Accurate 3D Reconstruction of Underwater Infrastructure using Stereo Vision

Brenton Leighton

Master of Engineering (Research)

Faculty of Engineering and Information Technology

University of Technology Sydney

2019

Certificate of Original Authorship

I, Brenton Leighton, declare that this thesis is submitted in fulfilment of the requirements for the award of Master of Engineering (Research) in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program (RTP), and by the Australian Research Council and Roads and Maritime Services (Linkage Project LP150100935).

Signature:

Production Note:
Signature removed prior to publication.

Date: 05/08/2019

Acknowledgements

I would like to thank my supervisor Associate Professor Shoudong Huang for his guidance of my research work and assistance in the preparation of this thesis. I would also like to thank my co-supervisor Distinguished Professor Dikai Liu and alternate supervisor Dr. Liang Zhao for their support and assistance.

I would like to thank members of the SPIR project Dr. Andrew To, Dr. Khoa Le, and Craig Borrows, for their work in developing the ROV and assistance with data collection.

Finally I would like to thank the staff and students at the Centre for Autonomous Systems, in particular Li Yang Liu, for any advice that helped with my research.

Table of Contents

List of Figures

List of Tables

Acronyms

AGAST adaptive and generic accelerated segment test.
AUV autonomous underwater vehicle.

BRIEF binary robust independent elementary features.
BRISK binary robust invariant scalable keypoints.

CLAHE contrast limited adaptive histogram equalisation.
CMVS clustering views for multi-view stereo.

DSO direct sparse odometry.
DTAM dense tracking and mapping.

ELAS efficient large-scale stereo matching.

FAST features from accelerated segment test.
FREAK fast retina keypoint.

HDR high dynamic range.

IMU inertial measurement unit.

KLT Kanade-Lucas-Tomasi.

LiDAR light detection and ranging.
LSD-SLAM large-scale direct SLAM.

MAPSAC maximum a posterior estimation sample concensus.
MLESAC maximum likelihood sample concensus.
MVE multi-view environment.

ORB oriented FAST and rotated BRIEF.

PMVS patch-based multi-view stereo.
PTAM parallel tracking and mapping.

RANSAC random sample concensus.
ROV remotely operated underwater vehicle.

SfM structure from motion.
SIFT scale-invariant feature transform.
SLAM simultaneous localisation and mapping.
SPIR Submersible Pile Inspection Robot.
SURF speeded-up robust features.
SVO semi-direct visual odometry.

USB universal serial bus.

VIO visual-inertial odometry.
VO visual odometry.

Abstract

Modern vehicle and pedestrian bridges over water are built on concrete piles; foundations that penetrate the soft soil of the bed of the body of water to sit on the solid rock below. Like all built structures these concrete piles require regular inspection to determine if any preventative maintenance is needed. Stereo vision and Structure from Motion techniques offer a cost effective method of creating a 3D reconstructions of a scene, but the underwater environment around a bridge pile has unique challenges. Poor visibility, strong and varying sunlight, and floating material in the water create difficulties for computer vision.

This thesis evaluates exposure control, image enhancement, and feature detection and description algorithms, for the purpose of localising images captured around a bridge pile. Stereo correspondence algorithms are evaluated and used to create a single viewpoint 3D reconstruction of a scene, then a visual SLAM system is used to localise the single viewpoint reconstructions, so that they can be merged together to create a 3D reconstruction of a bridge pile.

Visual odometry, using KAZE with CLAHE image enhancement for feature detection, was successfully performed in the underwater environment. ORB-SLAM2 can also perform well, and 3D reconstructions from a single viewpoint (created with block matching, semi-global matching, or ELAS) were merged to create 3D reconstructions of submerged bridge piles.

1

Introduction

1.1 Motivation

Modern vehicle and pedestrian bridges over water are built on concrete piles; foundations that penetrate the soft soil of the bed of the body of water to sit on the solid rock below. Like all built structures these concrete piles require regular inspection to determine if any preventative maintenance is needed. Structures in water will gradually accumulate marine growth (biofouling) which must be removed from the structure for inspection. This cleaning is currently performed by divers and can be hazardous, time consuming, and expensive. After cleaning the divers will take photos of any potential issues with the structure, which are later analysed by engineers.

A remotely operated underwater vehicle (ROV) could potentially replace the diver, allowing the cleaning and inspection to be performed by an operator from the safety of a boat. To reduce the workload on the operator, inspection and/or cleaning could be automated. This research is part of the Submersible Pile Inspection Robot (SPIR) project, exploring the feasibility of cleaning and inspecting submerged bridge piles using an ROV, which is shown in Figure 1.1.

For both remotely controlled and autonomous operation an accurate model of the structure is required. Sonar, laser, and structured light sensors are potential solutions for underwater perception, but stereo vision has been selected for its high resolution and low cost. Structure from motion (SfM) and visual simultaneous localisation and mapping (SLAM) are methods of estimating a 3D structure from images. SfM/visual SLAM in an underwater environment has some unique challenges, including camera calibration, lighting, image degradation, and the removal of unwanted objects. This research aims to determine if stereo vision can be used to create a 3D reconstruction of submerged concrete piles, under typical conditions, and with adequate accuracy for the purpose of cleaning and inspection.

There are numerous examples of underwater SfM and visual SLAM from monocular or stereo vision. The majority of these involve a nonholonomic vehicle, a torpedo shaped ROV or autonomous underwater vehicle (AUV) that moves forward and turns or changes depth gradually, mapping the seafloor or large object at a rage of several metres. This

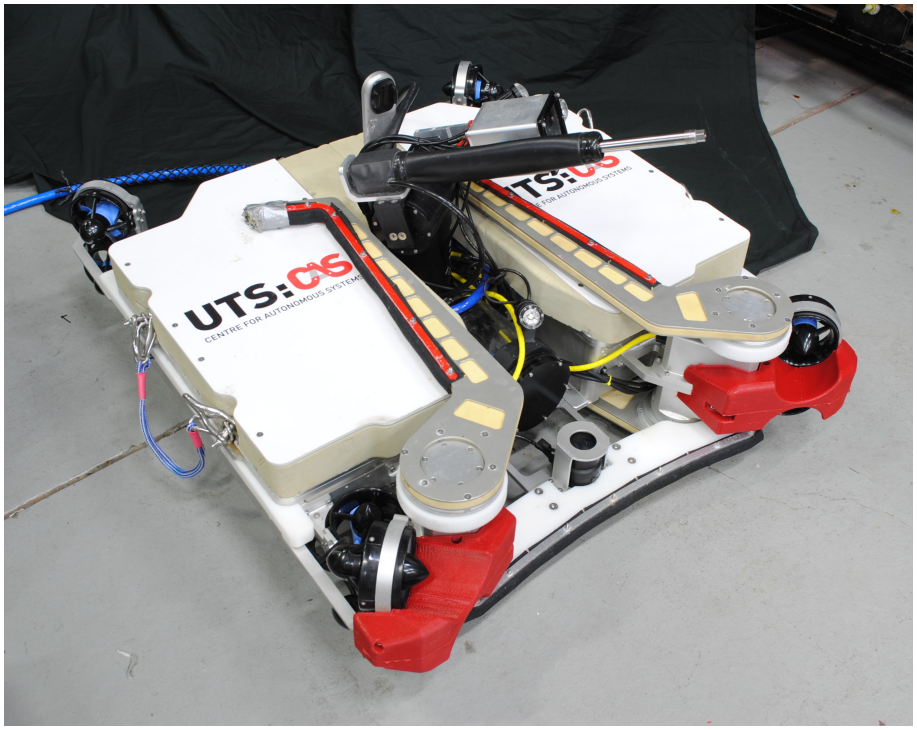


Figure 1.1: The third version of the SPIR ROV

research relates to creating a 3D reconstruction of a underwater infrastructure, from stereo vision gathered by a holonomic ROV that is expected to move erratically.

There are examples of underwater SfM using images taken at random positions around an object. These examples generally focus on accurate underwater camera calibration.

This research addresses a number of unique problems in order to perform underwater SfM/visual SLAM:

- Fast and erratic movement of the camera
- Difficult lighting conditions where sunlight may illuminate the object or dominate the background
- Highly turbid water with visibility of 1 m or less
- Varying amounts of floating material

1.2 Contribution

The main contributions of this thesis are:

- Evaluating feature detection and matching for underwater visual odometry in a difficult environment.
- Extending parallax parameterisation for bundle adjustment to stereo.
- Evaluating stereo correspondence algorithms for images captured in a difficult underwater environment.

- Creating a 3D reconstruction of underwater infrastructure from stereo images captured in a difficult underwater environment.

1.3 Publications

B. Leighton, L. Zhao, S Huang, and G. Dissanayake, "Extending Parallax Parameterised Bundle Adjustment to Stereo," in *Australasian Conference on Robotics and Automation (ACRA) 2017*, ARAA, 2017, pp. 1-9.

2

Literature Review

2.1 Methods of Underwater Perception

Sonar, laser, and vision are viable sensor modalities for underwater perception. Massot-Campos and Oliver-Codina [1] reviewed sensors and methods for underwater reconstruction. The sensors covered include:

- passive vision;
- structured light; and
- laser scanning.

Although they focus on optical sensing, there is some discussion of sonar sensing. Multi-beam or imaging sonar is capable of creating 3D reconstructions underwater, and compared to optical sensors has greater range, but at a lower resolution and with a longer acquisition time. For these reasons optical sensing is preferable.

Passive vision

Passive vision is the lowest cost method of two dimensional perception. A single camera produces a 2D image, but two cameras can perceive a scene in 3D. It is possible to perceive a scene in 3D from a monocular camera by combining images taken from multiple viewpoints, however additional information is required for accurate scale. This information can be from an additional sensor or sensors, or by making assumptions about the scene. The process of determining 3D structure from images captured at different viewpoints is called SfM.

Aside from low cost, passive vision has the advantage of fast acquisition time, because the entire frame of data is captured at one time. A disadvantage of passive vision is that is requires the scene to have sufficient visual features to determine depth. Because depth is determined by matching features in the image it can be less robust than active sensing methods.

Structured light

Structured light combines a monocular camera with a light pattern projector, and the light patterns are used to determine the depth of the scene. Geng [2] has written an overview of structured light. Binary code or phase shift patterns are commonly used, although other patterns are available that enable depth information capture with single pattern, using colour, greyscale, or complex binary patterns. A binary light pattern has alternating stripes of light and dark. A number of these patterns, each with decreasing stripe width, are projected and captured, and for each pixel of the image a binary code is generated based on whether or not the pixel was illuminated but each pattern. The binary code determines the depth of the pixel. Depth resolution is dependant on the number of stripes, with more, thinner stripes giving more depth resolution. Combining phase shift patterns with binary code patterns enables a higher depth resolution than possible with binary code patterns alone.

Structured light is capable of higher accuracy than passive vision. It is also able to determine the depth of surfaces with no visual features. In addition to the increased cost due to the light projector, it has the disadvantage that capturing depth requires multiple images when binary code or phase shift patterns are used. If the camera is moving, this may result is inaccuracies due to the images being taken from differing viewpoints. To an extent this can be mitigated by capturing at a very high frame rate.

Laser scanning

Laser scanning or light detection and ranging (LiDAR) involves a laser emitter and detector to measure distance. Distance is typically measured by time-of-flight; the time between emitting a laser pulse and detecting the reflection from an object. A single scan line is acquired by sweeping the emitter/detector, and a 3D scan is acquired using multiple sweeping emitter/detectors.

Although the terms laser scanning and LiDAR are often used interchangeably, Massot-Campos and Oliver-Codina [1] use the term LiDAR to refer to airborne laser scanning only. Airborne LiDAR is widely used for ocean surveying, which involves a LiDAR sensor mounted to a plane for the purpose of mapping a relatively shallow seabed [3]–[5]. Of the wavelengths available for laser emitters, green light is absorbed the least by water, and therefore has the furthest penetration. Reineman *et al.* [4] used airborne LiDAR with a green laser to penetrate 30 m into water.

If a laser scanner is submerged in water, the time-of-flight method of distance measurement is less suitable. At shorter ranges measurement error becomes a more significant percentage of the result. Continuous wave or modulation methods have greater accuracy at short ranges. Continuous wave laser scanning is based on triangulation. A laser emitter projects a beam of light, which is detected in an imaging sensor. Given a known distance between the emitter and imaging sensor, and known angles between the emitter and beam, and beam and sensor, the distance to the target can be triangulated. Modulated

laser scanning can involve frequency or amplitude modulation. Due to scattering, high frequencies tend to be lost in water, therefore amplitude modulation is the only viable type of modulation for use in water [1]. With amplitude modulation, the emitter modulates the amplitude of the laser beam, and the distance is determined by the difference between the original and returned signal.

Laser scanning overs high accuracy, range, and robustness over passive vision or structured light, but is significantly more expensive.

2.2 Underwater Vision

Due to its low cost, passive vision is a compelling sensing modality, however underwater vision has a number of challenges not typically addressed for in-air vision. Of particular interest is:

- image degradation caused by absorption, scattering, and backscattering;
- uneven lighting caused by artificial lighting, sunlight flicker, and varying direct sun; and
- camera calibration when using a flat port housing.

Image degradation

There are three major causes of underwater image degradation:

- absorption – the attenuation of light as it travels through the water;
- scattering – the redirection of light as it collides with microscopic particles in the water; and
- backscattering – the scattering of the artificial light from the camera system back into the camera.

Scattering and backscattering cause blurring and a reduction in contrast (hazing). Absorption attenuates light and therefore causes a reduction in visibility. Absorption also effects light frequencies differently, causing a change colour.

Histogram equalisation approaches are commonly used for enhancing underwater images. It involves redistributing pixel intensities so that the histogram is more evenly distributed. This simple approach performs poorly when the image has regions of varying brightness. Adaptive histogram equalisation [6] improves on simple histogram equalisation by dividing the image into blocks, computing histograms of the blocks, and equalising pixel intensity based on the nearest blocks. This approach can amplify noise in regions that have a very small intensity range. Contrast limited adaptive histogram equalisation (CLAHE) [7] minimises noise amplification by limiting contrast gain, and is used in many modern underwater vision systems. Rizzini *et al.* [8] used CLAHE in addition to a contrast mask, but found that the contrast mask contributed little. Zheng *et al.* [9] combined

CLAHE with an unsharp mask, which they found to be effective at enhancing underwater images. Hong and Kim [10] also found CLAHE effective.

Absorption, scattering, and backscattering are spatially variant: their effects increase with the distance to the observed object. CLAHE, while effective, is not spatially variant, and therefore cannot accurately correct for the underwater image degradation. Accurate underwater vision is often desired for the purpose of inspecting coral, underwater infrastructure, or performing archaeological surveys. A number of papers have looked at capturing accurate colour images in shallow water lit only by sunlight.

Schechner and Karpel [11] demonstrated that underwater image degradation is associated to light polarisation. They used a rotating polarisation filter to capture two images, one at minimum polarisation and one at maximum polarisation, which are used to estimate the depth of the scene and correct the image in a spatially variant way. Due to the rotating polarisation filter this method has a long acquisition time.

Carlevaris-Bianco *et al.* [12] determined scene depth using a single colour image, exploiting the fact that, underwater, red light is attenuated significantly more than green or blue. The depth of a pixel is estimated by comparing the maximum red and maximum green/blue values in a small surrounding region, and these estimates are refined using the image matting method proposed by Levin *et al.* [13]. The scene depth estimate is then used to dehaze the image by correcting for the scattering effect. They found that their method produced a more accurate image than histogram equalisation or CLAHE.

Queiroz-Neto *et al.* [14] used a stereo camera system and, after careful radiometric calibration of both the camera system and water, were able to reconstruct an underwater scene even in high turbidity. They note that the careful calibration required is difficult to achieve outside of a lab.

Roser *et al.* [15] also used a stereo camera system. They generated a coarse depth map from robustly matched points with efficient large-scale stereo matching (ELAS) [16], estimated visibility coefficients and then enhanced the image using an underwater light propagation model. They repeated this process, estimating depth and enhancing the image again, and found their result to be better than histogram equalisation or CLAHE.

Other methods have looked at determining visibility coefficients automatically. Bryson *et al.* [17] created a 3D reconstruction of an underwater scene with a monocular camera and, given a known lighting model, estimated visibility coefficients for image enhancement and colour correction. Skinner *et al.* [18] estimated visibility coefficients by including them in the optimisation problem that determines 3D structure from multiple images.

A number of methods for accurate image enhancement are available, however these are only necessary when accurate, and in particular colour, images are required. For the purpose of estimating camera motion from images, or for inspecting the condition of concrete piles, accurate images may not be needed.

Uneven lighting

Uneven lighting is more common with underwater vision than with in-air vision. The two main causes of this are artificial lighting and sunlight flicker.

Artificial lighting

With the exception of working in shallow, clear water, strong artificial lighting is often a requirement for images of underwater objects. Artificial lighting is directional and produces nonuniform illumination of the scene. Often the scene will be brighter in the centre of the image where the light is directed, and joining these images together for 3D reconstruction or photo mosaicing will result in an unevenly illuminated representation of the underwater scene.

Rzhanov *et al.* [19] observed that artificial illumination is associated with low frequencies in the Fourier transform of the image, compared to the higher frequency response of light reflected by objects in the scene. They used detrending to determine a low-order two-dimensional polynomial spline that represents the artificial illumination. The spline is averaged over multiple frames to improve accuracy, then used to remove the effect of artificial illumination in the original images.

Garcia *et al.* [20] compared CLAHE, homomorphic filtering (i.e. suppressing low frequencies in a Fourier transform with a high pass homomorphic filter), and the use of an illumination-reflectance model for illumination removal. They found CLAHE to be acceptable at removing uneven illumination in some cases, and homomorphic filtering to be effective as long as the objects in the scene have a high frequency response. However, the best results were achieved by using an illumination-reflectance model that considers the image as a product of the illumination and reflectance. The illumination model is generated by averaging a number of images that have been low-pass filtered (blurred) using a Gaussian kernel. The component of the image produced by reflectance is found by dividing an image by the illumination model.

Rzhanov and Gu [21] used median values to suppress the appearance of artificial illumination and seams in photo mosaics. First a robust feature detector such as scale-invariant feature transform (SIFT) [22] or speeded-up robust features (SURF) [23] is used to register the images, then micro warping is applied using a thin-plate spline algorithm [24] to remove any parallax distortion. For any overlapping areas, the value of the pixel in the mosaic image is the median of the values in the overlapping images. The median value was found to be less affected by outliers than a weighted average, assuming sufficiently dense coverage.

More recently, Zheng *et al.* [9] have found CLAHE to produce better results than homomorphic filtering. From a source image they produce an enhanced image with CLAHE as well as an unsharp mask, which are then combined with weighted blending. CLAHE offers a simple solution for mitigating the effect of uneven artificial lighting.

Sunlight flicker

In shallow water, a submerged scene is subject to spatial and temporal light fluctuations, caused by the refraction of sunlight by surface waves [25]. These caustics produce bright patterns in the underwater scene. Due to the changing nature of these patterns, Schechner and Karpel [26] demonstrated that taking the temporal median from multiple images would remove the bright spots, as long as the patterns were fast moving relative to the duration of the images captured. If a pixel is illuminated by a caustic over a majority of the images, the bright spot would not be removed by temporal median filtering. Based on research into shadow removal for in-air vision [27], Schechner and Karpel [26] calculate the median derivative for each pixel from a number of images, which are then integrated using a pseudo-inverse [28]. They show that using the median of derivatives is more effective at removing sunlight flicker than an average or median image.

Gracias *et al.* [29] looked at removing sunlight flicker from images taken by a moving camera. They use robust feature detection to register images but find that perfect registration is often not possible. They find that when registration is imperfect using the temporal average or median image, or the temporal median of the derivative [26], will result in a blurred image. Instead of estimating the reflectance field of a scene directly from the temporal median, they estimate the illumination field by applying a low-pass filter to the difference of a reference image and temporal median image. The estimated illumination field is then subtracted from the reference image to estimate the reflectance field: the image without the effect of sunlight flicker. They find that this method works well even in the presence of registration error.

Exposure and dynamic range

A common problem for outdoor computer vision is exposure control. An outdoor scene may have bright areas that are directly lit by the sun and dark areas that are in shadow. Typical imaging sensors have a limited range, so a choice must be made between overexposing bright areas or underexposing dark areas. Both options result in a loss of information. High dynamic range (HDR) imaging refers to capturing an image with a larger range of brightness than a typical sensor. Specialised HDR imaging sensors exist, but they are expensive and not readily available. HDR imaging is most commonly achieved by combining multiple images at different exposures [30]. This requires multiple images taken from the same position, something that may not be possible or practical from a camera attached to a moving robot.

Nuske *et al.* [31] describe a system for real-time HDR imaging with automatic exposure control for a mobile robot platform. Three different exposure levels were used. The middle exposure level aims for a mean pixel intensity of 50%, a commonly used metric for automatic exposure control. The darker exposure level aims for a mean pixel intensity of 50% for the darkest third of the pixels in the image. Similarly, the lightest exposure level aims for a mean pixel intensity of 50% for the lightest third of the pixels. Each image

is converted to the YCrCb colour space and the luminance channel is used to generate contour maps at various scales. The largest and smallest scale contour maps are used to find a 2D translation between two images, with the the largest scale contour map used for coarse registration, which is then refined with the smallest scale contour map. Image registration is performed between images of similar exposure, and the three results (from the three exposure levels) are interpolated to reduce error. The luminance channel of the HDR image is generated by taking the pixel values from the image with the largest contour value at that pixel. The idea is that the largest gradient magnitude represents the most information. The chrominance channels (Cr and Cb) are combined by taking the values with the maximum Euclidean distance from white.

Hrabar *et al.* [32] looked at HDR stereo vision with the goal of generating a more accurate and complete occupancy map of an environment. For each stereo image pair a disparity map is generated, which is projected into a local occupancy map. Given known camera poses, the local occupancy maps are combined into a single global occupancy map [33]. Compared to the image registration method of Nuske *et al.* [31] this method considers the full 3D pose of the camera but does not generate an HDR image.

Irie *et al.* [34] used multiple exposure levels to improve the number of key points detected from a sequence of images. The key points detected across a sequence of different exposures are combined into an HDR key point set, with matching key points found at different exposures combined into a single key point. They used these HDR key points to improve outdoor relocalisation (identifying images taken from similar pose) with images taken at different times of the day.

Instead of combining multiple images into an HDR image, Shim *et al.* [35] looked at improving auto exposure for outdoor mobile robots. Their auto exposure control aims to maximise the gradient information in the image, rather than balancing the overall intensity of the image. Given an input image, they simulate varying exposure levels with gamma correction, calculate the gradient information from each image, and select the exposure that maximises gradient information. Although less information may be captured than with HDR imaging, this method avoids image registration issues. Gradient information is derived from the magnitude of the gradient of each pixel in the image. A non-linear logarithm function maps gradient magnitude to gradient information.

Image entropy is another way to measure the amount of information in an image. It is a statistical measure of randomness that is derived from the histogram of a greyscale image [36].

The goal of exposure control and HDR imaging for outdoor robotics is to maximise visibility in the images collected. HDR imaging will provide more visual information in the images collected, however it adds a significant amount of cost or complexity to the system. If a method of exposure control produces adequately exposed images and allows for robust camera motion estimation, it would offer a simple solution for ensuring good visibility when collecting images under highly variable lighting.

Camera calibration

Accurate camera calibration is essential for accurate 3D reconstruction from images. It is needed to fix the radial and tangential distortion of the camera (undistortion), correct the misalignment of stereo cameras (rectification), and to determine scene depth from disparity information (reprojection).

An underwater vision system is characterised by an in-air camera in a waterproof housing. The front port of the housing can be domed or flat. A domed port is preferable, producing a more accurate image [37]. With a flat port, light is attenuated and the image blurred as the angle of view increases, as the effective thickness of the port increases [37], [38]. For a stereo camera with a narrow baseline, adequately small domed ports are not commercially available, making a flat port more practical. With a flat port camera calibration becomes more complicated.

For simplicity, cameras are usually simplified to a single viewpoint (pinhole) model. With a underwater camera in a flat port housing, this simplification becomes a less accurate representation of the camera. The real camera system involves a camera in air, looking through a flat interface (the front port) into a refractive medium (the water). The three mediums of differing refractive index cause 3D distortion that has significant error when corrected by the single viewpoint model [39].

It is possible to use the single viewpoint model when underwater by minimising the thickness of the port and placing the camera as close to the port as possible. This minimises the effect of the air and port, so that the system behaves like a pinhole camera in water. In this situation it is also possible to use the camera parameters determined from above water calibration by modifying the focal length by the ratio between refractive indices of the two mediums (for air to water this is approximately 1.33 [40]). Ideally calibration should be performed in-situ, as changes in pressure, temperature, and salinity can alter the refractive index of water [41].

Using a thin port is not always possible, for example ROVs operating in deep water require thick glass ports. There is also an issue with misalignment of the camera and front port, which becomes more significant as the camera resolution increases. An axial camera model represents an underwater camera system more accurately than the single viewpoint model, but calibration is significantly more challenging. Jordt-Sedlazeck and Koch [42] found axial model calibration parameters by synthesis, generating a 3D model of a calibration target and using evolutionary optimisation was used to find camera and housing parameters.

Another possibility for underwater calibration is to use the single viewpoint camera model and above water calibration to undistort and rectify images, and incorporate refraction into 3D reconstruction [43]. This reduces error and works with thick ports and camera misalignment.

Łuczyński *et al.* [44] have developed a model for the accurate calibration and rectification of underwater cameras that is based on a combination of the single viewpoint and

axial camera models. It uses precomputed lookup tables for fast processing and they claim it has higher accuracy than the current state of the art.

With a thin front pane the single viewpoint model can be used, avoiding the additional computation of the axial model. It will also allow for a wider range of computer vision software to be used, as these are generally based on the single viewpoint model.

2.3 Visual Odometry

It is possible to determine the movement of a camera from the images it captures. This is referred to as visual odometry (VO), visual SLAM, or SfM.

Davide Scaramuzza and Friedrich Fraundorfer have written a comprehensive two part introduction to VO [45], [46], which also covers SfM and visual SLAM. SfM, VO, and visual SLAM are all related to the process of determining camera pose and scene structure from images. SfM generally deals with a collection of unordered images, with either known or unknown camera parameters, and possibly even with images taken by different cameras. VO focuses on determining camera pose from sequential images, usually with know camera parameters, and in real time. Visual SLAM is similar to VO, but where VO is only concerned with local consistency of the camera poses and scene structure, visual SLAM aims to achieve global consistency. Visual SLAM is generally implemented as an extension to the process of VO.

SfM and VO/visual SLAM share the same initial steps:

- feature detection;
- descriptor extraction;
- feature matching; and
- motion estimation.

Feature detection, description, and matching

The issues of detecting salient points within an image and identifying corresponding points is a well researched topic, with applications including object recognition, image and texture classification, and camera motion estimation.

Hassaballah *et al.* [47] is a recent overview of feature detection, description, and matching. They divide feature detection into three types: single scale, multi-scale, and affine invariant. As the name suggests, single scale methods operate at a single scale only, and are unable to detect features larger than the detection region. Single scale detectors include the Harris detector [48], the features from accelerated segment test (FAST) detector [49] and the Hessian detector. Multi-scale detectors enable the detection of features at multiple scales by operating on multiple scales of the input images. Multi-scale detectors include Laplacian of Gaussian, Difference of Gaussian [50], Harris Laplace [51], Hessian Laplace, and Gabor wavelet [52]. Affine invariant detectors enable the detection of similar features that have undergone an affine transformation. Affine invariance can be considered as a

more general case of scale invariance: with scale invariance the x and y dimensions of the feature are scaled by the same amount, but with affine invariance the x and y dimensions may be scaled by different amounts. Harris and Hessian detectors have been extended to be affine invariant.

The aim of feature description is to generate a vector that describes the feature in a way that allows for the same feature to be identified across multiple images, despite changes to illumination, rotation, scale, or perspective. Methods of feature description can be divided into numerical, which describes a feature with a vector of (integer or floating point) values, or binary, which describes a feature with a binary code. Binary feature descriptors contain less information, and therefore tend to be less robust, but are faster in descriptor extraction and matching. Methods of feature description will also usually stipulate a method of feature detection and matching.

Examples of numerical descriptors include SIFT [50], SURF [23], and KAZE [53]. Lowe [50] created SIFT, a pioneering example of a scale, illumination, and (locally) affine invariant method of feature detection and description. It uses Difference of Gaussians for feature detection, which is similar to Laplacian of Gaussians, but faster to compute. Features are described by a histogram of gradient orientations around the key point. SURF claims performance on par with SIFT, but is much faster to compute. It uses a fast Hessian method of feature detection, and the sum of Haar wavelet responses for feature description. SIFT and SURF are generally the most commonly used feature detectors. KAZE features improve on the affine invariance of SIFT and SURF, but are slower to compute. Alcantarilla *et al.* [53] showed that KAZE features have better performance with deformable surfaces.

Examples of binary feature descriptors include binary robust independent elementary features (BRIEF) [54], oriented FAST and rotated BRIEF (ORB) [55], binary robust invariant scalable keypoints (BRISK) [56], and fast retina keypoint (FREAK) [57]. BRIEF describes features by comparing pairs of pixels, with the binary value for each pair indicating which has the greater intensity. Using the same feature detector as SURF, Calonder *et al.* [54] claim similar performance, although only when rotational invariance is not required. ORB extends on BRIEF, using a multi scale FAST detector for faster detection, and an orientated BRIEF descriptor for rotation invariance. The authors claim similar matching performance to SIFT and SURF, with significantly less computation time. Like ORB, BRISK is a binary descriptor with scale and rotation invariance. It uses the FAST based scale invariant adaptive and generic accelerated segment test (AGAST) [58] feature detector, and a sampling pattern made of concentric circles. FREAK features uses the same AGAST feature detector as BRISK, with a sampling pattern inspired by the human retina. Alahi *et al.* [57] claim that FREAK is more robust than SIFT, SURF, or BRISK, with lower computation and memory requirements.

To match features the difference between two numerical descriptors is typically measured by Euclidean distance, while the difference between binary descriptors is measured by Hamming distance. Hamming distance is the difference of the bits between two binary

codes, and is faster to compute than Euclidean distance.

Moreels and Perona [59] compared feature detectors and descriptors, but the paper is from some time ago and does not include methods developed since then. Garcia and Gracias [60] looked at feature detection in underwater images of varying turbidity, and found the detection method used by SURF to perform the best. The found that the detection methods used by SIFT performed acceptably, but worse than SURF. Noble [61] compared the feature detection and matching methods available in OpenCV [62]: SIFT, SURF, BRISK, ORB, KAZE, and AKAZE (Accelerated KAZE). In feature detection, he found the methods used by SURF and BRISK to perform well. SURF had the largest number of features detected, with a moderate run time, and BRISK had a reasonable number of features detected, with the shortest run time. In his experiments, SIFT and SURF resulted in the highest number of matches. BRISK produced about half the number of matches as SIFT and SURF, but in about half the run time.

Feature tracking and optical flow

Feature tracking and optical flow are methods of improving the speed of feature detection and matching for ordered, high frame rate images. Feature tracking uses a motion model to speed up matching. Given a set of landmarks, the location of the landmarks in a new image can be predicted from the previous camera pose and an estimate of the motion from the previous frame to the new frame. The predicted location of the landmarks in the image can be used to reduce the number of comparisons for feature matching. The motion model can be supplied by other sensors (wheel odometry, inertial measurement unit (IMU), etc.) or by extrapolating from previously estimated motion.

