S. Sandoval, F. Quisaguano, and A. Limaico, "Visual slam with a rgb-d camera on a quadrotor uav using on-board processing," in *Advances in Computational Intelligence: 14th International Work-Conference on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14-16, 2017, Proceedings, Part II 14*. Springer, 2017, pp. 596–606.

[45] M. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students*. Pearson education, 2009.

[46] Y. Reich, "Layered models of research methodologies," *AI EDAM*, vol. 8, no. 4, pp. 263–274, 1994.

[47] K. Williamson and G. Johanson, *Research methods: Information, systems, and contexts*. Chandos Publishing, 2017.

[48] A. Almami, "Investigating the antecedents and consequences of saudization in the construction sector," 2014.

[49] H. Nissen, H. Klein, and R. Hirschheim, "Choosing appropriate information systems research approaches: A revised taxonomy," 1991.

[50] D. M. Levin, *The opening of vision: Nihilism and the postmodern situation*. Routledge, 2008.

[51] J. Maksimovic and J. Evtimov, "Positivism and post-positivism as the basis of quantitative research in pedagogy," *Research in Pedagogy*, vol. 13, no. 1, pp. 208–218, 2023.

[52] C. Parsons, "Constructivism and interpretive theory," *Theory and methods in political science*, vol. 3, pp. 80–98, 2010.

[53] M. Greeff, "Interpretive research methods," *Improving health through nursing research*, vol. 129, 2009.

[54] D. Ortiz and J. Greene, "Research design: qualitative, quantitative, and mixed methods approaches," *Qualitative Research Journal*, vol. 6, no. 2, pp. 205–208, 2007.

[55] K. Dooley, "Simulation research methods," *The Blackwell companion to organizations*, pp. 829–848, 2017.

[56] S. E. Salterio and P. Mariestha, "Moving beyond the lab," *The Routledge Companion to Behavioural Accounting Research*, p. 149, 2017.

[57] C. R. Kothari, *Research methodology: Methods and techniques*. New Age International, 2004.

[58] P. Dugard, "Statistical applications for the behavioral sciences." 1994.

[59] D. Medak, L. Posilović, M. Subašić, M. Budimir, and S. Lončarić, "Defectdet: A deep learning architecture for detection of defects with extreme aspect ratios in ultrasonic images," *Neurocomputing*, vol. 473, pp. 107–115, 2022.

[60] M. Wall, "Human factors guidance to improve reliability of non-destructive testing in the offshore oil and gas industry," in *7th European-American Workshop on Reliability of NDE*, 2017.

[61] J. P. Steele, Q. Han, H. Karki, K. Al-Wahedi, A. A. Ayoade, M. R. Sweatt, D. P. Albert, and W. A. Yearsley, "Development of an oil and gas refinery inspection robot," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 46476. American Society of Mechanical Engineers, 2014, p. V04AT04A016.

[62] M. P. Bento, F. de Medeiros, I. C. de Paula Jr, and G. Ramalho, "Image processing techniques applied for corrosion damage analysis," in *Proceedings of the XXII Brazilian Symposium on Computer Graphics and Image Processing, Rio de Janeiro. RJ*, 2009.

[63] S. S. H. Hajjaj and I. B. Khalid, "Design and development of an inspection robot for oil and gas applications," 2018.

[64] S.-C. Her and S.-T. Lin, "Non-destructive evaluation of depth of surface cracks using ultrasonic frequency analysis," *Sensors*, vol. 14, no. 9, pp. 17 146–17 158, 2014.

[65] T. Wu, H. Wu, Y. Du, and Z. Peng, "Progress and trend of sensor technology for on-line oil monitoring," *Science China Technological Sciences*, vol. 56, pp. 2914–2926, 2013.

[66] P. J. Shull, "Introduction to nde," in *Nondestructive evaluation: theory, techniques, and applications*. CRC Press, 2016, pp. 1–15.

[67] A. D. Eslamlou, A. Ghaderiaram, E. Schlangen, and M. Fotouhi, "A review on non-destructive evaluation of construction materials and structures us-

ing magnetic sensors," *Construction and Building Materials*, vol. 397, p. 132460, 2023.

