Generation and Visualization of Large Scale 3D Reconstructions from Underwater Robotic Surveys

Matthew Johnson-Roberson, Oscar Pizarro, Stefan B. Williams, Ian Mahon
Australian Centre for Field Robotics
University of Sydney
Sydney, NSW 2006 Australia
m.roberson@acfr.usyd.edu.au

Abstract

Robust, scalable Simultaneous Localization and Mapping (SLAM) algorithms support the successful deployment of robots in real-world applications. In many cases these platforms deliver vast amounts of sensor data from large scale, unstructured environments. This data may be difficult to interpret by end-users without further processing and suitable visualization tools. This paper presents a robust, automated system for large-scale 3D reconstruction and visualization that takes stereo imagery from an Autonomous Underwater Vehicle (AUV) and SLAM-based vehicle poses to deliver detailed 3D models of the seafloor in the form of textured polygonal meshes. Our system must cope with thousands of images, lighting conditions that create visual seams when texturing, and possible inconsistencies between stereo meshes arising from errors in calibration, triangulation and navigation. Our approach breaks down the problem into manageable stages by first estimating local structure then combining these estimates to recover a composite geo-referenced structure using SLAM-based vehicle pose estimates. A texture mapped surface at multiple scales is then generated that is interactively presented to the user through a visualization engine. We adapt established solutions where possible, with an emphasis on quickly delivering approximate yet visually consistent reconstructions on standard computing hardware. This allows scientists on a research cruise to use our system to design follow-up deployments of the AUV and complementary instruments. To date, this system has been tested on several research cruises in Australian waters, and used to reliably generate and visualize reconstructions for over sixty dives covering diverse habitats and representing hundreds of linear kilometers of survey.

1 Introduction

As robotic platforms are successfully deployed in scientific (Bajracharya et al., 2008; German et al., 2008), industrial (Durrant-Whyte, 1996; Thrun et al., 2004), defense (Kim &

Sukkarieh, 2004), and transportation (Thrun et al., 2006) applications, the ability to visualize and interpret the large amounts of data they can collect has become a pressing problem. High resolution imaging of the seafloor using robotic systems presents a prime example of this issue. Optical imaging by robots has been used extensively to study hydrothermal vents (Kelley et al., 2005; Yoerger et al., 2007), document ancient and modern wrecks (Ballard et al., 2000; Howland, 1999), characterize benthic habitats (Armstrong et al., 2006; Singh, Eustice, et al., 2004; Webster et al., 2008) and inspect underwater man-made structures (Walter et al., 2008). Optical imagery is rich in detail and is easily interpretable by scientists. However, it is often difficult to acquire high quality geo-referenced imagery underwater, given that water strongly attenuates electromagnetic waves (including light and RF signals) (Duntley, 1963), which forces imaging close to the seafloor and precludes the use of high-bandwith communications and GPS-based positioning. Autonomous Underwater Vehicles (AUVs) can address the requirements for near-bottom high resolution imaging in a cost-effective manner. These robotic platforms closely follow rugged seafloor features to acquire well-illuminated imagery over controlled track lines covering hundreds to thousands of linear meters. Operating untethered and away from the surface also minimizes wave-induced motions, resulting in a steady, responsive sensor platform.

While having thousands of geo-referenced images of a site is useful, being able to easily visualize and interact with the imagery and associated structure at scales both larger and smaller than a single image can provide scientists with a powerful data exploration tool, potentially allowing them to observe patterns at scales much larger than that covered by a single image. Such a tool allows users to quickly build an intuitive understanding of the spatial relationships between substrates, morphology, benthos and depth. This might then be used to test hypotheses related to the distribution of benthic habitats that could inform further surveys and sampling.

Large-scale visualization underwater requires the creation of composite views through 2D or 3D reconstructions. Approaches for 2D mosaicking (Sawhney & Kumar, 1999; Sawhney et al., 1998) are significantly simpler than 3D approaches and are easy to visualize at multiple scales but can produce strong distortions in the presence of 3D relief. In terms of large scale underwater reconstructions, most mosaicking has been motivated largely by vision-based navigation and station keeping close to the sea-floor (Fleischer et al., 1996; Gracias & Santos-Victor, 2001; Negahdaripour et al., 1999; Negahdaripour & Xun, 2002). Additionally 2D mosaics with stereo compensation has been explored (Negahdaripour & Firoozfam, 2006). Large-area mosaicking with low overlap under the assumption of planarity is addressed by Pizarro and Singh (2003).

Since AUVs can operate in very rugged terrain we argue that a sounder approach is to account for 3D structure. In fact, AUV surveys are typically undertaken in environments that feature complex structure, such as reefs, canyons, and trenches, where a 2D seafloor model is not appropriate. The machinery to convert optical imagery into 3D representations of the environment has been studied extensively (Fitzgibbon & Zisserman, 1998; Hartley & Zisserman, 2000), including systems that operate reliably for large scale environments (Pollefeys et al., 2000). Some promising work has gone into 3D image reconstruction underwater (Negahdaripour & Madjidi, 2003) using a stereo-rig with high overlap imagery in a controlled environment or single moving cameras (Nicosevici & Garcia, 2008; Pizarro et al., 2004). Underwater stereo 3D reconstruction is shown by Jenkin et al. (2008) and Saez et al. (2006) on high frame-rate dense stereo imagery using SLAM and energy minimization to produce consistent 3D maps, but without explicitly addressing the fast reconstruction and visualization of thousands of images.

Most end-to-end systems for visualisating data collected by robotic systems have focused on reconstructing 3D models of urban environments (Fruh & Zakhor, 2004; Hu et al., 2003). The abundance of man-made structures supports strong priors on structure that result in simple or fast algorithms. One recent, state of the art system uses video rate imagery and a multi-view dense stereo solution with poses derived from high end navigation instruments (Pollefeys et al., 2008). The mesh fusion stage addresses minor inconsistencies and the implicit assumption is that the quality of the local data and navigation are sufficient for modelling purposes.

While there has been much work on outdoor vision-based Simultaneous Localization and Mapping (SLAM) (Agrawal et al., 2007; Lemaire et al., 2007; Ho & Jarvis, 2007; Steder et al., 2007), interactive visualization capabilities tend to be limited or non-existent, with results being used to validate reconstruction methods rather than to explore and understand the reconstructions. For unstructured scenes Shlyakhter presents some impressive results of 3D tree reconstruction but these involved human input and operate at relatively small scales (Shlyakhter et al., 2001).

In this paper we present a robust, automated system for large-scale, 3D reconstruction and visualization that combines stereo imagery with self-consistent vehicle poses to deliver dense 3D, texture mapped terrain reconstructions. This work takes advantage of recent advances in visual SLAM techniques proposed by Eustice et al. (2006) and extended by Mahon et al. (2008) that generate consistent estimates of the pose of an AUV during benthic survey missions. The novelty of this work arises from our capacity to process and render tens of thousands of images with sufficient speed to allow end user interaction with the reconstruction in time to inform further data gathering missions. Our sytem is geared towards delivering fast, approximate reconstructions that can be used during a research cruise and examples illustrating the utility of the reconstructions for deployment planning are discussed. Because of the system's focus on delivering timely results, we also examine robustness issues and several instances requiring tradeoffs between performance, accuracy, and the complexity of the reconstructed geometry.

The processing pipeline for our system can be broken down into the following main steps as shown in Figure 1:

1. **Data Acquisition and Preprocessing.** The stereo imagery is acquired by the AUV. The primary purpose of this step is to partially compensate for lighting and wavelength-dependent color absorption. This allows improved feature extraction and matching during the next stage.

2. **Stereo Depth Estimation.** Extracts 2D feature points from each image pair, robustly proposes correspondences and determines their 3D position by triangulation. The local 3D point clouds are converted into individual Delaunay triangulated meshes.

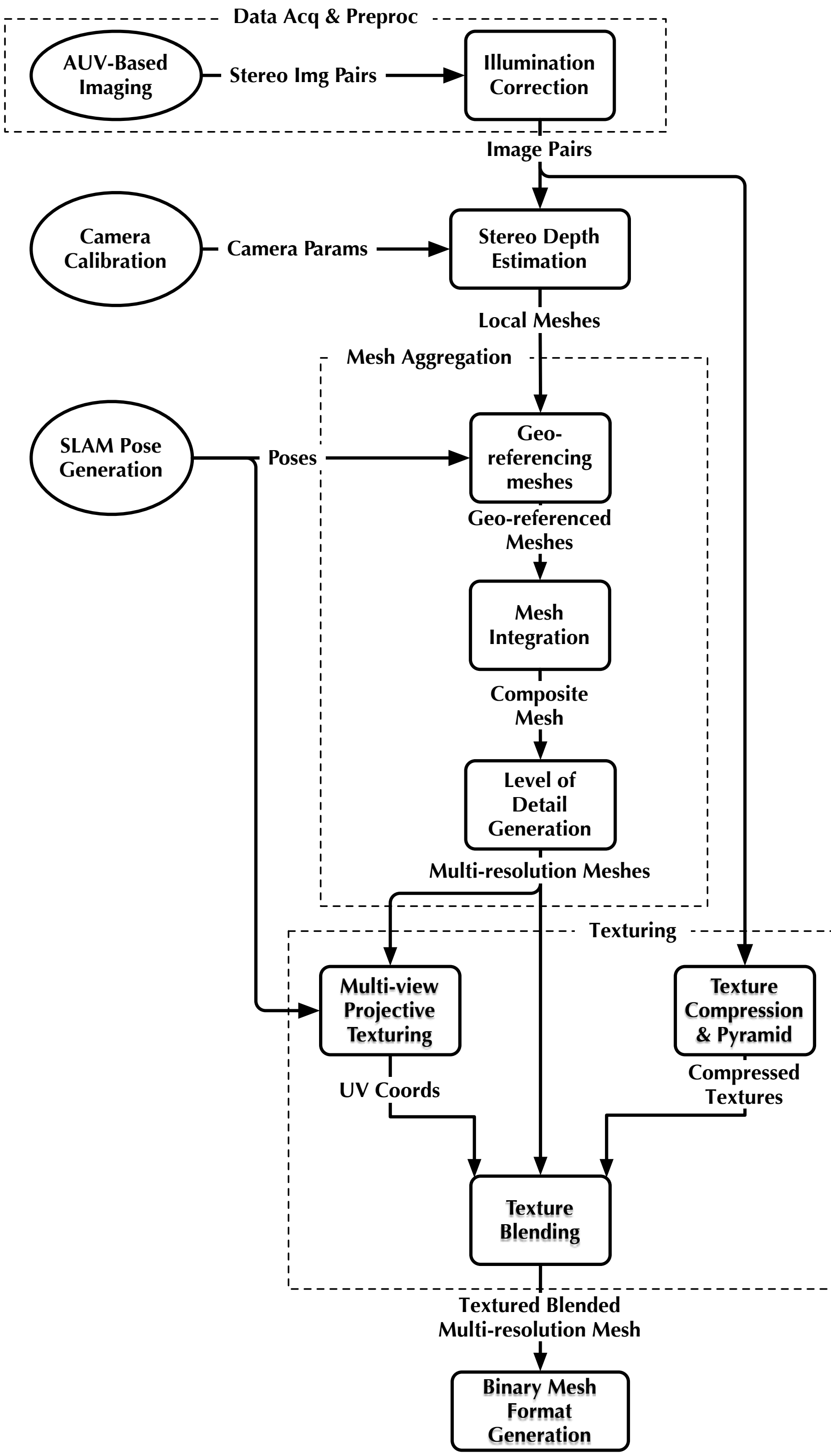


Figure 1: Processing modules and data flow for the reconstruction and visualization pipeline

3. **Mesh Aggregation.** Places the individual stereo meshes into a common reference frame using SLAM-based poses and then fuses them into a single mesh using Volumetric Range Image Processing (VRIP) (Curless & Levoy, 1996). The total bounding volume is partitioned so that standard volumetric mesh integration techniques operate over multiple smaller problems while minimizing discontinuities between integrated meshes. This stage also produces simplified versions of the mesh to allow for fast visualization at broad scales.

4. **Texturing.** The polygons of the complete mesh are assigned textures based on the overlapping imagery that projects onto them. Lighting and misregistration artifacts are reduced by separating images into spatial frequency bands that are mixed over greater extents for lower frequencies (Burt & Adelson, 1983).

The remainder of this paper is structured around the main steps in this processing pipeline and is organized as follows. Section 2 presents the AUV platform and preprocessing that enable the acquisition of geo-referrenced stereo imagery. Section 3 presents our approach to generating local structure while Section 4 describes how these local representations are merged into one consistent and readily viewable mesh. Section 5 details the application of visually consistent textures to the global mesh. Section 6 describes the practical considerations that enable the system to operate on the very large volumes of data collected by the vehicle. Section 7 illustrates the effectiveness of the system using data collected on a number of research cruises around Australia. Finally, Section 8 provides conclusions and discusses on-going work.