Optical flow is a direct method of feature matching. The methods covered so far are considered sparse and indirect: sparse because salient features are detected in the images, and indirect because a descriptor is extracted and used for matching. Dense methods consider all the pixels in the image, while semi-dense considers a large number of pixels in the image (typically edges). With such a large number of features descriptor extraction is too computationally expensive, so direct methods are required, where correspondences are identified by pixel intensity. The downside of direct methods are that they are very sensitive to lighting changes, which can be caused by shadows from moving objects or, in the underwater case, sunlight flicker. Direct methods also require accurate photometric camera calibration to remove vignetting, which is a reduction of brightness towards the periphery of an image. Artificial lighting mounted to the camera system also produces uneven lighting, but this can compensated for by measuring and modelling the effect of artificial light.

Direct methods can also be combined with sparse feature detection. A popular method is referred to as Kanade-Lucas-Tomasi (KLT) optical flow [63]. Initially features are detected in an image, then in subsequent images the movement of the features is determined by changes in pixel intensity. Gradually features are lost due to changes in appearance,

and when the number of tracked features drops below a certain threshold new features are detected and added to the set. By reducing the frequency of feature detection, and eliminating descriptor extraction and matching, sparse optical flow is significantly less computationally expensive than sparse, indirect methods [64]. It is also more robust than dense or semi-dense, direct methods, as features are tracked over a shorter time and have less appearance change.

Motion estimation

Given corresponding features, the relative pose between two images can be determined. There are three types of methods: 2D to 2D, 3D to 3D, and 3D to 2D. With 2D to 2D, features are parameterised by their position in the image, and the resulting relative pose is arbitrarily scaled. Typically Nistér's 5 point algorithm is used [65].

With 3D to 3D, features are parameterised by their three dimensional position in the world, and the relative pose is the pose that minimises the Euclidean distance between the corresponding points. 3D to 3D results in a pose that is correctly scaled, but requires a stereo camera to determine the three dimensional position of each point in every frame.

3D to 2D is also correctly scaled, but Nistér *et al.* [66] state that it is more accurate. With 3D to 2D the features from the previous frame are parameterised by their three dimensional position, and the features from the new frame are parameterised by their image position. The relative pose between frames is found by minimising the error of reprojecting the features into the new frame, which is a better representation of the real sensor error. 3D to 2D pose estimation is possible with both stereo and monocular cameras. In the monocular case, the 3D position of features can be determined by optimisation, after they have been viewed across a wide range of viewpoints. This is referred to as initialisation.

Often a number or correspondences identified by feature matching will be erroneous. To improve accuracy and robustness, random sample concensus (RANSAC) [67] is usually incorporated in the the motion estimation step. Rather than estimating relative pose with all corresponding features, a pose is estimated from a random sample of the correspondences. From the remaining correspondences, the correspondences that do not fit the model of the estimated pose are excluded as outliers. Using the inliers the process is repeated, estimating a pose from a sample and excluding outliers, until a maximum number of iterations is reached. The relative pose is then estimated using all the inliers. Maximum likelihood sample concensus (MLESAC) [68] and maximum a posterior estimation sample concensus (MAPSAC) [69] are related methods that improve upon RANSAC.

With a monocular camera, 2D to 2D or 3D to 2D (after initialisation) motion estimation is possible, however in either case the relative pose is arbitrarily scaled. There are a number of methods for correcting the scale.

For a wheeled vehicle, wheel odometry can be used to correct scale [70]. This is possible even if the robot is operating on undulating terrain [71]. For a wheeled vehicle operating

on a flat, level surface, the ground plane can be used to estimate the scale of motion [72]. This method has the advantage of relying on no other sensors, although the height of the camera above the ground must be known and constant.

If an IMU is available it can be used to provide scale. The simplest method of combining IMU data with VO is to use an extended Kalman filter for sensor fusion [73], although efficiency can be improved using a method called preintegration [74]. VINS-Mono [75] is a popular publicly available system for visual-inertial odometry (VIO).

Windowed bundle adjustment

The final step in VO is typically an optimisation of a fixed number of recent frames, which helps to reduce pose drift. Relative pose estimation considers information from two views only, however the landmarks that produce features in the image are often visible across multiple frames. There are two methods of optimisation, referred to as pose graph optimisation and bundle adjustment. Pose graph optimisation minimises the differences between the estimated relative poses between images, and the estimated position of landmarks in the images. Bundle adjustment optimises camera poses and landmark positions by minimising reprojection error: the difference between the expected position of a landmark in an image and the actual position in the image. In recent years bundle adjustment has become the preferred method of optimisation [76].

The process of bundle adjustment is covered in more detail in the next section. Although the next section deals with global bundle adjustment, which performs optimisation on all camera poses and landmarks, the process is identical to windowed bundle adjustment. Windowed bundle adjustment optimises only a fixed number of recent frames to ensure that the result is returned quickly, which is important for online operation.

2.4 Bundle Adjustment

Images suffer from inaccuracies due to sensor noise and calibration error, and therefore there is no exact solution for camera poses and landmark positions. Optimisation is used to find the best solution: the solution that minimises the difference between the actual positions of the landmarks in the images, and the expected positions in the images, given optimised camera poses and landmark positions in the world. This difference is referred to as residual error, and the optimisation process is referred to as bundle adjustment. The bundle adjustment problem is solved with a non-linear least squares algorithm, typically Levenberg-Marquadt [77]. g2o [78] and Ceres Solver [79] are popular libraries for non-linear least squares optimisation.

There are different parameterisations of bundle adjustment, that differ in the way they represent the landmark positions. Cartesian parameterisation is the most straight forward, and most commonly used. Landmarks are represented by an x, y and z coordinate, relative to a global origin point. In inverse depth parameterisation [80] landmarks are represented

by an azimuth and elevation angle in the frame of an observation, and by the inverse of the distance between the observation and the landmark. Inverse depth parameterisation performs better for distant features, that have a low parallax angle between observations. Parallax parameterisation [81] shares the azimuth and elevation angle from inverse depth, but represents the depth of a landmark using the angle between the landmark and camera positions of two observations of the landmark. Like inverse depth, parallax parameterisation performs better than Cartesian parameterisation for distant features. Zhao *et al.* [81] found that it performs better than inverse depth for another type of features that have low parallax: features that are in line with the motion of the camera.

Bundle adjustment can be monocular or stereo. In the monocular case the result is not correctly scaled, as with motion estimation, however all values do share a single common scale. In the stereo case the observations of the landmarks in both cameras are included in the optimisation, and the relative pose between the two cameras (which is determined through calibration) is included as a constraint, which forces the scale to be correct.

2.5 Structure from Motion and Visual SLAM

VO is focused on estimating the relative motion of a camera. If the accurate structure of a scene is desired, SfM or visual SLAM are required. SfM incorporates feature detection, description, and matching, as well as bundle adjustment to estimate globally consistent camera poses and scene structure. Visual SLAM extends VO with loop closure (detecting similar non sequential images for feature matching) and global bundle adjustment to refine camera trajectory and scene structure. A number of SfM and visual SLAM systems have been developed, some of which are available for evaluation.

Structure from Motion systems

Examples of SfM systems include Bundler [82], VisualSfM [83], multi-view environment (MVE) [84], OpenMVG [85], COLMAP [86], and AliceVision [87]. As is typical with SfM, these systems are monocular only, and use feature detection, description, and matching. All use SIFT feature detection and description, although some have additional methods available. Typically images are matched by exhaustively comparing each image with every other image in the data set. VisualSfM and AliceVision use a vocabulary tree to store the descriptors extracted from each image, which enables faster comparisons and speeds up the matching process. MVE improves the speed of matching by doing initial matching on lower resolution images, and OpenMVG has an option to only evaluate a certain number of neighbouring images in an ordered data set. All perform global bundle adjustment after camera poses and landmarks have been initialised. Some will also perform incremental bundle adjustment, bundle adjustment after each new image is added, to improve the robustness and computation time of the final bundle adjustment.

Visual SLAM systems

Examples of visual SLAM systems include:

- PTAM [88] and Stereo PTAM [89]
- DTAM [90]
- SVO [91]
- LSD-SLAM [92] and Stereo LSD-SLAM [93]
- ORB-SLAM [94] and ORB-SLAM2 [95]
- ProSLAM [96]
- DSO [64] and Stereo DSO [97]

Parallel tracking and mapping (PTAM) is an early example of online visual SLAM. It is intended for augmented reality applications, and targeted at mobile devices. It splits the process into two parallel threads: tracking and mapping. In the tracking thread features are initially detected with FAST. Then, using a motion model, the positions of the features in the next image are predicted. A small number of coarse features are compared by normalised patches around the features to find correspondences and update the pose estimation by the motion model. Using the updated pose, a larger number of features is projected into the current image and identified, then a refined pose is estimated from all correspondences. The mapping thread will insert a key frame if the tracking is good and the distance to the nearest existing key frame is greater than a certain threshold. It then performs bundle adjustment either locally or globally, if the local or global residual error is larger than a threshold. PTAM was extended to stereo by Pire *et al.* [89]. They used Shi-Tomasi [63] detection with a BRIEF descriptor, then identified matches by comparing descriptors. Stereo correspondences are used to constrain motion estimation and bundle adjustment to the correct scale.

Dense tracking and mapping (DTAM) is a system that uses all pixels in the image for tracking. It is a monocular method that, once a model of the scene has been initialised, estimates the camera pose by generating synthetic images of the model from new camera poses and selecting the image which best matches the actual image. The system has no bundle adjustment, but has the advantage that it creates a dense reconstruction of the scene online. Semi-direct visual odometry (SVO) is a monocular system that initialises landmarks in the world using feature detection, description, and matching, then tracks new images against the landmarks using a direct method. Tracking is similar to the model based image alignment of DTAM, however only sparse landmarks are used. When a new key frame is selected features are detected and descriptors are extracted, and the features are matched with existing features to add new landmarks to the map. In SVO the tracking thread is direct, while the mapping thread is indirect. Large-scale direct SLAM (LSD-SLAM) is another direct visual SLAM system and is semi-dense, using edges from the images rather than all pixels or salient features. A stereo version of LSD-SLAM is also available [93]. Direct sparse odometry (DSO) is by the same authors as LSD-SLAM. It is

also direct, but sparse. The authors have found that image data is highly redundant, and the benefit of more pixels degrades quickly [64]. DSO has also been extended to stereo by Wang *et al.* [97].

ORB-SLAM is a sparse, indirect system. It uses ORB features and a motion model for fast feature detection, description, and matching. ORB-SLAM2 adds support for stereo and depth cameras to ORB-SLAM, both of which enable correctly scaled visual SLAM. ProSLAM is a recent visual SLAM system, but it focuses on simplicity rather than performance or robustness. It is intended for teaching and learning. It is stereo only (to avoid complicated monocular initialisation) and uses FAST feature detection with BRIEF descriptors, and a motion model for faster matching. It does not have bundle adjustment, instead using pose graph optimisation.

2.6 3D Reconstruction

Although SfM and visual SLAM generate scene structure, the result is only sparse if a sparse system is used. There are a number of methods of creating a dense 3D reconstruction of a scene from images. They can be divided into single and multiple viewpoint, and monocular and stereo. Single viewpoint monocular methods typically rely on semantic cues from the image and typically only provide a relative depth estimate for the image, while stereo methods utilise a calibrated stereo camera for accurate 3D reconstruction. Multi-view reconstruction methods can, given known camera poses (such as from SfM or visual SLAM), generate a 3D reconstruction from images. In the monocular case the camera poses are scaleless, and therefore the 3D reconstruction is scaleless. In the stereo case an accurate 3D reconstruction is possible.

Single viewpoint 3D reconstruction

Underwater monocular methods

With underwater images it is possible to exploit image degradation effects to generate a coarse depth estimate. Schechner and Karpel [11] and Treibitz and Schechner [98] used a polarisation filter at multiple orientations to determine backscatter, and therefore scene depth. Carlevaris-Bianco *et al.* [12] exploited the fact that, underwater, absorption of red light is significantly greater than blue or green light. For a given patch of the image, the difference between the maximum red and maximum blue/green values gives an estimate for the absorption and depth of the patch.

Another method comes from outdoor (in-air) haze removal. He *et al.* [99] observe that, in haze free outdoor images, any patch of the image will tend to have a number of dark pixels (excluding patches of the sky). These dark pixels are shadowed areas, and will have an intensity close to zero. In a hazy outdoor image these dark regions will be brightened by haze, and the greater the depth of the region, the more haze and increase in intensity. These difference between the darkest pixels in an image patch and zero intensity

is a relative estimate of depth, called the dark channel prior. Drews *et al.* [100] improved the dark channel prior for use underwater, by ignoring the red channel, which suffers from significant attenuation.

Although these monocular methods generate a relative depth estimate suitable for image enhancement, they lack the accuracy, robustness and correct scale desired for 3D reconstruction.

Dense stereo correspondence

Scharstein and Szeliski [101] created a taxonomy for, and evaluated the performance of, dense stereo correspondence algorithms. Broadly speaking, dense stereo correspondence algorithms can be divided into two categories: local and global. Local algorithms calculate disparity, the pixel distance between corresponding points (which can be considered as the inverse of depth) using only the information in a local area. Global algorithms determine disparity for either a line or the whole image by solving an optimisation problem, which assumes a certain amount of smoothness in the disparity values. Global methods can be more accurate than local methods, but have greater computational complexity and memory usage.

Stereo correspondence algorithms typically require rectified images as input. Using parameters found by stereo camera calibration, the images can be transformed so that epipolar lines are horizontal: a point in the scene will have the same vertical position in one image as the other. This allows the search for correspondences to be restricted to the matching horizontal line (scan line) from the other image.

Local methods determine the disparity of a pixel using a fixed region around the pixel. The area (or block) around a pixel in one image is compared to areas around pixels along the scan line in the other image. The difference between the windows is calculated and aggregated, and the most similar window determines the disparity of the pixel. Further processing can give a sub-pixel resolution result.

The local stereo correspondence algorithm described by Konolige [102] is popular, as it is available in the OpenCV library [62]. It is an implementation of the work of Kanade *et al.* [103]. Images are first filtered with a Laplacian of Gaussian kernel, and window costs are compared by absolute difference. Additionally there is a left/right consistency check. The disparity is determined for the left image, then for the right image, and the disparity value for a pixel is only considered valid if the left and right values match.

A key issue with local methods is the need to define block size. Larger blocks give more reliable matching, since more information is contained in a block. However a large block size results in less detail, as the depth of any objects smaller than the block size tends to be lost.

Examples of global methods are dynamic programming, graph cuts [104], and belief propagation [105]. Dynamic programming is used to compute disparity by finding a lowest cost path between two corresponding scan lines. The solution enforces smoothness along

the horizontal line but there is no vertical consistency, leading to horizontal streaks. Graph cuts or belief propagation for stereo correspondence are consistent in two dimensions and typically produce the best results, but are computationally expensive and unsuitable for online operation or with large input images.

Hirschmuller [106] introduced a method they refer to as semi-global matching. Their method approximates 2D optimisation with multiple 1D optimisation problems. They show that semi-global matching is almost as accurate as global methods with much less computation. This algorithm is also available in the OpenCV library.

Geiger *et al.* [16] use sparse features, which they refer to as support points, to generate a dense disparity image. They show that their method is faster than global methods while producing a better result than local methods, without the need for a defined window size. First support points are found using Sobel filter responses. The matches are checked for consistency and any outliers or ambiguous matches are removed. Pixels in a small neighbourhood around a support point in the left image are copied to the matching location in the right image, then all the pixel intensities are averaged to create a predicted right image, which looks like a blurred version of the actual right image. The predicted right image is then refined with the actual right image to determine disparity estimations. This is repeated in the opposite direction, with a predicted left image and estimated disparities generated from the right image, and only the pixels with matching disparity values are considered valid. Furthermore, any segments with an area less than 50 pixels are removed. They refer to this method as efficient large-scale stereo matching.

Underwater dense stereo perception

Some research has focused on issues specific to stereo perception underwater. Swirski *et al.* [107] found that sunlight flicker patterns can be beneficial to stereo perception. The patterns add additional visual texture to the scene and, because these patterns are constantly changing, a number of disparity images taken by a fixed stereo camera can be combined to improve quality. Roser *et al.* [15] used a disparity map to perform spatially variant image enhancement, then repeated disparity estimation and image enhancement. They found that although image enhancement improved, the disparity image did not show significant improvement. Negahdaripour and Sarafraz [108] observed that while many underwater stereo vision systems account for underwater light attenuation, few had been designed to handle significant backscatter. They incorporated backscatter and attenuation into the cost calculation, and improved disparity estimation in turbid water.

Multi-view Reconstruction

SfM and visual SLAM estimate the camera poses of images. With this information a multi-view reconstruction can be generated. In the monocular case, multi-view stereo is used to generate a point cloud of the scene. Like (calibrated) stereo reconstruction, multi-view stereo compares image patches from two images along epipolar lines to determine

depth. With rectified stereo images the epipolar lines are horizontal, but with multi-view stereo there is significant variation. Patch-based multi-view stereo (PMVS) and clustering views for multi-view stereo (CMVS) are two popular libraries for multi-view stereo, and are commonly integrated into SfM systems. They are based on the work of Furukawa and Ponce [109] and Furukawa *et al.* [110].

In the stereo case the point clouds generated from single views can be merged into one multi-viewpoint cloud. From a point cloud there are a number of algorithms to generate a surface mesh. Popular algorithms include Poisson surface reconstruction [111], moving least squares [112], and marching cubes [113].

There are a number of examples of underwater stereo multi-view reconstruction, although most focus on seafloor mapping with an AUV, where the camera is moving smoothly at a relatively far distance from the target. Beall *et al.* [114] and Henderson *et al.* [115] share a similar process: capture stereo images of the seafloor, generate point clouds, determine camera poses with SLAM, combine the point clouds and generate a mesh, then generate a texture for the mesh. Johnson-Roberson *et al.* [116] also created a reconstruction of the seafloor, but generated a mesh from single viewpoint clouds, then combined the meshes.

Sedlazeck *et al.* [117] created a reconstruction of a hydrothermal vent and a shipwreck, from monocular images captured by an ROV. The ROV operates in very deep water and the camera housing requires a thick front port, so they integrated a correct underwater camera model into the SfM system.

3

Underwater Visual Odometry

3.1 Introduction

A goal of the SPIR project is to create a 3D reconstruction of submerged bridge piles, to aid in cleaning the piles of biofouling and inspecting for damage. Due to cost and simplicity, passive stereo vision has been selected as the sensing modality.

VO is a first step towards creating a 3D reconstruction of an object from images, but there are a number of difficulties caused by the environment in which the ROV operates. Visibility is often poor, ambient lighting is highly variable, and there can be a significant amount of floating material. This chapter looks at underwater stereo camera calibration, exposure control, and image enhancement, for the purpose of visual odometry. Finally, visual odometry in the underwater environment encountered by the ROV is evaluated.

The DUO M [118] stereo camera (shown in Figure 3.1) was selected for its compact design. The camera has a global shutter and 30 mm baseline that allows for a close sensing range. It is a monochrome camera with a relative large pixel size of 6.0 µm × 6.0 µm, which gives it good light sensitivity. Because of the camera's short baseline, domed front ports are not practical. Domed ports of a suitable size are not commercially available. A 3 mm flat front port was used, with the camera mounted as close to the port as possible.

The camera has been attached to the pan and tilt unit on the ROV, which directs the high pressure water jet used for cleaning. During data collection the pan and tilt unit is reset to its home position, with the camera aimed directly forward. The ROV has been designed with a front bumper that generally prevents the pile from being closer than approximately 0.3 m from the camera. The ROV also has grasping arms which grasp on to the pile, and ensure that the ROV remains in a fixed position during the cleaning process. The grasping arms can also be positioned in a loose grasp which allows the ROV to move around the pile, while preventing it from drifting away from the pile completely. The front bumper and grasping arms limit the expected distance from the camera to the pile to between approximately 0.3 m to 1 m.

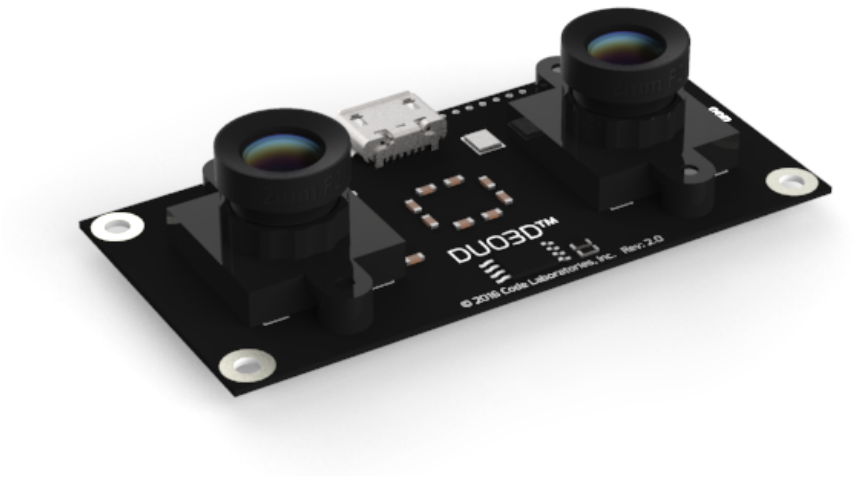


Figure 3.1: The DUO M stereo camera

3.2 Camera Calibration

Accurate camera calibration is essential for accurate results in VO, single viewpoint perception, SfM, and visual SLAM. For a monocular camera the aim of calibration is to find the intrinsic parameters (focal length and principal point) and distortion coefficients of the camera. Intrinsic parameters are used to convert coordinates in the image to a (scaleless) position in the world, and distortion coefficients are used to remove distortion in the image introduced by the lens system. For a stereo camera, camera calibration also finds extrinsic parameters (the relative rotation and translation between the two cameras) and rectification matrices. The extrinsic parameters are used to determine a correctly scaled world position from corresponding image points in each camera. Rectification matrices are used to transform the images so that epipolar lines (lines of corresponding points) between the stereo images are horizontal.

The DUO stereo camera comes factory calibrated, however for maximum accuracy a more recent calibration is preferred. The DUO includes a calibration application that will update the parameters stored in the device, however this calibration application presents a number of difficulties:

- It does not show all parameters to the user, so parameters from the calibration application cannot be used to rectify the images with other software.
- It is online only, which is difficult to do when the camera is on the ROV and in the water.
- It does not allow modifying the parameters that are saved into the device.

The application appears to be based on OpenCV, so a new calibration application was written using OpenCV that takes image files (rather than live data) and saves all parameters to text files. The DUO has a FOV of 165, which would be considered a fisheye

lens, however the DUO calibration application does not use the fisheye distortion model that is available in OpenCV. The DUO calibration application uses the rational polynomial distortion model, which has 6 radial parameters, so the custom OpenCV calibration application also uses this distortion model.

The instructions for the DUO calibration application state to hold the calibration board in front of the camera, directly facing the camera and in the middle of the frame, and then slowly move the board in an outward spiral. Other calibration software suggests moving the calibration board around the whole frame, with a wide range of skew and distance from the camera. Following this procedure with the DUO calibration application was found to consistently fail, and the same is true of the custom calibration application. If the board has too much skew, or is at the extremes of the frame, it was highly likely to fail. Calibration mages were collected with moderate skew and frame coverage, so that calibration would be successful, and using the custom calibration application calibration was successful performed in-air and underwater.

The custom calibration application could be improved by implementing the fisheye distortion model. The fisheye distortion model may be more suitable for the DUO camera in-air, however when the camera is underwater the field of view is reduced significantly, to the point that there may be little advantage over the rational polynomial model.

3.3 Exposure Control

The varying and high dynamic range lighting typically found in outdoor environments creates difficulty for computer vision. Scenes can range from dark to bright, or a combination of dark and light areas. A key question when capturing an image with a camera is what exposure parameters to use. Cameras commonly implement an auto exposure system, which selects parameters automatically. The simplest method of auto exposure is to control the exposure parameters to achieve a mean intensity value for the image, but this method tends to fail for outdoor robotics where scenes are often unevenly lit. If a scene is unevenly lit, setting exposure to achieve a target mean intensity can result in the dark areas being underexposed and the bright areas being overexposed, resulting in much of the visual information available in the scene being lost.

The camera system used for the SPIR project is a DUO stereo camera in an underwater housing, with two external lights. The camera system is used to collect images of submerged bridge piles, an environment where the scene lighting can vary significantly. The ideal case is to have the pile well lit by artificial lighting with weak background lighting. This occurs when the camera is observing the pile in deeper water or when sunlight is being blocked by the bridge, and in this case simple auto exposure tends to work well. Another case is that the pile is strongly lit by sunlight, with moderate background lighting. This occurs when the camera is observing the pile in shallow water with the sun behind the camera illuminating the pile and water around the pile. Simple auto exposure can work adequately in this case. The most difficult case is when the pile is weakly lit with

a strongly lit background. This occurs when the camera is in shallow water with the sun behind the pile. The pile is lit by artificial lighting, but this is weak relative to the sunlight illuminating the water behind the pile. In this case the simple auto exposure method can result in an underexposed pile, and little visual information from the image.

The aim of this section is to determine a method of controlling exposure that captures adequate visual information for further computer vision tasks. The camera used on the ROV allows for controlling exposure through shutter speed and digital gain. While digital gain can improve the appearance of an image to an observer it does not add any information to the image, and actually results in the loss of information in high intensity areas, so it has been disabled with only the shutter speed used to control exposure. For a moving camera, slower shutter speeds can result in a blurry image. Through experimentation it was found that a exposure time of 5 ms would result in moderate but acceptable blur, given the fastest expected movement of the ROV and camera. Experiments were limited to less than 5 ms exposure time.

Data collection

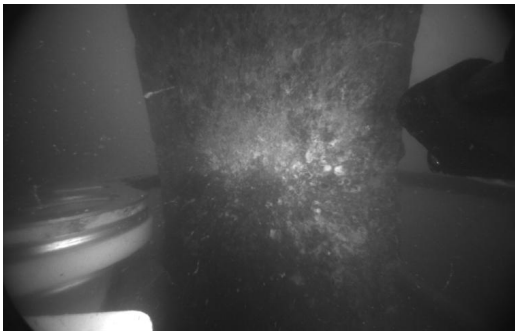
Data was collected on site at a real bridge pile, in various positions to give varying ambient lighting. All images were captured with 0 digital gain, with only camera position, artificial lighting level, and exposure time varied. Due to the time and cost required to test the ROV on site, data was only collected at one location and, for the purpose of this research, useful data was only collected on one occasion. Although water turbidity has a significant effect on underwater visibility it was not possible to vary the turbidity of the water for testing, and the turbidity of the water was not measured. The turbidity of the water during testing is considered to be typical of the water in which the ROV will operate.

Four camera positions were evaluated:

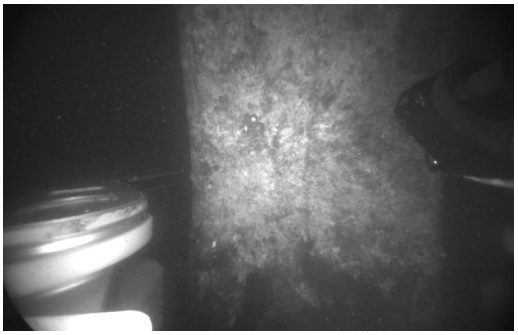
- Deep (approximately 1.5 m) with the sun behind the camera (pile moderately illuminated by sun, background weakly illuminated by sun).
- Deep with the sun behind the pile (pile and background weakly illuminated by sun).
- Shallow (approximately 0.5 m) with the sun behind the camera (pile strongly illuminated by sun, background moderately illuminated by sun).
- Shallow with the sun behind the pile (pile weakly illuminated by sun, background strongly illuminated by sun).

The positions were chosen to represent the extreme cases for ambient lighting. Each position is shown in Figure 3.2.

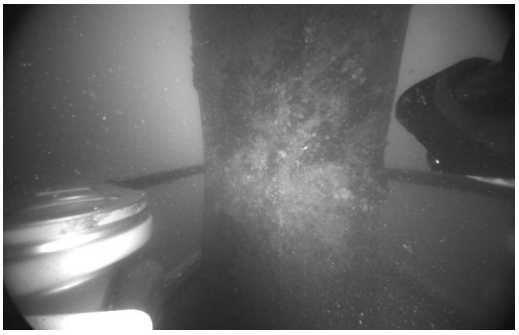
Three artificial lighting levels were evaluated: off, low, and high. Low is the minimum power of lights when on and high is the maximum power of the lights. Finally six exposure times were evaluated: 0.5, 1, 2, 3, 4, and 5 milliseconds. At each position the ROV grasps onto the pile, so that the position remains fixed while the artificial lighting level and exposure time are varied.



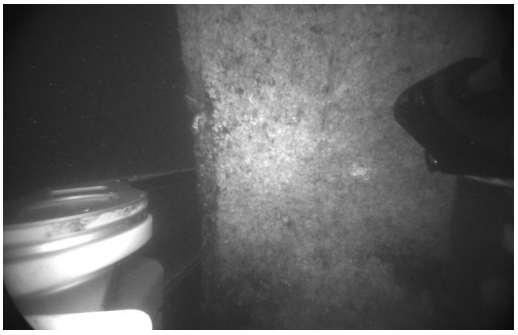
(a) Position 1



(b) Position 2



(c) Position 3



(d) Position 4

Figure 3.2: Examples of well exposed images at each position

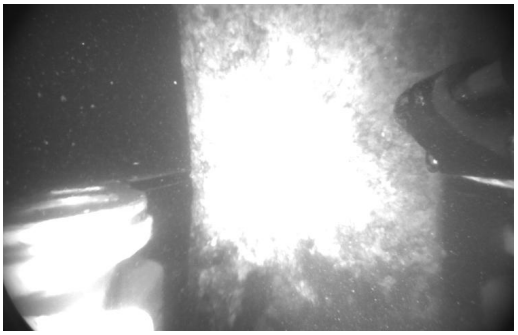
The power of the artificial lights fluctuates slightly, and the fluctuation is more significant at low power and shorter exposure times. To ensure that images compared are captured under similar conditions, for each configuration a number of images were captured and the image with the highest mean intensity was selected to represent the configuration.

Evaluation

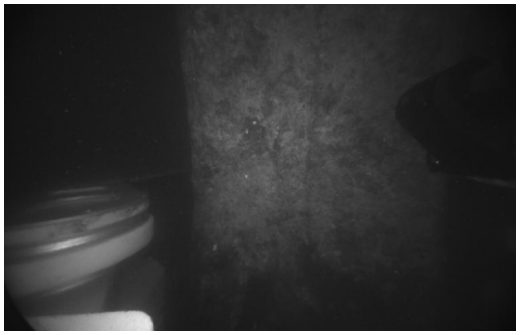
The quality of exposure is measured by:

- mean pixel intensity (as a percentage of the maximum pixel intensity);
- mean of the pixel gradient magnitudes;
- image entropy;
- number of SURF features detected in image;
- number of BRISK features detected in image; and
- subjective assessment of exposure ("poor", "average", "good", "very good")

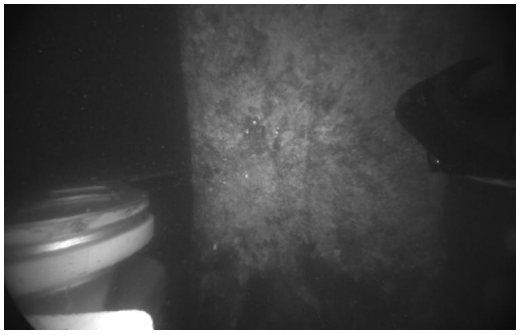
Pixel gradient magnitude were calculated with MATLAB's *imgradient* function, and image entropy was calculated with MATLAB's *entropy* function. SURF features were detected using MATLAB's *detectSURFFeatures* function, with a "MetricThreshold" value of 100. BRISK features were detected using MATLAB's *detectBRISKFeatures* function, with a "MinContrast" value of 0.05 and a "MinQuality" value of 0.1.