[68] R. Barnes and D. Atherton, "Effects of bending stresses on magnetic flux leakage patterns," *NDT & E International*, vol. 26, no. 1, pp. 3–6, 1993.

[69] Y. Wang, Y. Xu, B. Wang, S. Ding, J. Xu, and M. Zheng, "Research on metal atmospheric storage tank inspection method for standard in china," in *ASME Pressure Vessels and Piping Conference*, vol. 43642, 2009, pp. 447–452.

[70] Y. Shi, C. Zhang, R. Li, M. Cai, and G. Jia, "Theory and application of magnetic flux leakage pipeline detection," *Sensors*, vol. 15, no. 12, pp. 31 036–31 055, 2015.

[71] J. Hansen, "The eddy current inspection method," *Insight*, vol. 46, no. 5, pp. 279–281, 2004.

[72] C. Ye, Y. Huang, L. Udpa, and S. S. Udpa, "Differential sensor measurement with rotating current excitation for evaluating multilayer structures," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 782–789, 2015.

[73] M. Safizadeh and M. Hasanian, "Gas pipeline corrosion mapping using pulsed eddy current technique," *ADMT Journal*, vol. 5, no. 1, 2011.

[74] S. Bagavathiappan, B. B. Lahiri, T. Saravanan, J. Philip, and T. Jayakumar, "Infrared thermography for condition monitoring–a review," *Infrared Physics & Technology*, vol. 60, pp. 35–55, 2013.

[75] Z. Ali, Y. Addepalli, and Y. Zhao, "Through transmission thermography-a review of the state-of-the-art," 2022.

[76] S. Doshvarpassand, C. Wu, and X. Wang, "An overview of corrosion defect characterization using active infrared thermography," *Infrared physics & technology*, vol. 96, pp. 366–389, 2019.

[77] Y. He, B. Deng, H. Wang, L. Cheng, K. Zhou, S. Cai, and F. Ciampa, "Infrared machine vision and infrared thermography with deep learning: A review," *Infrared physics & technology*, vol. 116, p. 103754, 2021.

[78] A. Kylili, P. A. Fokaides, P. Christou, and S. A. Kalogirou, "Infrared thermography (irt) applications for building diagnostics: A review," *Applied Energy*, vol. 134, pp. 531–549, 2014.

[79] M. F. Silva, J. T. Machado, and J. K. Tar, "A survey of technologies for climbing robots adhesion to surfaces," in *2008 IEEE international conference on computational cybernetics*. IEEE, 2008, pp. 127–132.

[80] M. F. Silva, R. S. Barbosa, and A. L. Oliveira, "Climbing robot for ferromagnetic surfaces with dynamic adjustment of the adhesion system," *Journal of Robotics*, vol. 2012, 2012.

[81] Y. Guan, H. Zhu, W. Wu, X. Zhou, L. Jiang, C. Cai, L. Zhang, and H. Zhang, "A modular biped wall-climbing robot with high mobility and manipulating function," *IEEE/ASME transactions on mechatronics*, vol. 18, no. 6, pp. 1787–1798, 2012.

[82] Y. Du, Q.-m. Zhu, S. Ghauri, J.-h. Zhai, H.-r. Jia, and H. Nouri, "Progresses in study of pipeline robot," in *2012 Proceedings of International Conference on Modelling, Identification and Control*. IEEE, 2012, pp. 808–813.

[83] J. M. M. Tur and W. Garthwaite, "Robotic devices for water main in-pipe inspection: A survey," *Journal of Field Robotics*, vol. 4, no. 27, pp. 491–508, 2010.

[84] R. Bogue, "Robots in the offshore oil and gas industries: a review of recent developments," *Industrial Robot: the international journal of robotics research and application*, vol. 47, no. 1, pp. 1–6, 2020.

[85] M. Bengel, K. Pfeiffer, B. Graf, A. Bubeck, and A. Verl, "Mobile robots for offshore inspection and manipulation," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3317–3322.

[86] videoray, "Underwater remotely operated vehicles you can trust wherever your mission takes you," http://www.Videoray.com/, 2022.

[87] Nova Ray, "Subsea systems rov (remotely operated vehicle)," http://www.novaray.com/, 2022.

[88] Saab Seaeye, "Electric underwater robotics (rovs)," https://www.saabseaeye.com, 2022.