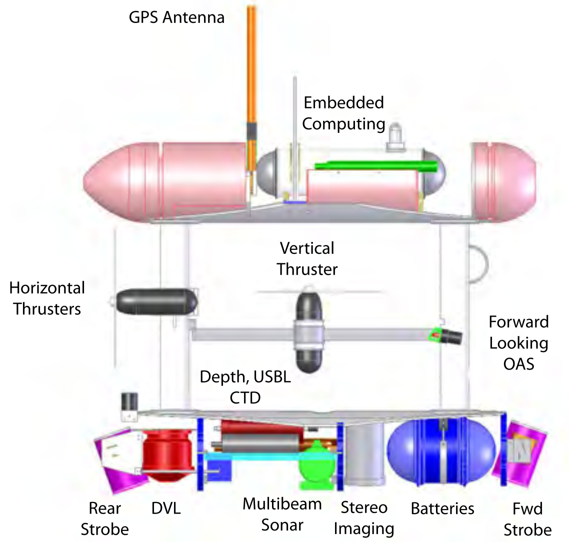
2 Data Acquisition and Preprocessing

2.1 AUV-based Imaging

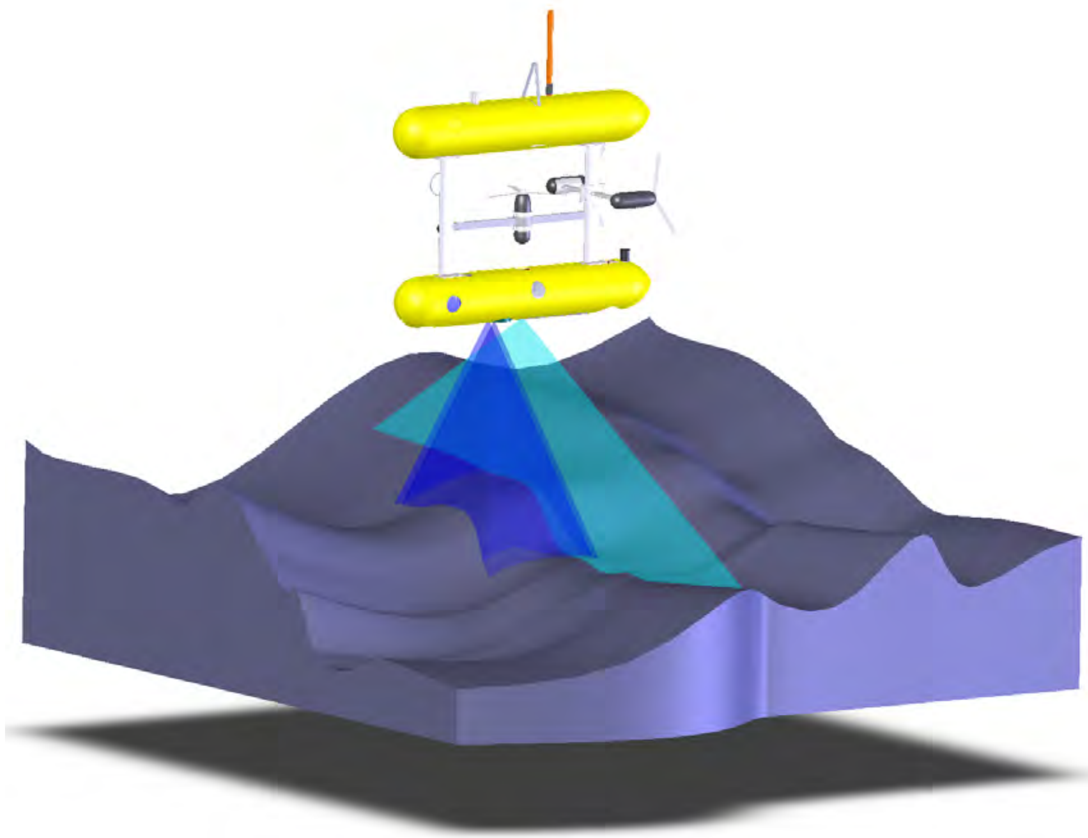
The University of Sydney's Australian Centre for Field Robotics (ACFR) operates an ocean going AUV called *Sirius* capable of undertaking high resolution, geo-referenced survey work. *Sirius* is part of the Integrated Marine Observing System (IMOS) AUV Facility, with funding available on a competitive basis to support its deployment as part of marine studies in Australia. *Sirius* is a modified version of the autonomous underwater vehicle *SeaBED* built at the Woods Hole Oceanographic Institution (Singh, Can, et al., 2004). This class of AUV is designed specifically for near-bottom, high resolution imaging and is passively stable in pitch and roll. In addition to a stereo camera pair and a multibeam sonar, the submersible is equipped with a full suite of oceanographic sensors (see Figure 2 and Table 1). The two 1360×1024 cameras are configured as down-looking pair with a baseline of approximately 7 cm and $42^\circ \times 34^\circ$ field of view, while the down-looking multibeam returns can be beamformed to 480 beams in a 120° fan across track.

The AUV is typically programmed to maintain an altitude of 2m above the seabed while traveling at 0.5 m/s (1 knot approx.) during surveys. Missions last up to five hours with 2 Hz stereo imagery and 5-10 Hz multibeam data, resulting in approximately 40 GB/hr of raw imagery, sonar data and navigation data.

(a)



(b)



(c)

Figure 2: (a) The AUV *Sirius* being retrieved after a mission aboard the R/V Southern Surveyor (b) layout of internal components and (c) the imaging configuration of the stereo cameras 42° x 34° Field of View (FOV) (depicted in dark blue) and multibeam 120° x 0.75° FOV (depicted in teal).

Table 1: Summary of the *Sirius* AUV specifications.

Vehicle	
Depth rating	800 meters
Size	2.0 m (L) $\times$ 1.5 m (H) $\times$ 1.5 m (W)
Mass	200 kg
Maximum Speed	1.0 m/s
Batteries	1.5 kWh Li-ion pack
Propulsion	3 x 150 W brushless DC thrusters
Navigation	
Attitude+Heading	Tilt $\pm 0.5°$, Compass $\pm 2°$
Depth	Digiquartz pressure sensor, 0.01%
Velocity	RDI 1200 kHz Navigator DVL $\pm 2mm/s$
Altitude	RDI Navigator four beam average
USBL	TrackLink 1500 HA (0.2m range, 0.25°)
GPS receiver	uBlox TIM-4S
Optical Imaging	
Camera	Prosilica 12bit 1360$\times$1024 CCD stereo pair
Lighting	2 x 4 J strobe
Separation	0.75m between camera and lights
Acoustic Imaging	
Multibeam sonar	Imagenex DeltaT 260 kHz
Obstacle avoidance	Imagenex 852 675 kHz
Tracking and comms	
Radio	Freewave RF Modem / Ethernet
Acoustic Modem	Linkquest 1500 HA Integrated Modem
Other Sensors	
Conductivity and Temperature (CT)	Seabird 37SBI
Chlorophyll-A, CDOM and Turbidity	Wetlabs Triplet Ecopuck

The vehicle navigates using the Doppler Velocity Log (DVL) measurements of both velocity and altitude relative to the sea-floor. Absolute orientation is measured using a magneto-inductive compass and inclinometers, while depth is obtained from a pressure sensor. Absolute position information from a GPS receiver is fused into the position estimate when on the surface. Acoustic observations of the range and bearing from the ship are provided by an Ultra Short Baseline (USBL) tracking system which includes an integrated acoustic modem. USBL observations are communicated to the vehicle over the acoustic link and the vehicle returns a short status message, including battery charge, estimated position and mission progress, so its performance can be monitored while it is underway.

2.2 Illumination Correction

Range and wavelength dependent attenuation of light through water implies that the appearance of a scene point will have a strong dependence on the range to the light source(s) and camera. For example, underwater imagery typically has darker edges because of stronger attenuation associated with the viewing angle and longer path lengths (Jaffe, 1990). An image patch being tracked on a moving camera will therefore violate the brightness constancy constraint (BCC) that underlies many standard image matching algorithms.

Lighting correction for underwater imagery has received some attention as a way of improving the general appearance of imagery or to aid in establishing correspondences between images (Garcia et al., 2002). The simplest approaches increase contrast by stretching the histogram of intensities. This can offer some visual improvement over individual images but can result in significant changes in mean over a sequence of images. In the case of non-uniform lighting, the resulting histogram may already be broad and stretching the whole image histogram may fail to adequately correct for illumination artefacts. Adaptive histogram equalization operates over subregions of the image and can be used to account to some extent for variation of illumination across an image (Zuiderveld, 1994). Homomorphic processing variants decompose images into a low frequency component assumed to be related to the lighting pattern and invert that field before reassembling the image (Singh et al., 2007). These techniques do not, however, enforce consistency across an ensemble of images which would lead to seams in the texture maps used in our 3D reconstructions.

We have addressed the illumination issue in two ways: optimizing camera strobe configuration and performing post processing. In the current configuration the vehicle is programmed to maintain a constant altitude above the seafloor with the cameras pointed downwards. The vehicle carries a pair of strobes separated along the length of the frame that are synchronised with the image capture. The fore-aft arrangement of strobes partially cancels shadowing effects while reducing the impact of backscatter in the water column between the cameras and the seafloor. In post processing, we construct an approximate model of the resulting lighting pattern by calculating the mean and variance for each pixel position and channel over a representative sample of images. A gain and offset for each pixel position and channel is then calculated to transform the distribution associated with that position and channel to a target distribution with high variance (i.e., contrast) and mid-range mean. This is a form of the 'gray world' assumption (Barnard et al., 2002), where each pixel position and channel is treated independently and the samples of the world are acquired over many images. More sophisticated versions of this approach could identify multimodal distributions and correct them accordingly. However, we have found this straightforward approach to be sufficient for improving the feature matching process and the consistency of illumination of the resulting images for most situations. An example of a set of sample images and associated lighting pattern can be seen in Figure 3.

In addition to improving the visual quality of the images from a user perspective, applying this normalization yields significant improvements in the reliability of feature matching. Feature extraction and description can be made robust or even invariant to some changes in lighting (Burghouts & Geusebroek, 2009). To illustrate the effect of lighting compensation we apply the stereo matching algorithm described in Section 3 to two pairs of images, as shown in Figure **??**. The first has no lighting correction applied to the images prior to matching of stereo features while the second has had the proposed lighting correction algorithm applied. The results are displayed in Figures 4(c) and 4(d). As can be seen, feature matching performs significantly better when the illumination is corrected for, particularly in the dark corners where the contrast is poor.

3 Stereo Depth Estimation

There is a large body of work dedicated to two-view and multi-view stereo (Scharstein & Szeliski, 2002; Seitz et al., 2006) but dense stereo results tend to be too complex for our application and for limited overlap can produce incorrect surfaces. Other approaches have examined the use of Structure from Motion (SFM) (Tomasi & Kanade, 1992; Hartley &

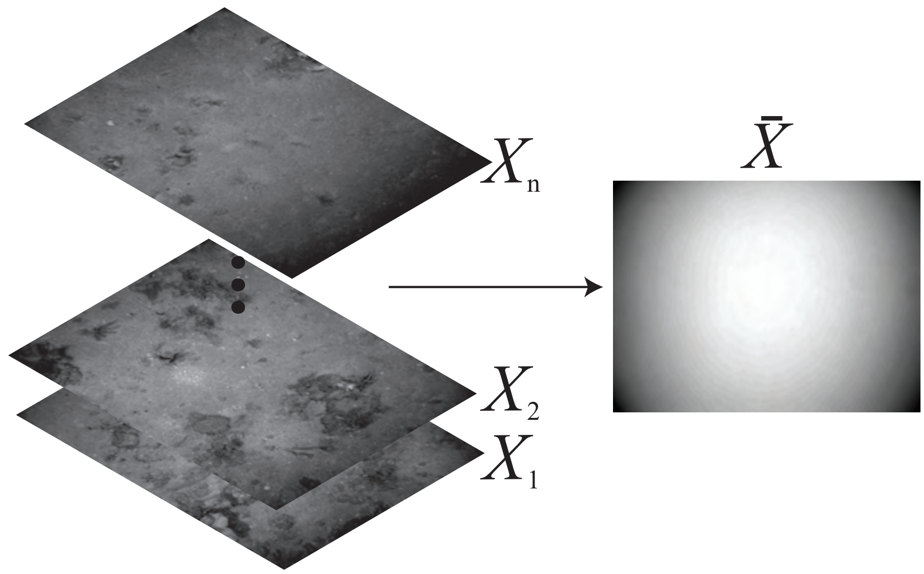


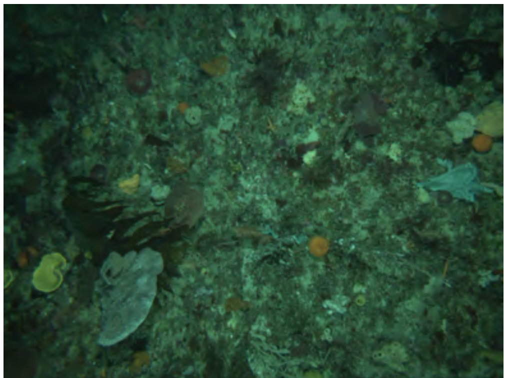
Figure 3: Illustration of a stack of over 5,000 grayscale images from a mission averaged across each pixel creating the mean lighting pattern image on the right. (Note the contrast has been stretched slightly to enhance viewing)

Zisserman, 2000) to recover scene structure, utilizing the location, matching, and tracking of feature points over sequences of images to recover the 3D structure of the underlying scene as well as the associated camera poses. SFM's simple hardware requirements make it popular but scale is lost if a single camera is used. It is difficult to build a robust system based solely on monocular SFM since it is sensitive to configurations of motions and surfaces that can not be solved uniquely. SFM modified to rely on resection (Nicosevici & Garcia, 2008) or navigation instruments (Pizarro et al., 2004) has been applied successfully in an underwater context.