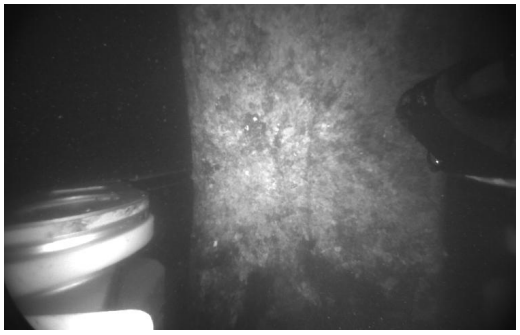
(a) "Poor"



(b) "Average"



(c) "Good"



(d) "Very good"

Figure 3.3: Examples of subjective descriptions of exposure

The subjective assessment of exposure refers to the exposure of the pile only. "Very good" means that there is no over or underexposure of the pile. "Good" means that there is a small amount of over or underexposure, "average" means that there is a moderate amount of over or underexposure, and "poor" means that there is significant over or underexposure. Figure 3.3 shows examples of the subjective descriptions of exposure.

Results

Table 3.1 shows the results of varying artificial lighting and exposure time with the camera in deeper water (approximately 1.5 m) and the sun behind the pile. In this position there is minimal sunlight illuminating the pile, and without artificial lighting the pile is barely visible. With the lights at their lowest setting, the pile was well exposed with a 4 or 5 ms shutter time. With the lights at their highest setting, the pile was well exposed with a 2 ms shutter time.

Table 3.2 shows the results of varying artificial lighting and exposure time with the camera in deeper water and the sun behind the camera. In this position there is a moderate amount of sunlight illuminating the pile and even without artificial lighting the pile is somewhat visible at a 4 or 5 ms shutter speed. With the lights at their lowest setting, the pile was well exposed with a 4 or 5 ms shutter time. With the lights at their highest setting, the pile was well exposed with a 2 or 3 ms shutter time.

Table 3.3 shows the results of varying artificial lighting and exposure time with the

Artificial lighting	Exposure (ms)	Mean intensity (%)	Mean gradient magnitude	Image entropy	SURF features	BRISK features	Subjective assessment
off	0.5	4.93	3.09	2.67	0	2	poor
off	1.0	5.77	3.34	3.44	0	2	poor
off	2.0	7.41	3.89	4.31	1	10	poor
off	3.0	9.18	4.55	4.88	6	18	poor
off	4.0	10.94	5.23	5.29	24	27	poor
off	5.0	12.73	5.89	5.61	42	41	poor
low	0.5	7.00	3.96	3.98	3	4	poor
low	1.0	10.16	5.36	4.97	15	18	average
low	2.0	15.71	8.60	5.91	126	251	average
low	3.0	21.72	11.60	6.49	254	618	good
low	4.0	27.83	15.62	6.88	439	1444	very good
low	5.0	33.67	18.75	7.17	592	2292	very good
high	0.5	11.00	6.49	5.31	60	79	poor
high	1.0	17.66	11.16	6.26	247	681	good
high	2.0	31.65	18.48	7.20	635	1297	very good
high	3.0	45.12	23.36	7.40	878	2297	average
high	4.0	55.10	22.25	7.08	817	3266	poor
high	5.0	64.55	22.32	6.68	816	2487	poor

Table 3.1: Results of varying artificial lighting and exposure time with the pile and background weakly illuminated by the sun

Artificial lighting	Exposure (ms)	Mean intensity (%)	Mean gradient magnitude	Image entropy	SURF features	BRISK features	Subjective assessment
off	0.5	5.38	3.35	3.12	0	1	poor
off	1.0	6.80	4.02	4.05	0	3	poor
off	2.0	9.59	5.42	5.01	2	20	poor
off	3.0	12.56	7.30	5.61	36	41	average
off	4.0	15.23	8.63	5.99	130	99	good
off	5.0	18.32	11.21	6.34	233	279	good
low	0.5	7.44	4.28	4.37	0	15	poor
low	1.0	10.92	6.33	5.34	16	33	average
low	2.0	17.66	10.21	6.31	172	212	good
low	3.0	24.67	13.88	6.87	466	545	good
low	4.0	31.07	18.75	7.24	709	2019	very good
low	5.0	36.90	21.60	7.44	822	3257	very good
high	0.5	12.23	7.17	5.62	48	74	average
high	1.0	19.87	12.17	6.54	328	527	average
high	2.0	33.62	20.42	7.36	790	1151	very good
high	3.0	43.45	23.21	7.49	945	2637	very good
high	4.0	52.60	21.13	7.12	840	1769	average
high	5.0	60.18	18.82	6.66	703	2197	poor

Table 3.2: Results of varying artificial lighting and exposure time with the pile moderately illuminated, and the background weakly illuminated, by the sun

Artificial lighting	Exposure (ms)	Mean intensity (%)	Mean gradient magnitude	Image entropy	SURF features	BRISK features	Subjective assessment
off	0.5	6.04	3.38	3.74	0	1	poor
off	1.0	8.20	4.04	4.69	1	3	poor
off	2.0	12.51	5.47	5.66	35	36	poor
off	3.0	17.16	7.14	6.28	73	100	poor
off	4.0	21.63	8.74	6.69	92	144	poor
off	5.0	25.64	10.22	6.98	109	185	poor
low	0.5	7.72	3.96	4.33	1	6	poor
low	1.0	11.51	5.31	5.34	8	16	poor
low	2.0	18.78	8.13	6.31	62	101	poor
low	3.0	26.56	11.21	6.91	120	208	average
low	4.0	35.39	14.18	7.37	202	187	good
low	5.0	42.93	16.38	7.41	277	327	very good
high	0.5	11.31	5.97	5.17	11	44	poor
high	1.0	18.10	9.37	6.09	57	80	average
high	2.0	29.69	14.26	6.89	245	261	good
high	3.0	39.34	18.41	7.27	403	765	very good
high	4.0	51.41	22.45	7.54	679	1529	very good
high	5.0	61.48	23.77	7.29	726	2738	good

Table 3.3: Results of varying artificial lighting and exposure time with the pile weakly illuminated, and the background strongly illuminated, by the sun

Artificial lighting	Exposure (ms)	Mean intensity (%)	Mean gradient magnitude	Image entropy	SURF features	BRISK features	Subjective assessment
off	0.5	6.18	3.71	3.73	0	3	poor
off	1.0	8.53	5.04	4.73	0	8	poor
off	2.0	13.28	8.06	5.75	43	50	average
off	3.0	18.53	11.62	6.39	199	299	good
off	4.0	24.14	15.36	6.84	393	762	good
off	5.0	29.46	18.90	7.15	636	1984	very good
low	0.5	8.21	4.84	4.65	1	10	poor
low	1.0	12.31	7.41	5.61	36	49	average
low	2.0	20.24	12.68	6.56	253	402	good
low	3.0	27.98	17.63	7.08	547	1221	very good
low	4.0	35.37	22.08	7.39	790	1758	very good
low	5.0	42.26	25.09	7.51	937	2302	very good
high	0.5	11.75	7.13	5.52	25	38	poor
high	1.0	20.04	12.67	6.55	250	225	good
high	2.0	35.68	22.51	7.41	768	1673	very good
high	3.0	49.91	24.76	7.25	900	2962	average
high	4.0	59.58	22.44	6.68	771	3046	poor
high	5.0	66.36	19.37	6.06	620	3131	poor

Table 3.4: Results of varying artificial lighting and exposure time with the pile strongly illuminated, and the background moderately illuminated, by the sun

camera in shallow water (approximately 0.5 m) and the sun behind the pile. In this position there is minimal sunlight illuminating the pile and a significant amount of light behind the pile, and without artificial lighting the pile is barely visible. With the lights at their lowest setting, the pile was well exposed with a 5 ms shutter time. With the lights at their highest setting, the pile was well exposed with a 3 or 4 ms shutter time.

Table 3.4 shows the results of varying artificial lighting and exposure time with the camera in shallow water and the sun behind the camera. In this position the pile is strongly illuminated by the sun, with a moderate amount of light behind the pile. Without artificial lighting the pile is visible with a shutter speed longer than 2 ms, and well illuminated at 5 ms. With the lights at their lowest setting the pile is well illuminated at 3, 4 or 5 ms shutter speed, and with the lights at their highest setting the pile is well illuminated at 2 ms.

Discussion

The tables show that a mean intensity target could be an effective method of controlling exposure for an underwater ROV navigating around a pile. A target mean intensity of 50% would result in an overexposed image in most cases, however a mean intensity target of around 30% would result in well exposed images, for the images evaluated. The camera positions evaluated cover various ambient lighting conditions, however more evaluation is required to confirm that a mean intensity target is an adequate method of auto exposure for all ambient lighting conditions encountered.

A simple method of controlling exposure is to choose a fixed artificial lighting level and exposure time. The results suggest that a low lighting level with 4 ms exposure, or a high lighting level with 2 ms exposure would give well exposed images of the pile in most cases of ambient lighting. Again, more evaluation is needed to confirm that these values work for all ambient lighting conditions. For subsequent data collection a high artificial lighting level with 3 ms exposure time was found to work well in most conditions.

Mean gradient magnitude and image entropy appear to have a high correlation, with the peak mean gradient magnitude for a set generally corresponding to the peak image entropy. The peaks also generally correspond to the images that are subjectively well exposed, although there is a tendency towards overexposure. The results suggest that either mean gradient magnitude or image entropy would be a suitable metric for automatic exposure control.

The results also show that the maximum number of SURF and BRISK features detected tends to correspond to overexposed images, especially for detecting BRISK features. Both feature detectors operate at multiple scales, so this could be due to overexposed images generating more features at larger scales. Regardless, the computational cost of feature detection makes it unsuitable as a metric for automatic exposure control.

The conditions in which the camera operates are unique and somewhat predictable, compared to conventional outdoor robotics. The camera moves around a pile, which is the

only object of interest. Anything else in the image is ignored for subsequent processing. As such, only the pile needs to be well exposed. Auto exposure could be improved by limiting calculation of the metric used to only the parts of the image which contain the pile. Section 3.5 discusses feature detection and matching for the purpose of estimating the motion of the camera. The features that result from this process generally belong to the pile. Auto exposure could be improved by only sampling around these features when calculating a metric that measures the visibility of the pile.

3.4 Image Enhancement

CLAHE is a popular method for improving the appearance of underwater images. The aim of this section is to determine if CLAHE is useful in improving feature detection and matching for images of a submerged pile in varying lighting positions.

Data collection

Stereo images were collected with the ROV in various positions around a number of bridge piles. Six stereo image pairs were selected that are typical of the underwater environment, with varying ambient lighting and amounts of biofouling, to evaluate the effect of CLAHE on feature detection and matching.

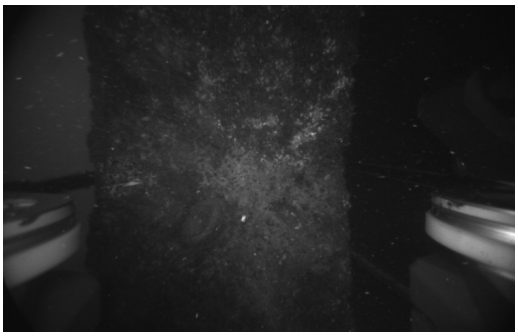
For each image, SURF, KAZE, FAST, and BRISK features were detected using the corresponding functions in MATLAB. The default parameters were used for each function. The process was repeated for the same images with CLAHE applied using the *adapthisteq* function in MATLAB. The results show the number of features detected in the left image of each pair.

Evaluation

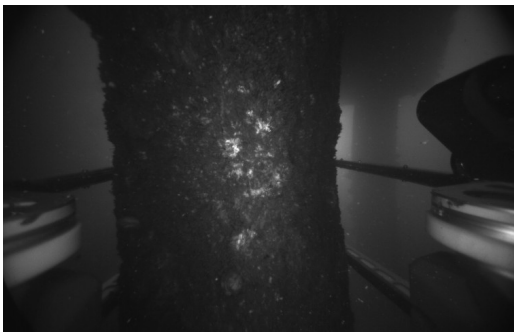
To evaluate feature matching, for all features (detected with and without CLAHE) feature descriptors were extracted with and without CLAHE applied to the image. Descriptors were extracted using MATLAB's *extractFeatures* function with the same method as used for detection, except for FAST detection which uses FREAK for description. The features from the left and right image were matched using MATLAB's *matchFeatures* function. Then the feature locations were rectified using the parameters generated by camera calibration, so that matching features would lie on the same horizontal epipolar line. Any matches with more than 2 pixels vertical distance were considered outliers and discarded. The results tables show the number of features matched before and after epipolar outlier removal, as well as the ratio between the two.

Results

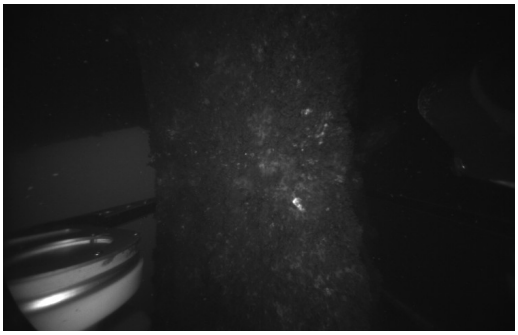
Table 3.5 shows the number of features detected in the left image of each pair, with and without CLAHE. In every image and for every method of feature detection CLAHE



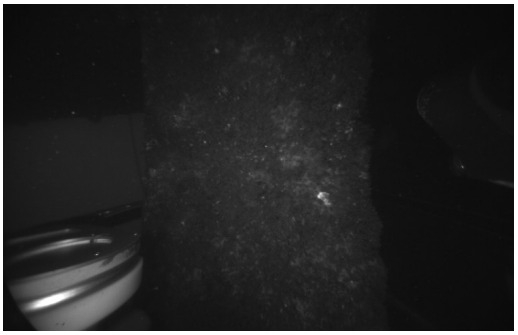
(a) Image pair 1



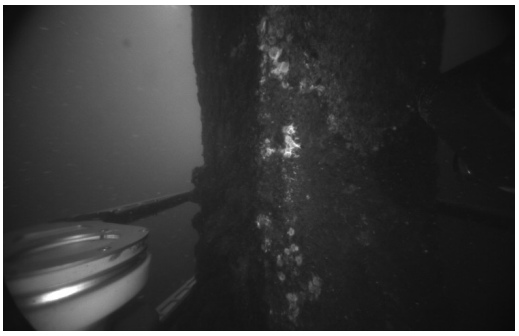
(b) Image pair 2



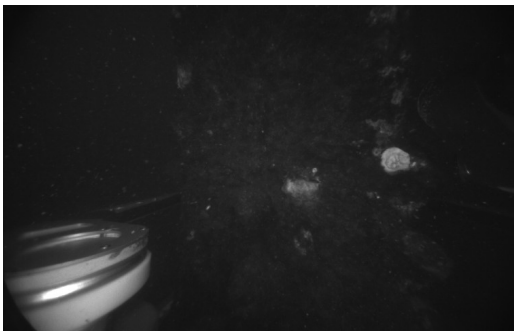
(c) Image pair 3



(d) Image pair 4



(e) Image pair 5



(f) Image pair 6

Figure 3.4: The left images of each image pair used to evaluate image enhancement

Image	SURF		KAZE		FAST		BRISK	
	Without CLAHE	With CLAHE	Without CLAHE	With CLAHE	Without CLAHE	With CLAHE	Without CLAHE	With CLAHE
1	11	355	796	3731	14	294	28	600
2	39	259	780	2680	50	222	113	500
3	10	126	386	2347	13	56	34	154
4	10	131	405	2250	15	66	43	168
5	59	238	739	2343	36	116	94	352
6	25	73	403	1792	10	22	46	95

Table 3.5: Number of features detected with and without CLAHE

Image	Without CLAHE			With CLAHE		
	Initial matches	Final matches	Ratio	Initial matches	Final matches	Ratio
1	9	8	0.89	6	5	0.83
2	26	24	0.92	21	19	0.90
3	1	1	1.00	1	1	1.00
4	2	0	0.00	2	0	0.00
5	36	35	0.97	31	31	1.00
6	10	9	0.90	10	9	0.90

Table 3.6: Number of SURF features (detected without CLAHE) matched with and without CLAHE before descriptor extraction

Image	Without CLAHE			With CLAHE		
	Initial matches	**Final matches**	**Ratio**	**Initial matches**	**Final matches**	**Ratio**
1	115	106	0.92	98	93	0.95
2	91	80	0.88	74	66	0.89
3	12	3	0.25	13	3	0.23
4	21	3	0.14	24	3	0.13
5	103	97	0.94	105	97	0.92
6	23	21	0.91	26	22	0.85

Table 3.7: Number of SURF features (detected with CLAHE) matched with and without CLAHE before descriptor extraction

Image	Without CLAHE			With CLAHE		
	Initial matches	**Final matches**	**Ratio**	**Initial matches**	**Final matches**	**Ratio**
1	235	220	0.94	210	204	0.97
2	240	228	0.95	179	172	0.96
3	21	9	0.43	26	11	0.42
4	41	8	0.20	45	13	0.29
5	323	307	0.95	267	258	0.97
6	100	91	0.91	97	86	0.89

Table 3.8: Number of KAZE features (detected without CLAHE) matched with and without CLAHE before descriptor extraction

Image	Without CLAHE			With CLAHE		
	Initial matches	**Final matches**	**Ratio**	**Initial matches**	**Final matches**	**Ratio**
1	793	731	0.92	696	659	0.95
2	645	601	0.93	468	440	0.94
3	48	14	0.29	68	24	0.35
4	137	23	0.17	127	27	0.21
5	742	678	0.91	571	521	0.91
6	243	204	0.84	260	227	0.87

Table 3.9: Number of KAZE features (detected with CLAHE) matched with and without CLAHE before descriptor extraction

Image	Without CLAHE			With CLAHE		
	Initial matches	**Final matches**	**Ratio**	**Initial matches**	**Final matches**	**Ratio**
1	1	1	1.00	3	3	1.00
2	17	17	1.00	14	14	1.00
3	2	0	0.00	2	0	0.00
4	2	0	0.00	2	0	0.00
5	11	11	1.00	11	11	1.00
6	3	3	1.00	2	0	0.00

Table 3.10: Number of FREAK features (detected without CLAHE) matched with and without CLAHE before descriptor extraction

Image	Without CLAHE			With CLAHE		
	Initial matches	**Final matches**	**Ratio**	**Initial matches**	**Final matches**	**Ratio**
1	35	34	0.97	32	31	0.97
2	22	22	1.00	22	22	1.00
3	3	1	0.33	2	0	0.00
4	2	0	0.00	1	2	2.00
5	17	17	1.00	19	19	1.00
6	0	0	–	0	0	–

Table 3.11: Number of FREAK features (detected with CLAHE) matched with and without CLAHE before descriptor extraction

Image	Without CLAHE			With CLAHE		
	Initial matches	**Final matches**	**Ratio**	**Initial matches**	**Final matches**	**Ratio**
1	1	1	1.00	1	1	1.00
2	22	20	0.91	17	15	0.88
3	1	1	1.00	0	0	–
4	0	0	–	0	0	–
5	24	22	0.92	19	18	0.95
6	9	8	0.89	3	3	1.00

Table 3.12: Number of BRISK features (detected without CLAHE) matched with and without CLAHE before descriptor extraction

Image	Without CLAHE			With CLAHE		
	Initial matches	**Final matches**	**Ratio**	**Initial matches**	**Final matches**	**Ratio**
1	43	42	0.98	31	31	1.00
2	52	50	0.96	25	22	0.88
3	1	1	1.00	2	0	0.00
4	1	1	1.00	1	1	1.00
5	51	48	0.94	31	28	0.90
6	8	6	0.75	5	5	1.00

Table 3.13: Number of BRISK features (detected with CLAHE) matched with and without CLAHE before descriptor extraction

increased the number of features detected.

Tables 3.6 to 3.13 show the results of feature matching with and without CLAHE. With the exception of KAZE, feature detection without CLAHE usually results in too few features to evaluate feature matching. Nonetheless the results have been included, as some images do have an acceptable number of matches. Generally, matching features using descriptors extracted from images enhanced with CLAHE tends to reduce the number of matches while demonstrating similar or a small reduction in accuracy (where accuracy is defined as the ratio of initial matches to matches after epipolar outliers have been removed). CLAHE for feature detection generally increases the number of initial matches with a small reduction in accuracy.

Discussion

The results suggest that the number of matching features found can be maximised by applying CLAHE image enhancement for feature detection, then using the original images for descriptor extraction. The KAZE algorithm offers the best data for comparing detection and extraction with and without CLAHE, as it can detect an adequate number of features even without CLAHE, which other algorithms have difficulty with. All detection algorithms have difficult with images 3 and 4, so they are an unreliable source of comparison.

Comparing the KAZE results without and with CLAHE for detection (Tables 3.8 and 3.9 respectively), it can be seen that with or without CLAHE for descriptor extraction, the number of matches more than doubles for all images when using CLAHE for detection, with a similar or small decrease in accuracy. When an adequate number of features have been detected by other algorithms without CLAHE, the data shows a similar trend.

It is understandable that image enhancement negatively effects descriptor extraction

and matching. Descriptors are generated from an area around the feature and, as they are designed to be robust to illumination changes, are concerned only with the relative intensity of the pixels in the region and not the absolute intensity, which CLAHE tends to improve. CLAHE enhances images by equalising the intensity histogram in a block around each pixel. CLAHE may alter the relative intensity in the region around a feature differently in different images, and may cause the descriptors that are generated for the same feature in different images to deviate.

3.5 Visual Odometry

Visual odometry is the process of determining camera trajectory from an ordered set of images. It produces camera poses and landmarks (features matched across multiple views), which are used as initialisation for global bundle adjustment. More accurate camera poses for initialisation will mean faster convergence, with less chance of converging to an incorrect local minimum. Better feature correspondences with fewer outliers will mean a more correct result from bundle adjustment.

Windowed bundle adjustment and key frame selection are commonly used to reduce drift, however the aim of the section is to compare feature detection and description algorithms, and for simplicity these have been excluded. Instead, the resulting camera trajectory is a concatenation of the estimated relative motion between each pair of sequential frames.

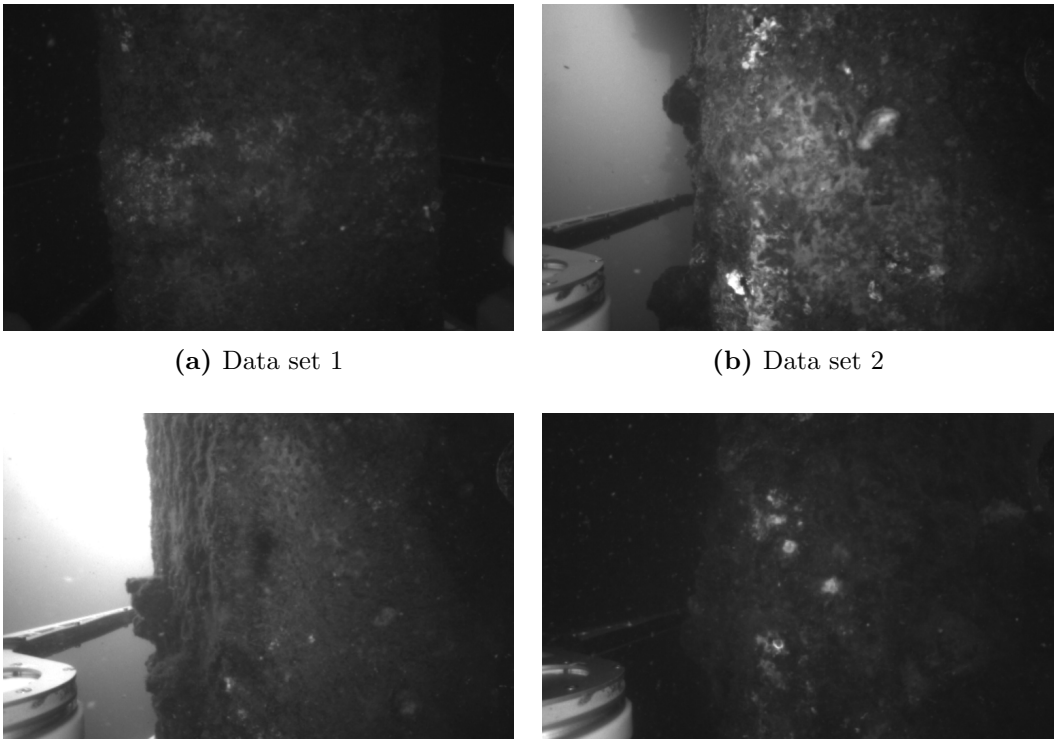
Data collection

Data sets were collected while an ROV moved around a concrete bridge pile. Four data sets were selected to cover a range of ambient lighting and water conditions. In all data sets the camera is remains between 0.3 m and 1 m from the pile, and the ROV sways left or right while rotating to keep the pile in the view of the camera. As collecting ground truth data for evaluation was not practical, all data sets start on one face and end on the opposite face, so that the performance of each feature detection and description algorithm can be compared by the alignment of the opposite faces of the pile.

Typical images from each data set are shown in Figure 3.5. In data set 1 the pile ranges from acceptably lit to well lit, with dark to moderate background ambient light and moderate floating material. In data set 2 the pile is generally well lit, with dark to moderate background ambient light and minimal floating material. In data set 3 the pile is generally acceptably lit, with moderate to strong background ambient light and moderate floating material. In data set 4 the pile ranges from poorly lit to acceptably lit, with a dark background and significant floating material. Data set 4 also has the most difficult movement, with the camera stopping or moving in the opposite direction in several places.

Four feature detection and description algorithms were compared:

- SURF



(a) Data set 1 **(b)** Data set 2

(c) Data set 3 **(d)** Data set 4

Figure 3.5: Typical images from each data set

- KAZE
- FREAK (with FAST feature detection)
- BRISK

The process of determining the camera poses of the data set is shown in Algorithm 1.

Features were detected using MATLAB's *detectSURFFeatures*, *detectKAZEFeatures*, *detectFASTFeatures*, or *detectBRISKFeatures* functions. The default parameters were used, with two exceptions:

- "MetricThreshold" for *detectSURFFeatures* was reduced from 1000 to 100, to increase the number of features detected; and
- "MinContrast" for *detectFASTFeatures* and *detectBRISKFeatures* was reduced from 0.2 to 0.1, to increase the number of features detected.

The parameters used for the *detectSURFFeatures* function were:

- "MetricThreshold": 100
- "NumOctaves": 3 (default)
- "NumScaleLevels": 4 (default)

The parameters used for the *detectKAZEFeatures* function were:

- "Diffusion": Region (default)

```
for Each stereo image pair do
    Enhance each stereo image;
    Detect features in each enhanced stereo image;
    Extract descriptors for all features using the original images;
    Rectify features;
    Remove any features that are out of the region of interest;
    Match features from left and right images;
    for Each stereo match do
        if Difference in left and right vertical postion > 2 pixels then
            Discard match;
        else
            Triangulate world position;
            if Distance from camera < 0.2 m or > 1.2 m then
                Discard match;
            else
                Keep landmark;
            end
        end
    end
    if Not first stereo image pair then
        Find matches between the features of the current and previous left images;
        Estimate the motion from the previous image pair to the current image
         pair using matching features;
    end
end
```

Algorithm 1: Stereo visual odometry

- "Threshold": 0.0001 (default)
- "NumOctaves": 3 (default)
- "NumScaleLevels": 4 (default)

The parameters used for the *detectFASTFeatures* function were:

- "MinQuality": 0.1 (default)
- "MinContrast": 0.1

The parameters used for the *detectBRISKFeatures* function were:

- "MinQuality": 0.1 (default)
- "MinContrast": 0.1
- "NumOctaves": 4 (default)

Feature descriptors were extracted using MATLAB's *extractFeatures* function, using the default parameters. The region of interest used removes features less than 128 pixels from the left of the image, or less than 20 pixels from the right of the image, because these areas usually contain the background and grasping arms of the ROV. For both stereo and sequential matching, MATLAB's *matchFeatures* function was used with the default parameters. MATLAB's *estimateWorldCameraPose* function was used to estimate the relative pose between corresponding landmarks from the previous stereo image pair and image features from the current left image. The default parameters were used. *estimateWorldCameraPose* uses the perspective-three-point algorithm and also removes erroneous correspondences using the M-estimator sample consensus algorithm.

Evaluation

To visualise the results a reduced number of camera poses and landmarks were plotted for each method and data set. Only 21 camera poses and the landmarks that belong to them were plotted.

The performance of each feature detection and description algorithm was evaluated by:

- plotting the number of matches between sequential image pairs; and
- plotting the landmarks generated against a model of the three faces of the pile.

The bridge piles in the data sets have a width of 0.36 m however, due to the thickness of the biofouling and possibly inaccuracies in camera calibration, a model with a width of 0.38 m was found to better match the landmarks generated. The landmarks were aligned with the model using MATLAB's *pcregistericp* function (using the default parameters), and the landmarks and model were plotted from a top down view so that the alignment of the three faces can be seen.

In some cases the landmarks generated were too inaccurate to align correctly. Instead of aligning to the model, all landmarks were plotted from as close to a top down view as possible. In the cases where the algorithm was not able to process the entire data set, the results were not plotted.

Results

Figure 3.6 shows plots of the number of matches between sequential image pairs (frames) for each feature detection and description algorithm and data set. In all data sets KAZE found significantly more matches than the other algorithms. For data sets 1 and 2 SURF and BRISK performed similarly, with FREAK moderately behind. For data set 3 FREAK and BRISK are not able to continue past approximately 30 frames, and for data set 4 they are not able to process past the first sequential pair of frames. Matches can be used as an indicator of the visibility of the image, and the plot for data set 4 suggests that visibility is poor until around frame 600, then significantly improves.

Figure 3.8 shows a top down view of all landmarks generated from data set 1 after alignment with a model. KAZE and BRISK appear to perform well, with the opposite faces generally parallel, and perpendicular to the second face. SURF also performs well, but the opposite faces are not as parallel as KAZE and BRISK. FREAK performs the worst, with the opposite faces moderately out of parallel alignment.