[89] Deep Ocean Engineering, "Underwater drones (rov, usv) for sale in california," https://www.deepocean.com/, 2022.

[90] F-e-t, "Forum energy technologies," https://www.f-e-t.com/, 2022.

[91] K. Sun, W. Cui, and C. Chen, "Review of underwater sensing technologies and applications," *Sensors*, vol. 21, no. 23, p. 7849, 2021.

[92] P. K. Paim, B. Jouvencel, and L. Lapierre, "A reactive control approach for pipeline inspection with an auv," in *Proceedings of OCEANS 2005 MTS/IEEE*. IEEE, 2005, pp. 201–206.

[93] J. Albiez, D. Cesar, C. Gaudig, S. Arnold, R. Cerqueira, T. Trocoli, G. Mimoso, R. Saback, and G. Neves, "Repeated close-distance visual inspections with an auv," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 2016, pp. 1–8.

[94] C. Wang and L. Cui, "The implementation of automatic inspection algorithm for underwater vehicles based on hough transform," in *2018 7th International Conference on Sustainable Energy and Environment Engineering (ICSEEE 2018)*. Atlantis Press, 2019, pp. 459–464.

[95] T. R. Wanasinghe, R. G. Gosine, O. De Silva, G. K. Mann, L. A. James, and P. Warrian, "Unmanned aerial systems for the oil and gas industry: Overview, applications, and challenges," *IEEE access*, vol. 8, pp. 166 980–166 997, 2020.

[96] Offshore Energy, "Drones halve the cost of inspection, cyberhawk says," https://www.offshore-energy.biz/drone-inspections-halve-the-cost-of-inspection-cyberhawk/, 2017.

[97] Intel, "Intel and cyberhawk inspect gas terminal through lens of commercial drone technology," https://newsroom.intel.com/news/intel-cyberhawk-inspect-gas-terminal-lens-commercial-drone-technology/, 2017.

[98] ——, "Intel® falcon™ 8+ system," https://www.intel.co.uk/content/www/uk/en/products/drones/falcon-8.html, 2022.

[99] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A uav system for inspection of industrial facilities," in *2013 IEEE Aerospace Conference*. IEEE, 2013, pp. 1–8.

[100] Flyability, "Elios inspectand explore indoors and confined spaces," https://www.flyability.com/elios/.

[101] L. Xu, C. Feng, V. R. Kamat, and C. C. Menassa, "An occupancy grid mapping enhanced visual slam for real-time locating applications in indoor gps-denied environments," *Automation in Construction*, vol. 104, pp. 230–245, 2019.

[102] S. A. Umoren, M. M. Solomon, and V. S. Saji, *Polymeric Materials in Corrosion Inhibition: Fundamentals and Applications*. Elsevier, 2022.

[103] M. G. Kadhim and M. T. Ali, "A critical review on corrosion and its prevention in the oilfield equipment," *Journal of petroleum research and studies*, vol. 7, no. 2, pp. 162–189, 2017.

[104] X. Huang, K. Zhou, Q. Ye, Z. Wang, L. Qiao, Y. Su, and Y. Yan, "Crevice corrosion behaviors of cocrmo alloy and stainless steel 316l artificial joint materials in physiological saline," *Corrosion Science*, vol. 197, p. 110075, 2022.

[105] T. Hoar and J. Agar, "Factors in throwing power illustrated by potential-current diagrams," *Discussions of the Faraday Society*, vol. 1, pp. 162–168, 1947.

[106] G. NirmalaDevi, R. Viswanath, G. Suresh, K. Shunmuganathan, T. Mathews, and T. Sampath Kumar, "Synthesis and microstructure influenced antimicrobial properties of dispersed nanoporous gold rods," *Transactions of the Indian Institute of Metals*, vol. 75, no. 10, pp. 2737–2747, 2022.

[107] A. Donald, "Encyclopedia of materials: Science and technology," *Guide-Wave Optical Communications: Materials, 2nd ed.; Elsevier: Amsterdam, The Netherlands*, pp. 3231–3233, 2001.