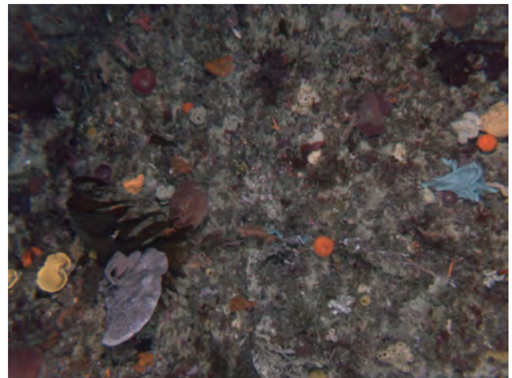
3.1 Sparse Feature Based Stereo

For computational reasons we require a simple representation that captures the coarse structure of the scene. One approach would be to use a dense stereo algorithm and then simplify the resulting depth map using a mesh simplification technique such as quadric error metric simplification (Garland & Heckbert, 1998) that preserves detail at the expense of extra computations. Instead, we have chosen to extract a sparse set of 2D features, robustly triangulate their positions, and then fit a mesh to the 3D points using Delaunay triangulation (Hartley & Zisserman, 2000). By focusing on a sparse set of well localized points we expect to minimize gross errors down the pipeline while keeping computational demands low.

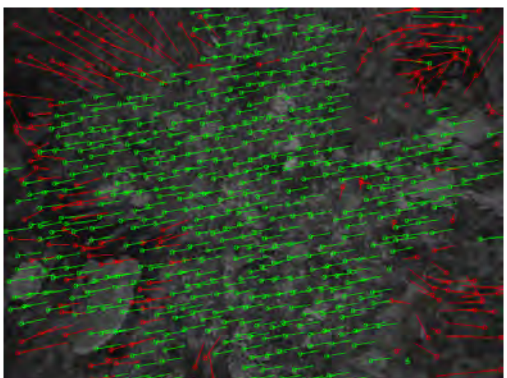
The choice of feature in a correspondence based stereo method is heavily dependent on the camera geometry. The cameras on the AUV are in a small baseline configuration, resulting in negligible change in scale of corresponding features, and close to complete overlap between



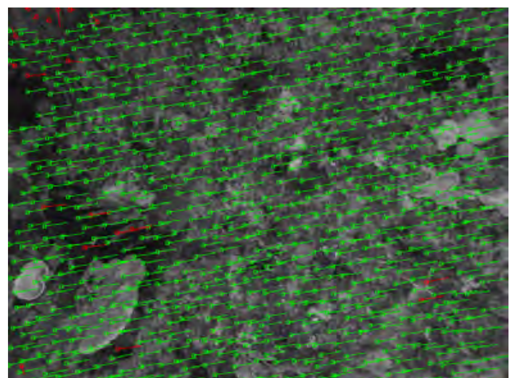
(a) Image w/o illumination compensation



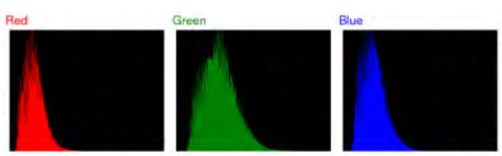
(b) Image with illumination compensation



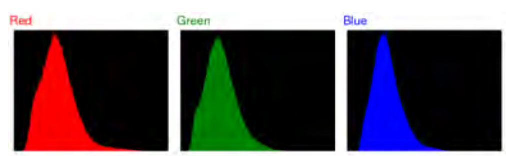
(c) Feature matches w/o illumination compensation



(d) Feature matches with illumination correction



(e) Image Histogram w/o illum. comp.



(f) Image Histogram with illum. comp.

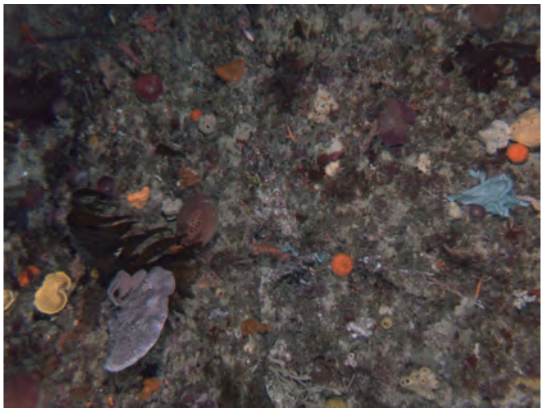
Figure 4: (a) Uncorrected image illustrating the lighting pattern induced by the strobes, with darker corners and a bright central region. Stronger attenuation of the red channel also causes the image to appear green (we have applied a constant gain to the three channels to brighten the overall image for easier viewing while preserving the relationship between channels and the lighting falloff towards the edges) (b) the lighting is considerably more consistent in the compensated image (c) Left image from stereo pair without lighting correction showing matched Harris corners as circles, green are valid, red have been rejected based upon epipolar geometry (d) image feature matches with lighting compensation applied. Note the increased number of matches especially in the corners. A wider range of color, in particular in the red channel, can be seen in the corrected image on the right. The histograms of the red, green, and blue channels of the corrected and uncorrected images appear in (e) and (f) respectively.

left and right frames. This means that the majority of pixels in one image frame should have matches in the corresponding view except where portions of the surface are occluded. The downside of the small baseline configuration is an increased uncertainty in depth. An overview of the feature matching process is as follows:

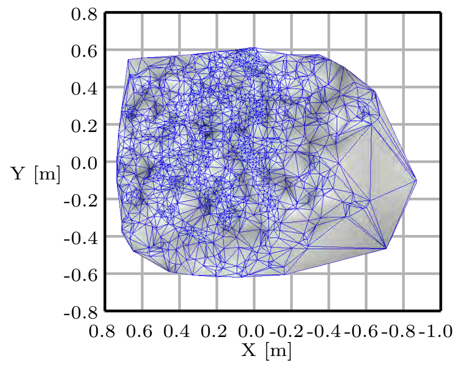
1. Feature points are extracted from the left-side source imagery using a Harris Corner detector (Harris & Stephens, 1988).
2. Correspondences are proposed using a Lucas-Kanade tracker (Lucas & Kanade, 1981) seeded into right-side images by intersecting the associated epipolar line with a plane at the altitude given by a sonar altimeter (the DVL).
3. Proposed matches that are not consistent with the epipolar geometry derived from stereo calibration, i.e. outliers, are then rejected.
4. Remaining feature points can then be triangulated using the midpoint method (Hartley & Zisserman, 2000).

An example of Harris corners that have been matched from the left to right frame of a sample stereo pair can be seen in Figures 4(c) and 4(d). The red points correspond to rejected associations based upon the constraints of epipolar geometry while the green points represent features that have been successfully triangulated. This example illustrates the distribution of features typically recovered with our imagery. In most cases it it possible to recover on the order of two to three thousand triangulated points per image pair with the camera geometry and distance to the seabed used by our system.

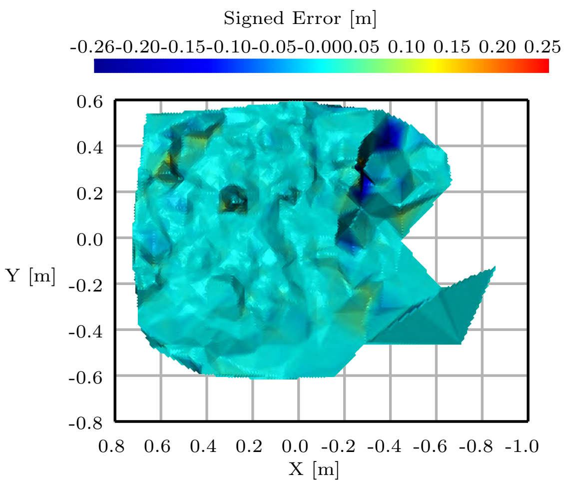
The density of features extracted is a crucial consideration in feature based stereo. Too many features will result in complex meshes with large memory requirements particularly when dealing with thousands of other images, while too few features will result in loss of detail in the relief and more ghosting when reprojecting images onto the over-simplified mesh. Figure 5 illustrates the change in the quality of the scene reconstruction when using varying numbers of features. As the number of features decreases, the model is less able to capture variation in the terrain and differences in the estimated depth of the scene points become more pronounced, particularly around areas of relatively high relief. A set of 100 randomly selected images were triangulated using between 200 and 1700 features relative to a benchmark reconstruction using 2000 points. The objective was to determine if the effect of the sparsity of points is apparent across a number of sample images. As shown in Figure 6, there is an approximately linear relationship between number of features and error induced. Additionally there is a linear relationship in computation time which increases with number of features. In practice 800 feature strikes a balance between computation time and the quality of the output mesh that has been sufficient in our experience. Alternatively, if there were less emphasis on the computation time of the stereo calculation we could extract dense stereo meshes and simplify them to the desired level of detail in a similar fashion to that described in Section 4.3. Instead of dense stereo, a large number of features could be triangulated and then simplified to the desired level, again at a higher computational cost than directly selecting the desired number of features to triangulate. We are currently investigating these approaches to characterize their performance.



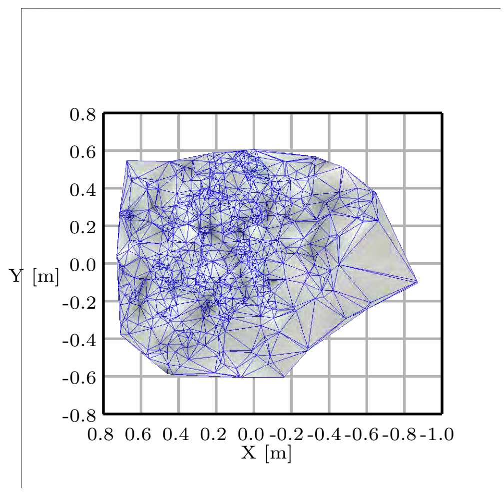
(a) Left color image



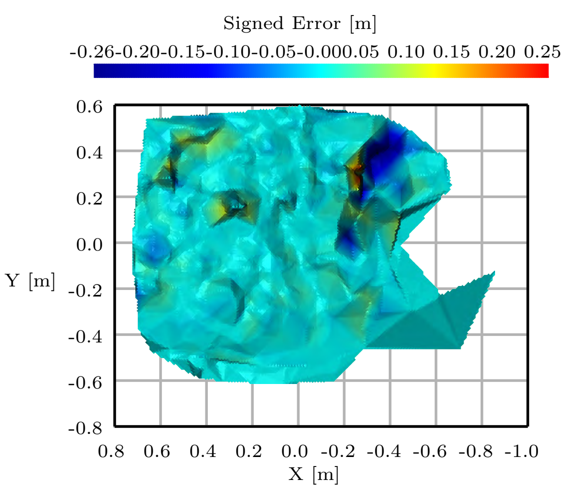
(b) 2000 feature mesh



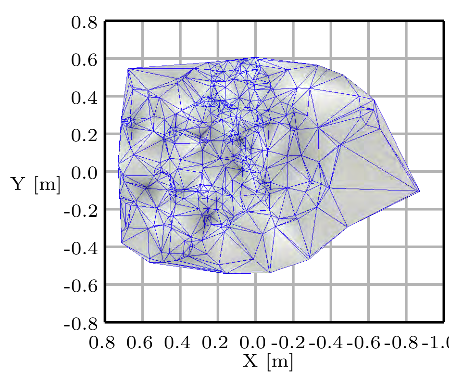
(c) ΔZ 2000 feature mesh vs. 1000 feature mesh



(d) 1000 feature mesh



(e) ΔZ 2000 feature mesh vs. 500 feature mesh



(f) 500 feature mesh

Figure 5: An example of a 3D mesh and the effect of extracting a decreasing number of features on the surface geometry. As the mesh complexity decreases, greater differences in height are seen as the mesh is no longer able to model variability in terrain.