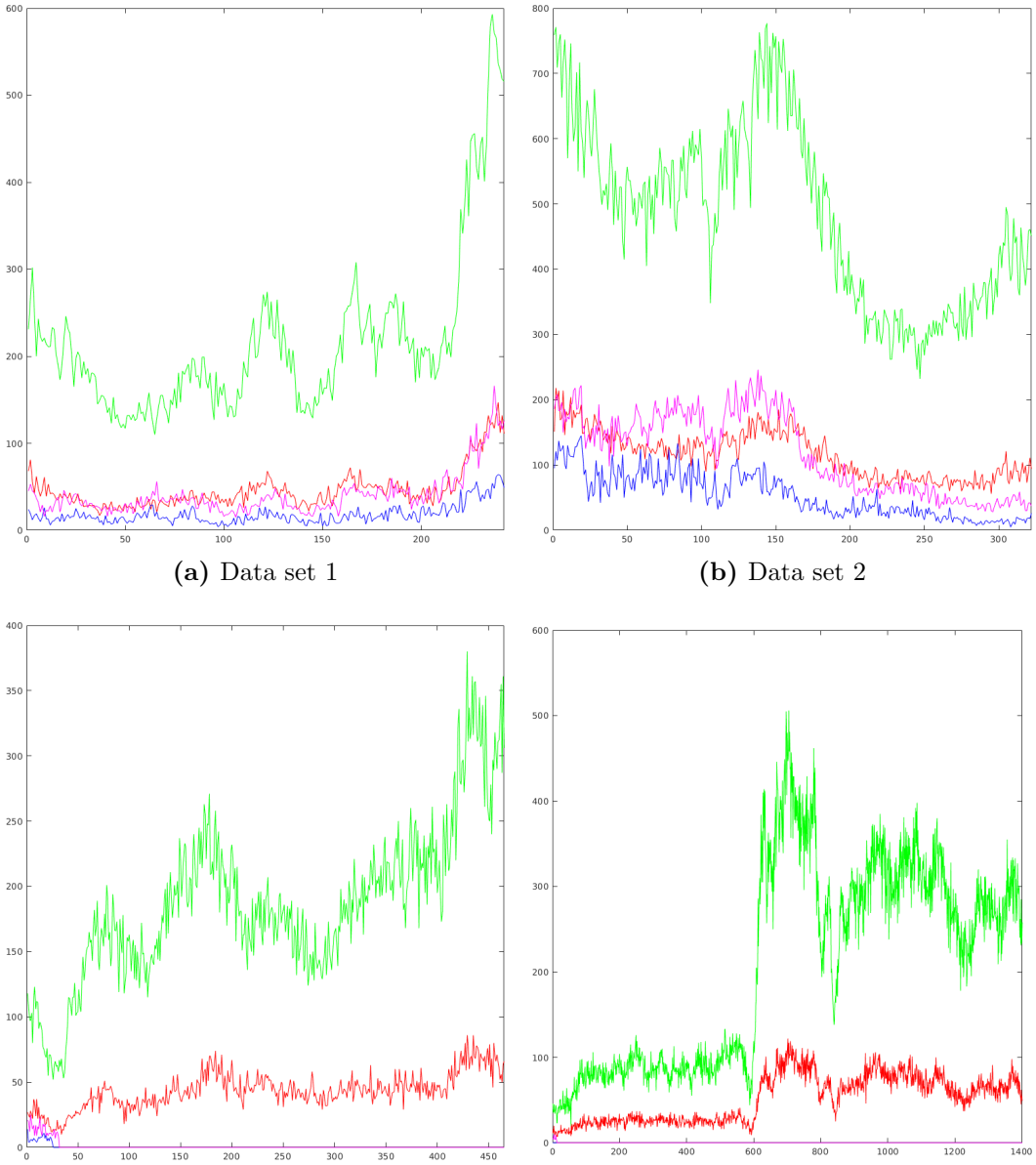
Figure 3.10 shows a top down view of all landmarks generated from data set 2 after alignment with a model, for the SURF and KAZE algorithms only. The landmarks from FREAK and BRISK are too out of alignment to be registered with the model, so the landmarks have been plotted from a generally top view in Figure 3.11. SURF and KAZE both perform very well, with the opposite faces generally parallel. BRISK also performs well, with the first and second faces mostly perpendicular, however the third face is out of alignment which causes the landmarks to fail to register with the model. FREAK performs the worst of the algorithms, with all three faces out of alignment.

FREAK and BRISK were not able to process all of data set 3, so their results have been omitted. Figure 3.13 shows that both SURF and KAZE perform well, although with some misalignment in the opposite faces.

FREAK and BRISK were also unable to process all of data set 4 and their results have again been omitted. SURF and KAZE were able to process the whole data set, however the landmarks generated are not accurate enough to register with the model. Looking at the plots in Figure 3.15 it is apparent that they have difficulty with the first face (the face on the right of the plot), and with moving from the first to second face. This likely corresponds to segment of the data set with a low number of matches shown in Figure 3.6d. The second and last faces appear to be perpendicular with both algorithms.

Discussion

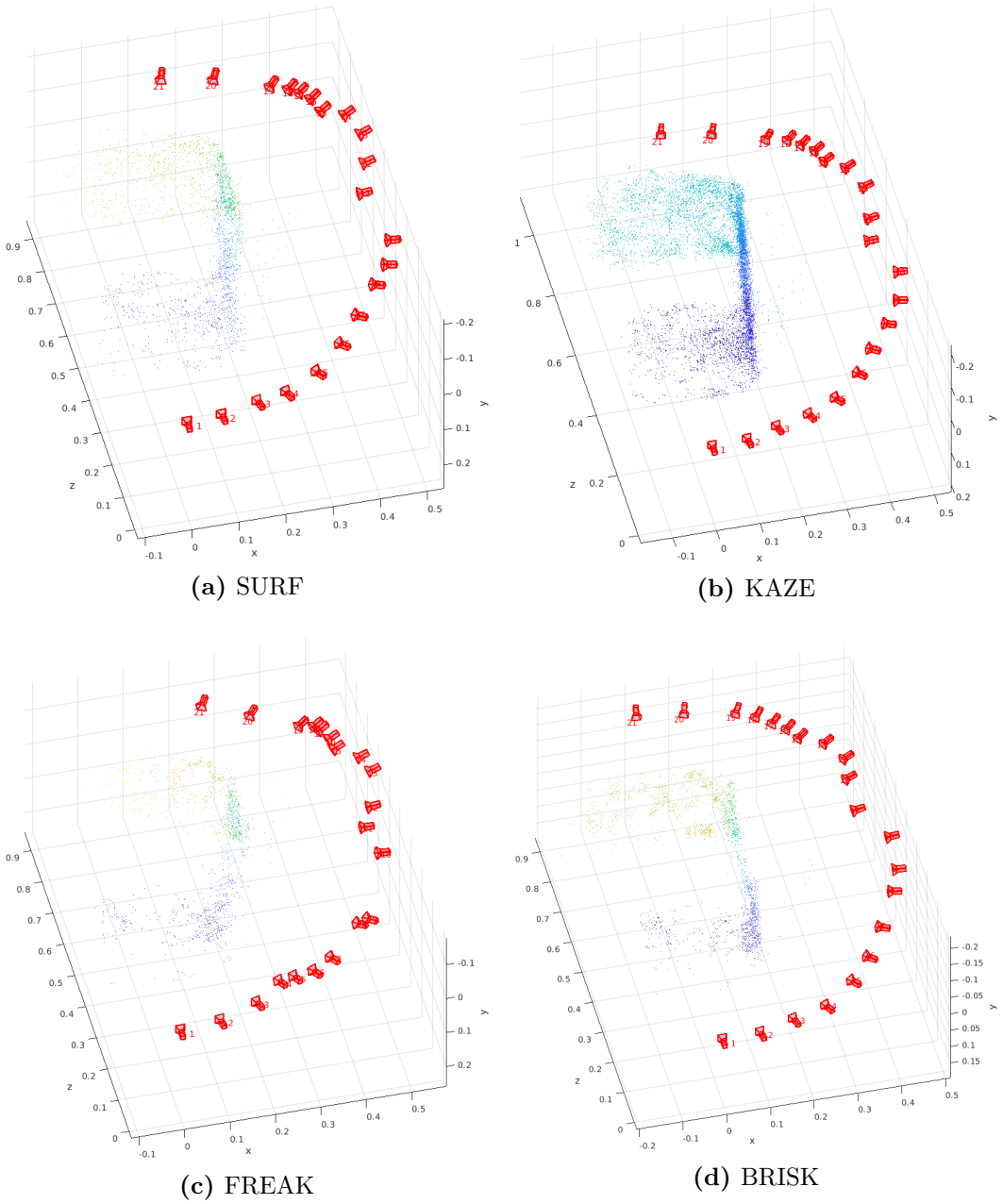
KAZE was able to process all data sets, matched significantly more features than the other algorithms, and the landmarks generated are well aligned. Although SURF finds significantly fewer matches than KAZE it was also successful on all data sets, with similar alignment to KAZE. The binary algorithms were less robust, with both failing on data sets 3 and 4. FREAK did not perform well, with significant misalignment on both data sets 1 and 2. BRISK did perform well on data set 1 and acceptably on data set 2, and



(a) Data set 1 **(b)** Data set 2

(c) Data set 3 **(d)** Data set 4

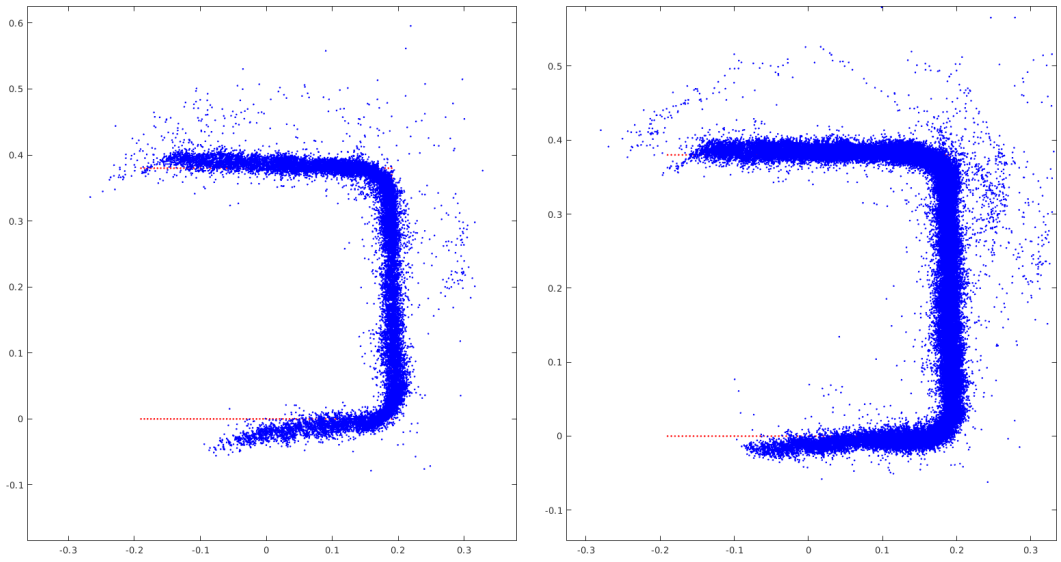
Figure 3.6: Number matches between sequential image pairs for each data set using SURF, KAZE, FREAK, and BRISK



(a) SURF **(b)** KAZE

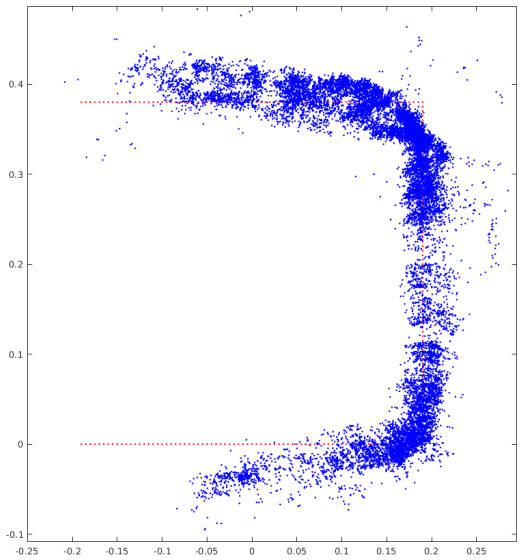
(c) FREAK **(d)** BRISK

Figure 3.7: Reduced camera viewpoints and landmarks from data set 1

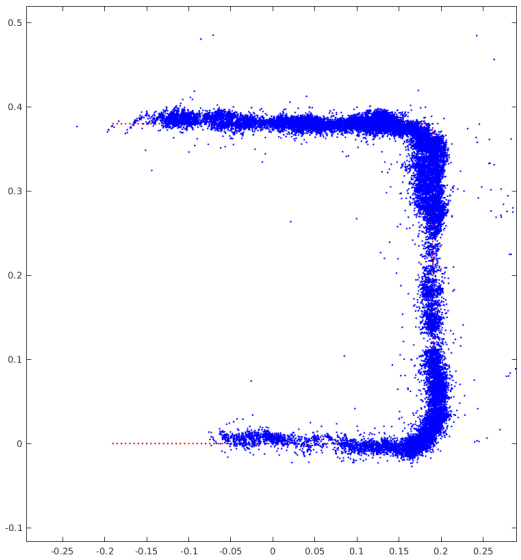


(a) SURF

(b) KAZE

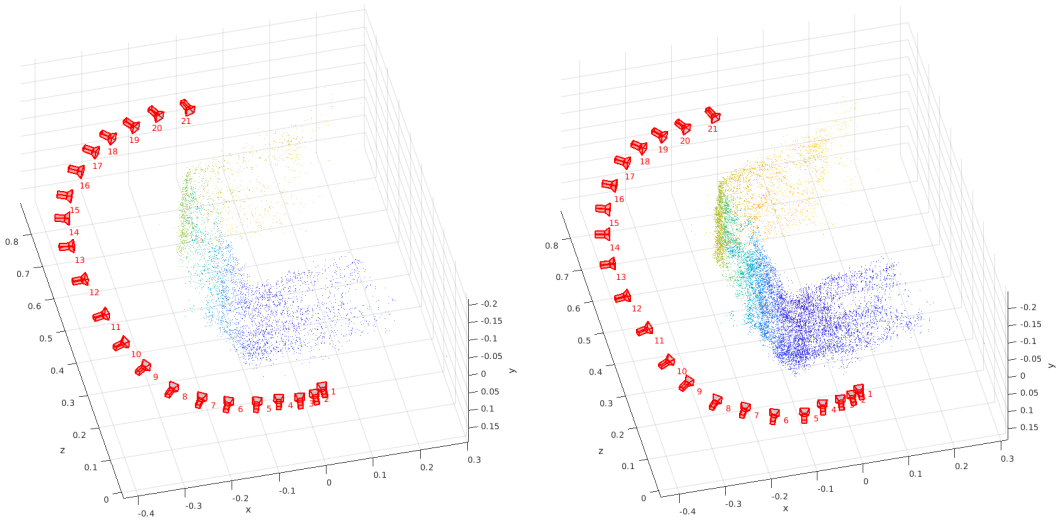


(c) FREAK



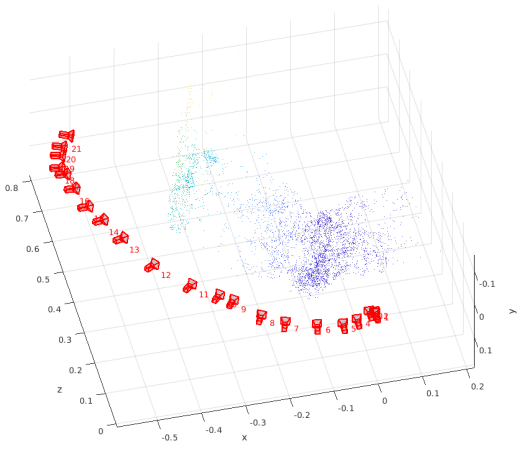
(d) BRISK

Figure 3.8: All landmarks (shown in blue) generated from data set 1, viewed top down after aligning with a model (shown in red)

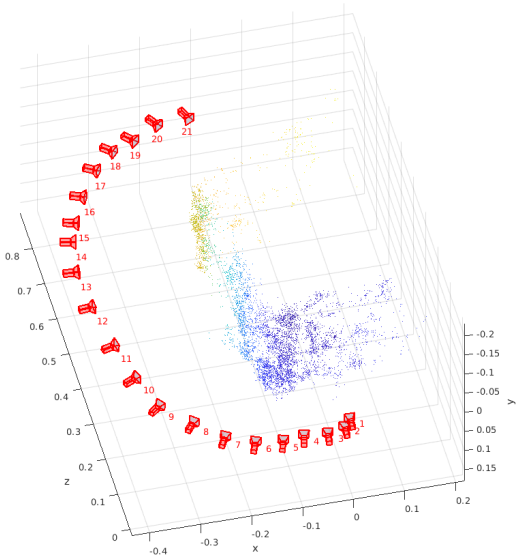


(a) SURF

(b) KAZE

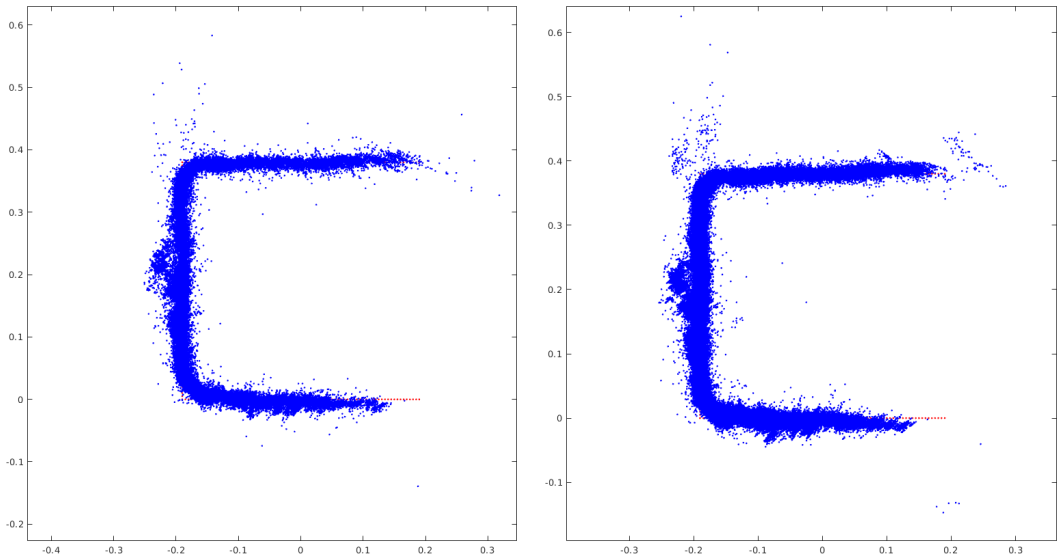


(c) FREAK



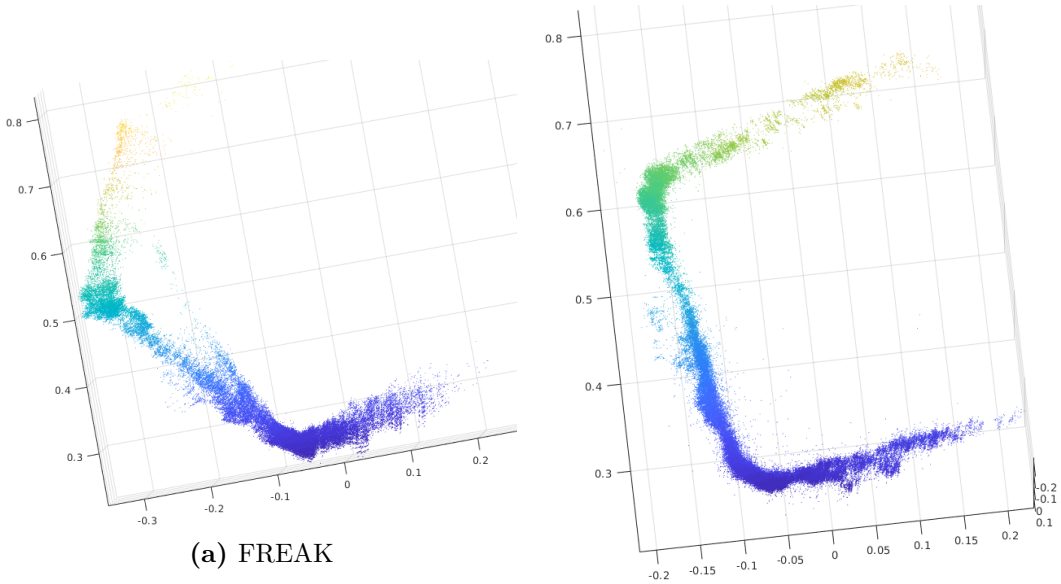
(d) BRISK

Figure 3.9: Reduced camera viewpoints and landmarks from data set 2



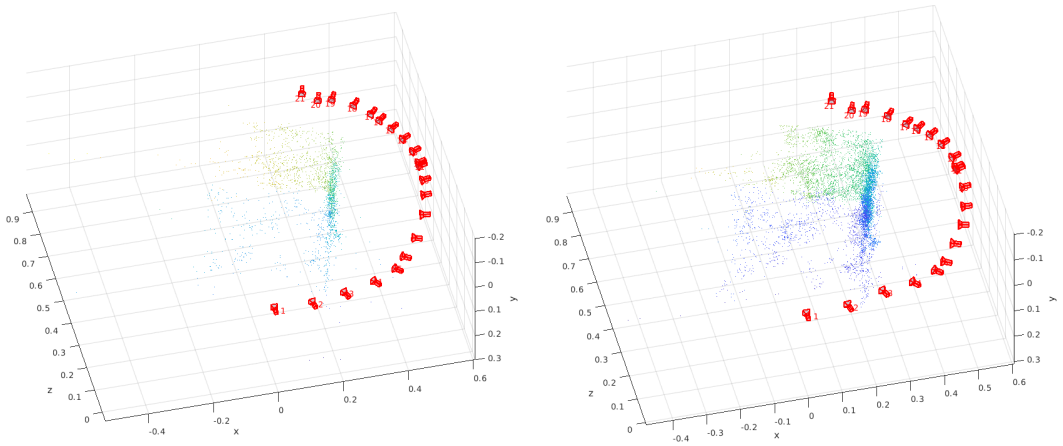
(a) SURF **(b)** KAZE

Figure 3.10: All landmarks (shown in blue) generated from data set 2 (SURF and KAZE only), viewed top down after aligning with a model (shown in red)



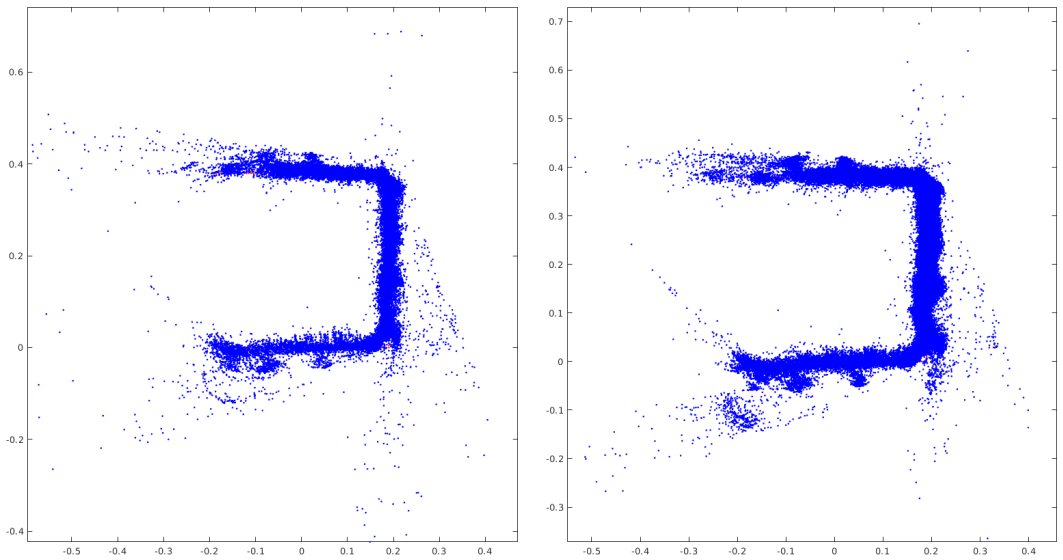
(a) FREAK **(b)** BRISK

Figure 3.11: All landmarks generated from data set 2 (FREAK and BRISK only)



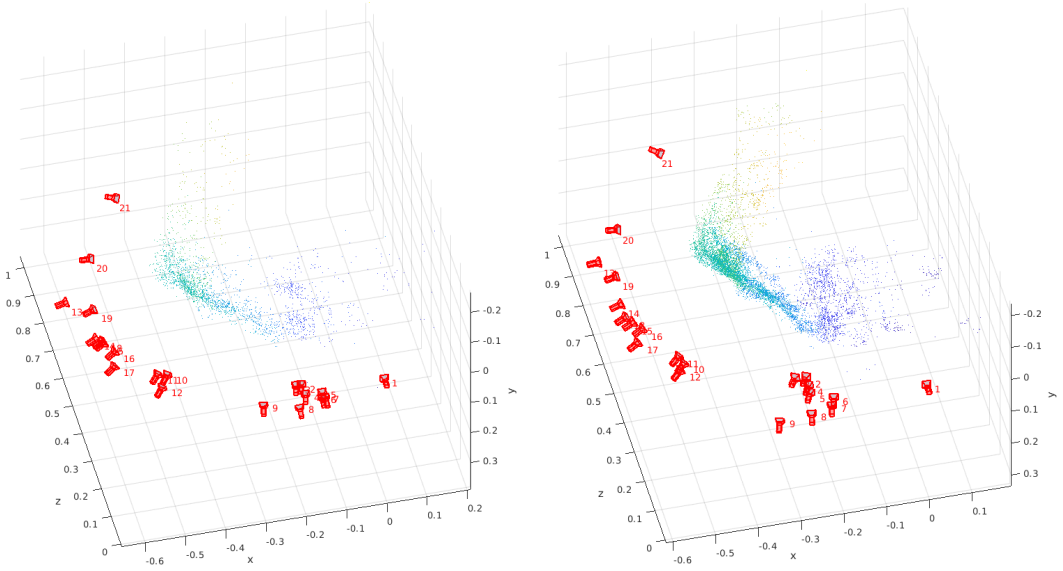
(a) SURF **(b)** KAZE

Figure 3.12: Reduced camera viewpoints and landmarks from data set 3



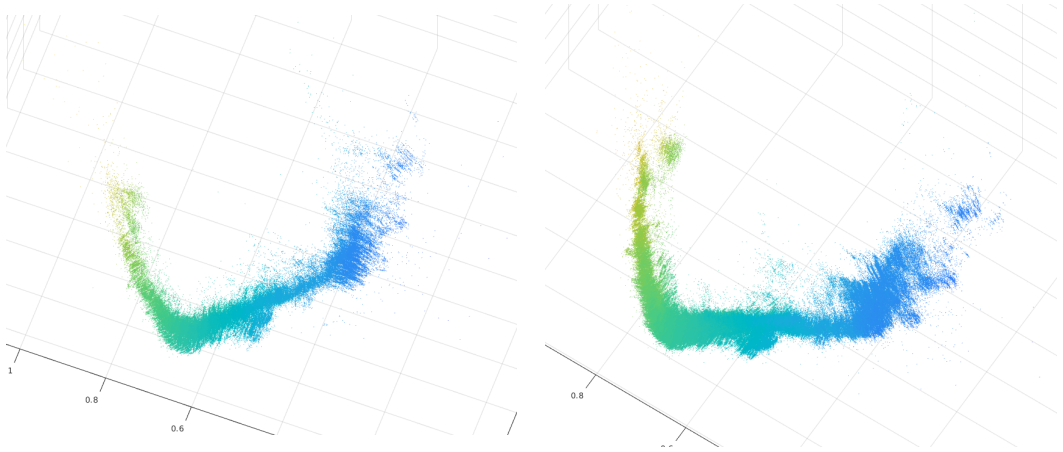
(a) SURF **(b)** KAZE

Figure 3.13: All landmarks (shown in blue) generated from data set 3 (SURF and KAZE only), viewed top down after aligning with a model (shown in red)



(a) SURF **(b)** KAZE

Figure 3.14: Reduced camera viewpoints and landmarks from data set 4



(a) SURF **(b)** KAZE

Figure 3.15: All landmarks generated from data set 4 (SURF and KAZE only)

may be an acceptable option if visibility is good and processing time is critical.

A major difference between FREAK and the other algorithms evaluated is that FAST feature detection operates at a single scale only. The other feature detection algorithms operate at multiple scales, essentially downsampling the image to find larger features. This may mean that FREAK is more sensitive to floating material, and therefore performs poorly compared to other algorithms.

There are numerous ways this process of visual odometry can be improved. Key frame selection and windowed bundle adjustment are commonly used, as these significantly reduce drift. These were excluded for the purpose of evaluating feature and description algorithms, however in a practical visual odometry system including them would be preferable. Visual odometry may also benefit from photometric camera calibration to mitigate the effect of vignetting and the light distribution pattern of the artificial lights. This may help to improve matching.

Many visual odometry systems use a motion model for feature matching. Known landmarks are projected into the current image by assuming constant motion of the camera. The landmarks are then compared to the features detected in regions around the predicted location. The motion model is used to improve the speed of matching however, given the highly repetitive texture of the pile, it is possible that it may also improve matching accuracy.

Stereo motion estimation was implemented using the landmarks (stereo matches) from the previous frame and matching image features from the current left image. This limits sequential matches to only those features which have a stereo match in the previous frame. Significantly more matches are found in only the left or right sequential image pairs. Robustness and accuracy could be improved by implementing motion estimation and outlier rejection that incorporates monocular features in addition to the stereo landmarks.

3.6 Summary

Stereo visual odometry has been successfully performed in the environment in which the ROV operates. A fixed exposure time was found to be acceptable, and image enhancement for feature detection was found to improve the number of matches between sequential frames. KAZE was found to be the best performing algorithm for feature detection and description.

The accuracy of the camera trajectory and landmarks generated by VO can be improved with windowed bundle adjustment and key frame selection. The next chapter describes how to perform parallax parameterised bundle adjustment on the data generated by stereo visual odometry. Parallax parameterisation has been shown to perform better than other parameterisations under some conditions.

With the addition of loop closure and global bundle adjustment, VO can be extended into a visual SLAM system, which maintains a globally consistent map and camera trajectory. This is evaluated in Chapter 5.

4

Extending Parallax Parameterised Bundle Adjustment to Stereo

4.1 Introduction

Bundle adjustment is an important step in VO, SfM, and visual SLAM. It reduces drift in the camera trajectory and corrects the positions of landmarks by solving an optimisation problem: find the camera poses and landmark positions that minimises the error between the landmarks' predicted positions in the images, and the observed positions in the images.

Bundle adjustment in SfM and visual SLAM systems commonly use Cartesian coordinates to define landmarks. Some use inverse depth parameterisation [80], where the landmark is defined by bearing angles (azimuth and elevation) and an inverse depth parameter, in the frame of the first observation. Inverse depth parameterisation improves the performance of bundle adjustment with distant landmarks. Distant landmarks have a low parallax angle and do not provide much information on the translation of the camera.

There is another type of low parallax feature, features that are coaxial with the movement of the camera. Zhao *et al.* [81] introduced parallax angle parameterisation for monocular bundle adjustment, which achieves better and faster convergence than Cartesian or inverse depth parameterisation, in the presence of low parallax features. Parallax parameterisation is similar to inverse depth parameterisation, sharing the azimuth and elevation parameters. Depth however is parameterised as a parallax angle. This chapter describes how to extend parallax parameterised bundle adjustment to stereo, so that the resulting camera poses and landmark locations will be correctly scaled.

4.2 Algorithm

Problem Formulation

Bundle adjustment is a non-linear optimisation problem. Viewpoint poses and landmark coordinates are produced by VO, which bundle adjustment improves by minimising reprojection error: the difference between the actual observations and the predicted observations

determined from optimised poses and landmark coordinates.

Bundle adjustment can be described by

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| o_j^i - \tilde{o}_j^i \right\|^2 \tag{4.1}$$

where o_j^i is the observation of the jth landmark observed from the ith viewpoint

$$o_j^i = [u_1 \ v_1 \ u_2 \ v_2]^\intercal \tag{4.2}$$

and $\tilde{o}_j^i$ is the predicted observation of a landmark j from pose i

$$\tilde{o}_j^i = [\tilde{u}_1 \ \tilde{v}_1 \ \tilde{u}_2 \ \tilde{v}_2]^\intercal \tag{4.3}$$

Predicted observations are determined by:

$$\begin{bmatrix} \tilde{u} \\ \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} \hat{u}/\lambda \\ \\ \hat{v}/\lambda \end{bmatrix} \tag{4.4}$$

where

$$\begin{bmatrix} \hat{u} \\ \\ \hat{v} \\ \\ \lambda \end{bmatrix} = K \ \mathbf{x}_j^i \tag{4.5}$$

for the first camera, and

$$\begin{bmatrix} \hat{u} \\ \\ \hat{v} \\ \\ \lambda \end{bmatrix} = K \ (R \ \mathbf{x}_j^i + \mathbf{t}) \tag{4.6}$$

for the second camera, where K is the intrinsic matrix of the camera, R and $\mathbf{t}$ are the rotation and translation of the first camera in relation to the second, and $\mathbf{x}_j^i$ is the predicted Cartesian coordinate of the jth landmark in the ith viewpoint, which is determined by (4.9) for Cartesian parameterisation, and (4.19) or (4.20) for parallax parameterisation.