[108] R. Kumar, R. Kumar, and S. Kumar, "Erosion corrosion study of hvof sprayed thermal sprayed coating on boiler tubes: a review," *IJSMS*, vol. 1, no. 3, pp. 1–6, 2018.

[109] F. Bonnin-Pascual and A. Ortiz, "Corrosion detection for automated visual inspection," in *Developments in Corrosion Protection*. IntechOpen, 2014.

[110] N.-D. Hoang, "Image processing-based pitting corrosion detection using metaheuristic optimized multilevel image thresholding and machine-learning approaches," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[111] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *Journal of King Saud University-Computer and Information Sciences*, 2019.

[112] Z. Yue, F. Gao, Q. Xiong, J. Wang, T. Huang, E. Yang, and H. Zhou, "A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition," *Cognitive Computation*, vol. 13, no. 4, pp. 795–806, 2021.

[113] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of automatic feature extraction in deep learning architecture," in *2016 International conference on digital image computing: techniques and applications (DICTA)*. IEEE, 2016, pp. 1–8.

[114] J. Du, L. Yan, H. Wang, and Q. Huang, "Research on grounding grid corrosion classification method based on convolutional neural network," in *MATEC Web of Conferences*, vol. 160. EDP Sciences, 2018, p. 01008.

[115] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[116] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.

[117] J. Li, Z. Su, J. Geng, and Y. Yin, "Real-time detection of steel strip surface defects based on improved yolo detection network," *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 76–81, 2018.

[118] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[119] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[120] Y. Dai, R. Han, L. Liu, X. Jiang, S. Qian, Q. Hong, and S. Gao, "Research on substation equipment rust detection method based on improved ssd," in *2021 International Conference on Advanced Electrical Equipment and Reliable Operation (AEERO)*. IEEE, 2021, pp. 1–3.

[121] R. E. Andersen, L. Nalpantidis, and E. Boukas, "Vessel classification using a regression neural network approach," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4480–4486.

[122] R. Ali, D. Kang, G. Suh, and Y.-J. Cha, "Real-time multiple damage mapping using autonomous uav and deep faster region-based neural networks for gps-denied structures," *Automation in Construction*, vol. 130, p. 103831, 2021.

[123] X. Cheng and J. Yu, "Retinanet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[124] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[125] T. Li, L. Xing, H. Fan, and H. Zhu, "Surface defect detection of aluminum material based on deep learning," in *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*. IEEE, 2022, pp. 1375–1379.

[126] C. Luo, L. Yu, J. Yan, Z. Li, P. Ren, X. Bai, E. Yang, and Y. Liu, "Autonomous detection of damage to multiple steel surfaces from 360 panoramas using deep neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 12, pp. 1585–1599, 2021.

[127] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[128] NVIDIA DEVELOPER, "Jetson tx2 module," https://developer.nvidia.com/embedded/jetson-tx2, 2017.

[129] S. Jung and Y.-J. Kim, "Mils and hils analysis of power management system for uavs," *IEEE Access*, 2023.

[130] L. Sun, D. Adolfsson, M. Magnusson, H. Andreasson, I. Posner, and T. Duckett, "Localising faster: Efficient and precise lidar-based robot localisation in large-scale environments," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 4386–4392.

[131] B. W. Parkinson and J. J. Spilker, *Progress in astronautics and aeronautics: Global positioning system: Theory and applications*. Aiaa, 1996, vol. 164.

[132] M. Y. Arafat, M. M. Alam, and S. Moh, "Vision-based navigation techniques for unmanned aerial vehicles: Review and challenges," *Drones*, vol. 7, no. 2, p. 89, 2023.

[133] I. A. Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual slam," *Expert Systems with Applications*, vol. 205, p. 117734, 2022.

[134] A. Annaiyan, M. A. Olivares-Mendez, and H. Voos, "Real-time graph-based slam in unknown environments using a small uav," in *2017 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2017, pp. 1118–1123.

[135] M. Hassanalian, M. Radmanesh, and S. Ziaei-Rad, "Sending instructions and receiving the data from mavs using telecommunication networks," in *Proceeding of International Micro Air Vehicle Conference (IMAV2012), Braunschweig, Germany*, 2012, pp. 3–6.