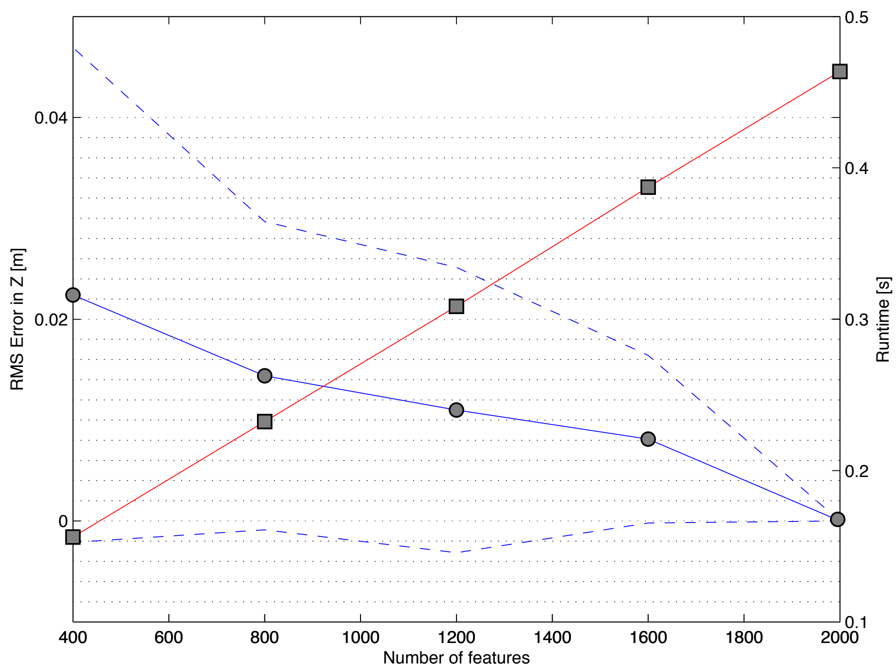
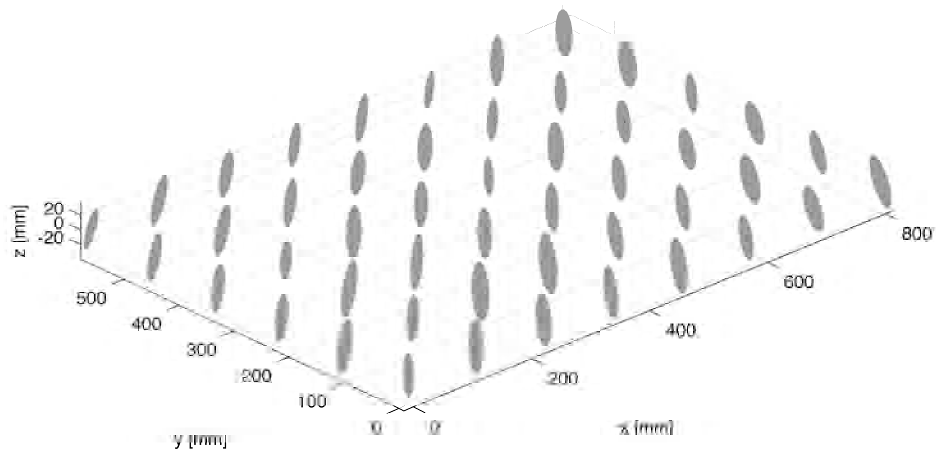


Figure 6: RMS Error in reconstructed meshes vs. the number of extracted features. The results are generated using 100 different feature based meshes. A 2000 feature mesh is used as a reference in each case and errors in height are calculated for meshes with between 400 and 2000 vertices. The solid blue line (circles) represents the mean error in height and the dotted lines are one sigma above and below the mean. Additionally the time required to extract and triangulate the features is plotted in red (squares) on the same graph. The relationship is near linear between number of features and error. It also appears linear between number of features and computation time. Based upon both these observations the user may safely tune this number depending on time requirements, but we use 800 as a compromise between quality and computational cost for constructing the local stereo meshes.

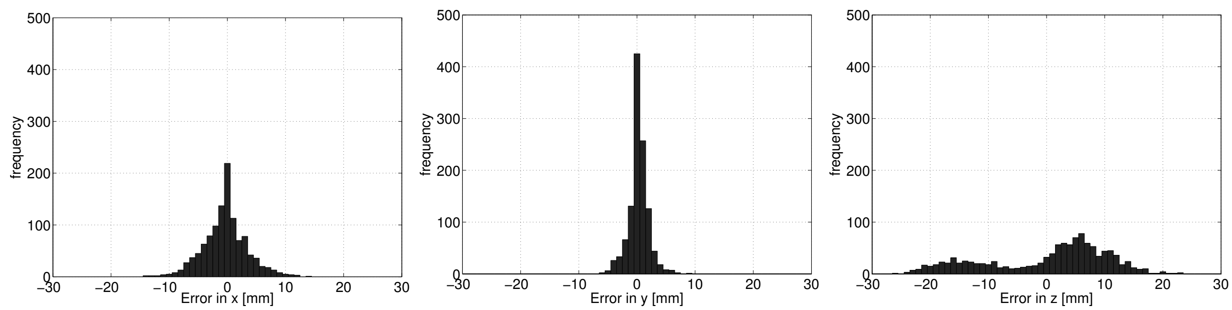
3.2 Reconstruction Accuracy

The accuracy of the stereo camera triangulation is difficult to determine for general underwater imagery as ground truth is not available for the natural underwater scenes we image. We present the results for the estimation of the corners of the checkerboard target used to calibrate our stereo rig. This calibration is undertaken in a pool prior to deployment of the vehicle. Figure 7 shows that the system is successfully able to estimate the positions of the corners on our calibration target, with a maximum error in the z position on the order of 2 cm. While these results are generated using an ideal corner feature, it suggests that the calibration of our camera is of reasonable quality for the purposes of imaging the seafloor.

A complementary approach that would be easily applicable in the field would be to use multibeam sonar data to assist in the calibration or validation of the stereo system. We performed a preliminary comparison between the 3D surface generated from the stereo imagery and the multibeam sonar. While the results are in general agreement, we are still characterizing the performance of our sonar and as such we can not draw strong conclusions from the



(a)



(b)

Figure 7: (a) Triangulation of calibration board corners from 24 views. Camera intrinsic and extrinsic parameters were estimated. Ellipses represent the three-sigma covariance of the triangulations. The blue points are the actual position of the corners. (b) Histograms of triangulation errors in x, y, z for all 48 corners in 24 stereo views. Consistent with a narrow baseline configuration, the highest variability is along z (depth away from the camera). The baseline is approximately parallel to the x-axis, resulting in triangulation with higher uncertainty in x (along epipolar lines) than y.

comparison. Some of the issues we are addressing are the calibration of the camera-sonar offsets, the consistency of the sonar beamforming and outlier rejection.

4 Mesh Generation

The mesh generation stage transforms the individual stereo meshes into a common reference frame and integrates them into a single approximate mesh. This is a necessary step in the process of generating a model because the separate stereo meshes can have errors in the estimated structure and in geo-referencing. There is also redundant data in the overlapping meshes that may fill holes resulting from occlusion or poor viewing geometry. It is therefore desirable to integrate several aligned meshes into a single global model. This stage is also responsible for generating multiple decimated meshes to allow for use in a level-of-detail system.

4.1 Geo-referencing stereo meshes

An estimate of the vehicle's trajectory is required to place all data collected by it into a common reference frame. Navigation underwater is a challenging problem since absolute position observations such as those provided by GPS are not readily available. Acoustic positioning systems (Yoerger et al., 2007) can provide absolute positioning but typically at lower precision and update rates than observations from environmental instruments on-board the AUV (i.e., cameras and sonars). Using a naive approach, the mismatch between navigation and sensor precision results in 'blurred' maps. A more sophisticated approach uses the environment to aid in the navigation process and ensure poses that are consistent with observations of the environment. Simultaneous Localisation and Mapping (SLAM) is the process of concurrently building a map of the environment and using this map to obtain estimates of the location of the vehicle using its on-board sensors. The SLAM problem has seen considerable interest from the mobile robotics community as a tool to enable fully autonomous navigation (Dissanayake et al., 2001; Durrant-whyte & Bailey, 2006). Earlier work at the ACFR demonstrated SLAM machinery in an underwater setting (S. Williams & Mahon, 2004). Work at the Woods Hole Oceanographic Institution has also examined the application of SLAM (Eustice et al., 2006; Roman & Singh, 2007) and SFM (Pizarro et al., 2004) methods to data collected by ROVs and AUVs.

In order to provide self-consistent, geo-referenced reconstructions, the imagery and navigation data acquired by our AUV is processed by an efficient SLAM system to estimate the vehicle state and trajectory (Mahon et al., 2008; Mahon, 2008). Our approach extends the Visual Augmented Navigation (VAN) methods proposed by Eustice et al. (2006). This technique uses an extended information filter (EIF) to estimate the current vehicle state along with a selection of past vehicle poses, typically the poses at the instant a stereo pair was acquired. An appealing property of this technique is that it does not rely explicitly on features to be maintained within the filter framework, sidestepping the issue of deciding which features are likely to be revisited and used for loop closure observations.

The information matrix for a view-based SLAM problem is exactly sparse, resulting in a significant reduction in the computational complexity of maintaining the correlations in the state estimates when compared with dense covariance or information matrices caused by marginalising past vehicle poses. Recovering the state estimates is required in the EIF prediction, observation, and update operations, while state covariances are required for data association or loop-closure hypothesis generation. Efficient state estimate and covariance recovery is performed using a modified Cholesky factorization to maintain a factor of the VAN information matrix (Mahon et al., 2008).

4.1.1 Loop-Closures: Wide Baseline Matching

Visual feature extraction and matching is an expensive process relative to the entire pipeline. Therefore it is important to be able to evaluate if a pair of poses are likely candidates for a loop-closure. This evaluation will be run many times on a large number of candidate poses and therefore must be efficient. A simplified sensor, vehicle, and terrain model is used to asses the likelihood of overlap. The terrain is assumed to be planar, the vehicle's pitch and roll are assumed to be zero, and the field of view of the vehicle is treated as a cone. Using this model the altitude and XY position of the vehicle define the overlap. As the vehicle's position is uncertain in the VAN framework the likelihood of image overlap is calculated by integrating the probability distribution of the 2D separation of the poses in question. This conservative test allows a large number of potential loop closure candidates to be rejected without performing the feature extraction.

Wide-baseline feature extraction and matching is a well studied field and several algorithms robust to changes in scale and rotation are now available. A number of techniques have been proposed to improve the speed of such techniques to make them applicable for real-time systems. Speeded Up Robust Features (SURF) (Bay et al., 2008) is a wavelet-based extension (primarily for speed) of the popular Scale-Invariant Feature Transform (SIFT) algorithm (Lowe, 2004). When used after lighting correction, both SURF and SIFT features have been successfully used to identify loop closure observations in our AUV benthic survey imagery. An example of such a loop closure is presented in Figure 10(c).

Recently there has been research into utilizing Graphics Processing Units (GPUs) to provide additional speed improvements to SURF and SIFT (Cornelis & Gool, 2008; Sinha et al., 2007). We have explored the benefits of GPU-based and multi-threaded feature extraction to further increase the speed of this step in the pipeline. With this selection of tools we can generate a fast SLAM loop-closure system on a variety of platforms. A comparison of the performance of the various systems is beyond the scope of this paper, however all afford speedups over the single-threaded non-GPU solutions. This implies that missions can be re-navigated on the order of tens of minutes making a real-time SLAM system feasible.

4.1.2 Stereo Relative Pose Estimation

Once a loop-closure has been hypothesized, the likelihood that the pairs of stereo images are imaging the same patch of seafloor is evaluated and, if a match is identified, the relative poses from which the images were acquired must be estimated. Wide-baseline feature descriptors allow matches to be proposed (i.e., corresponding features) between loop-closure images but mis-associations arising from visual self-similarity and low contrast are still possible. While most proposed matches are correct, a few incorrect ones can create gross errors in pose estimation if not recognized as outliers (i.e., proposed correspondences that are inconsistent with a motion model or a geometric constraint such as the epipolar geometry). To address this problem we generate relative pose estimates using a robust pipeline to process the stereo pairs. The steps involved in the process are illustrated in Figure 8 and are summarised here, with full details appearing in Mahon (2008).

1. Features are extracted in the images using one of the wide-baseline feature descriptors discussed in Section 4.1.1.
2. Features are matched within each stereo pair constrained by epipolar geometry and the resulting 3D points are triangulated.
3. The features are then associated across the two stereo pairs using wide-baseline descriptors. The majority of outliers or mis-associations can be rejected by applying epipolar constraints between each of the first and second pairs of images.