When images are rectified the rotation between two cameras R is the identity matrix (i.e. no rotation) and the translation between two cameras $\mathbf{t}$ is only the translation in the x axis.

Cartesian Parameterisation

Initialisation

With Cartesian parameterisation, bundle adjustment uses the Cartesian coordinates of the landmark in the world reference frame as optimisation parameters. Landmarks in the world reference frame $\mathbf{x}_j$ are determined from the landmark in the coordinate frame of the first observation $\mathbf{x}_j^i$ with

$$\mathbf{x}_j = R_i \, \mathbf{x}_j^i + \mathbf{t}_i \tag{4.7}$$

where R_i and $\mathbf{t}_i$ are the absolute rotation and translation of the ith viewpoint respectively, and $\mathbf{x}_j^i$ is the Cartesian coordinates of the jth landmark in the coordinate frame of the ith viewpoint, which was determined by triangulation with

$$\mathbf{x}_j^i = K^{-1} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \left(\frac{f \; b}{u_1 - u_2} \right) \tag{4.8}$$

where f is the focal length of the cameras and b is the baseline. After the images are rectified the vertical image coordinates of the features will be equal (i.e. $v_1 = v_2$).

Reprojection

To find reprojection error the Cartesian coordinates of a landmark in a viewpoint $\mathbf{x}_j^i$ is determined from the Cartesian coordinates of the landmark in the world reference frame $\mathbf{x}_j$ with

$$\mathbf{x}_j^i = R_i^\intercal (\mathbf{x}_j - \mathbf{t}_i) \tag{4.9}$$

which is the inverse of (4.7) used for initialisation.

The Cartesian coordinates of the landmark in the viewpoint are reprojected into the first camera with (4.4) and (4.5), and into the second camera with (4.4) and (4.6).

Parallax Parameterisation

Initialisation

Parallax parameterisation differs from Cartesian parameterisation in the way landmarks are defined. In parallax parameterisation the jth landmark is defined by the parameters ψ_j, θ_j, and ω_j. These parameters are determined in relation to a main and associate anchor. The main anchor is the pose of the first observation and the associate anchor is the pose of another observation. The associate anchor is chosen based on the resulting parallax: either the pose of the earliest observation that results in a parallax greater than

a certain threshold or, if no pose results in a parallax greater than the threshold, the pose that results in the largest parallax. For the results in this paper the threshold used was 0.5 radians. [81] found that the threshold value was not critical to the performance of ParallaxBA.

For simplicity, only landmarks that were observed in both stereo images and from multiple viewpoints were used.

ψ_j and θ_j are determined with the equations

$$\psi_j = \text{arctan2}\left(\vec{x}_j^m,\ \vec{z}_j^m\right) \tag{4.10}$$

$$\theta_j = \text{arctan2}\left(\vec{y}_j^m,\ \sqrt{(\vec{x}_j^m)^2 + (\vec{z}_j^m)^2}\right) \tag{4.11}$$

and ω_j is determined with the equation

$$\omega_j = \arccos\left(\frac{\vec{\mathbf{x}}_j^m \cdot \vec{\mathbf{x}}_j^a}{\left\|\vec{\mathbf{x}}_j^m\right\| \left\|\vec{\mathbf{x}}_j^a\right\|}\right) \tag{4.12}$$

where $\vec{\mathbf{x}}_j^i$ is a vector from the absolute pose translation $\mathbf{t}_i$ to the jth landmark (where $i = m$ for the main anchor and $i = a$ for the associate anchor), and

$$\vec{\mathbf{x}}_j^i = \begin{bmatrix} \vec{x}_j^i \\ \vec{y}_j^i \\ \vec{z}_j^i \end{bmatrix} \tag{4.13}$$

To initialise bundle adjustment, the vector from the main anchor to landmark is determined by rotating the Cartesian coordinates of the landmark with

$$\vec{\mathbf{x}}_j^m = R_m\ \mathbf{x}_j^m \tag{4.14}$$

where $\mathbf{x}_j^m$ is triangulated from observations using (4.8).

The vector from the associate anchor is determined from the landmark in the main anchor with

$$\vec{\mathbf{x}}_j^a = R_m\ \mathbf{x}_j^m + \mathbf{t}_m - \mathbf{t}_a \tag{4.15}$$

Reprojection

To find reprojection error the predicted parallax parameters are converted back into Cartesian coordinates. From the parallax parameters ψ_j, θ_j, and ω_j, a unit vector from the main anchor $\mathbf{t}_m$ to the jth landmark is determined with

$$\vec{\mathbf{x}}_j^m = \begin{bmatrix} \sin\,\psi_j\,\cos\,\theta_j \\ \sin\,\theta_j \\ \cos\,\psi_j\,\cos\,\theta_j \end{bmatrix} \tag{4.16}$$

Then the angle φ_j between the vector $\vec{\mathbf{x}}_j^m$ and vector $\mathbf{t}_a - \mathbf{t}_m$ is determined with

$$\varphi_j = \arccos\left(\vec{\mathbf{x}}_j^m \cdot \frac{\mathbf{t}_a - \mathbf{t}_m}{\|\mathbf{t}_a - \mathbf{t}_m\|}\right) \tag{4.17}$$

To scale the unit vector correctly the Euclidean distance d_j^m between the main anchor and jth landmark is determined with

$$d_j^m = \frac{\sin\,(\omega_j + \varphi_j)}{\sin\,\omega_j}\,\|\mathbf{t}_a - \mathbf{t}_m\| \tag{4.18}$$

If the pose of the observation is the main anchor, the unit vector is scaled and rotated to find the predicted coordinates of the landmark in the main anchor with

$$\mathbf{x}_j^m = {R_m}^\intercal\left(\vec{\mathbf{x}}_j^m d_j^m\right) \tag{4.19}$$

If the observation pose is not the main anchor then a vector from the main anchor to observation pose $(\mathbf{t}_i - \mathbf{t}_m)$ is subtracted from the scaled vector before it is rotated with

$$\mathbf{x}_j^i = {R_i}^\intercal\left(\vec{\mathbf{x}}_j^m d_j^m - (\mathbf{t}_i - \mathbf{t}_m)\right) \tag{4.20}$$

The Cartesian coordinates of the landmark in the viewpoint are reprojected into the first camera with (4.4) and (4.5), and into the second camera with (4.4) and (4.6).

Implementation

The non-linear optimisation problem of bundle adjustment is solved using Ceres Solver [79]. The form of the bundle adjustment problem solved by Ceres is

$$\min\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|o_j^i - \tilde{o}_j^i\right\|^2 \tag{4.21}$$

Both Cartesian and parallax parameterisation are solved using a trust region method. For parallax parameterisation the dogleg [119] strategy with sparse normal Cholesky solver was used for all data sets. The dogleg strategy can converge significantly faster than the Levenberg-Marquardt (LM) strategy [120]. For Cartesian parameterisation the dogleg strategy with sparse normal Cholesky solver was used where possible, but if the optimisation did not converge within 300 iterations the LM strategy and sparse Schur solver was used instead.

The stopping criteria used were:

- Parameter tolerance: 1e-9
- Function tolerance: 1e-9
- Gradient tolerance: 1e-9

All tests were performed on a computer with:

- An Intel i5-6300U (a dual core CPU with simultaneous multithreading, and a clock speed of 2.4 GHz base, 3.00 GHz boost); and
- 8 GB of DDR4 RAM

4.3 Evaluation

Simulated data set

A simulated data set was used to evaluate the bundle adjustment algorithms. The simulation generates a number of viewpoints and a number of landmarks. The landmarks are projected into the viewpoints as observations, sensor noise is added to the observations, and local landmarks are triangulated from the observations. To initialise bundle adjustment motion estimation error is added to the poses.

First the field of view is determined to generate landmarks that will be within the simulated image of a viewpoint. From focal length f, image width w and image height h, the field of view can be determined with

$$x_{\text{fov}} = 2 \arctan(\frac{w}{2}/f) \tag{4.22}$$

$$y_{\text{fov}} = 2 \arctan(\frac{h}{2}/f) \tag{4.23}$$

For the first viewpoint, a number of landmarks are randomly generated within the field of view and within a specified distance range. From each landmark $\mathbf{x}_j^i$ the image coordinates for each camera k are determined with

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \hat{u}/\lambda \\ \hat{v}/\lambda \end{bmatrix} \tag{4.24}$$

where $\hat{u}$, $\hat{v}$ and λ are determined by (4.5) for the first camera and (4.6) for the second camera.

Subsequent viewpoints are created by generating a relative rotation R_{i-1}^i and translation $\mathbf{t}_{i-1}^i$ which are applied to the previous viewpoint's global rotation R_{i-1} and global translation $\mathbf{t}_{i-1}$, to get the new viewpoint's global rotation R_i and translation $\mathbf{t}_i$ with the equations

$$R_i = R_{i-1}\ R_{i-1}^i \tag{4.25}$$

$$\mathbf{t}_i = \mathbf{t}_{i-1} + \left(R_{i-1}\ \mathbf{t}_{i-1}^{i}\right) \tag{4.26}$$

Using the relative rotation and translation, the Cartesian coordinates of each landmark in the previous viewpoint $\mathbf{x}_j^{i-1}$ are transformed into the new viewpoint with

$$\mathbf{x}_j^i = {R_{i-1}^i}^\intercal \left(\mathbf{x}_j^{i-1} - \mathbf{t}_{i-1}^i\right) \tag{4.27}$$

The image coordinates of the landmarks in the new viewpoint are determined with (4.24). Any landmarks that are outside of the field of view are removed from the viewpoint. Additionally a third of the landmarks are randomly removed from the viewpoint. New landmarks for the viewpoint are generated to replace the removed landmarks in the same way as for the first viewpoint. A random amount of error is added to the image coordinates of all landmarks to simulate sensor noise, and the Cartesian coordinates of the landmarks are triangulated from the observations with (4.8).

For the initialisation of bundle adjustment, a random amount of error is added to the rotation and translation of each viewpoint to simulate motion estimation error.

The camera parameters used to generate the simulation data were:

- Camera focal length: $f = 300$ px
- Stereo camera baseline: $b = 30$ mm
- Relative rotation between the first and second camera: $R = I$
- Relative translation between the first and second camera: $\mathbf{t} = [b\ 0\ 0]^\intercal$
- Image width: $w = 800$ px
- Image height: $h = 600$ px

All simulated data sets have 100 viewpoints and 100 landmarks visible. There is no loop closure in the simulated data sets.

The rotation of each viewpoint relative to the previous viewpoint was $+\pi/64$ radians about the Y axis, with an additional uniformly distributed random value from $-\pi/32$ to $+\pi/32$ radians about each axis. The translation of each viewpoint relative to the previous viewpoint was +60 mm in the X axis, +2 mm in the Y axis, and +2 mm in the Z axis, with an additional uniformly distributed random value from −30 to +30 mm on each axis.

Uniformly distributed random values between −1 and +1 pixels is added to the image coordinates of every observation to simulate sensor noise.

To simulate motion estimation error the rotation of each viewpoint has uniformly distributed random values from $-0.3\pi/32$ to $+0.3\pi/32$ radians added to each axis, and the translation of each viewpoint has uniformly distributed random values from −18 to +18 mm added to each axis.

The distance range of generated landmarks was varied between data sets. For each range four data sets were generated and used to evaluate the bundle adjustment algorithms. The distance ranges used are:

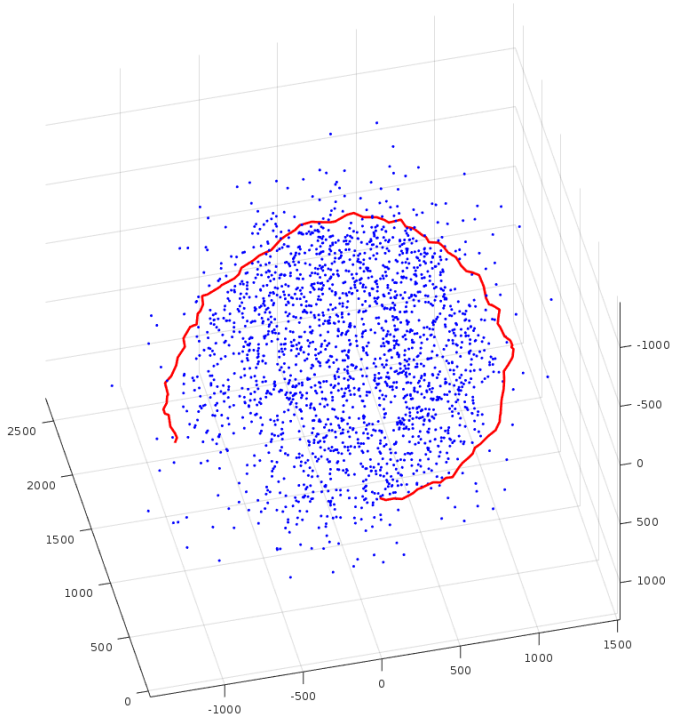


Figure 4.1: Estimated landmarks (blue points) and viewpoints (red line) of a simulated data set with landmarks between 0.1 m and 2 m.

- 0.1 to 2 m
- 1 to 3 m
- 2 to 5 m
- 3 to 10 m

Results

Table 4.1 shows the results of four data sets with landmark distance range of 0.1 to 2 m and Table 4.2 shows the results of four data sets with a landmark distance range of 1 to 3 m. The dogleg strategy was used for all data sets. In these data sets Cartesian and parallax parameterisation converged to the same final cost, in a similar number of iterations. Because of the complexity of parallax parameterisation it generally took a slightly longer time to run.

Table 4.3 shows the results of four data sets with a landmark distance range of 2 to 5 m. For these data sets Cartesian parameterisation with the dogleg strategy was unable to converge within 300 iterations. In two of the four data sets Cartesian and parallax converged to the same final cost, with parallax needing fewer iterations and slightly less time. In the other two data sets parallax converged to a lower final cost with fewer iterations (around 2 to 10 times fewer) and in less time.

Table 4.4 shows the results of four data sets with a landmark distance range of 3 to 10 m. For these data sets Cartesian parameterisation with the dogleg strategy was unable to

Figure 4.2: An example stereo image pair from the New College data set.

converge within 300 iterations. In these data sets parallax converges to a final cost lower than Cartesian with significantly fewer iterations (around 10 to 20 times fewer) and in significantly less time (around 10 times less).

The simulation data sets show that, as the parallax angle of landmarks becomes low, Cartesian parameterisation has difficulty converging. In these cases parallax parameterisation converges to a lower final cost, in fewer iterations and less time.

New College data set

A section of the New College data set [121] that has been processed for use with g2o [78] was used to evaluate the bundle adjustment algorithms. The data set contains 3,500 viewpoints, 491,640 landmarks, and 2,124,449 observations. An example of a stereo image pair used to generate the data for bundle adjustment is shown in Figure 4.2.

Results

The results of Cartesian and parallax bundle adjustment on the New College data set are presented in Table 4.5. Parallax parameterisation converged to a lower final cost, in fewer iterations and less time.

Cardboard box data set

A small data set was captured using a stereo camera while walking around a cardboard box. Salient features were extracted from the images using the SIFT algorithm [50]. The features are then rectified using the camera calibration parameters. Features are matched in the stereo image pairs and any epipolar outliers (where the vertical image coordinates of the matching features differ by more than two pixels) are removed, and the Cartesian coordinates of the landmarks are triangulated from the observations with (4.8).

Features from the temporal image pairs of the first camera are matched and, using the previous landmarks (Cartesian coordinates) and the current features (image coordinates), the relative motion of the camera is estimated using the perspective-three-point algorithm

Simulation data set	1		2		3		4	
Parameterisation	Cartesian	Parallax	Cartesian	Parallax	Cartesian	Parallax	Cartesian	Parallax
Strategy	Dogleg	Dogleg	Dogleg	Dogleg	Dogleg	Dogleg	Dogleg	Dogleg
Initial cost	7.16e+6	7.16e+6	7.71e+6	7.71e+6	1.46e+8	1.46e+8	3.24e+6	3.24e+6
Final cost	4.16e+3	4.16e+3	4.18e+3	4.18e+3	4.11e+3	4.11e+3	4.15e+3	4.15e+3
Iterations	8	8	7	7	13	12	5	5
Time (seconds)	0.40	0.85	0.42	0.65	0.66	1.10	0.32	0.54

Table 4.1: Results of the simulated data sets with 0.1 to 2 m landmarks

Simulation data set	5		6		7		8	
Parameterisation	Cartesian	Parallax	Cartesian	Parallax	Cartesian	Parallax	Cartesian	Parallax
Strategy	Dogleg	Dogleg	Dogleg	Dogleg	Dogleg	Dogleg	Dogleg	Dogleg
Initial cost	1.83e+6	1.83e+6	7.92e+6	7.92e+6	2.43e+6	2.43e+6	1.87e+6	1.87e+6
Final cost	3.92e+3	3.92e+3	4.04e+3	4.04e+3	3.96e+3	3.96e+3	4.01e+3	4.01e+3
Iterations	7	6	17	21	7	7	8	6
Time (seconds)	0.40	0.62	0.67	2.01	0.40	0.72	0.42	0.62

Table 4.2: Results of the simulated data sets with 1 to 3 m landmarks

Simulation data set	9		10		11		12	
Parameterisation	Cartesian	Parallax	Cartesian	Parallax	Cartesian	Parallax	Cartesian	Parallax
Strategy	LM	Dogleg	LM	Dogleg	LM	Dogleg	LM	Dogleg
Initial cost	1.94e+6	1.94e+6	1.64e+6	1.64e+6	2.17e+6	2.17e+6	1.51e+6	1.51e+6
Final cost	3.89e+3	3.89e+3	3.89e+3	3.89e+3	4.10e+3	3.85e+3	4.49e+3	3.89e+3
Iterations	21	9	34	9	83	8	104	8
Time (seconds)	1.16	0.92	1.75	0.94	4.69	0.80	5.60	0.81

Table 4.3: Results of the simulated data sets with 2 to 5 m landmarks

Simulation data set	13		14		15		16	
Parameterisation	Cartesian	Parallax	Cartesian	Parallax	Cartesian	Parallax	Cartesian	Parallax
Strategy	LM	Dogleg	LM	Dogleg	LM	Dogleg	LM	Dogleg
Initial cost	2.27e+6	2.27e+6	1.57e+6	1.57e+6	1.58e+6	1.58e+6	2.26e+6	2.26e+6
Final cost	5.85e+3	3.90e+3	5.75e+3	3.94e+3	6.30e+3	4.03e+3	5.36e+3	3.90e+3
Iterations	220	12	199	10	217	10	199	11
Time (seconds)	12.03	1.22	10.94	1.05	12.26	1.03	11.05	1.10

Table 4.4: Results of the simulated data sets with 3 to 10 m landmarks

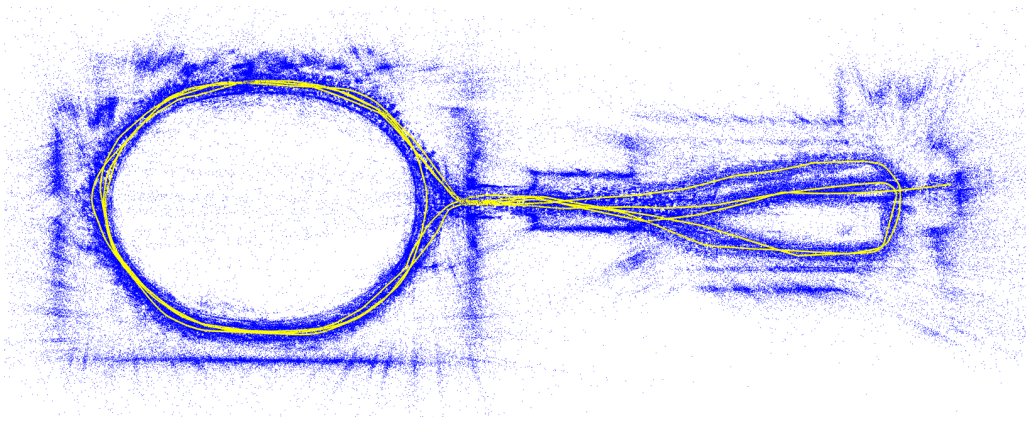


Figure 4.3: Estimated landmarks (blue points) and viewpoints (yellow line) of the New College data set. Distant landmarks are not shown.

	Cartesian	**Parallax**
Strategy	LM	Dogleg
Initial cost	5.120590e+7	5.120590e+7
Final cost	2.253884e+6	2.243549e+6
Iterations	74	10
Time (seconds)	1990.65	290.53

Table 4.5: Results of the New College data set

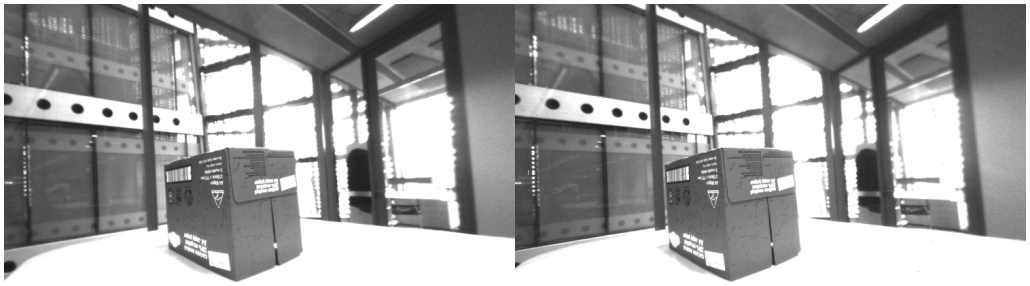


Figure 4.4: An example stereo image pair from the cardboard box data set.

	Cartesian	**Parallax**
Strategy	LM	Dogleg
Initial cost	9.763550e+4	9.763550e+4
Final cost	5.546365e+3	3.693787e+3
Iterations	99	6
Time (seconds)	6.62	0.78

Table 4.6: Results of the cardboard box data set

[122] while eliminating erroneous matches with the M-estimator sample consensus algorithm [68].

Key frame selection is used to reduce the amount of data for bundle adjustment. Frames are only added if the estimated motion is greater than 40 mm or 5° from the previous key frame. The data used for both parameterisations of bundle adjustment is identical.

Feature matching is also performed on the first and last viewpoints. Since the camera has travelled back to its starting position this adds a loop closure to the data.

The data for bundle adjustment contains 60 viewpoints, 2,237 landmarks, and 12,933 observations. The stereo camera has a baseline of 30 mm, with the camera approximately 300 to 500 mm away from the cardboard box. The background provided more distant landmarks. An example of a stereo image pair used to generate the data for bundle adjustment is shown in Figure 4.4.

Results

The results of Cartesian and parallax bundle adjustment on the cardboard box data set are presented in Table 4.6. Cartesian and Parallax parameterisation converged to the same final cost, however parallax converged in fewer iterations and less time.

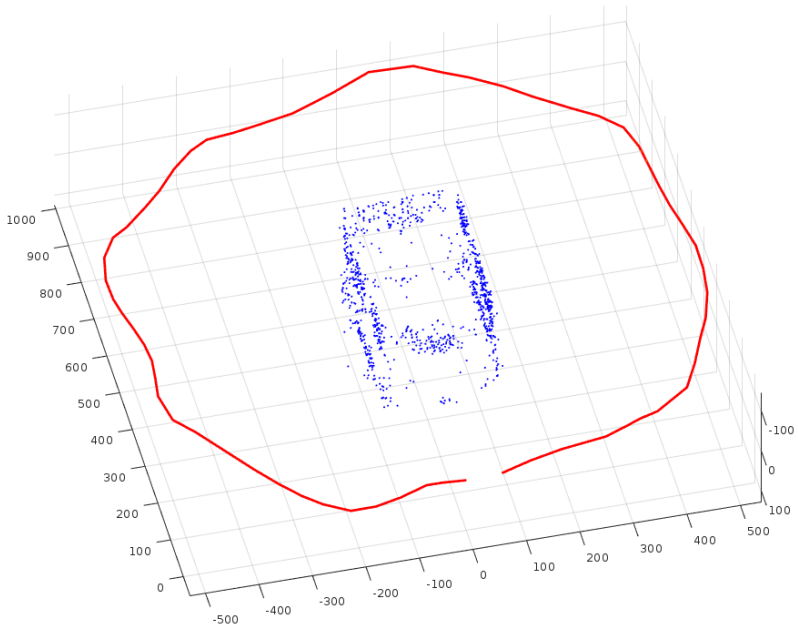


Figure 4.5: Estimated landmarks (blue points) and viewpoints (red line) of the cardboard box data set. Distant landmarks are not shown.

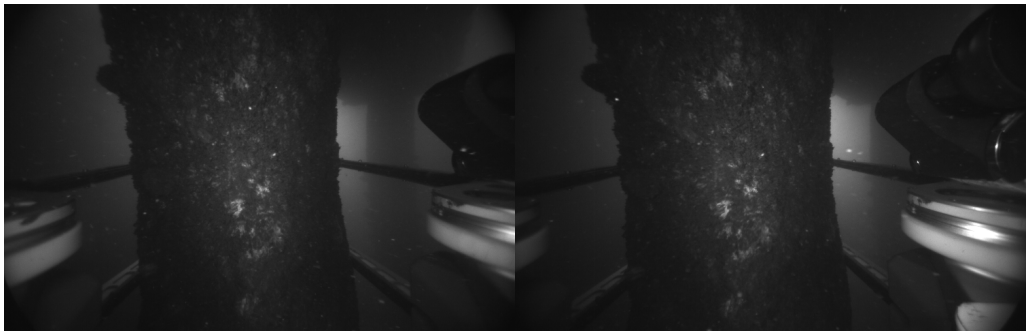


Figure 4.6: An example stereo image pair from the underwater data set.

Underwater data set

A data set was captured underwater in a similar way to the cardboard box data set, with the stereo camera in an underwater housing attached to an ROV, which rotates around a concrete bridge pile. The data set was processed almost identically to the cardboard box data set.

The data for bundle adjustment contains 61 viewpoints, 1,155 landmarks, and 3,053 observations. The stereo camera has a baseline of approximately 30 mm, with the camera approximately 0.3 to 1 m away from the bridge pile. In the underwater environment features not belonging to the pile tend to be unreliable, so any landmarks further than 1.2 m from the camera have been excluded. Landmarks closer than 0.2 m have also been excluded, as the design of the ROV should prevent the pile being closer than about 0.3 m from the camera. An example of a stereo image pair used to generate the data for bundle adjustment is shown in Figure 4.6.

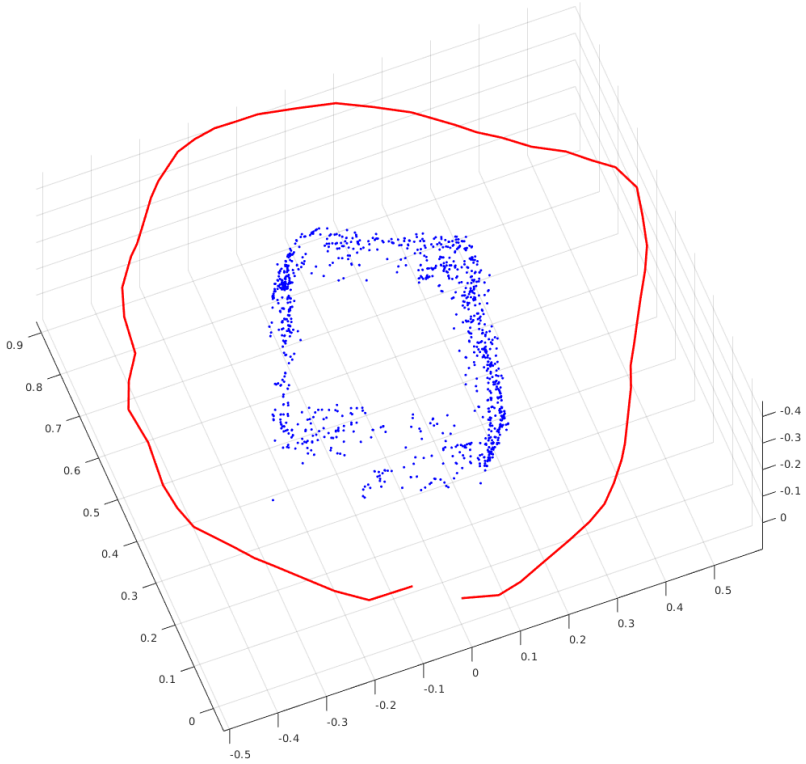


Figure 4.7: Estimated landmarks (blue points) and viewpoints (red line) of the underwater data set.

Results

The results of Cartesian and parallax bundle adjustment on the underwater data set are presented in Table 4.7. Cartesian and Parallax parameterisation converged to the same final cost, however parallax converged in fewer iterations and less time. The improvement in computation time is less dramatic than with the cardboard box data set because distant features have been removed, resulting in fewer low parallax features for bundle adjustment.

	Cartesian	**Parallax**
Strategy	LM	Dogleg
Initial cost	4.176352e+06	4.176352e+06
Final cost	2.086290e+06	2.086290e+06
Iterations	33	7
Time (seconds)	0.4014	0.1057

Table 4.7: Results of the underwater data set

4.4 Summary

Zhao *et al.* [81] proposed ParallaxBA: a monocular bundle adjustment algorithm using parallax angles for landmark parameterisation. In this chapter it has been demonstrated that parallax parameterisation in stereo offers the same advantages as monocular while providing correct scale.

In all data sets evaluated parallax parameterisation converged to a cost equal to or less than Cartesian parameterisation. Generally parallax parameterisation can converge in fewer iterations however, because of the complexity of parallax parameterisation, each iteration takes longer to compute. For the data sets with landmarks near to the viewpoint (and thus well defined), parallax parameterisation converges to the same cost as Cartesian parameterisation, in slightly longer time. For the data sets with landmarks further from the viewpoint (and thus poorly defined), parallax parameterisation can converge to a lower cost than Cartesian parameterisation, in significantly shorter time.

Stereo parallax parameterisation is good choice for bundle adjustment in VO and visual SLAM, particularly when dealing with distant landmarks. The next chapter evaluates underwater visual SLAM, for the purpose of creating a multiple viewpoint 3D reconstruction.

5

Underwater 3D Reconstruction

5.1 Introduction

This chapter looks at how to create a 3D reconstruction of a submerged bridge pile with stereo images captured from a single viewpoint, and how to determine the camera poses of images so that single viewpoint reconstructions can be merged into a multi-view reconstruction.

The single viewpoint 3D reconstruction may be used to aid with cleaning the pile. The pile is cleaned using a high pressure water jet from a nozzle attached to a pan-and-tilt unit, with an additional linear actuator. To clean effectively, the nozzle must be within a specific distance range from the pile. Too far and the water jet will not remove biofouling, too close and the water jet could damage the concrete. The stereo camera can be used to perceive the pile to generate a path for the blasting nozzle that maintains the correct distance to the pile while avoiding any obstacles.

The multiple viewpoint reconstruction of the pile may be used for inspecting the pile (checking for and measuring damage to the concrete) or for keeping a record of the condition of the concrete or amount of biofouling.

SfM and visual SLAM are methods of determining camera poses (and scene structure) from images, and build upon VO and bundle adjustment covered in the previous chapters.

5.2 Single Viewpoint Perception

For the SPIR project, a 3D reconstruction of the pile from a single viewpoint is desired to assist with cleaning the pile, and for combining together with reconstructions from other viewpoints to create a multi-view reconstruction of the entire pile. For cleaning, the depth of all pixels in the image is not required, as long as any obstacles are adequately identified so that they can be avoided. Sparse (feature based) methods or dense methods may be used. For inspection, the depth of all or the majority of pixels is desired so that, when multiple reconstructions are combined, the result will be a complete dense reconstruction of the pile. Therefore a dense method is required. Feature based methods will tend to produce less noise than dense methods, since only salient points will be matched.