[136] B. Renfro, M. Stein, E. Reed, J. Morales, and E. Villalba, "An analysis of global positioning system (gps) standard positioning service performance for 2019," *The University of Texas at Austin*, 2020.

[137] M. Perez-Ruiz, D. C. Slaughter, C. Gliever, and S. K. Upadhyaya, "Tractor-based real-time kinematic-global positioning system (rtk-gps) guidance system for geospatial mapping of row crop transplant," *Biosystems engineering*, vol. 111, no. 1, pp. 64–71, 2012.

[138] M. Lu, W. Chen, X. Shen, H.-C. Lam, and J. Liu, "Positioning and tracking construction vehicles in highly dense urban areas and building construction sites," *Automation in construction*, vol. 16, no. 5, pp. 647–656, 2007.

[139] G. Mao, S. Drake, and B. D. Anderson, "Design of an extended kalman filter for uav localization," in *2007 Information, Decision and Control*. IEEE, 2007, pp. 224–229.

[140] L. Arreola, A. M. De Oca, A. Flores, J. Sanchez, and G. Flores, "Improvement in the uav position estimation with low-cost gps, ins and vision-based system: Application to a quadrotor uav," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 1248–1254.

[141] A. Nemra and N. Aouf, "Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering," *IEEE Sensors Journal*, vol. 10, no. 4, pp. 789–798, 2010.

[142] Y. J. Choi, I. N. A. Ramatryana, and S. Y. Shin, "Cellular communication-based autonomous uav navigation with obstacle avoidance for unknown indoor environments," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 2, pp. 344–352, 2021.

[143] J. Ho, S. Phang, and H. Mun, "2-d uav navigation solution with lidar sensor under gps-denied environment," in *Journal of Physics: Conference Series*, vol. 2120, no. 1. IOP Publishing, 2021, p. 012026.

[144] X. Gao, T. Zhang, Y. Liu, and Q. Yan, "14 lectures on visual slam: from theory to practice," *Publishing House of Electronics Industry*, 2017.

[145] S. Park, T. Schöps, and M. Pollefeys, "Illumination change robustness in direct visual slam," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4523–4530.

[146] P. Sun and H. Y. Lau, "Rgb-channel-based illumination robust slam method," *Journal of Automation and Control Engineering Vol*, vol. 7, no. 2, 2019.

[147] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1935–1942.

[148] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2502–2509.

[149] Y. Fang, G. Shan, T. Wang, X. Li, W. Liu, and H. Snoussi, "He-slam: A stereo slam system based on histogram equalization and orb features," in *2018 Chinese Automation Congress (CAC)*. IEEE, 2018, pp. 4272–4276.

[150] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[151] W. Yang and X. Zhai, "Contrast limited adaptive histogram equalization for an advanced stereo visual slam system," in *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, 2019, pp. 131–134.

[152] Q. Gu, P. Liu, J. Zhou, X. Peng, and Y. Zhang, "Drms: Dim-light robust monocular simultaneous localization and mapping," in *2021 International Conference on Computer, Control and Robotics (ICCCR)*. IEEE, 2021, pp. 267–271.

[153] Y. Miao, D. Song, W. Shi, H. Yang, Y. Li, Z. Jiang, W. He, and W. Gu, "Application of the clahe algorithm based on optimized bilinear interpolation in near infrared vein image enhancement," in *Proceedings of the 2nd international conference on computer science and application engineering*, 2018, pp. 1–6.

[154] S. Rahman, M. M. Rahman, M. Abdullah-Al-Wadud, G. D. Al-Quaderi, and M. Shoyaib, "An adaptive gamma correction for image enhancement," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, pp. 1–13, 2016.

[155] K. Luo, M. Lin, P. Wang, S. Zhou, D. Yin, and H. Zhang, "Improved orb-slam2 algorithm based on information entropy and image sharpening adjustment," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[156] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4503–4508.

[157] J. Huang and S. Liu, "Robust simultaneous localization and mapping in low-light environment," *Computer Animation and Virtual Worlds*, vol. 30, no. 3-4, p. e1895, 2019.

[158] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza, "Learning-based image enhancement for visual odometry in challenging hdr environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 805–811.