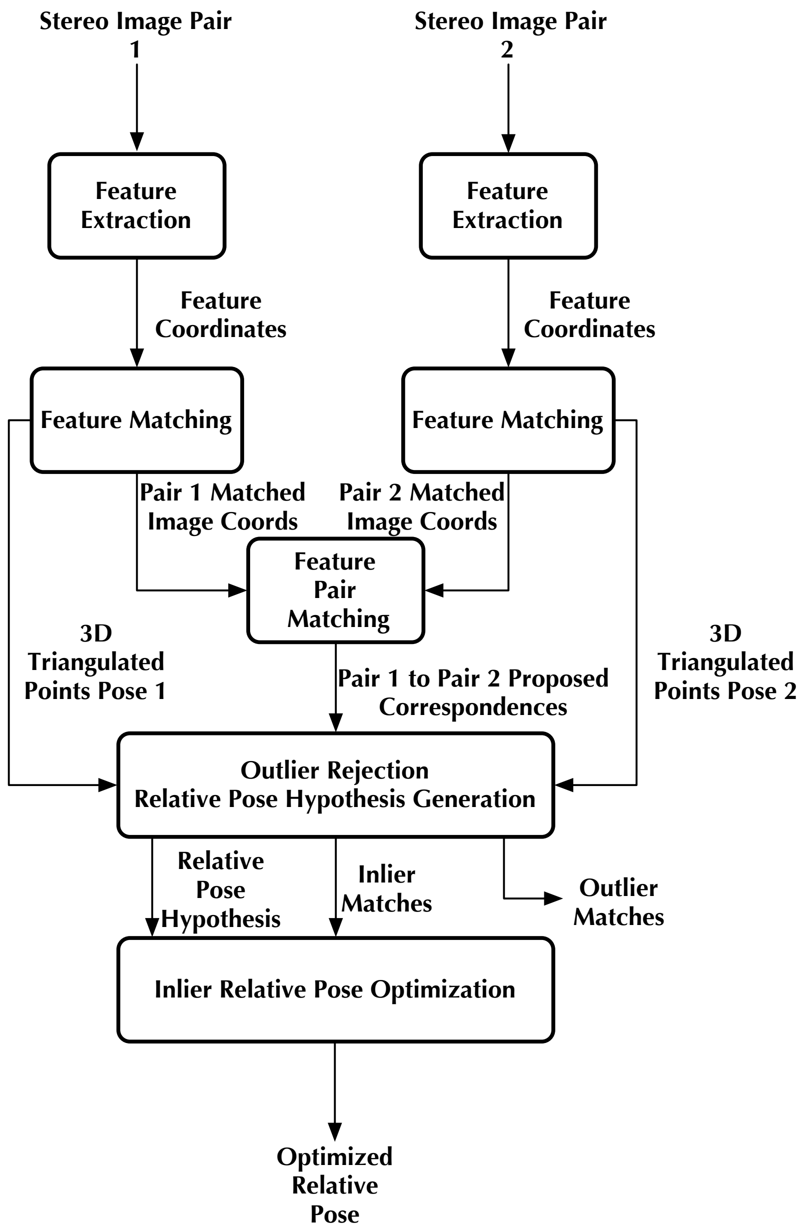


Figure 8: The stereo-vision relative pose estimation process.

4. Remaining outliers are rejected by calculating a robust relative pose estimate using the Cauchy ρ-function (Huber, 1981), then using a Mahalanobis outlier rejection test (Matthies & Shafer, 1987) designed to accept 95% of inliers. The robust estimate is calculated using the random sample initialisation method, where each initial hypothesis is calculated by maximum likelihood 3D registration on a minimal set of 3 randomly selected features.

5. A final relative pose estimate and covariance is produced from the remaining inlier features using maximum likelihood 3D registration initialised at the robust relative pose estimate.

A loop-closure event comprises an observation of the relative pose between the current and a past pose. Given the availability of feature observations from our stereo cameras, we can compute a full 6 DOF relationship between poses rather than a 5DOF constraint (attitude and direction of motion) available from a monocular camera.

4.1.3 Decoupling SLAM and Reconstruction

We have explicitly elected to decouple the SLAM and reconstruction steps in our pipeline. As shown in Figure 1, SLAM is considered as a first step in the process, in which the poses of the vehicle throughout the dive are estimated using the navigation sensors available on the vehicle and the constraints imposed using the matched features in the imagery. The reconstruction phase, described in Section 4.2, uses these poses to project the stereo meshes into space and to then compute a single, aggregate mesh. Any inconsistencies remaining at this point are assumed to be small and are dealt with using the mesh aggregation (Section 4.2) and texture blending (Section 5.1) techniques. While in principle it is possible to formulate a representation of pose, structure and visual appearance such that adjustments performed to the 3D structure and texture could propagate to corrections in pose estimates and calibration parameters, the complexity of such a problem is significantly greater than the one we are addressing in this paper. By decoupling the reconstruction problem from pose estimation (SLAM), stereo estimation, and texturing, the entire pipeline can be made to handle extremely large reconstruction problems, featuring on the order of 10k image pairs. As outlined previously, our goal was to produce a reconstruction system that can process whole missions in a timely manner, yielding approximate reconstructions in less time than they can be acquired to allow future missions to be planned. We believe some tradeoffs were required in order to achieve these goals.

4.1.4 Sample SLAM results

A comparison of the estimated trajectories produced by dead-reckoning and SLAM is shown in Figure 9 for one of the deployments undertaken on the Great Barrier Reef in October 2007. Both filters integrate the DVL velocity (relative to the bottom), attitude and depth observations. The SLAM filter is also capable of incorporating loop closure constraints from the stereo imagery. The dead-reckoning filter is not able to correct for drift which accumulates in the vehicle navigation solution. In contrast, loop closures identified in the imagery allow for this drift to be identified and for the estimated vehicle path to be corrected. This particular deployment comprised a total of over 6500 image pairs and a state vector that includes the six state pose estimates for each image location. Loop-closure observations were applied to the SLAM filter, shown by the red lines joining observed poses. Applying the loop-closure observations results in a trajectory estimate that suggests the vehicle drifted approximately 10 meters north of the desired survey area. Figure 10 illustrates the role of SLAM in providing self-consistent camera poses for 3D model generation.

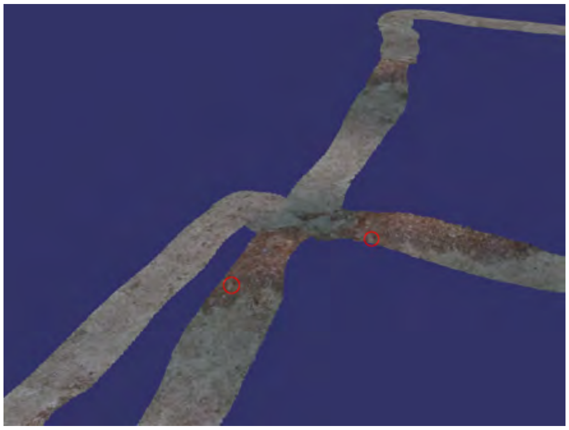
Figure 9: Comparison of dead-reckoning and SLAM vehicle trajectory estimates. The mission begins near 0,0 and ends near 40,60 and covers a distance of approximately 1.5 km. The SLAM trajectory is shown in black with dots marking positions where a stereo pair was acquired, while the dead-reckoning estimates are shown in blue. The SLAM estimates suggest the vehicle has drifted approximately 10 m north of the desired survey area. The red lines connect vehicle poses for which loop closure constraints have been applied. The red circle shows a loop closure area highlighted in Figure 10.

4.2 Mesh Aggregation

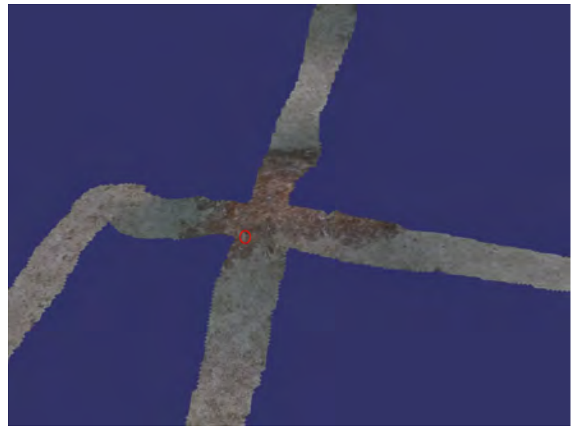
Once the individual stereo meshes have been placed in a common reference frame, they must be combined to create a single mesh from the set of geo-referenced stereo meshes. While care is taken to use self-consistent camera poses and to generate stereo meshes from feature points, errors in the structure and pose of the meshes are still possible. The main issues to address when aggregating meshes are (Campbell & Flynn, 2001) :

Error In the geo-referencing and range estimation leading to inconsistencies in the overlapping meshes.

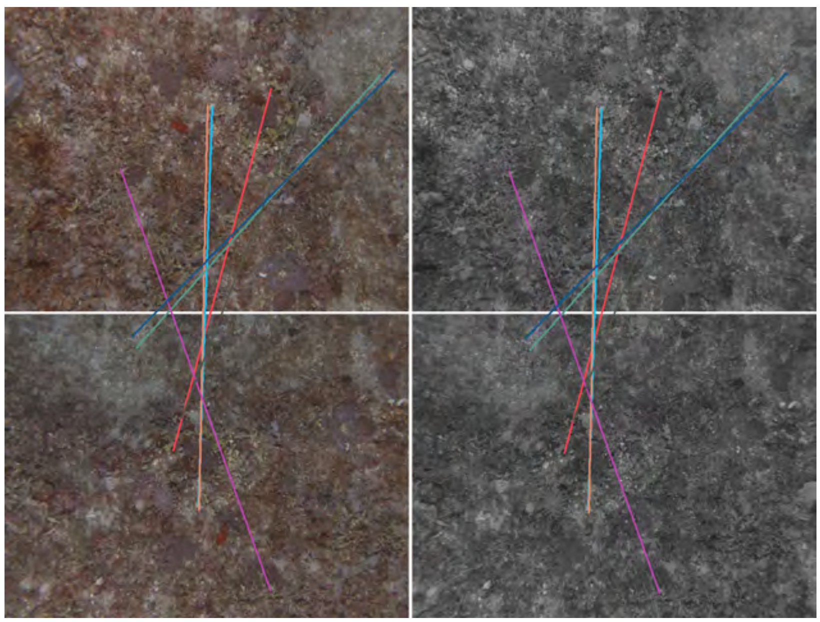
Redundancy The set of stereo meshes includes redundant information depending on the amount of overlap. A technique that removes this allows for more efficient storage and rendering.



(a) Overlap Dead Reckoning poses



(b) Overlap SLAM poses



(c) Loop closure feature matches

Figure 10: Mesh errors induced when local consistency is not enforced using SLAM. The red dots in (a) represent common features that have not been correctly placed in space due to drift in the estimated vehicle pose while (b) shows the same intersections when SLAM has been used to correct for navigation errors. (c) Loop closure feature associations. The first stereo pair is shown on top, with the second stereo pair below. The lines join the positions of associated features between left and right frames of the two pairs. The relative pose estimate based on these features is incorporated into the SLAM filter as an observation that constrains the vehicle's trajectory.

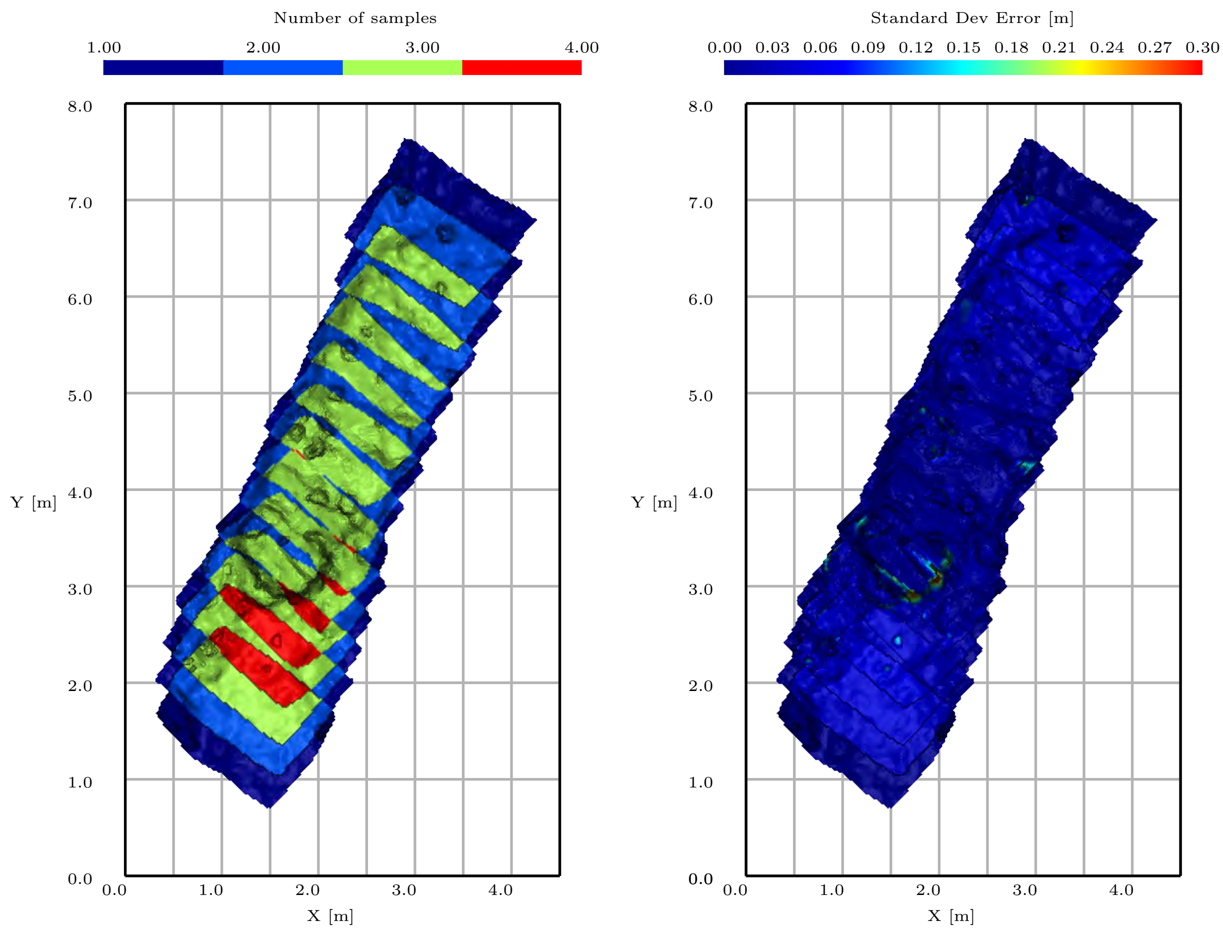
Occlusion/Aperture Sensors with limited sensor aperture are not capable of capturing the entirety of an arbitrary scene. Any particular view of that scene may have occluded sections that result in 'holes' in the associated mesh. Combining multiple views allows these holes to be filled in.