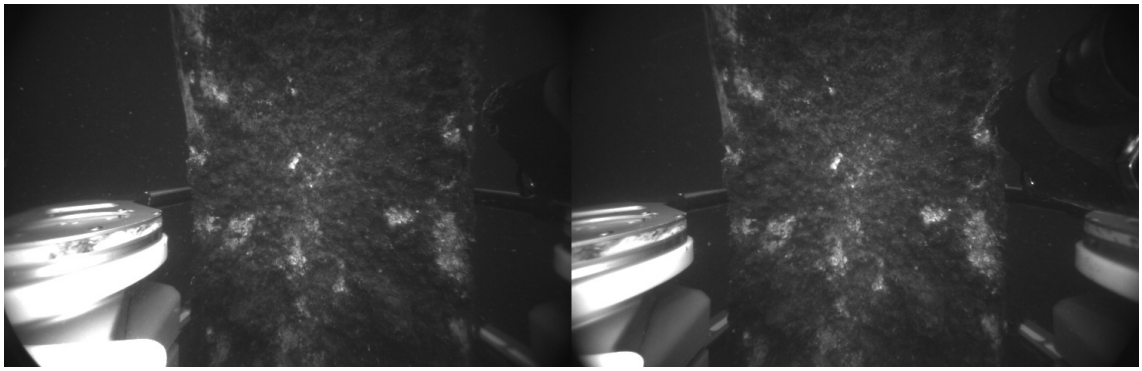


Figure 5.1: Image pair 1 before rectification

The main challenge for 3D reconstruction of a pile in the underwater environment is ignoring or removing anything in the scene that is not the pile. For example, there can be floating material in the water, especially after cleaning the pile or when the ROV is moving close to the bed of the body of water. Ambient light can illuminate the background depending on the position of the sun and, if the grasping arms are used to hold the ROV in position, the grasping arms will be visible to the sides of the pile. The aim of this section is to determine which methods of stereo perception produce the best model of the pile in the typical water and lighting conditions encountered by the ROV.

Method

Six stereo image pairs were selected from the data set for evaluation. They cover a range conditions, with differing:

- position with respect to the pile;
- ambient lighting;
- amounts of biofouling; and
- amounts of floating material

Unfortunately the left and right sides of the original images are difficult to work with and will be removed by rectification and cropping. This is because the camera lenses have a wide field of view, and the positioning of the camera results in parts of the ROV being visible in the sides of the images. Figure 5.1 shows the original images from the camera for image pair 1. During underwater camera calibration the calibration board is not visible at the sides of the image, as it is blocked by the ROV itself. The resulting undistortion and rectification will crop the parts of the image which were not covered by the calibration board. Even after undistortion and rectification some parts of the ROV were still visible, and for simplicity the images have been cropped further to remove them.

Some parts of the ROV cannot simply be cropped out. The images were captured before and after cleaning the pile, while the ROV was grasping onto the pile. The grasping arms have varying positions and wrap around the pile, which makes removing them more

complicated. Removing the grasping arms from the image or point cloud has not been attempted at this stage. The image pairs are shown (after rectification) in subsection 5.2.

For each image pair point clouds were generated with sparse feature matching and dense stereo correspondence algorithms.

Sparse feature matching

Four algorithms for feature detection and description were compared:

- SURF
- KAZE
- FREAK (with FAST feature detection)
- BRISK

The process to generate a point cloud from sparse feature was:

1. Apply CLAHE to images using MATLAB's *adapthisteq* function.
2. Detect features in the enhanced images using MATLAB's *detectSURFFeatures*, *detectKAZEFeatures*, *detectFASTFeatures*, or *detectBRISKFeatures* functions.
3. Extract feature descriptors from the original images using MATLAB's *extractFeatures* function (with default parameters).
4. Rectify the feature positions using rectification matrices from camera calibration.
5. Crop 128 pixels from the left of the images and 20 pixels from the right of the images, to remove parts of the ROV from the images.
6. Match features using MATLAB's *matchFeatures* function.
7. Remove epipolar outliers: matching features with more than 2 pixels difference in vertical position.
8. Triangulate world locations from image locations using OpenCV's *triangulatePoints* function.
9. Remove points closer than 0.2 m and further than 1.2 m from the camera.

For the *detectKAZEFeatures* function, MATLAB's default parameters were used. For the *detectSURFFeatures* function, the "MetricThreshold" parameter was reduced from 1000 to 100, to increase the number of features detected. For the *detectFASTFeatures* and *detectBRISKFeatures* functions, the "MinContrast" parameter was reduced from 0.2 to 0.1, also to increase the number of features detected.

The parameters used for the *detectSURFFeatures* function were:

- MetricThreshold: 100
- NumOctaves: 3 (default)
- NumScaleLevels: 4 (default)

The parameters used for the *detectKAZEFeatures* function were:

- Diffusion: Region (default)

- Threshold: 0.0001 (default)
- NumOctaves: 3 (default)
- NumScaleLevels: 4 (default)

The parameters used for the *detectFASTFeatures* function were:

- MinQuality: 0.1 (default)
- MinContrast: 0.1

The parameters used for the *detectBRISKFeatures* function were:

- MinQuality: 0.1 (default)
- MinContrast: 0.1
- NumOctaves: 4 (default)

Points closer than 0.2 m from the camera were removed because the ROV has a front bumper that prevents the camera being closer than approximately 0.3 m from the pile. Points further than 1.2 m were removed because the stereo camera has a baseline of around 30 mm, and generally the disparity of points further than 40 times the baseline is too small to give a reliable distance measurement.

The point clouds generated from sparse features were evaluated by:

- number of points;
- area of the image that the points cover; and
- plotting the point clouds to visualise noise and outliers.

The area of the image that the point clouds cover is the area of a convex hull of the points. The units of this area are pixels, which is listed in the results as megapixels.

Dense stereo correspondence

Three dense stereo correspondence algorithms were compared:

- block matching;
- semi-global matching; and
- ELAS.

For block matching and semi-global matching the OpenCV library was used. The process to generate a point cloud was:

1. Rectify the images.
2. Compute disparity map using the relevant OpenCV method.
3. Reproject the disparity map to a point cloud using the reprojection matrix generated by camera calibration and OpenCV's *reprojectImageTo3D* function.
4. Remove points closer than 0.2 m and further than 1.2 m from the camera.
5. Apply the colour of the corresponding pixel from the left image to the points.

The quality of the point cloud generated is dependent on the parameters used. Parameters for block matching and semi-global matching were found through experimentation. The best parameters found for block matching were:

- Minimum disparity: 0
- Number of disparites: 128
- Block size: 19
- Uniqueness ratio: 15
- Pre-filter type: Sobel
- Pre-filter size: 5
- Pre-filter cap: 15
- Speckle window size: 10
- Speckle range: 2

The best parameters found for semi-global matching were:

- Minimum disparity: 0
- Number of disparites: 128
- Block size: 15
- Uniqueness ratio: 15
- P1: 1,000
- P2: 10,000
- Speckle window size: 10
- Speckle range: 2

Any other parameters not listed above were disabled.

The process for ELAS was slightly different than for block matching and semi-global matching. The library used to generate the disparity map is *libELAS*, which is distributed by the authors of ELAS. The process to generate a point cloud was:

1. Rectify the images.
2. Crop 128 pixels from the left of the images and 20 pixels from the right of the images, to remove parts of the ROV from the images.
3. Compute disparity with libELAS.
4. Reproject the disparity map to a point cloud using the reprojection matrix generated by camera calibration and OpenCV's *reprojectImageTo3D* function.
5. Remove points closer than 0.2 m and further than 1.2 m from the camera.
6. Apply the colour of the corresponding pixel from the left image to the points.

For ELAS the parameters used were:

- Minimum disparity: 0
- Number of disparities: 96
- Subsampling: disabled

Evaluation

The point clouds can be compared subjectively by plotting and inspecting for noise and outliers, and by comparing the number of points. Because of the differences in the processes to generate the point clouds it is difficult to compare block matching/semi-global matching and ELAS by the number of points. Block matching and semi-global matching will not compute disparities for the left side of the left image, and right side of the right image, up to the maximum disparity distance given by the parameters (*number of disparities* − *minimum disparity*). For the images being evaluated this is useful, since it crops out parts of the image that contain the ROV, however it is difficult to achieve the same effect with ELAS. Although ELAS has a maximum disparity distance, it first uses support points (sparse features) to initialise the disparity search, and does not crop the sides of the images like block matching and semi-global matching. Instead, to remove the parts of the image that contain the ROV, the left 128 pixels and right 20 pixels were cropped from the images. This is similar, but not identical, to the parts of the image that are cropped by block matching and semi-global matching. Sparse features were cropped in the same way so it is possible to compare the area covered by sparse feature matching methods and ELAS.

Results

Table 5.1 lists the number of points and the area covered by each sparse feature method for each image pair. For every image pair KAZE has the highest number of points detected and matched, and (with the exception of image pair 6) largest convex hull area. Generally SURF has significantly fewer points than KAZE, however the convex hull area is similar or only moderately smaller. With the exception of image pair 6, BRISK detects more features than SURF, while FREAK often detects a similar number of features, however both consistently have a smaller convex hull area.

Table 5.2 lists the number of points generated by each dense stereo matching method for each image pair. The table shows that semi-global matching generates more points than block matching. As mentioned previously, the number of points generated by block matching/semi-global matching and ELAS are not directly comparable due to differences in the processes used, however the table suggests that the number of points generated by ELAS is similar to the other methods. Since the image cropping used for the sparse feature methods is the same as used for ELAS, the area covered by sparse features and the number of points generated by ELAS can be compared. The area covered by KAZE and the points generated by ELAS are very similar, with KAZE ahead in some image pairs, and ELAS ahead in others.

To evaluate the performance of the algorithms the point clouds can be evaluated subjectively. The following pages show the images after rectification and the point clouds generated by each algorithm, for each of the image pairs.

Image pairs 1 and 2 are likely the easiest to process. They are looking at a well lit

Image	SURF		KAZE		FREAK		BRISK	
	Quantity	Area (MP)	Quantity	Area (MP)	Quantity	Area (MP)	Quantity	Area (MP)
1	326	0.248	1267	0.256	314	0.227	608	0.231
2	240	0.239	859	0.251	180	0.227	304	0.233
3	97	0.174	373	0.179	100	0.122	136	0.094
4	175	0.142	638	0.175	250	0.136	364	0.139
5	272	0.223	1035	0.237	383	0.204	480	0.206
6	104	0.189	246	0.182	25	0.107	57	0.141

Table 5.1: Quantity and area (in megapixels) of landmarks produced by stereo feature detection and matching

Image	Block Matching	Semi-Global Matching	ELAS
1	0.260	0.290	0.249
2	0.250	0.286	0.245
3	0.109	0.215	0.177
4	0.153	0.224	0.193
5	0.215	0.269	0.237
6	0.110	0.235	0.195

Table 5.2: Quantity (in millions) of points produced by dense stereo matching methods

pile face with even biofouling, minimal background light and minimal floating material. For image pair 1 (Figure 5.2) all methods work well with few erroneous points. ELAS (Figure 5.3d) and KAZE (Figure 5.3a) have the best appearance. With image pair 2 (Figure 5.4) an adjacent face of the pile is barely visible on the right of the image. The sparse methods (Figure 5.5a) do not show any points from this face however the dense methods do. Block matching (Figure 5.5b) gives several clusters of points belonging to the face, while ELAS (Figure 5.5d) extends a smooth line of points along the face. These appear to be outliers but could be a correct representation of part of the face. Semi-global matching (Figure 5.5c), however, definitely has clusters of outliers, as the points extend much further than the width of the face.

Image pairs 3 and 4 contain a substantial amount of floating material. Image pair 3 (Figure 5.6) also has a substantial amount of ambient light in the background, is looking at the corner of two pile faces, and is probably the most difficult to process. On the other hand image pair 4 (Figure 5.8) is looking directly at a pile face with minimal background lighting. Block matching (Figures 5.7b and 5.9b) and semi-global matching (Figures 5.7c and 5.9c) have difficulty with both image pairs. There are a substantial number of clusters of outliers, most likely due to the floating material. The sparse methods (Figures 5.7a and 5.9a) and ELAS (Figures 5.7d and 5.9d) work well for image pair 4, but the background ambient light of image pair 3 causes a problem for all algorithms. The background light is not featureless and not identical in the left and right image, which causes points to be generated. The grasping arm is also visible to the left of the pile and creates points in the point clouds, however these are not erroneous. With ELAS the background light and grasping arm cause the surface of the pile in the point cloud to extend beyond the face of the actual pile.

Image pair 5 (Figure 5.10) are images of a pile mostly cleaned of biofouling, looking towards the corner of a face. The sparse methods, block matching, and semi-global matching (Figures 5.11a, 5.11b, and 5.11c respectively) generally do not identify much from the adjacent face to the right. ELAS (Figure 5.11d) fills in the adjacent right face very well. All methods generate points to the right side above the face pile, which could be due to the grasping arm or background light.

Image pair 6 (Figure 5.12) has thick biofouling that appears to be poorly lit, and moderate background light. KAZE and SURF (Figure 5.13a) generate a fair number of points without significant outliers, although the distribution across the image is uneven. Block matching (Figure 5.13b) is also quite patchy across the image, with a few outlier clusters. Semi-global matching (Figure 5.13c) fills in more of the face than block matching, but also adds a significant number of outlier clusters. ELAS (Figure 5.13d) perceives the biofouling well, generating a mostly complete point cloud, however there appears to be an erroneous support point which has caused a line of points to extend out to the left.

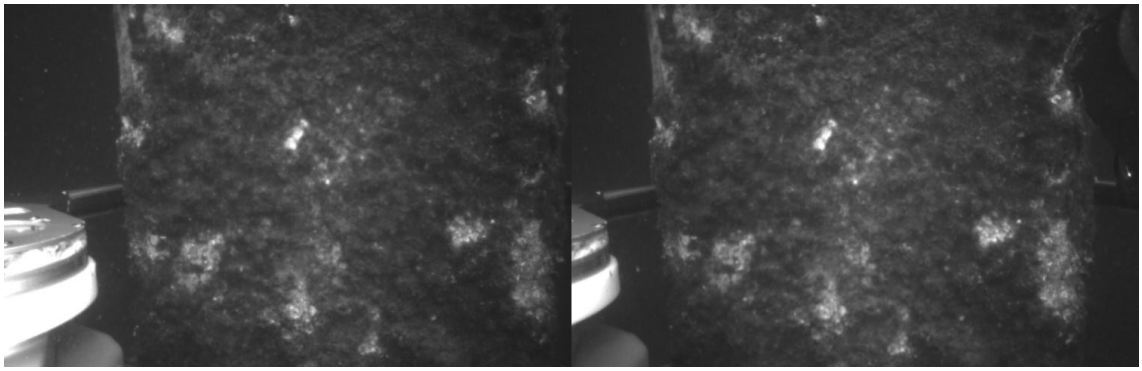
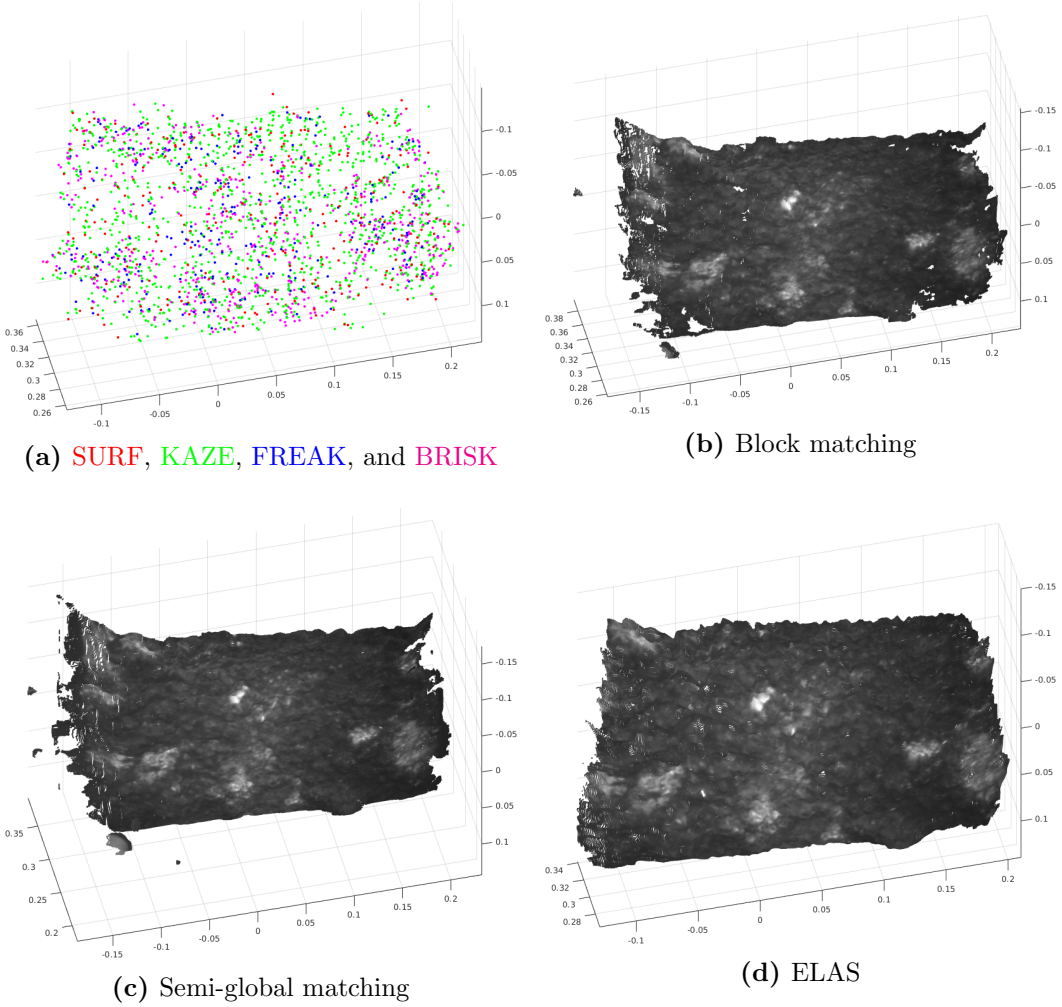


Figure 5.2: Image pair 1 after rectification



(a) SURF, KAZE, FREAK, and BRISK

(b) Block matching

(c) Semi-global matching

(d) ELAS

Figure 5.3: Point clouds generated from image pair 1

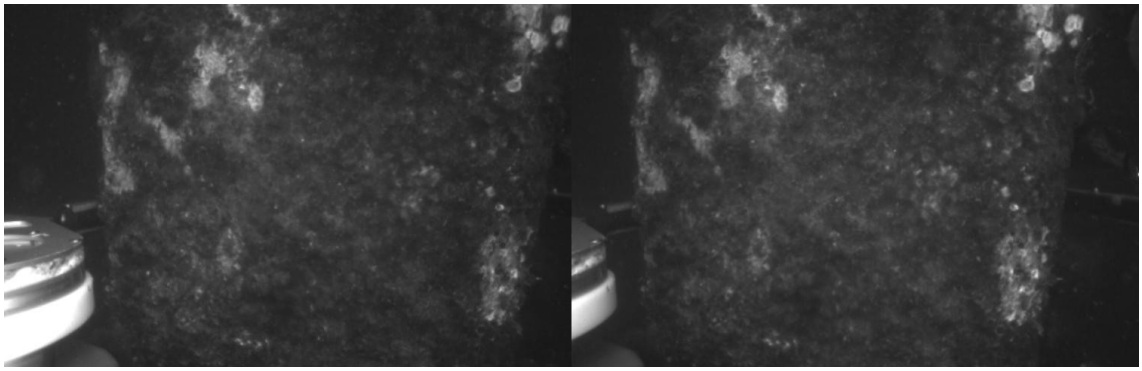
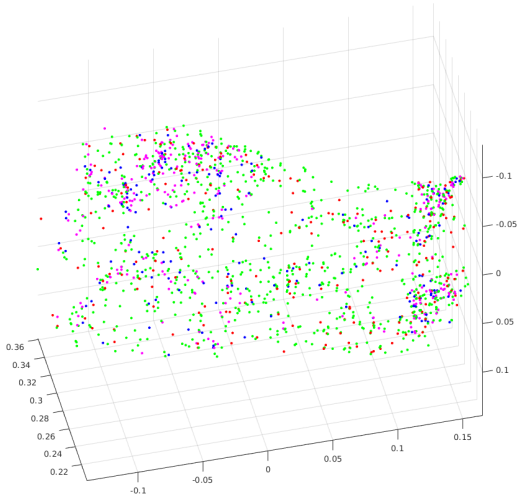
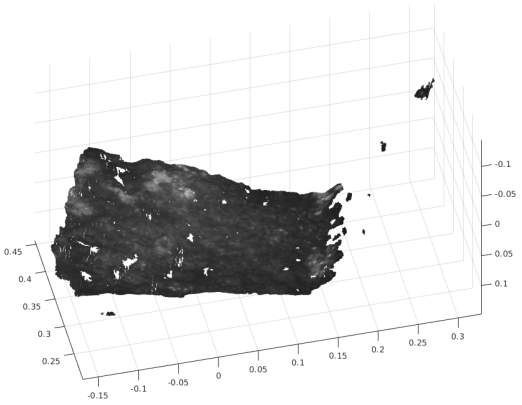


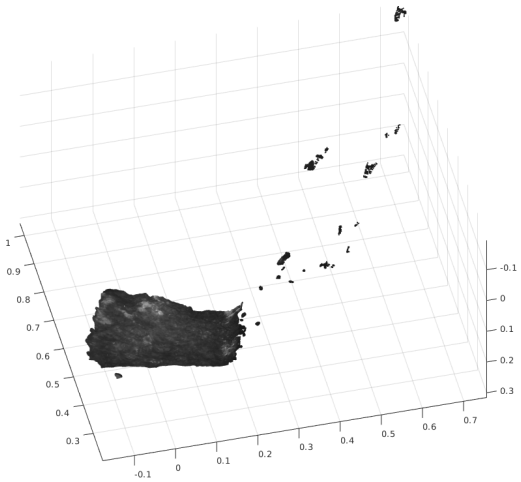
Figure 5.4: Image pair 2 after rectification



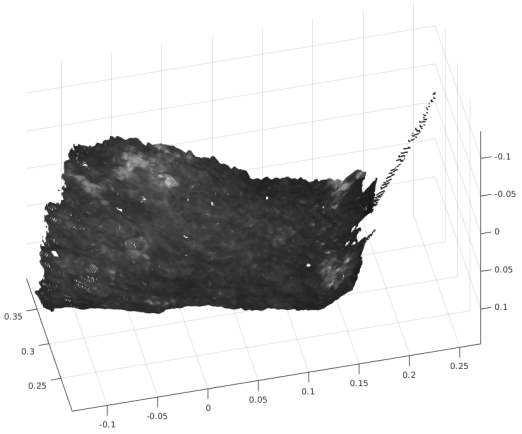
(a) SURF, KAZE, FREAK, and BRISK



(b) Block matching



(c) Semi-global matching



(d) ELAS

Figure 5.5: Point clouds generated from image pair 2

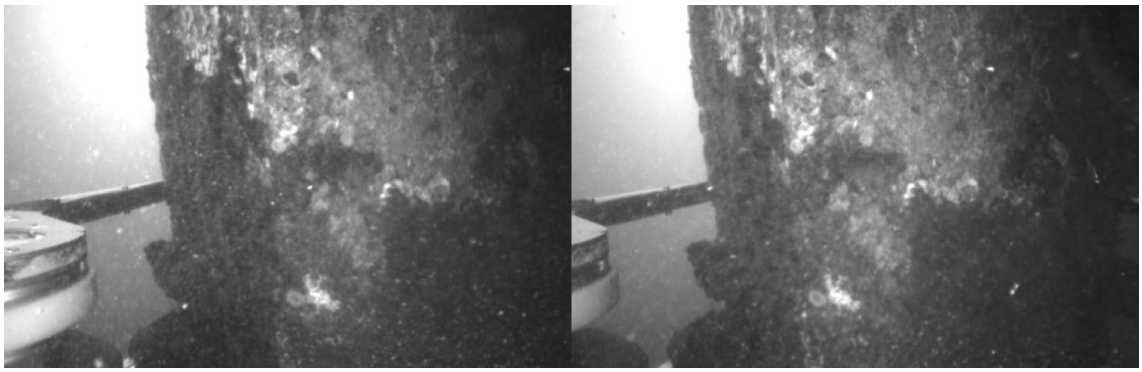
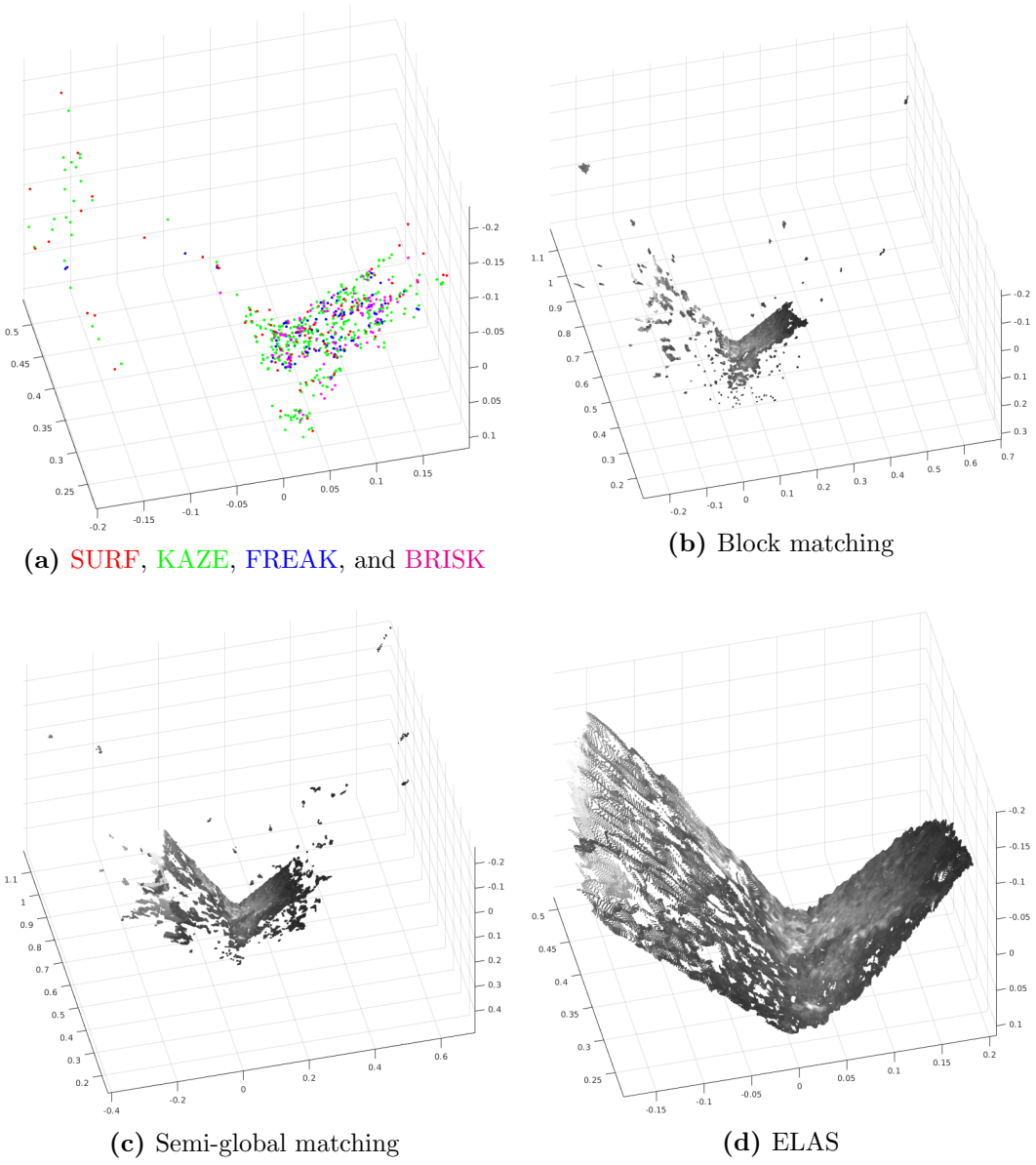


Figure 5.6: Image pair 3 after rectification



(a) SURF, KAZE, FREAK, and BRISK

(b) Block matching

(c) Semi-global matching

(d) ELAS

Figure 5.7: Point clouds generated from image pair 3

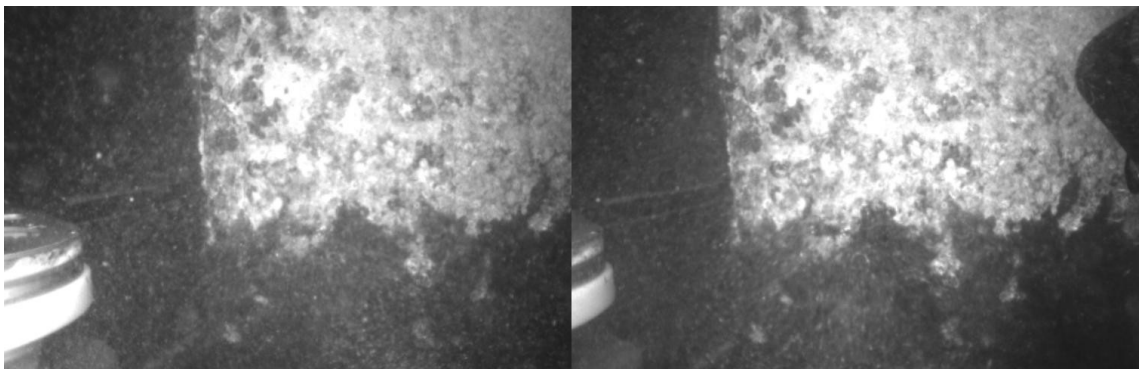
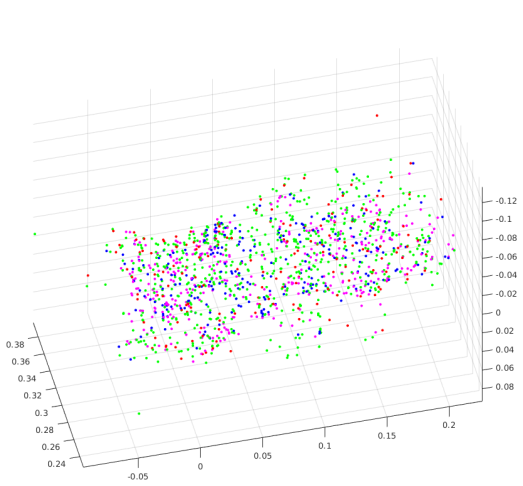
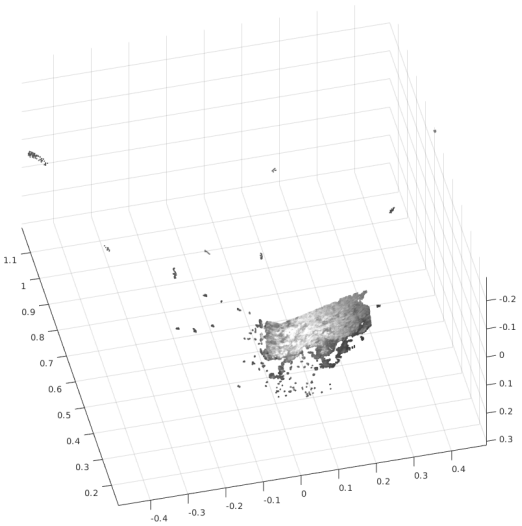