[159] A. Savinykh, M. Kurenkov, E. Kruzhkov, E. Yudin, A. Potapov, P. Karpyshev, and D. Tsetserukou, "Darkslam: Gan-assisted visual slam for reliable operation in low-light conditions," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*. IEEE, 2022, pp. 1–6.

[160] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang, "Enlightengan: Deep light enhancement without paired supervision," *IEEE transactions on image processing*, vol. 30, pp. 2340–2349, 2021.

[161] J. Tang, J. Folkesson, and P. Jensfelt, "Geometric correspondence network for camera motion estimation," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1010–1017, 2018.

[162] R. Kang, J. Shi, X. Li, Y. Liu, and X. Liu, "Df-slam: A deep-learning enhanced visual slam system based on deep local features," *arXiv preprint arXiv:1901.07223*, 2019.

[163] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks." in *Bmvc*, vol. 1, no. 2, 2016, p. 3.

[164] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[165] D. Li, X. Shi, Q. Long, S. Liu, W. Yang, F. Wang, Q. Wei, and F. Qiao, "Dxslam: A robust and efficient visual slam system with deep features," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4958–4965.

[166] X. Han, Y. Tao, Z. Li, R. Cen, and F. Xue, "Superpointvo: A lightweight visual odometry based on cnn feature extraction," in *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*. IEEE, 2020, pp. 685–691.

[167] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[168] H. M. S. Bruno and E. L. Colombini, "Lift-slam: A deep-learning feature-based monocular visual slam method," *Neurocomputing*, vol. 455, pp. 97–110, 2021.

[169] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European conference on computer vision*. Springer, 2016, pp. 467–483.

[170] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous lo-

calization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[171] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[172] L. Chun, L. Hongfei, Z. Qi, M. Zhenzhen, T. Sisi, and W. Yaping, "Binocular slam based on learning-based feature extraction," in *Proceedings of the 2020 3rd International Conference on Robot Systems and Applications*, 2020, pp. 25–29.

[173] S. Li, S. Liu, Q. Zhao, and Q. Xia, *IEEE/ASME Transactions on Mechatronics*, 2021.

[174] G. Li, L. Yu, and S. Fei, "A deep-learning real-time visual slam system based on multi-task feature extraction network and self-supervised feature points," *Measurement*, vol. 168, p. 108403, 2021.

[175] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, "Gcnv2: Efficient correspondence prediction for real-time slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3505–3512, 2019.

[176] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[177] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[178] S. B. Damelin and W. Miller, *The mathematics of signal processing*. Cambridge University Press, 2012, no. 48.

[179] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *International workshop on artificial neural networks*. Springer, 1995, pp. 195–201.

[180] A. Smith, *Oxford dictionary of biochemistry and molecular biology: Revised Edition*. Oxford University Press, 2000.

[181] K. Hara, D. Saito, and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," in *2015 international joint conference on neural networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[182] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[183] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, pp. 1–74, 2021.

[184] K. S. Vepuri and N. Attar, "Improving the performance of deep learning in facial emotion recognition with image sharpening," *International Journal of Computer and Information Engineering*, vol. 15, no. 4, pp. 234–237, 2021.

[185] F. Gao, X. Xue, J. Sun, J. Wang, and Y. Zhang, "A sar image despeckling method based on two-dimensional s transform shrinkage," *IEEE Transac-*

tions on Geoscience and Remote Sensing, vol. 54, no. 5, pp. 3025–3034, 2016.

[186] S.-C. Huang, F.-C. Cheng, and Y.-S. Chiu, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *IEEE transactions on image processing*, vol. 22, no. 3, pp. 1032–1041, 2012.

[187] M. Kim and M. G. Chung, "Recursively separated and weighted histogram equalization for brightness preservation and contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1389–1397, 2008.

[188] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6235–6240.

[189] S. Bennett, "Development of the pid controller," *IEEE Control Systems Magazine*, vol. 13, no. 6, pp. 58–62, 1993.

[190] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[191] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[192] J. Vanne, E. Aho, T. D. Hamalainen, and K. Kuusilinna, "A high-performance sum of absolute difference implementation for motion esti-

mation," *IEEE transactions on circuits and systems for video technology*, vol. 16, no. 7, pp. 876–883, 2006.