An example of a number of stereo meshes gathered from *Sirius* is illustrated in Figure 11. As can be seen, there are a number of strips of seabed that have been imaged from multiple positions and there is some inconsistency in the estimated height of the sea-floor, particularly around areas of high structure. Merging these multiple estimates of seabed height requires a technique for fusing multiple, noisy observations of the height in a consistent manner.

There are a number of techniques that consider the problem of merging source geometry into a single surface. The most basic of these techniques simply generate a new mesh from all the available source points, resulting in a single mesh. Generating a single interpolated mesh that incorporates all the data may be achieved using Delaunay triangulation (Boissonnat, 1984), Voronoi diagrams (Amenta & Bern, 1999), or DEM greedy insertion (Fowler & Little, 1979), however such approaches create jagged models in response to noise in the data. Other techniques stitch together a set of source meshes, remove overlapping vertices and average out the resulting surface (Turk & Levoy, 1994) but again this tends to be sensitive to inconsistencies and noise. We investigated the use of these techniques but found that they lacked the robustness to the level of noise in our data and produced poor results. We therefore selected a class of volumetric techniques that provided the robustness to error that was required.

Volumetric techniques create a subdivision of the 3D space to integrate many views into a single volume. Volumetric Range Image Processing (VRIP) (Curless & Levoy, 1996) provides a weighted average of meshes in voxel space, creating an averaged surface from several noisy samples. This technique was used to generate a large, highly detailed model of Michelangelo's David (Levoy et al., 2000). Their approach is used as a benchmark and as a standard tool for reconstruction (Seitz et al., 2006; Kazhdan et al., 2006). This technique is limited by the constant resolution of the grid requiring large amounts of memory to be capable of generating detailed models. This limitation inspired the use of adaptable grids to allow for greater memory efficiency. Ohtake et al. introduced octree structures as a means of adaptively subdividing space (Ohtake et al., 2003). The version of our system presented in this paper uses VRIP and a simple strategy to subdivide the total volume into manageable problems that include spatially adjacent meshes even if they are temporally distant (i.e., loop closures).

We have also explored the use of more recent techniques based on Fast Fourier Transform (FFT) convolution of the points with a filter solving for an implicit surface (Kazhdan, 2005),which was later reformulated as solving for a Poisson distribution (Kazhdan et al., 2006). However these techniques are more complex than VRIP and allowed us less flexibility in multithreading and control over the output. Both techniques were poorly suited as they are intended to estimate closed surfaces and make no use of the visibility space carving of VRIP and as such produce overconfident interpolations creating large amounts of data where none exists. Another promising technique is the Irregular Triangular Mesh representation (Rekleitis et al., 2007), which provides a variable resolution model which can incorporate overhangs, however the outlier rejection is based upon edge length and therefore lacks the full 3D voxel averaging and outlier control of VRIP. For these reasons we decided to utilize VRIP for the integration of the meshes.



(a) Number of samples

(b) Standard Deviation

Figure 11: Number of samples and standard deviation along the vertical (Z) axis in 10cm cells.

When using VRIP the quality of the integrated mesh is dependent on the selection of an appropriate ramp function used to weight meshes. The length of this ramp determines the distance over which points influence a voxel cell. The amount of noise in the data and the resolution of the grid help dictate the length of the ramp, trading off smoothness for detail. In other words on noisy data averaging a large number of samples, or a large ramp, will produce smoother results while a short ramp averages only a few samples preserving high frequency data. For our data a ramp value was selected experimentally in proportion to our largest estimated misregistration. An example of the typical standard deviation in Z is shown in Figure 11(b). Grid resolution is another important factor to consider when using this algorithm. We chose to limit the onscreen polygon count to approximately 20,000 to guarantee smooth rendering on a laptop. Another requirement was to be able to view sections at least 10m long (approx. 20m^2) at the highest level of detail. A grid resolution of 33mm produced meshes of 10m transects with approximately 20,000 faces.

VRIP is a fixed resolution technique and even though it uses run-length encoding of the voxel space, which can offer a 1:10 savings in memory usage, integrating entire mission areas

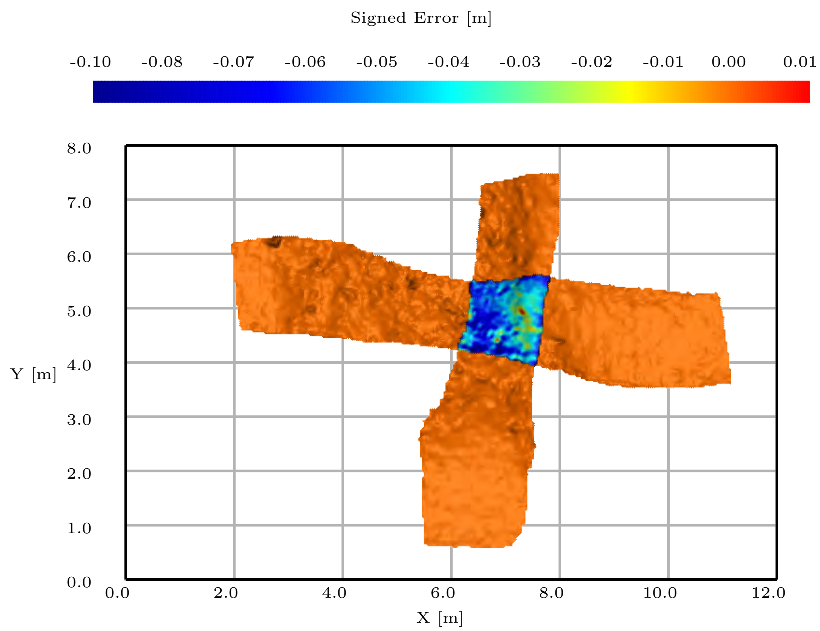
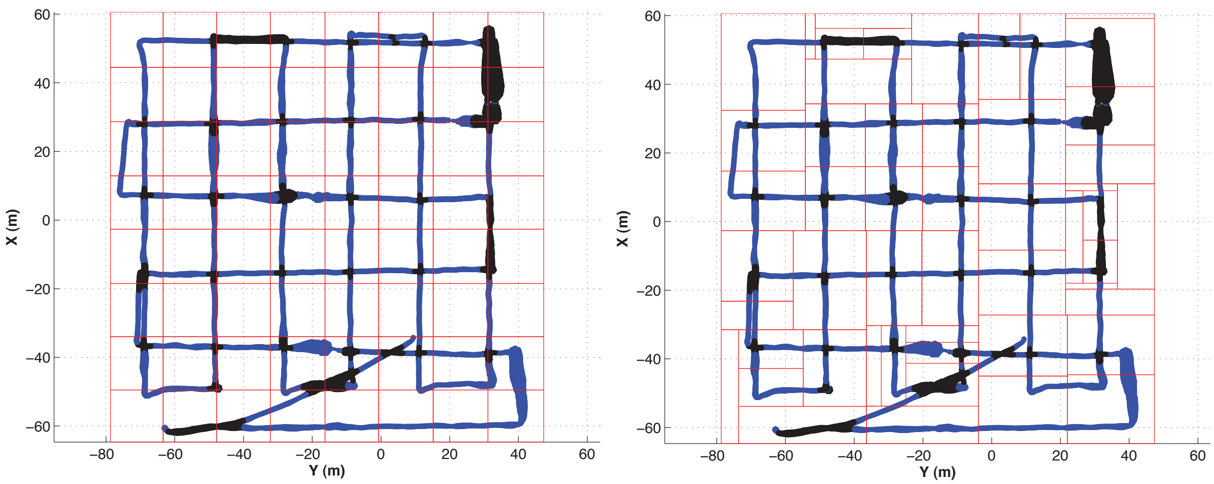


Figure 12: Mesh integrated using temporal splitting. The two crossing transects have first been merged individually. The resulting aggregate mesh features an area at the crossover that is not consistently merged when the final meshes are assembled. The inconsistency here can most likely be attributed to stereo triangulation errors or tidal effects which are currently not modeled.

is infeasible. For a 4 hour dive at 0.5m/s the vehicle covers 7200 linear meters, assuming a 2 m wide swath and a 30m vertical excursion, the volume of space to discretize is 432,000m^3. At 33 mm voxel resolution there are 27826 voxels per m^3 resulting in over 12×10^9 voxels for that volume. Even if a voxel was encoded as 1 byte, this is already 12GB of RAM which exceeds the limits of 32-bit systems. In addition, the bounding volume of a survey will grow with greater depth excursions and survey patterns that deviate from a simple linear transect. One possible solution to this sparse problem is to use adaptive grids, such as octrees, to manage the computational requirements of the map building process. We have started exploring an integration technique using quadtrees, the two-dimensional analog of an octree. The quadtree method use a 2.5D representation which is a reasonable approximation given our imaging geometry (Johnson-Roberson et al., 2009).

In this paper we present a more mature approach that uses constant resolution grids but subdivides the problem into several subtasks to perform integration within available memory. A number of methods to achieve subdivision of the imaged space were considered. Splitting the meshes based on temporal constraints is not appropriate in this case as many of the AUV deployments feature overlapping grids and portions of the survey that are temporally separated may in fact be imaging spatially nearby regions. As shown in Figure 12, if two meshes are first merged based on temporal constraints, the resulting aggregate mesh features relatively large errors when the meshes are finally assembled. A spatial subdivision is therefore more appropriate and two potential approaches were considered. The first is a trivial even division of space where grid lines are evenly distributed across the entire modeling space. An



(a) Even subdivision

(b) Heuristic subdivision

Figure 13: Subdivision of the mesh integration tasks to allow VRIP to operate on subsets of the problem. Divisions are made using (a) an even subdivision of space and (b) a heuristic based subdivision of space that penalises having boundaries along a transect or at an intersection. Stereo meshes are shown in blue with crossover points depicted in black and red lines indicating where subdivisions have been created.

example can be seen in Figure 13(a). This can introduce errors if the subdivisions are laid down along a transect or at an intersection point as portions of individual meshes may be integrated into different subdivisions. This may introduce seams at the intersection when the final meshes are combined. The second approach uses a cost function which penalizes putting boundaries over loop closures and an example subdivision can be seen in Figure 13(b). These loop closure intersections are points at which there is considerable redundancy in the available meshes and errors in geo-referencing associated with the navigation solution may be most pronounced. By ensuring that the subdivision of the meshes around these cross-overs is avoided, VRIP is able to more consistently aggregate the meshes. This type of approach is only appropriate for overlapping grid and transect mission profiles. In the case of dense mission trajectories the even splitting technique is used.

4.3 Level of Detail Generation

Level of detail (LOD) techniques enable the viewing of extremely large models with limited computational bandwidth (Clark, 1976). The underlying concept is to reduce the complexity of a 3D scene in proportion to the viewing distance or relative size in screen space. The scale and density of mission reconstruction requires some LOD processing to allow for rendering the models on current hardware. Some AUV missions have upwards of 10,000 pairs of images which expand to hundreds of millions of vertices when the individual stereo pairs are processed. This would require multiple GB of RAM if kept in core, which is impractical on conventional hardware. To view this data, a discrete paged level of detail scheme is used in which several discrete simplifications of geometry and texture data are generated and stored on disk. These are paged in and out of memory based upon the viewing distance to the object.

Integral to the discrete level of detail scheme is a method of simplification of the full resolution mesh. We use a quadric error method of decimation first introduced in (Garland & Heckbert, 1997) and extended in (Garland & Heckbert, 1998). It is based on collapsing edges, a process in which two vertices connected by an edge are collapsed into a new vertex location. This simplification step moves the vertices v_1 and v_2 to the new position $\hat{v}$, connects all their incident edges to $\hat{v}$ and deletes the vertices v_2 and v_1. The selection of which vertices to remove from a mesh is done using a quadric error metric, $Q(v)$, that describes the cost of removing a vertex. This cost is equivalent to the distance from the vertex to all of its incident planes. The process of mesh simplification in outline is as follows,

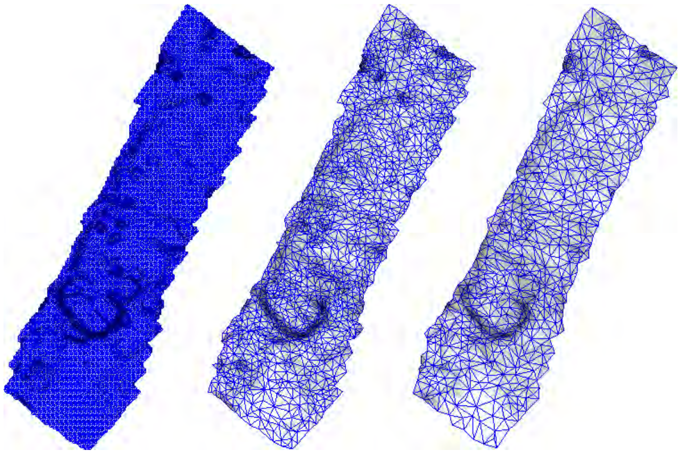
1. Compute cost for all vertices
2. Place vertices in a priority queue with min cost at the root
3. Collapse the root vertex and recompute all costs
4. Repeat algorithm until desired mesh complexity is reached.