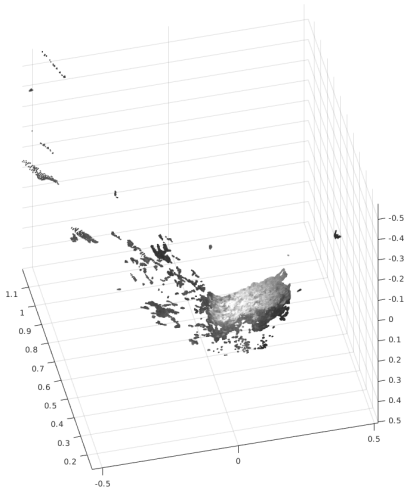
Figure 5.8: Image pair 4 after rectification



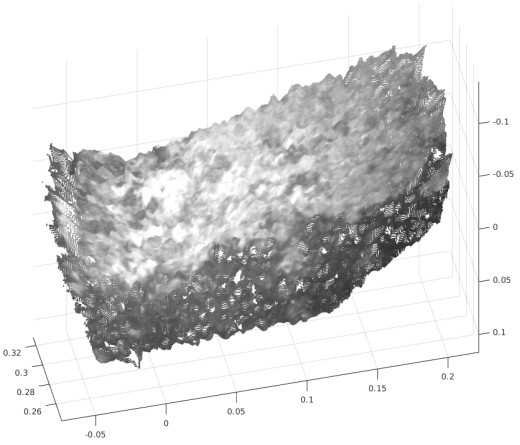
(a) SURF, KAZE, FREAK, and BRISK



(b) Block matching



(c) Semi-global matching



(d) ELAS

Figure 5.9: Point clouds generated from image pair 4

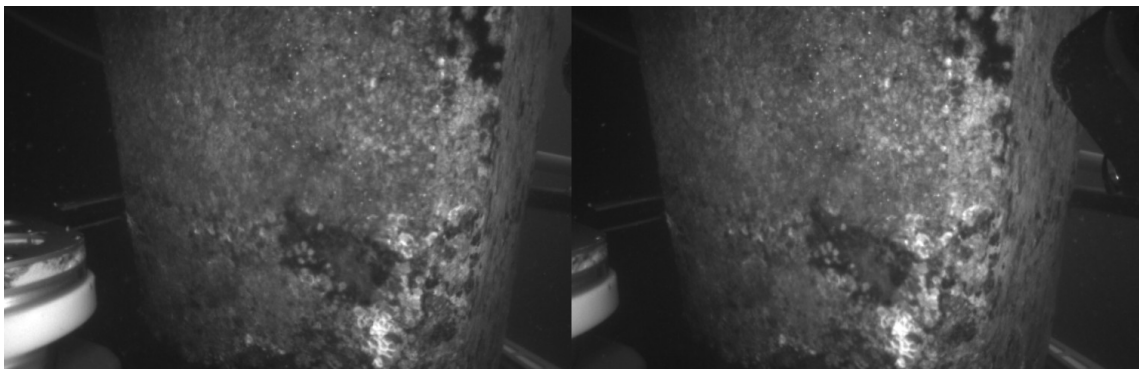
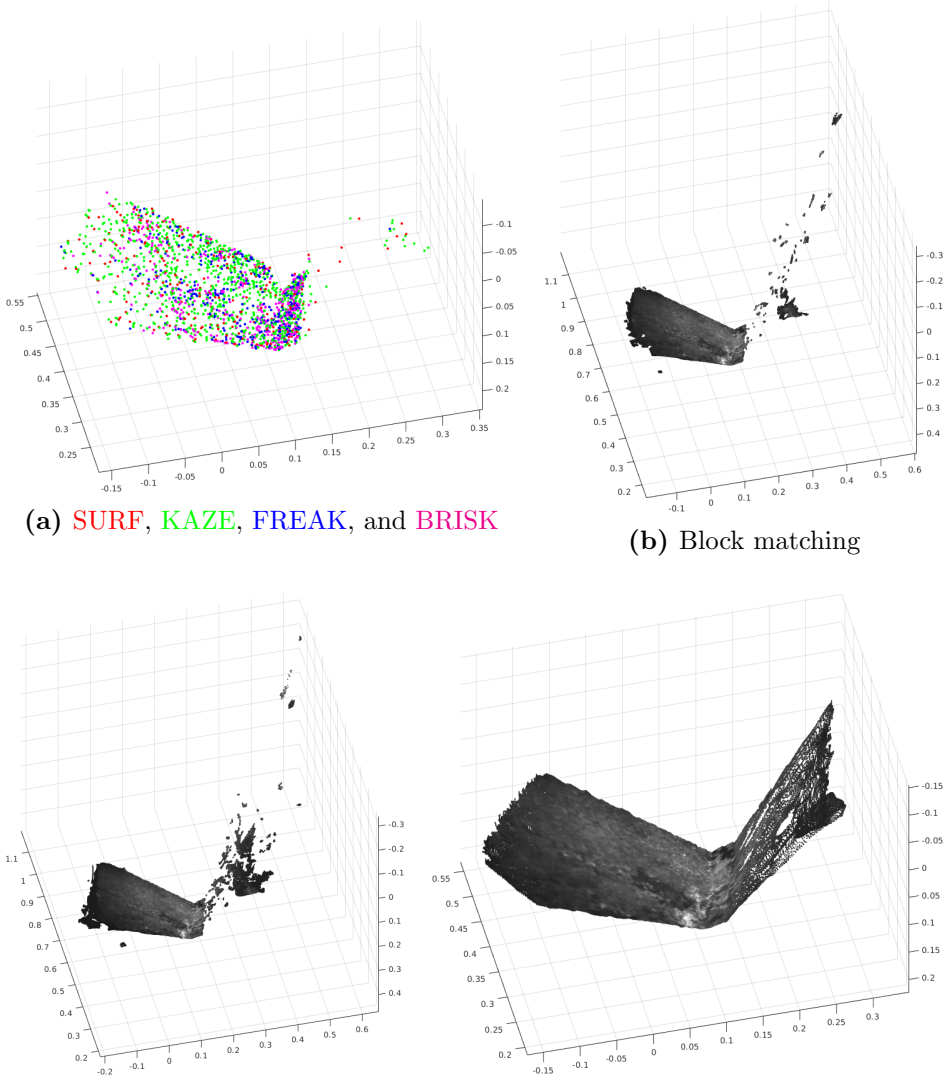


Figure 5.10: Image pair 5 after rectification



(a) SURF, KAZE, FREAK, and BRISK

(b) Block matching

(c) Semi-global matching

(d) ELAS

Figure 5.11: Point clouds generated from image pair 5

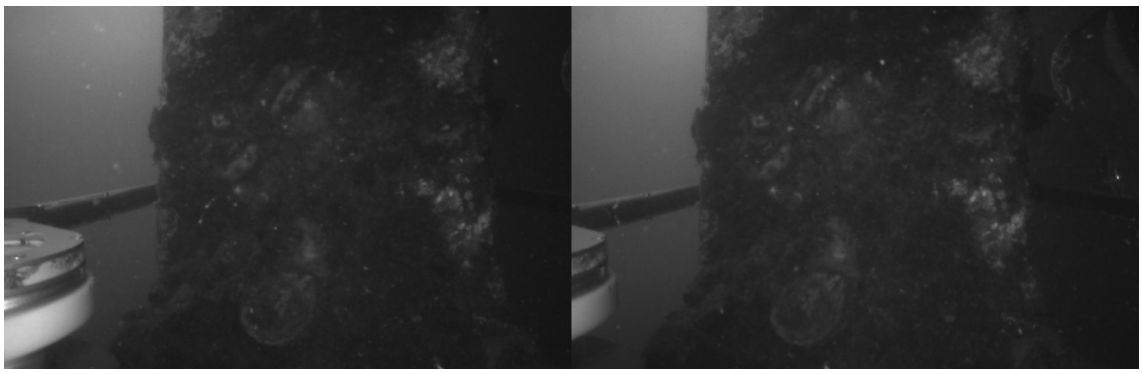
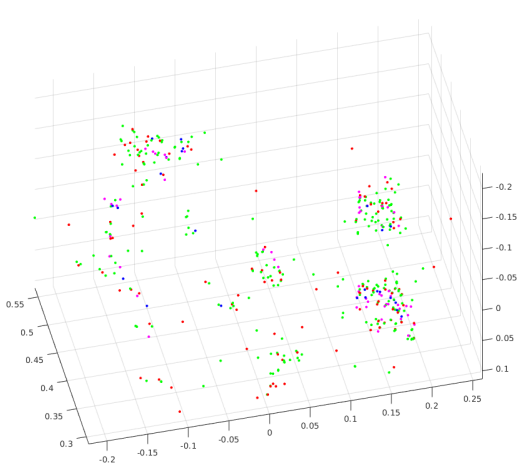
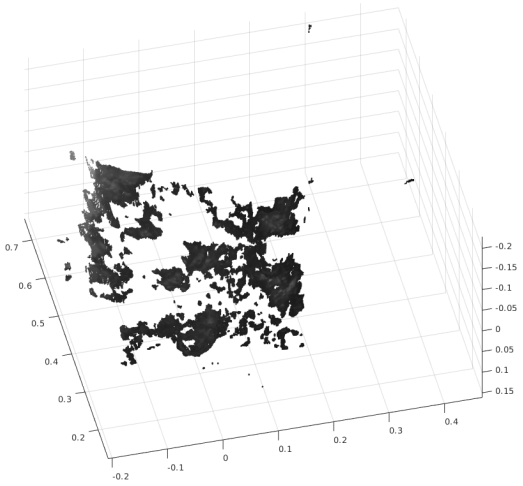


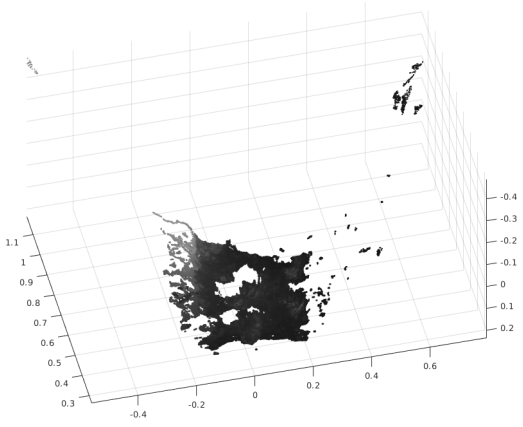
Figure 5.12: Image pair 6 after rectification



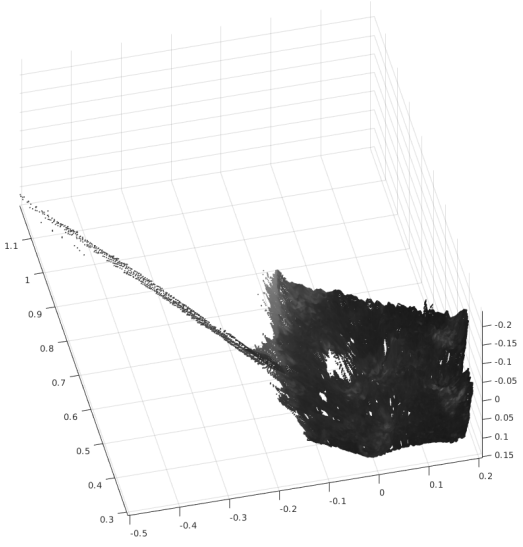
(a) SURF, KAZE, FREAK, and BRISK



(b) Block matching



(c) Semi-global matching



(d) ELAS

Figure 5.13: Point clouds generated from image pair 6

Discussion

Of the sparse methods, KAZE performs better than the rest. Although KAZE matches significantly more features than the other algorithms, for all algorithms the amount of points generated and area covered are generally adequate for determining a cleaning path, and there were no significant outliers or failures for the image pairs evaluated. The sparse methods were not able to perceive an adjacent pile face at a steep angle, but this should not matter for the purpose of cleaning, since the blasting arm will be unable to reach that face.

Of the dense methods, ELAS performs better than block matching or semi-global matching. The point clouds ELAS generated were a more complete reconstruction of the surface, with fewer outliers. ELAS was able to generate the surface of a steep face where block matching and semi-global matching would only generate patchy clusters. However, if an erroneous support point is generated in a poorly textured area, ELAS can create an erroneous surface out from the correct support points to the erroneous support point. Of the other dense methods, semi-global matching filled in the surface of the pile better than block matching, but also had more clusters of outliers.

Block matching and semi-global matching had significant problems with floating material, but this was ignored by ELAS and the sparse methods. Feature methods extract a descriptor from the region around a feature, which is used to identify matches. Given the size of the floating material it is likely that any descriptor extracted includes pixels from the pile. Because the floating material is closer than the surface of the pile, it is likely the same floating material will have a different background in the left and right camera, and therefore a different descriptor, preventing a point being generated. The same would be true of ELAS, as it uses sparse features to initialise a pixelwise correspondence search.

All the methods had problems with a bright background and the gasping arms of the ROV. This is understandable, since they should not be expected to ignore objects which exist in the scene. It may be possible to know the position of the grasping arms, and therefore subtract them from the resulting point cloud. More generally, it may be possible to segment the image to isolate the region that contains the pile from everything else, or to segment the features detected from the pile from the features detected from anything else.

In Sections 3.5 and 5.3 visual odometry and visual SLAM are evaluated. A product of these is robust sparse features. As the features are tracked over multiple positions of the camera, erroneous features or features belonging to the ROV itself can be eliminated, as the motion of these features will not match the motion of the pile in relation to the camera. These robust features could be used to help single viewpoint perception ignore the background or grasping arms.

The next section evaluates combining multiple single viewpoint reconstructions into a reconstruction of a larger section of the pile.

5.3 Visual SLAM and Multi-View 3D Reconstruction

Like any built structure, concrete bridge piles require regular inspection to confirm that the structures are safe. The aim of this section is to create an accurate 3D reconstruction of a bridge pile from stereo images collected by an ROV, for the purpose of inspecting the condition of a pile. Section 5.2 evaluated methods of generating a point cloud from a pair of stereo images captured from a single viewpoint. A simple method of creating a 3D reconstruction of a larger section of the pile is to merge the single viewpoint reconstructions together. This requires an accurate pose for each viewpoint. SfM and visual SLAM are two techniques for determining accurate camera poses from images.

Data collection

Four data sets were selected for the evaluation of visual SLAM and 3D reconstruction. All were collected on site at a real bridge. The calibration parameters used were also generated from images collected on site. On site, poor visibility and the flow of water make it difficult to navigate around the pile. During data collection the grasping arms of the ROV were used to prevent the ROV from moving too far from the pile. The grasping arms are positioned in a loose grasp that allows the ROV to rotate around the pile without detaching from the pile completely. Unfortunately this makes smooth movement impossible, and it is also difficult to remove the grasping arms from the images collected.

In each data set the goal is for the ROV to start on a face of the pile, rotate around the pile to come back to the first face, descend approximately 0.25 m, and repeat. Occasionally the grasping arms catch on the pile or the tether cable prevents movement, and the ROV must move backwards or shake free of the obstruction before continuing. The first data set was collected by manually controlling the ROV. For convenience the other data sets were collected with a simple automated program. Initially the ROV is manually positioned looking at a face of the pile, and the grasping arms are closed to loosely hold onto the pile. The automated program moves left or right until a magnetometer indicates that the ROV has returned to its original position, then descends until the depth sensor indicates that the ROV has descended 0.25 m, and then moves back in the opposite direction. This repeats several times. Because the grasping arms hold the pile in a loose grasp, a simple left or right translation causes the ROV to rotate around the pile. If the magnetometer indicates that the ROV is not rotating around the pile, the program will move the ROV in the opposite direction, then attempt to continue, until the obstruction is passed.

All data sets have varying background ambient lighting, floating material, and amounts of biofouling. Background lighting is dependant on the position of the ROV on the pile, the time of day that the data was recorded, and the position of the pile in relation to the bridge. Data sets 1 and 3 have sections of strong background light, data set 2 has sections of at worst moderate background light, and data set 4 generally has minimal background light. The thickness of biofouling tends to increase at greater depths, although in data set

4 there are areas of the pile that have been cleaned of biofouling.

The approximate length of each data set is listed in the results section. All data sets were captured at 20 frames per second, and a fixed exposure time of 3 ms.

ORB-SLAM2

Although numerous SfM and visual SLAM systems are freely available, compiling and using them can be challenging and time consuming, as they often depend on specific versions of other libraries that also need to be installed or compiled. SfM was attempted following MATLAB's tutorial on SfM and also with MVE, however neither were able to process the data sets. ORB-SLAM2 was compiled and run successfully, and was able to process the data sets. Unlike MVE or MATLAB's SfM tutorial, ORB-SLAM2 is able to process stereo vision so that the output is correctly scaled.

As the name implies, ORB-SLAM2 uses ORB features. ORB is similar to BRISK, which was evaluated in Section 3.5. Both are binary, rotationally invariant, improve on the computational performance of SIFT and SURF, use FAST for feature detection, and were introduced in 2011.

A number of modifications were made to ORB-SLAM2, primarily for ease of use. It was also modified to output key frame images and poses to files, for further processing in MATLAB.

The following parameters were used for ORB-SLAM2:

- Depth threshold: 35
- Number of features: 2000
- Scale factor: 1.2
- Number of levels: 8
- Initial FAST threshold: 12
- Minimum FAST threshold: 7

The parameters were selected after a brief evaluation of the default parameters in the example configuration files included with ORB-SLAM2. For testing, the "number of features" parameter was set to 1200, 2000, and 4000, and the "initial FAST threshold" was set to 12 and 20. Lowering the "initial FAST threshold" to 12 was found to substantially improve the number landmarks tracked over a section of data. A larger value for "number of features" also increased the number of landmarks tracked, however a value of 2000 was found to provide good number of features, while a value of 4000 substantially increased computation cost for processing each frame.

The effect of CLAHE image enhancement was also evaluated. In Section 3.4, enhancing images with CLAHE was found to improve the number of features matched when applied to feature detection only. CLAHE for feature detection only was not attempted with ORB-SLAM2, as it was believed that this would require significant modification of the code. CLAHE for both feature detection and description was evaluated, and was not

found to improve the number of landmarks tracked, when using the "number of features" and "initial FAST threshold" parameters that had been selected.

To improve performance and reliability, the input images were cropped after rectification, before being processed by ORB-SLAM2. 200 pixels were removed from the left and 100 pixels from the right of both stereo images. These parts of the image usually only contain the background and grasping arms, which should be ignored so that ORB-SLAM2 can localise in relation to the pile.

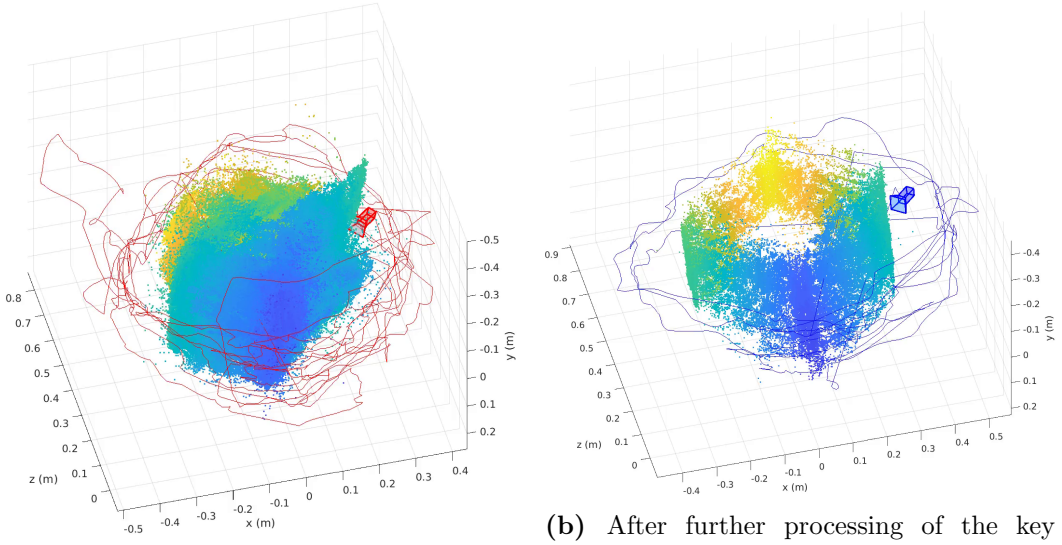
In early testing ORB-SLAM2 was found to perform poorly, so the information from ORB-SLAM2 was processed further to improve the result. SIFT features were detected in the key frames selected by ORB-SLAM2, then matched by comparing key frame pairs which ORB-SLAM2 indicates are connected. Bundle adjustment was performed with the new features, initialised with the original camera poses from ORB-SLAM2. The result improved significantly, as shown in Figure 5.14. Subsequently it was found that the poor performance of ORB-SLAM2 was cause by an issue with data capture from the stereo camera. The camera connects to the ROV computer by universal serial bus (USB). A number of other devices were also connected by USB, which caused contention on the bus and resulted in an uneven frame rate and occasionally corrupted frames. ORB-SLAM2 requires a consistent frame rate, as it uses a motion model that assumes the motion of the camera between the current and previous frame is similar to the previously estimated motion. Once the issue with data capture was corrected ORB-SLAM2 was able to process the data sets, removing the need for additional feature detection and bundle adjustment.

ORB-SLAM2 uses g2o for bundle adjustment, which uses Cartesian parameterisation. Although this has been shown to be inferior to inverse depth or parallax parameterisations, replacing the bundle adjustment component of ORB-SLAM2 would require significant work, and this was not attempted.

Multi-view 3D reconstruction

The key frames that ORB-SLAM2 selected were used to create a 3D reconstruction of the pile, by merging the reconstructions from each stereo image pair. Each key frame reconstruction was generated with block matching, semi-global matching, and ELAS, as described in Section 5.2. Each point cloud is then transformed into the global frame then merged with MATLAB's *pcmerge* function, using a grid step of 0.001 m for the filter.

Although ELAS was subjectively the best performing method for single viewpoint 3D reconstruction, its tendency to fill in untextured areas to outlier support points can produce a poor result when multiple reconstructions are merged together. A support point detected in the background or on the grasping arm can generate an erroneous surface from the pile to the point, which adds a significant amount of noise when multiple point clouds are merged. To mitigate this, the images processed by ELAS were cropped significantly, removing 150 pixels from the left, 100 pixels from the right, and 50 pixels from the top and bottom. This is in addition to the cropping for ORB-SLAM2, leaving an image that



(a) Initial result from ORB-SLAM2

(b) After further processing of the key frames selected by ORB-SLAM2

Figure 5.14: Camera trajectory and landmarks of underwater visual SLAM with data capture issues

is just 202 pixels wide and 380 pixels high. Even so, there are still erroneous surfaces generated in the point cloud, and cropping further reduces the amount of the pile surface in the final multi-view reconstruction.

Block matching and semi-global matching also generate points from the background and grasping arms, but these are much more sparse and generally isolated from the pile. As shown in Section 5.2, these methods generate points from floating material which, although they may not be incorrect, are unwanted. To improve the quality of the reconstruction, distance segmentation was applied to each single viewpoint reconstruction using MATLAB's *pcsegdist* and a distance of 0.01 m. After segmentation only the largest cluster of points is kept and merged into the multi view reconstruction. This is effective at removing noise and points generated from floating material. Valid points may also be removed, but as multiple reconstructions are merged it is likely that the surface of the pile will be adequately covered.

Results

Table 5.3 lists the duration of each data set and the number of key frames and landmarks that ORB-SLAM2 has generated, and Figure 5.15 shows a visualisation of the landmarks. Data set 1 has the longest duration, largest number of key frames and landmarks, and most coverage of a pile. Data set 1 was captured with manual control, while the other sets were captured with an automated program. Although manual control is ideal, the automated program is generally adequate and much more convenient.

Data set 2 covers a reasonable height of the pile, however there is a gap towards the bottom where there is not enough overlap when rotating around the pile a different depths.

Data set	Duration (sec)	Key frames	Landmarks
1	385	232	18307
2	275	96	10043
3	320	100	7230
4	350	128	9865

Table 5.3: Duration of each data set and number of key frames and landmarks generated by ORB-SLAM2

Data sets 3 and 4 generally have good coverage although they are shorter than data set 1. Data set 4 was captured on a deeper section of a pile which has thicker biofouling. The landmarks are less dense than the landmarks generated from the other data sets, as the pile was not as well lit.

Each data set has some outliers, but they are relatively few compared to the correct landmarks from the pile.

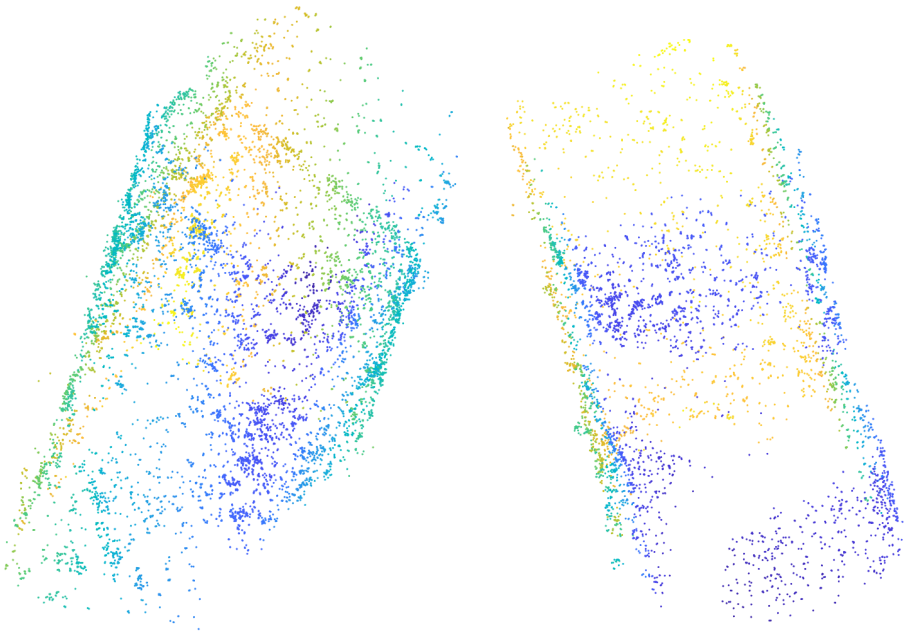
Figures 5.16 to 5.19 show the 3D reconstruction of each data set generated by block matching, semi-global matching, and ELAS. The block matching and semi-global matching reconstructions are created by merging the largest cluster from a single viewpoint point cloud. The reconstructions from the full point clouds are not shown as they contain a significant amount of noise. Generally block matching and semi-global matching produce similar results, with semi-global matching filling in more of he surface, but also producing more erroneous points. ELAS can generate a smoother surface than the other methods, but also often produces streaks of erroneous points.

For data set 1 the results by block matching and semi-global matching are similar. The surface of the pile is adequately reconstructed however there are points, typically around the edges, which have been generated from the background or from the grasping arms. With semi-global matching there are more of these points, creating protrusions from the pile surface that do not exist. ELAS generates the surface of the pile very well, however in some views it has detected support points in the background and generated streaks of points out from the pile surface.

The reconstructions produced from data set 2 are similar to data set 1, although with less erroneous points as there is less background light in this data set, and with a section of the pile missing which the ROV did not adequately cover. The missing section in the reconstruction produced by ELAS is larger than with the other methods because of the significant cropping of the input images.

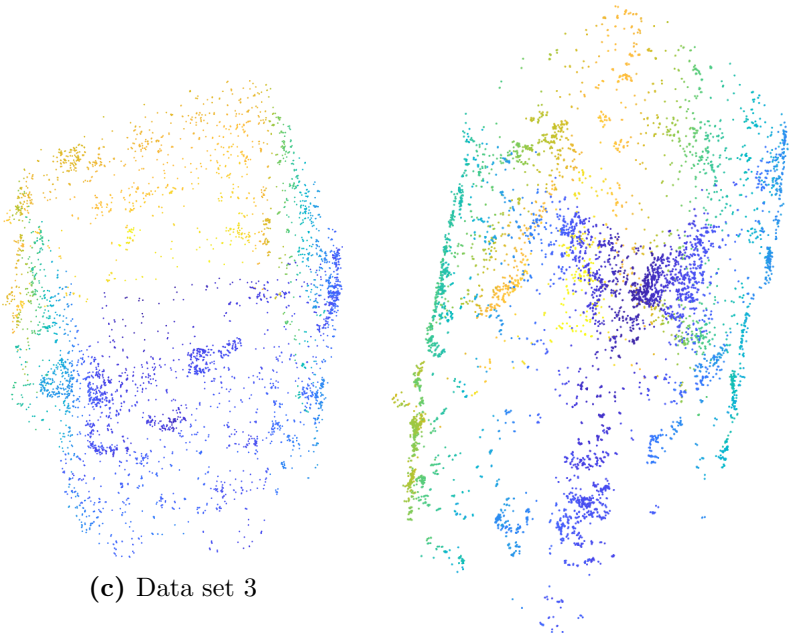
For data set 3 the reconstructions by all methods are very good. The result from ELAS is smoother than the other methods, but misses some parts of the surface due to cropping.

With data set 4 the reconstruction from semi-global matching is better than block matching. There is minimal noise from the background and there is better coverage



(a) Data set 1

(b) Data set 2



(c) Data set 3

(d) Data set 4

Figure 5.15: Landmarks from ORB-SLAM2

towards the bottom of the pile, however both results are very good. On the other hand ELAS has a number of streaks and more of the surface missing due to cropping.

Discussion

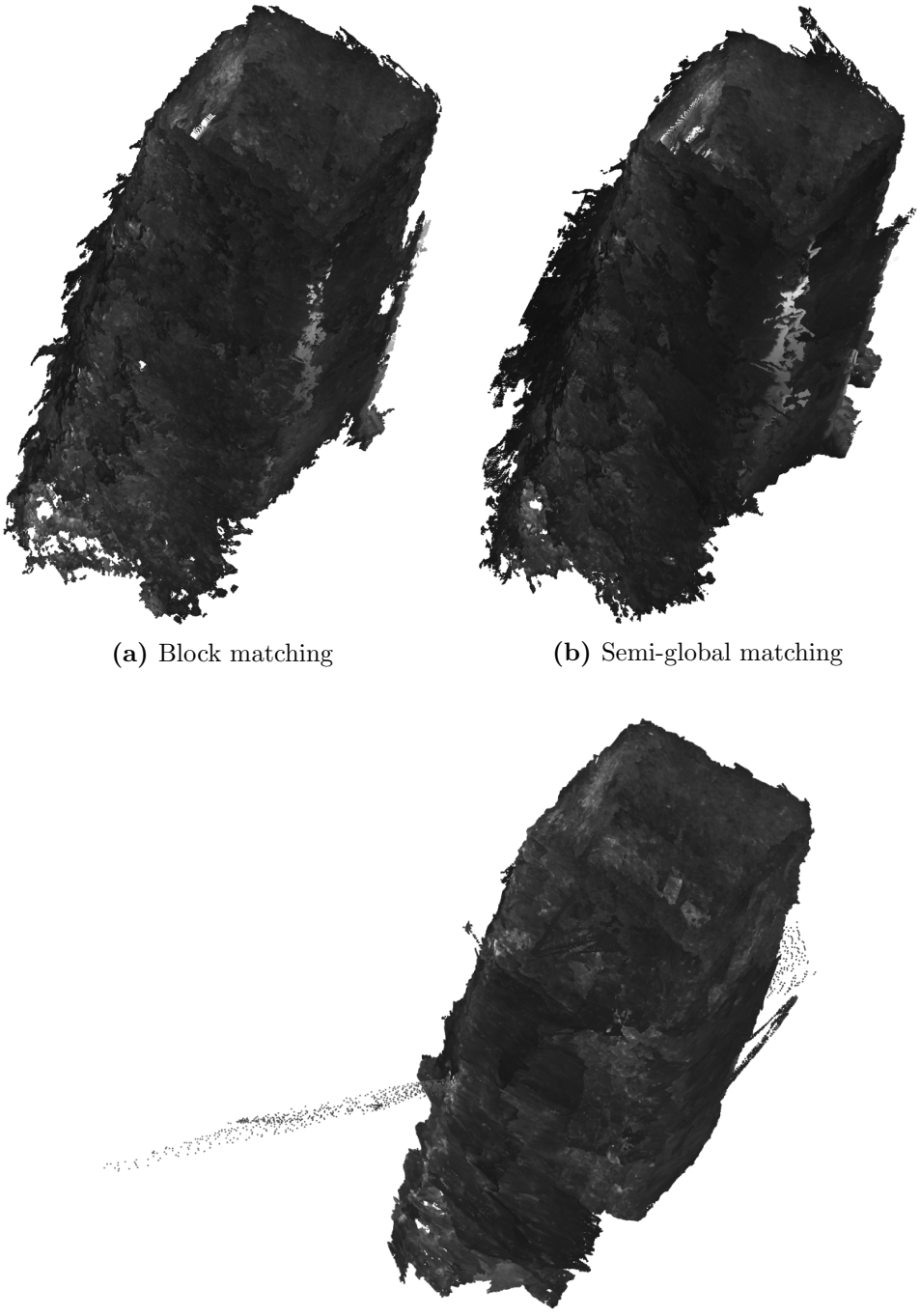
Both ORB-SLAM2 and multi-view 3D reconstruction by merging point clouds can perform well on the data sets captured. Like visual odometry evaluated in Section 3.5, ORB-SLAM2 handles moderate amounts of floating material well. Most floating material that generates a landmark in ORB-SLAM2 is removed as an outlier. Point clouds generated with block matching have a significant amount of noise but keeping only the largest cluster of points is an effective way to remove them, and merging multiple point clouds provides good coverage of the pile. There are, however, a number of ways accuracy and robustness could be improved.