[193] M. Chu and N. Thuerey, "Data-driven synthesis of smoke flows with cnn-based feature descriptors," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–14, 2017.

[194] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[195] V. Kumar, D. R. Recupero, D. Riboni, and R. Helaoui, "Ensembling classical machine learning and deep learning approaches for morbidity identification from clinical notes," *IEEE Access*, vol. 9, pp. 7107–7126, 2021.

[196] F. Gao, T. Huang, J. Wang, J. Sun, A. Hussain, and E. Yang, "Dual-branch deep convolution neural network for polarimetric sar image classification," *Applied Sciences*, vol. 7, no. 5, p. 447, 2017.

[197] W. Chen, Y. Qiao, and Y. Li, "Inception-ssd: An improved single shot detector for vehicle detection," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–7, 2020.

[198] D. Sarkar and S. K. Gunturi, "Wind turbine blade structural state evaluation by hybrid object detector relying on deep learning models," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2020.

[199] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui, and X. Chen, "Automatic bridge crack detection using a convolutional neural network," *Applied Sciences*, vol. 9, no. 14, p. 2867, 2019.

[200] C. Guindel, D. Martín, and J. M. Armingol, "Modeling traffic scenes for intelligent vehicles using cnn-based detection and orientation estimation," in *Iberian Robotics conference*. Springer, 2017, pp. 487–498.

[201] W. Fang, L. Wang, and P. Ren, "Tinier-yolo: A real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2019.

[202] S. Hossain and D.-j. Lee, "Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with gpu-based embedded devices," *Sensors*, vol. 19, no. 15, p. 3371, 2019.

[203] F. Gao, F. Ma, J. Wang, J. Sun, E. Yang, and H. Zhou, "Visual saliency modeling for river detection in high-resolution sar imagery," *IEEE Access*, vol. 6, pp. 1000–1014, 2017.

[204] X. Jin, P. Deng, X. Li, K. Zhang, X. Li, Q. Zhou, S. Xie, and X. Fang, "Sun-sky model estimation from outdoor images," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2020.

[205] F. Gao, W. Shi, J. Wang, A. Hussain, and H. Zhou, "A semi-supervised synthetic aperture radar (sar) image recognition algorithm based on an attention mechanism and bias-variance decomposition," *IEEE Access*, vol. 7, pp. 108 617–108 632, 2019.

[206] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge 2007 (voc2007) results," 2007.

[207] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (voc2012) development kit," *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, vol. 8, 2011.

[208] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[209] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[210] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[211] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[212] D. L. Olson and D. Delen, "Performance evaluation for predictive modeling," in *Advanced data mining techniques*. Springer, 2008, pp. 137–147.

[213] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[214] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[215] B. Fang, G. Mei, X. Yuan, L. Wang, Z. Wang, and J. Wang, "Visual slam for robot navigation in healthcare facility," *Pattern Recognition*, vol. 113, p. 107822, 2021.

[216] B. Fang and Z. Zhan, "A visual slam method based on point-line fusion in weak-matching scene," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, p. 1729881420904193, 2020.

[217] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, "Semantic monocular slam for highly dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 393–400.

[218] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, "Visual and visual-inertial slam: State of the art, classification, and experimental benchmarking," *Journal of Sensors*, vol. 2021, 2021.

[219] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, 2021.

[220] J. Zubizarreta, I. Aguinaga, and J. M. M. Montiel, "Direct sparse mapping," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1363–1370, 2020.

[221] G. Maragatham and S. M. M. Roomi, "A review of image contrast enhancement methods and techniques," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 9, no. 5, pp. 309–326, 2015.

[222] A. Bhandari, A. Kumar, and P. Padhy, "Enhancement of low contrast satellite images using discrete cosine transform and singular value decomposition," *World Academy of Science, Engineering and Technology*, vol. 79, pp. 35–41, 2011.

[223] D. F. Malin, "Unsharp masking." *AAS Photo Bulletin*, vol. 16, pp. 10–13, 1977.

[224] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010: 11th Euro-*

pean Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11. Springer, 2010, pp. 778–792.

[225] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[226] M. Kasper, S. McGuire, and C. Heckman, "A benchmark for visual-inertial odometry systems employing onboard illumination," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5256–5263.