For the current setup we generate three simplified versions from the original mesh at approximately 1000, 100, and 10 polygons per m^2. Figure 14 shows an example of the mesh simplification process, demonstrating the reduction in complexity of the mesh as the number of triangles included are reduced by an order of magnitude at each step. Figure 14(c) shows a texture mapped version of the most simplified mesh, illustrating the fact that viewing the mesh from a distance can still be informative even at a relatively low level of mesh complexity. As the user zooms in on a particular section of mesh, the increasingly detailed meshes are loaded from disk and presented for detailed inspection of the sea-floor structure.

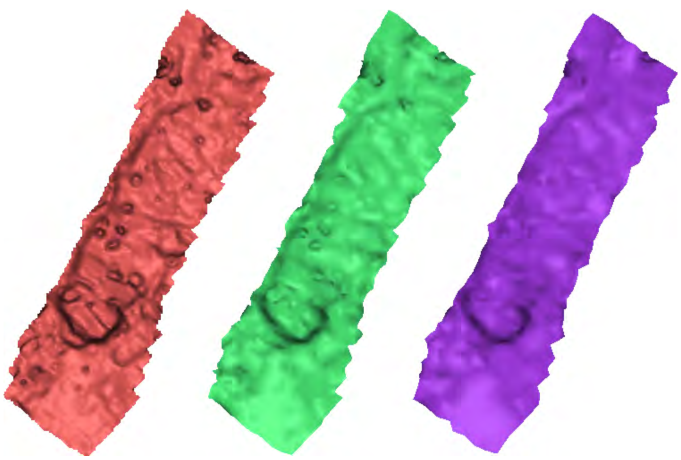
Suitable LOD down-sampling ranges were selected to allow for satisfactory operation on laptops released in the last three years. This make the system accessible to most users on a ship but requires that the system be capable of running on a system with limited graphics processing power (as most laptop use integrated GPUs), limited graphics RAM (usually 16-64MB) but access to most newer shader functionality $\geq$ OpenGL 1.5 support (Segal & Akeley, 2003). These requirements dictate the tuning of the lower levels of detail, however we wanted to maintain the ability to visualize single full-resolution images and all associated feature points at the highest LOD. When zoomed in there is minimal loss of detail, minor compression artifacts but full resolution imagery. With respect to geometry we have tuned the highest LODs to be at the limit of the GPUs at frame-rate. Higher density models are possible but require a change in the target hardware platform.

5 Texturing

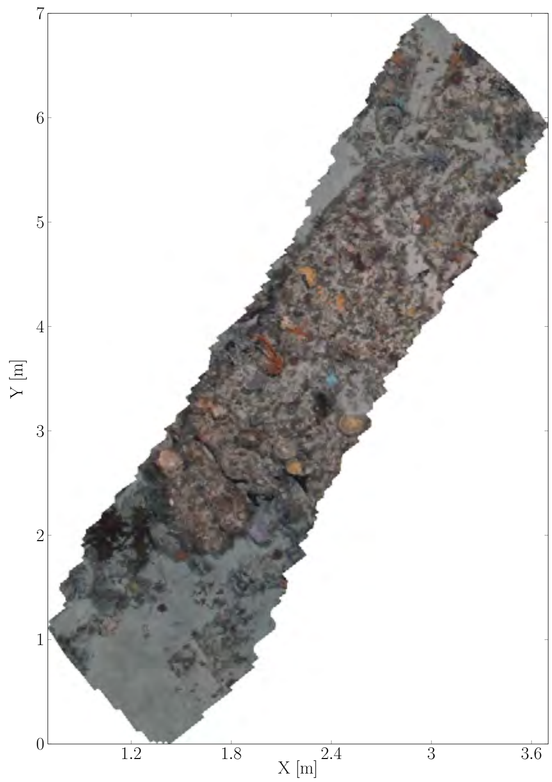
It is often desirable to display detail beyond that which is modeled by the three dimensional shape of a mesh. Texture mapping projects images onto the surface of a mesh allowing finer detail than the structure contains to be displayed (Heckbert, 1986). Traditional techniques of visualizing AUV images utilize 2D image mosaicing to display the imagery in a spatially consistent fashion but eliminate structure and result in strong distortions (Singh, Howland, & Pizarro, 2004). Through parametric mapping of the imagery onto the meshes we can effectively mosaic the images while accounting for the structure in the scene. The process



(a) Three meshes in wireframe displaying the reduction in mesh complexity



(b) Three increasingly simplified meshes in shaded relief



(c) Texture Mapped mesh

Figure 14: The meshes above display the result of the simplification process described in Section 4.3. (a) meshes from left to the right each being a reduction in triangles of one order of magnitude the first is 100% followed by 10% and finally 1%. Each corresponds to a level of detail in the hierarchy described in Section 4.3. The shaded images in (b) highlight the loss of relief but the overall persistence of outline and shape. (c) A texture mapped version of the most simplified mesh. While the mesh itself is relatively simple, the texture mapped images allow the gross structure to be inferred, even at a large viewing distance.

determines the projective camera viewpoints that have imaged a particular triangle on the mesh and then assigns two varying parameters (u, v) to each vertex that is a mapping into the corresponding image.

Using survey images directly as texture maps for a 3D mesh can create distracting visual artifacts and destroy the impression of a photorealistic model. These issues arise primarily from visible lighting patterns and misregistration. While our system compensates partially for non-ideal moving light sources and strong lighting attenuation, any residual differences in appearance of the same scene point when viewed from a different viewpoint will produce seam-like artifacts when switching to textures from a different view. Radiance maps can restore the dynamic range of images (Debevec & Malik, 1997), which in part mitigates this problem for texturing, but require highly redundant views at different exposure settings. This is impractical underwater as it would require significantly more lighting energy and data storage.

In the same way that lighting patterns cause visual inconsistency, registration error can also introduce artefacts in the reconstruction. Registration errors occur when the camera poses and 3D structure have errors that result in images of the same scene point being reprojected on different parts of the 3D model. These errors are unavoidable when using approximate 3D structure in the form of meshes derived from a sparse set of features. This type of problem is common in mosaicking applications when camera motion induces parallax but the scene is assumed to be planar. In order to produce a visually consistent texture most approaches exploit the redundancy in views by fusing the views in such a manner that high frequency components have a narrow mixing region, reducing the chances of ghosting (Uyttendaele et al., 2001). We adapt band-limited blending (Burt & Adelson, 1983) to be used on 3D meshes with calculations performed on a GPU and without having to explicitly determine the boundary on which blending is performed. This technique allows for the blending to be computed in real-time transparent to the rendering process.

Blending several textures at a vertex requires the calculation of the projection of that vertex into all cameras in which it was seen. In computer graphics this is the parameterization of images known as texture coordinates (Heckbert, 1986). Because the mesh integration step combines several meshes to produce the final mesh the original one-to-one mapping that existed between feature points and image coordinates has been lost. However the new merged vertices can be assigned to image coordinates in all cameras that view them through a process of back projection. Naively one would traverse through all projections and check which were valid but this is a costly procedure when performed on all mesh vertices. We create a bounding box tree (Smits, 1999) to allow all camera frames that view a mesh vertex to be quickly located. This bounding box tree contains all of the bounding volumes of the original triangulated meshes and produces a fast query of all of the cameras that include a point. This operation is performed for every vertex of the mesh, generating multiple image coordinates for all cameras. This allows us to describe an image pixel's correspondence to the world for each view which in turn allows us to blend all pixels associated with a particular face in the mesh. In the following section we disscuss the blending in more detail.

5.1 Texture Blending

The mechanism that performs the blending is based on the use of *image splines* (Burt & Adelson, 1983). The image spline is used to blend the image seam smoothly without losing fine detail from either image. Assuming that there is some misregistration between the images, blending can result in some ghosting (multiple instances of objects, particularly noticeable at higher spatial frequencies). Prior work has shown that for most real-world images it is impossible to have one spline which appropriately blends all the frequency components of an image (Burt & Adelson, 1983). Therefore in order to perform blending, the images must be decomposed into frequency bands and an appropriate blending spline selected for each band. We choose to blend three non-overlapping frequency bands that

Algorithm 1 The color of a pixel on the mesh.

for $vert_i$ in the set of all vertex **do**
  for k = (5,10,50) **do** {Each band limited image}
    for $(u, v)_j$ in the set of all projections of $vert_i$ **do**
      $\{(u_{center}, v_{center})$ is the center of the image for projection $j\}$
      $r = dist((u, v)_j - (u_{center}, v_{center}))$
      Calculate Non-normailzed weight $B_k(r)$ using Eq. 1.
    end for
    Calculate normalized weight W_k^i for all B_k for $vert_i$ using eq. 2
    $\sum$ all normalized weight colors at $(u, v)_j$
  end for
  Recombine each band into final pixel color
end for

comprise the entire image. Three bands were selected empirically after limiting factors such as floating point precision and image size in the GPU showed diminishing returns for any additional band use. The frequency decomposition can be represented as a set of low pass filters applied to a series of images in one-octave steps. Burt and Adelson propose the use of an image pyramid to perform the filtering using band pass component images. We extend this work by using a novel implementation on a GPU which allows for efficient and simple processing to achieve a similar result. Graphics cards are setup to handle images for the purposes of texturing geometry and can quickly load and manipulate such texture data (Oka et al., 1987; Catmull, 1974). Specifically the GPU's hardware mipmapping (L. Williams, 1983) is leveraged to create the texture frequency decomposition and then a shader is used to perform the blending.

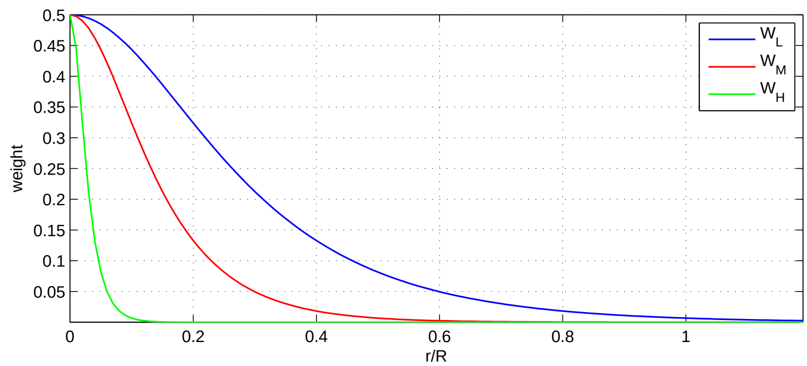
The steps used to calculate the color of a pixel on the mesh are shown in Pseudocode in Algorithm 1. The weighting function that determines the degree to which differing source pixels contribute to the blend is optimised for pixels from the center of images. Since the images display significant radial distortion and illumination falloff away from the center this weighting favors pixels with better imaging geometry (Jaffe, 1990). The formula to derive the non-normalized weighting value B_k for an image at a pixel is shown in Eq. 1,

$$B_k(r) = \frac{e^{-k \cdot \frac{r}{R}}}{1 + e^{-2 \cdot k \cdot \frac{r}{R}}} \tag{1}$$

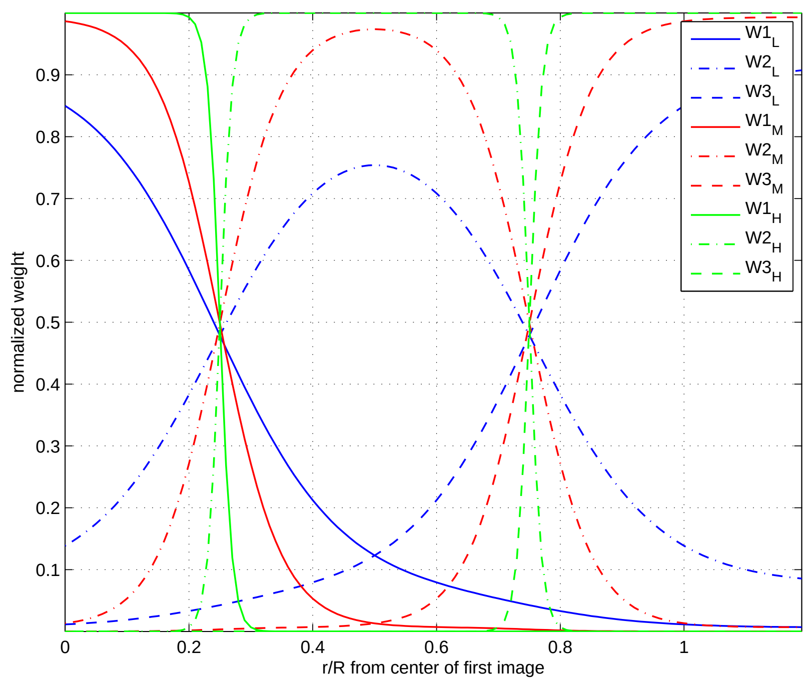
where r is the distance to the center of the image and R is a reference distance (typically the maximum distance to be considered). Each frequency band is shaped by k, with larger k inducing a sharper drop-off. Figure 15(a) illustrates the shapes of the weighting function for $k = 5, 10, 50$. The actual weights applied to the textures are normalized by the total weights contributed by the images nearby as shown in Eq 2,

$$W_k^i(r_i) = \frac{B_k^i(r_i)}{\sum_j B_k^j(r_j)} \tag{2}$$

where W_k^i is the normalized weight for image i with drop-off shaped by k. Figure 15(b) shows an example of the normalized weights for three partially overlapping images.