The field of view of the lenses of the stereo camera is very wide, but because of the position of the camera much of the image is occupied by the ROV. These parts of the image are removed by rectification and cropping, significantly lowering the resolution of the scene. Increasing the focal length of the lenses on the stereo camera will result in a narrower field of view and a higher resolution for the useful part of the scene, which should improve the accuracy of both visual SLAM and 3D reconstruction.

In data set 4 there are images where the pile is poorly lit. This results in fewer landmarks from ORB-SLAM2 and holes in the point clouds generated by block matching. The cause of this is the ROV being in deeper water where there is minimal ambient light, and being positioned too far from the pile, where the artificial lights cannot properly illuminate the pile. Implementing auto exposure (as discussed in Section 3.3) may improve visibility of the pile in these situations.

The ORB descriptors used by ORB-SLAM2 are binary, however Section 3.5 showed that floating point descriptors were more successful than binary descriptors. Binary descriptors are used primarily for speed, but currently online operation is not required. Even if it were, with the computing power available on the ROV it is possible that a floating point descriptor could be processed fast enough for online operation. It is also possible that an unoriented descriptor could improve matching and processing time, since the ROV is generally always upright.

The main issue for both visual SLAM and 3D reconstruction is separating the pile from everything else in the image, mainly the background and grasping arms of the ROV. For visual SLAM any features detected from the effect of background ambient light or from the ROV will lower the accuracy of estimating the motion of the camera in relation to the pile. If erroneous features were to dominate the image, motion estimation would fail completely. Machine learning techniques could be useful in separating the pile from the background, either through semantic segmentation of the whole image, or by clustering the features descriptors. Both methods would require a large amount of training data. For clustering descriptors it may be possible to use data generated by ORB-SLAM2. After



(a) Block matching

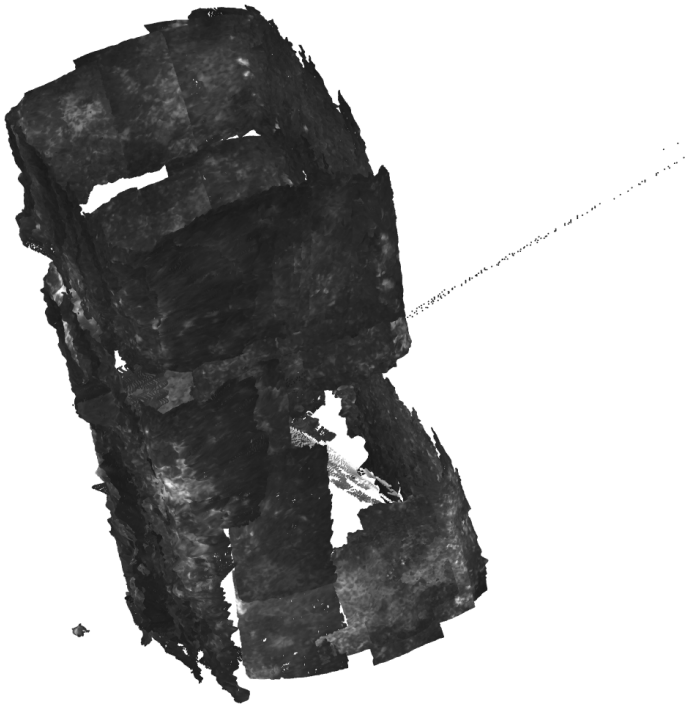
(b) Semi-global matching

(c) ELAS

Figure 5.16: 3D reconstructions of data set 1

(a) Block matching

(b) Semi-global matching



(c) ELAS

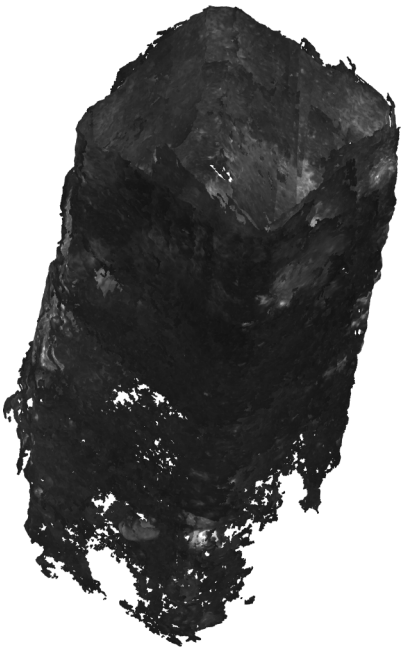
Figure 5.17: 3D reconstructions of data set 2

(a) Block matching

(b) Semi-global matching

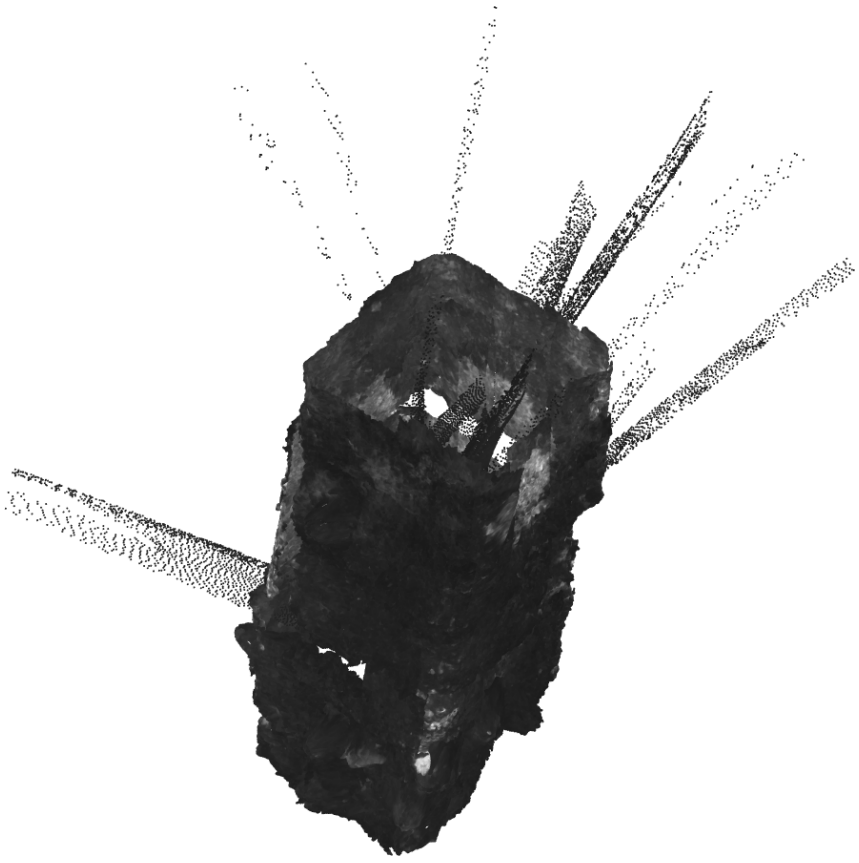
(c) ELAS

Figure 5.18: 3D reconstructions of data set 3



(a) Block matching

(b) Semi-global matching



(c) ELAS

Figure 5.19: 3D reconstructions of data set 4

processing a data set ORB-SLAM2 will have features that were kept as landmarks and are therefore likely to belong to the pile, and features that were removed as outliers. Training data for semantic segmentation is more difficult to generate, as the whole image needs to be labelled.

Another way to improve 3D reconstruction would be to use the information that is already available from visual SLAM. ELAS could be improved by using reliable landmarks from ORB-SLAM2 (ones that have been matched over multiple key frames) instead of having ELAS detect its own support points. This could eliminate the problem of generating points from the background or grasping arms, and reduce misalignment in point clouds taken at different viewpoints.

Currently when multiple point clouds are merged, there is no processing to normalise the colour of points that have been generated from different viewpoints. MATLAB's *pcmerge* function averages the location and colour of points within a grid box. When dealing with illumination variance, using the median value for colour may reduce the visibility of seams between point clouds. Correcting for illumination variance would become more important if auto exposure was used, or if the images were applied to a mesh as textures.

Finally, rather than controlling the ROV manually or with a simple automated program, it may be possible to use the information from an visual SLAM system (running online) to move the ROV more effectively. Currently the grasping arms are used to ensure that the ROV does not leave the pile, and in the simple automation program, to cause the ROV to rotate around the pile. If the thrusters of the ROV are able to move the ROV around the pile regardless of the flow of water, it should be possible to have the ROV automatically move around the pile without the use of the grasping arms. This would eliminate the grasping arms as an issue for visual SLAM and 3D reconstruction, and also speed up data collection as the ROV will not have to free itself when the arms are caught on the pile. Automating ROV control with visual SLAM could also ensure that there is adequate coverage of the pile, and that the correct distance from the camera to the pile is maintained, so that the lights are able to illuminate the pile adequately.

5.4 Summary

There are a number of methods that are effective at perceiving a submerged bridge pile from a stereo image pair captured at a single viewpoint. Sparse feature detection and matching (with the KAZE algorithm), block matching, semi-global matching and ELAS can all create a model adequate to assist with cleaning the pile. Block matching and semi-global matching generate a significant amount of noise, but this can be removed by segmenting the point cloud into clusters, and keeping only the largest cluster. All methods have an issue generating points from sunlight in the background, however this should not interfere with generating a path for cleaning the pile.

ORB-SLAM2 accurately and robustly estimates camera trajectory (as well as the struc-

ture of the pile), which can be used to merge single viewpoint reconstructions into a 3D reconstruction of an entire bridge pile.

6

Conclusion

ROVs are increasingly being used for the inspection and cleaning of underwater assets. Typically they are capable of little to no autonomy. This research looked at improving underwater perception, localisation, and mapping, using vision only. This has the potential to reduce the cost, or improve the autonomy, of ROVs.

Although vision has a shorter range, is less robust to poor water conditions, and requires more complicated processing, it has the potential to replace other more expensive sensors for some applications. With a reduced initial cost, the use of ROVs for asset inspection and maintenance could increase, and the corresponding reduction in the use of divers could improve safety.

The ongoing costs of asset maintenance could be reduced by improving the efficiency of asset maintenance operations. For example vision based localisation and mapping could be used to assist an ROV operator by providing information on where the ROV is in relation to an asset, reducing the duration of an operation. Robust visual perception could allow the automation of certain tasks, reducing the load on the operator and possibly improving the quality of work. Reduced ongoing costs may lead to an increase in the frequency of asset maintenance. Fully autonomous operation is another potential outcome, further reducing operational costs.

Stereo vision is a cost effective sensing modality, however the underwater environment that the ROV operates in presents a number of challenges. The environment has highly variable sunlight, poor visibility, and significant amounts of floating material.

Camera calibration is important for accuracy in computer vision tasks. A pinhole camera model is commonly used to represent a camera in-air, but this model does not accurately represent an underwater camera. The error can be minimised by mounting the camera lens as close to the front port of the underwater camera housing, and minimising the thickness of the front port. With this configuration good results are possible with the pinhole model, after the camera has been calibrated in the water.

VO is the process of estimating the motion of a camera from images. It involves feature detection, description, and matching to estimate the relative camera pose between two images. Well exposed images are critical for good feature detection and description, but this can be difficult in an outdoor environment with highly variable sunlight. Ideally

exposure would be controlled automatically to maximise the amount of information in the image, however using a fixed exposure level was shown to give adequate results. CLAHE improves visibility for an operator, and also improves the number of features detected, although using CLAHE enhanced images for descriptor extraction tends to reduce the performance of feature matching. Four feature detection and description algorithms were evaluated and KAZE was found to perform the best, detecting the most features and providing the most matches between images. Stereo visual odometry, using KAZE with CLAHE for feature detection, was successfully performed in the underwater environment. Although stereo visual odometry was shown to perform well, it may be improved with the addition of an IMU, particularly when dealing with poor visibility.

Bundle adjustment reduces camera trajectory drift and improves the consistency of landmarks in VO, SfM, and visual SLAM. Parallax parameterisation improves robustness to low parallax and coaxial landmarks, and using stereo information ensures that the scale is correct.

Visual SLAM builds upon VO and bundle adjustment to maintain a globally consistent scene structure. Due to the considerable work required to write a robust visual SLAM system, it was decided to use a publicly available system, ORB-SLAM2. Although ORB-SLAM2 uses binary features, it can perform well in the underwater environment around the bridge piles. The landmarks generated by ORB-SLAM2 provide a good representation of the pile, but a dense reconstruction can be created by merging single viewpoint stereo reconstructions. Block matching, semi-global matching and ELAS all produce a reasonable reconstruction. Block matching and semi-global matching both have significant noise when the scene contains floating material, but this can be reduced or eliminated by segmenting the point cloud into clusters and keeping on the largest cluster.

All three stereo correspondence algorithms have difficulty with erroneous features in the background of the image, and it might be beneficial to use the information available from visual SLAM to identify which parts of the image are valid. Visual SLAM tracks features across multiple viewpoints. Any features that are detected from the effect of sunlight in the background or floating material would not match the motion of the camera relative to the pile, and can be identified after a sufficient change in viewpoint.

There are machine learning methods of segmenting a whole image, and both stereo correspondence and visual SLAM could benefit from this. Motion estimation in visual SLAM would be improved by excluding erroneous features before motion estimation and outlier removal.

Future work for the SPIR project may involve:

- Implementing auto exposure or HDR imaging.
- Creating a new visual SLAM system using KAZE feature detection and description, or another well performing algorithm.
- Integrating IMU data into visual odometry/SLAM
- Improving stereo correspondence with information from visual SLAM or image seg-

mentation.

- Improving visual SLAM with image segmentation.
- Controlling the ROV autonomously using information from visual SLAM, to improve data capture.

7

Bibliography

[1] M. Massot-Campos and G. Oliver-Codina, "Optical sensors and methods for underwater 3D reconstruction," *Sensors*, vol. 15, no. 12, pp. 31 525–31 557, 2015.

[2] J. Geng, "Structured-light 3D surface imaging: A tutorial," *Advances in Optics and Photonics*, vol. 3, no. 2, pp. 128–160, 2011.

[3] N. Cadalli, P. J. Shargo, D. C. Munson and A. C. Singer, "Three-dimensional tomographic imaging of ocean mines from real and simulated LiDAR returns," in *Ocean Optics: Remote Sensing and Underwater Imaging*, International Society for Optics and Photonics, vol. 4488, 2002, pp. 155–167.

[4] B. D. Reineman, L. Lenain, D. Castel and W. K. Melville, "A portable airborne scanning LiDAR system for ocean and coastal applications," *Journal of Atmospheric and Oceanic Technology*, vol. 26, no. 12, pp. 2626–2641, 2009.

[5] F. Pellen, V. Jezequel, G. Zion and B. Le Jeune, "Detection of an underwater target through modulated LiDAR experiments at grazing incidence in a deep wave basin," *Applied Optics*, vol. 51, no. 31, pp. 7690–7700, 2012.

[6] D. J. Ketcham, "Real-time image enhancement techniques," in *Image Processing*, vol. 74, 1976, pp. 120–125.

[7] K. Zuiderveld, "Contrast limited adaptive histogram equalization," *Graphics Gems*, pp. 474–485, 1994.

[8] D. L. Rizzini, F. Kallasi, F. Oleari and S. Caselli, "Investigation of vision-based underwater object detection with multiple datasets," *International Journal of Advanced Robotic Systems*, vol. 12, no. 6, p. 77, 2015.

[9] L. Zheng, H. Shi and S. Sun, "Underwater image enhancement algorithm based on CLAHE and USM," in *2016 IEEE International Conference on Information and Automation*, IEEE, 2016, pp. 585–590.

[10] S. Hong and J. Kim, "Efficient visual SLAM using selective image registration for autonomous inspection of underwater structures," in *2016 IEEE/OES Autonomous Underwater Vehicles*, IEEE, 2016, pp. 189–194.

[11] Y. Y. Schechner and N. Karpel, "Clear underwater vision," in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 1, 2003, pp. I–I.

[12] N. Carlevaris-Bianco, A. Mohan and R. M. Eustice, "Initial results in underwater single image dehazing," in *OCEANS 2010*, IEEE, 2010, pp. 1–8.

[13] A. Levin, D. Lischinski and Y. Weiss, "A closed form solution to natural image matting," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 1, 2006, pp. 61–68.

[14] J. P. Queiroz-Neto, R. Carceroni, W. Barros and M. Campos, "Underwater stereo," in *17th Brazilian Symposium on Computer Graphics and Image Processing*, IEEE, 2004, pp. 170–177.

[15] M. Roser, M. Dunbabin and A. Geiger, "Simultaneous underwater visibility assessment, enhancement and improved stereo," in *2014 IEEE International Conference on Robotics and Automation*, IEEE, 2014, pp. 3840–3847.

[16] A. Geiger, M. Roser and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision*, Springer, 2010, pp. 25–38.

[17] M. Bryson, M. Johnson-Roberson, O. Pizarro and S. B. Williams, "True color correction of autonomous underwater vehicle imagery," *Journal of Field Robotics*, vol. 33, no. 6, pp. 853–874, 2016.

[18] K. A. Skinner, E. Iscar and M. Johnson-Roberson, "Automatic color correction for 3D reconstruction of underwater scenes," in *2017 IEEE International Conference on Robotics and Automation*, IEEE, 2017, pp. 5140–5147.

[19] Y. Rzhanov, L. M. Linnett and R. Forbes, "Underwater video mosaicing for seabed mapping," in *2000 International Conference on Image Processing*, IEEE, vol. 1, 2000, pp. 224–227.

[20] R. Garcia, T. Nicosevici and X. Cufí, "On the way to solve lighting problems in underwater imaging," in *OCEANS 2002*, MTS/IEEE, vol. 2, 2002, pp. 1018–1024.

[21] Y. Rzhanov and F. Gu, "Enhancement of underwater videomosaics for post-processing," in *OCEANS 2007*, IEEE, 2007, pp. 1–6.

[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[23] H. Bay, T. Tuytelaars and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision*, Springer, 2006, pp. 404–417.

[24] G. Wahba, "How to smooth curves and surfaces with splines and cross-validation," University of Wisconsin–Madison Department of Statistics, Tech. Rep., 1979.

[25] A. Fraser, R. Walker and F. Jurgens, "Spatial and temporal correlation of underwater sunlight fluctuations in the sea," *IEEE Journal of Oceanic Engineering*, vol. 5, no. 3, pp. 195–198, 1980.

[26] Y. Y. Schechner and N. Karpel, "Attenuating natural flicker patterns," in *OCEANS 2004*, MTS/IEEE, vol. 3, 2004, pp. 1262–1268.

[27] Y. Matsushita, K. Nishino, K. Ikeuchi and M. Sakauchi, "Shadow elimination for robust video surveillance," in *Workshop on Motion and Video Computing*, IEEE Computer Society, 2002, p. 15.

[28] Y. Weiss, "Deriving intrinsic images from image sequences," in *8th IEEE International Conference on Computer Vision*, IEEE, vol. 2, 2001, pp. 68–75.

[29] N. Gracias, S. Negahdaripour, L. Neumann, R. Prados and R. Garcia, "A motion compensated filtering approach to remove sunlight flicker in shallow water images," in *OCEANS 2008*, IEEE, 2008, pp. 1–7.

[30] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 369–378.

[31] S. Nuske, J. Roberts and G. Wyeth, "Extending the dynamic range of robotic vision," in *2006 IEEE International Conference on Robotics and Automation*, IEEE, 2006, pp. 162–167.

[32] S. Hrabar, P. Corke and M. Bosse, "High dynamic range stereo vision for outdoor mobile robotics," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 430–435.

[33] S. Hrabar, "3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2008, pp. 807–814.

[34] K. Irie, T. Yoshida and M. Tomono, "A high dynamic range vision approach to outdoor localization," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 5179–5184.

[35] I. Shim, J.-Y. Lee and I. S. Kweon, "Auto-adjusting camera exposure for outdoor robotics using gradient information," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 1011–1017.

[36] The MathWorks, Inc. (2018). Entropy of grayscale image - MATLAB entropy - MathWorks Australia, [Online]. Available: `https://au.mathworks.com/help/images/ref/entropy.html` (visited on 03/01/2019).

[37] F. Menna, E. Nocerino and F. Remondino, "Flat versus hemispherical dome ports in underwater photogrammetry," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 481, 2017.

[38] C. Kunz and H. Singh, "Hemispherical refraction and camera calibration in underwater vision," in *OCEANS 2008*, IEEE, 2008, pp. 1–7.

[39] T. Treibitz, Y. Schechner, C. Kunz and H. Singh, "Flat refractive geometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 51–65, 2012.

[40] J.-M. Lavest, G. Rives and J.-T. Lapresté, "Dry camera calibration for underwater applications," *Machine Vision and Applications*, vol. 13, no. 5-6, pp. 245–253, 2003.

[41] F. Menna, E. Nocerino, F. Fassi and F. Remondino, "Geometric and optic characterization of a hemispherical dome port for underwater photogrammetry," *Sensors*, vol. 16, no. 1, p. 48, 2016.

[42] A. Jordt-Sedlazeck and R. Koch, "Refractive calibration of underwater cameras," in *European Conference on Computer Vision*, Springer, 2012, pp. 846–859.

[43] A. Jordt-Sedlazeck and R. Koch, "Refractive structure-from-motion on underwater images," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 57–64.

[44] T. Łuczyński, M. Pfingsthorn and A. Birk, "The pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings," *Ocean Engineering*, vol. 133, pp. 9–22, 2017.

[45] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I - the first 30 years and fundamentals," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[46] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II - matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[47] M. Hassaballah, A. A. Abdelmgeid and H. A. Alshazly, "Image features detection, description and matching," in *Image Feature Detectors and Descriptors*, Springer, 2016, pp. 11–45.

[48] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, vol. 15, 1988, pp. 10–5244.

[49] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, Springer, 2006, pp. 430–443.

[50] D. G. Lowe, "Object recognition from local scale-invariant features," in *7th IEEE International Conference on Computer Vision*, Ieee, vol. 2, 1999, pp. 1150–1157.

[51] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[52] W. N. J. H. W. Yussof and M. S. Hitam, "Invariant Gabor-based interest points detector under geometric transformation," *Digital Signal Processing*, vol. 25, pp. 190–197, 2014.

[53] P. F. Alcantarilla, A. Bartoli and A. J. Davison, "KAZE features," in *European Conference on Computer Vision*, Springer, 2012, pp. 214–227.

[54] M. Calonder, V. Lepetit, C. Strecha and P. Fua, "BRIEF: Binary robust independent elementary features," in *European Conference on Computer Vision*, Springer, 2010, pp. 778–792.

[55] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 IEEE International Conference on Computer Vision*, IEEE, 2011, pp. 2564–2571.

[56] S. Leutenegger, M. Chli and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *2011 IEEE International Conference on Computer Vision*, IEEE, 2011, pp. 2548–2555.

[57] A. Alahi, R. Ortiz and P. Vandergheynst, "FREAK: Fast retina keypoint," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 510–517.

[58] E. Mair, G. D. Hager, D. Burschka, M. Suppa and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *European Conference on Computer Vision*, Springer, 2010, pp. 183–196.

[59] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007.

[60] R. Garcia and N. Gracias, "Detection of interest points in turbid underwater images," in *OCEANS 2011*, IEEE, 2011, pp. 1–9.

[61] F. K. Noble, "Comparison of OpenCV's feature detectors and feature matchers," in *23rd International Conference on Mechatronics and Machine Vision in Practice*, IEEE, 2016, pp. 1–6.

[62] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[63] J. Shi and C. Tomasi, "Good features to track," Cornell University, Tech. Rep., 1993.

[64] 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, 2018.

[65] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[66] D. Nistér, O. Naroditsky and J. Bergen, "Visual odometry," in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Ieee, vol. 1, 2004, pp. I–I.

[67] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[68] P. H. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[69] P. H. S. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp. 35–61, 2002.

[70] D. Nistér, O. Naroditsky and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[71] J. Zhang, S. Singh and G. Kantor, "Robust monocular visual odometry for a ground vehicle in undulating terrain," in *Field and Service Robotics*, Springer, 2014, pp. 311–326.

[72] B. M. Kitt, J. Rehder, A. D. Chambers, M. Schonbein, H. Lategahn and S. Singh, "Monocular visual odometry using a planar road model to solve scale ambiguity," in *European Conference on Mobile Robots*, 2011.

[73] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, 2007, pp. 3565–3572.

[74] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2012.

[75] T. Qin, P. Li and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[76] H. Strasdat, J. Montiel and A. J. Davison, "Real-time monocular SLAM: Why filter?" In *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 2657–2664.

[77] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*, Springer, 1978, pp. 105–116.

[78] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 3607–3613.

[79] S. Agarwal, K. Mierle *et al.* (2018). Ceres solver — a large scale non-linear optimization library, [Online]. Available: `http://ceres-solver.org` (visited on 03/01/2019).

[80] J. Civera, A. J. Davison and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[81] L. Zhao, S. Huang, Y. Sun, L. Yan and G. Dissanayake, "ParallaxBA: Bundle adjustment using parallax angle feature parametrization," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 493–516, 2015.

[82] N. Snavely, S. M. Seitz and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," in *ACM Transactions on Graphics*, ACM, vol. 25, 2006, pp. 835–846.

[83] C. Wu, "Towards linear-time incremental structure from motion," in *2013 International Conference on 3D Vision*, IEEE, 2013, pp. 127–134.

[84] S. Fuhrmann, F. Langguth and M. Goesele, "MVE - a multi-view reconstruction environment," in *EUROGRAPHICS Workshop on Graphics and Cultural Heritage*, 2014, pp. 11–18.

[85] P. Moulon, P. Monasse, R. Marlet *et al.* (2018). GitHub - openMVG/openMVG: open multiple view geometry library. basis for 3D computer vision and structure from motion, [Online]. Available: `https://github.com/openMVG/openMVG` (visited on 03/01/2019).

[86] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[87] P. Moulon, P. Monasse and R. Marlet, "Adaptive structure from motion with a contrario model estimation," in *Asian Computer Vision Conference*, Springer Berlin Heidelberg, 2012, pp. 257–270.

[88] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE/ACM International Symposium on Mixed and Augmented Reality*, IEEE, 2007, pp. 225–234.

[89] T. Pire, T. Fischer, J. Civera, P. De Cristóforis and J. J. Berlles, "Stereo parallel tracking and mapping for robot localization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2015, pp. 1373–1378.

[90] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 IEEE International Conference on Computer Vision*, IEEE, 2011, pp. 2320–2327.

[91] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation*, IEEE, 2014, pp. 15–22.

[92] J. Engel, T. Schöps and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *European Conference on Computer Vision*, Springer, 2014, pp. 834–849.

[93] 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*, IEEE, 2015, pp. 1935–1942.

[94] 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.

[95] 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.

[96] D. Schlegel, M. Colosi and G. Grisetti, "ProSLAM: Graph SLAM from a programmer's perspective," in *2018 IEEE International Conference on Robotics and Automation*, IEEE, 2018, pp. 1–9.

[97] R. Wang, M. Schwörer and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *2017 IEEE International Conference on Computer Vision*, vol. 42, 2017.

[98] T. Treibitz and Y.-Y. Schechner, "Active polarization descattering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 385–399, 2009.

[99] K. He, J. Sun and X. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2011.

[100] P. Drews, E. Nascimento, F. Moraes, S. Botelho and M. Campos, "Transmission estimation in underwater single images," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 825–830.

[101] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[102] K. Konolige, "Small vision systems: Hardware and implementation," in *Robotics Research*, Springer, 1998, pp. 203–212.

[103] T. Kanade, A. Yoshida, K. Oda, H. Kano and M. Tanaka, "A stereo machine for video-rate dense depth mapping and its new applications," in *1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 1996, pp. 196–202.

[104] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *8th IEEE International Conference on Computer Vision*, IEEE, vol. 2, 2001, pp. 508–515.

[105] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.

[106] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 2, 2005, pp. 807–814.

[107] Y. Swirski, Y. Y. Schechner, B. Herzberg and S. Negahdaripour, "CauStereo: Range from light in nature," *Applied Optics*, vol. 50, no. 28, F89–F101, 2011.

[108] S. Negahdaripour and A. Sarafraz, "Improved stereo matching in scattering media by incorporating a backscatter cue," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5743–5755, 2014.

[109] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[110] Y. Furukawa, B. Curless, S. M. Seitz and R. Szeliski, "Towards internet-scale multi-view stereo," in *2010 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 1434–1441.

[111] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," in *ACM Transactions on Graphics*, ACM, vol. 32, ACM, 2013, p. 29.

[112] D. Levin, "The approximation power of moving least-squares," *Mathematics of Computation*, vol. 67, no. 224, pp. 1517–1531, 1998.

[113] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *ACM SIGGRAPH*, ACM, vol. 21, 1987, pp. 163–169.

[114] C. Beall, B. J. Lawrence, V. Ila and F. Dellaert, "3D reconstruction of underwater structures," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 4418–4423.

[115] J. Henderson, O. Pizarro, M. Johnson-Roberson and I. Mahon, "Mapping submerged archaeological sites using stereo-vision photogrammetry," *International Journal of Nautical Archaeology*, vol. 42, no. 2, pp. 243–256, 2013.

[116] M. Johnson-Roberson, O. Pizarro, S. B. Williams and I. Mahon, "Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys," *Journal of Field Robotics*, vol. 27, no. 1, pp. 21–51, 2010.

[117] A. Sedlazeck, K. Koser and R. Koch, "3D reconstruction based on underwater video from ROV Kiel 6000 considering underwater imaging conditions," in *OCEANS 2009*, IEEE, 2009, pp. 1–10.

[118] Code Laboratories Inc. (2018). DUO M - configurable ultra-compact stereo camera, [Online]. Available: `https://duo3d.com/product/duo-mini-lv1` (visited on 03/01/2019).

[119] M. J. Powell, "A new algorithm for unconstrained optimization," *Nonlinear Programming*, pp. 31–65, 1970.

[120] K. Madsen, H. B. Nielsen and O. Tingleff, *Methods for non-linear least squares problems*, 2004.

[121] M. Smith, I. Baldwin, W. Churchill, R. Paul and P. Newman, "The New College vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, 2009.

[122] X.-S. Gao, X.-R. Hou, J. Tang and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.