[227] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[228] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige, "g2o: A general framework for (hyper) graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, 2011, pp. 9–13.

[229] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[230] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[231] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.

[232] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[233] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[234] X. Tao, L. Han, M. Paoletti, S. Roy, J. Plaza, J. M. Haut, and A. Plaza, "Multiple incremental kernel convolution for land cover classification of remotely sensed images," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 2548–2551.

[235] F. Gao, Q. Liu, J. Sun, A. Hussain, and H. Zhou, "Integrated gans: Semi-supervised sar target recognition," *IEEE Access*, vol. 7, pp. 113 999–114 013, 2019.

[236] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[237] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[238] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in con-

text," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[239] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[240] Y.-Y. Jau, R. Zhu, H. Su, and M. Chandraker, "Deep keypoint-based camera pose estimation with geometric constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4950–4957.

[241] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[242] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

Appendix A

Hardware Specifications

Table A.1: **Detailed Hardware Components Utilised in This Work**

Hardware type	Quantity	Specification
UAV frame	1	DJI F450 Flame Wheel Quadcopter Frame
Motor	4	Multistar 2212 920KV Motor
Electronic Speed Control	4	HobbyWing 40A
Propellers	4	DJI 9450 Self-tightening Propellers
Power module	1	Holybro PM07
UAV battery	1	YouMe 6000mAh 4S 50C Lipo Battery
Flight Controller	1	Pixhawk 4
Companion computer	1	Nvidia Jetson TX2
Carrier board for companion computer	1	Auvidea J120 Carrier Board
Telemetry Radio	1	Holybro 100mW Telemetry Radio Set V3
Remote control receiver	1	Turnigy TGY-iA6C Receiver
Remote control transmitter	1	Turnigy TGY-i6S Digital Proportional Radio Control System
RGB Camera	1	Logitech C270
Lighting	1	OLIGHT Baton 3

Appendix B

Communication between UDP and ROS

The example of sending position information obtained from the VO system is listed as follows:

```
int client_sockfd;
int len;
std::ostringstream ss;
struct sockaddr_in remote_addr;
int sin_size;
char buf[BUFSIZ];
char vslam_x[BUFSIZ];
char vslam_y[BUFSIZ];
char vslam_z[BUFSIZ];
memset( & remote_addr, 0, sizeof(remote_addr));
remote_addr.sin_family = AF_INET;
remote_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
```

```
remote_addr.sin_port = htons(8000);
if ((client_sockfd = socket(PF_INET, SOCK_DGRAM, 0)) < 0)
perror("socket error");
}
tf::poseTFToMsg(new_transform, pose.pose);
x = pose.pose.position.x;
y = pose.pose.position.y;
z = pose.pose.position.z;
float ix, iy, iz;
ix = x;
iy = z;
iz = -y;
sprintf(vslam_x, "%f", ix);
sprintf(vslam_y, "%f", iy);
sprintf(vslam_z, "%f", iz);
strcat(vslam_x," ");
strcat(vslam_x, vslam_y);
strcat(vslam_x," ");
strcat(vslam_x, vslam_z);
strcpy(buf, vslam_x);
printf("sending: '%s' \n", buf);
sin_size = sizeof(struct sockaddr_in);
if ((len = sendto(client_sockfd, buf, strlen(buf), 0, (struct sockaddr * ) &
remote_addr, sizeof(struct sockaddr))) < 0)
perror("recvfrom");
}
```

close(client_sockfd);

The example of receiving position information and publishing it through the ROS topic is shown as follows:

UDP_IP = "127.0.0.1"

UDP_PORT = 8000

sock = socket.socket(socket.AF_INET,

socket.SOCK_DGRAM)

sock.bind((UDP_IP, UDP_PORT))

rate = rospy.Rate(30)

self.orbpose = PoseStamped()

self.position = rospy.Publisher('/mavros/vision_pose/pose', PoseStamped,

queue_size=10)

data, addr = sock.recvfrom(1024)

data1 = data.split()

self.orbpose.pose.position.x = float(data1[0])*scale;

self.orbpose.pose.position.y = float(data1[1])*scale;

self.orbpose.pose.position.z = float(data1[2])*scale;

self.position.publish(self.orbpose)