(a) Weighting as a function of normalized distance to center



(b) Actual weightings for three images

Figure 15: Plot of three weighting functions used to combine frequency bands (a) Weighting as a function of normalized distance to center for Low (blue), Medium (red) and High (green) frequency bands. (b) Actual weightings for three images with centers at $r/R = \{0, 0.5, 1\}$. The weights for Image 1 centered at $r/R = 0$ are shown with a solid line. Image 2 is centered on $r/R = 0.5$ and its weights are shown as a dotted line. Image 3 is centered on $r/R = 1$ and its weights correspond to the dashed lines. The weights for the low frequency bands are in blue, for the medium frequency band in red and for the high frequency bands in green. Notice the sharp transition zone for the high frequency components, whereas there is a more gradual blending of the low frequency components.

The GPU shader code for this technique is novel in its use of texture arrays which allows for simultaneous access to many 2D textures enabling blending to be performed in real time. The technique produces visually appealing results. Figure 16 presents three views of the same section of mesh with a different texture blending algorithm applied in each. Figure 16(a) shows an unblended approach of selecting the closest image and the characteristic seams that exist without blending in projective texturing from multiple images. A naively blended mesh can be seen in Figure 16(b) where each pixel is the average of all views of that point. The results of the proposed technique are displayed in Figure 16(c). As can be seen, the proposed approach results in a blended, textured mesh with fewer visible seams without a loss of high detail texture. Figure 16(d) illustrates a short section of blended mesh.

The algorithm uses a single programmable shader and takes approximately 7 ms per render update using the GeForce 8600GS, a modest dedicated graphics card that supports texture arrays. It is possible to maintain over 120 fps with this technique making it more than suitable for realtime pipeline rendering, however because the blending is static this rendering is only necessary once as the resulting texture is saved and replaces the original unblended texture. This means the entire process requires only an additional one time cost of 7 ms per image (along with the time to write new images to disk) to produce a final blended mesh. This also removes the constraint of requiring a graphics card which supports the extension of 2D texture arrays to display the blended results, making this technique accessible to virtually all modern computer hardware once the blended textures have been generated.

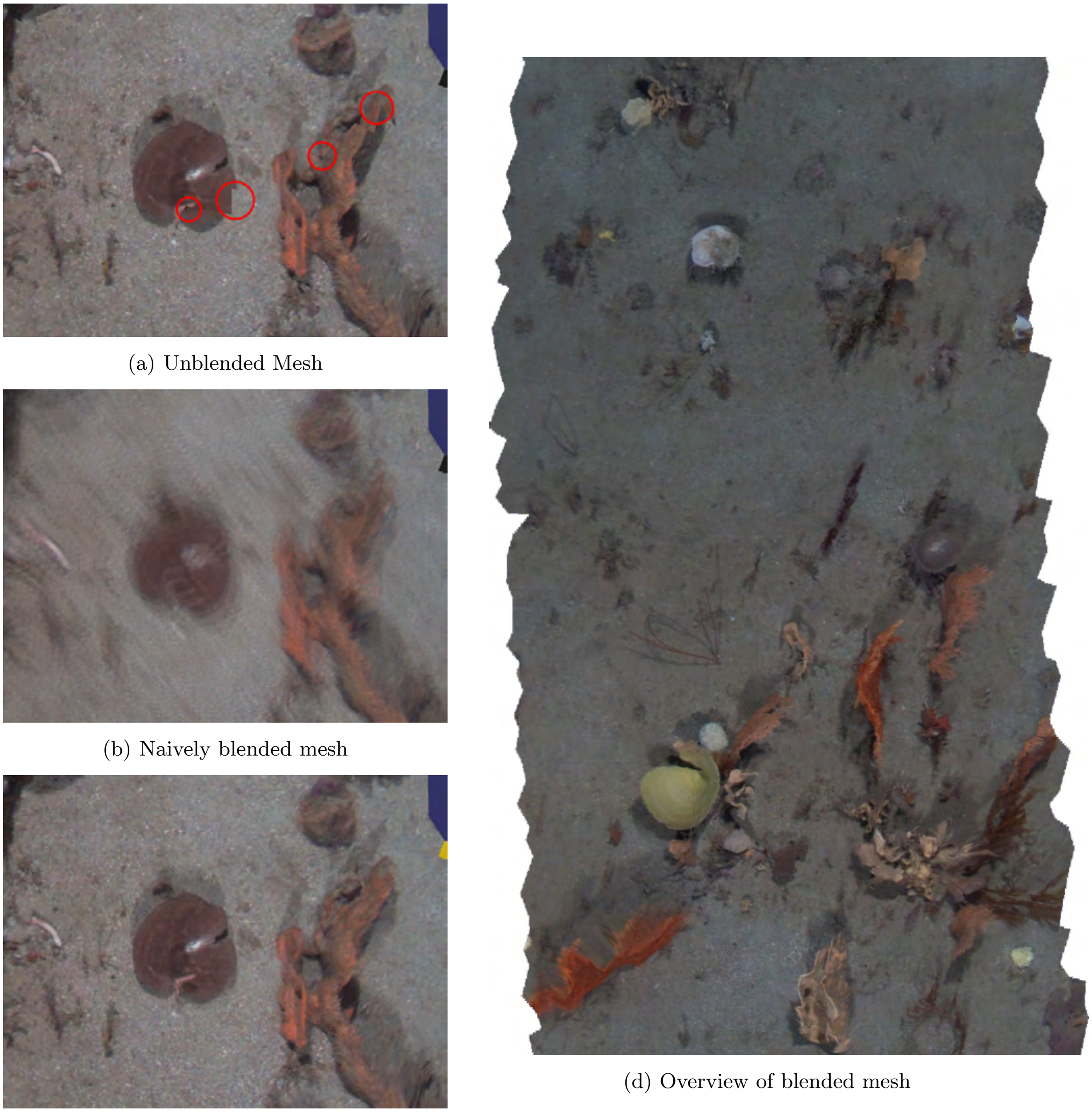
6 Practical Considerations

While considerable work has gone into establishing the pipeline required to generate detailed, texture mapped models of the seafloor using the techniques described here, a number of practical considerations have been addressed in order to allow these models to be generated and displayed in a timely fashion. This section examines some of these issues and how they have affected the design decisions and performance of the system.

6.1 Texture Compression

One of the central goals of the visualization of large data sets is the ability to display all images and structure simultaneously. This is particularly challenging if the visualization is to run on commodity hardware. Two issues dominate the visualization of tens of thousands of images. First, keeping all images in system memory. At approximately two megabytes per image, loading 10,000 images would require 20GB of system memory which is beyond the capacity of most current desktops and laptops. Second, modern graphics cards have limited memory and processing power despite great advances in the last five years. Thus only 10 or 15 full resolution images can be held in the 32 MB of graphics memory as well as a limited number of vertices and faces.

Texture compression serves to increase the number of images that can be viewed (Liou et al., 1990). Hardware implementations are now fairly ubiquitous in commodity graphics cards. We are utilizing the DXT1 variant of texture compression which represents a 4×4 block of pixels using 64 bits. DXT1 is a block compression scheme, in which each block of 16 pixels in the image is represented with a start and end color in 5 bits for red, 6 for green and 5 for blue and a 4×4 2-bit lookup table to determine the color level for each pixel. This compression algorithm achieves an 8:1 compression ratio.



(a) Unblended Mesh

(b) Naively blended mesh

(c) Blended Mesh

(d) Overview of blended mesh

Figure 16: Blended and Unblended Meshes displaying visual results from the proposed technique (a) Note the seams highlighted in red on the unblended image that selects the closest texture for each surface (b) A naive blending that averages the textures for each surface results in significant blurring. (c) The band limited blending proposed here preserves significant detail while avoiding seams in the blended image. (d) Overview of a section of blended mesh

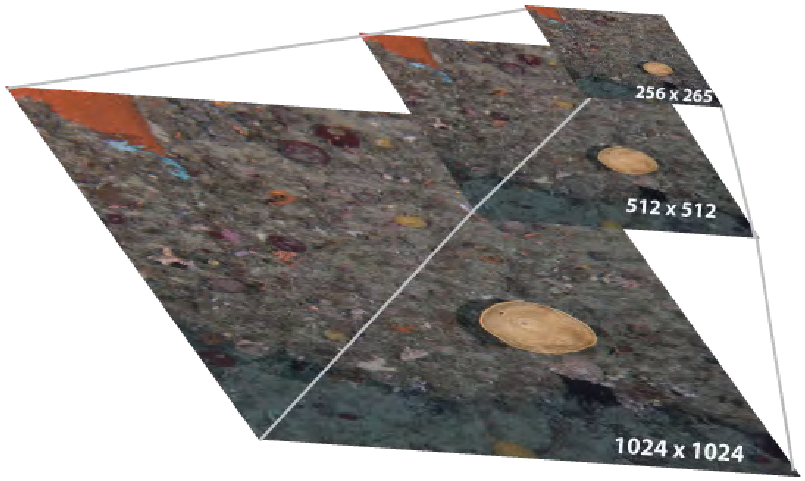


Figure 17: An example of texture LOD images

6.2 Texture Pyramid

With the limitations in system memory and GPU power, textures must be managed to maintain the performance of the system. Just as LOD schemes are used for geometry (Clark, 1976), textures can be represented in a multi-resolution pyramid. An example of image sizes can be seen in Figure 17. Furthermore a technique traditionally harnessed for distance dependent display of textures known as mipmapping exists as a hardware feature of all modern GPUs. Mipmapping is the generation of the aforementioned image pyramid (used for texture blending) from a high-resolution texture at reduced resolutions (L. Williams, 1983; Strengert et al., 2006). Traditionally these images are generated in quarter resolution. If the initial image is 256×256 a total of eight mipmap images will be generated at 128×128 pixels, 64×64, 32×32, 16×16, 8×8, 4×4, 2×2, 1×1. By using the hardware generated pyramid, computation time can be save in the LOD generation step. Then we store these automatically generated texture pyramids in an explicit discrete level of detail (DLOD) textured model. Levels are created prior to runtime and the system selects the LOD most appropriate for the viewing distance. This makes effective use of the screen's limited resolution when viewing large numbers of images. DLOD schemes can suffer from the introduction of visual error when switching levels when compared with some recent continuous level of detail schemes (Ma et al., 2007; Ramos et al., 2006). However, we consider that these disadvantages are outweighed by the simpler requirements on hardware, making the visualization system more accessible.

6.3 Binary Mesh Format Generation

We optimize the storage of these meshes in individual binary format meshes with textures stored internally in compressed format allowing for the minimum of transformation to load data files into system and graphics memory. The images are stored in their natively compressed DirectDraw Surface (DDS) format and the meshes can stream directly into vertex buffers in the graphics card. We also use the binary mesh format of Open Scene Graph to aid in efficient geometry storage (*OpenSceneGraph*, n.d.). We utilize a multi-threaded paging system which pulls each sub-mesh as created in Section 4.2 into memory when the viewing distance is close. This paging allows for the entire mesh to be seen simultaneously while only high detail sections are paged in when necessary. The final binary mesh with compressed textures takes up approximately 2.5 MB per image on average. A typical 100 image strip is about 25MB and a typical 19000 image complete mission is about 4.8GB, which is well within the storage capabilities of current computers.

6.4 GPU

GPUs are naturally suited to manipulating texture data and significant speed gains can be achieved by re-implementing the texture processing segments of the pipeline in graphics hardware. In the current implementation we perform all the texture compression and blending in the GPU. This allows for greater parallelization as these tasks can be performed without overloading the CPU, leaving it free to continue processing the mesh geometry. We make use of NVIDIA CUDA for texture compression which offers a speedup of over 12x over CPU based texture compression (NVIDIA, 2008). The texture blending (Section 5.1) is performed in realtime and creates a negligible slowdown in processing.

6.5 Multi-threading and Distributed Processing

The introduction of multi-core CPUs to the desktop market has brought symmetric multiprocessing to the main stream. The challenge is now to write code that can take advantage of this parallelism. We have taken several sections of the pipeline and made them parallel so that they can take advantage of these modern CPUs. In addition we have taken this a step further and implemented a system for distributing the processing across multiple machines further decreasing processing time. The basis of both techniques is that there is no need for synchronization between frames in each individual stage of the pipeline. The stereo processing of each pair is independent of the previous pair in the current implementation, therefore the task can be completely divided along with the data. Thread synchronization is only needed in between each pipeline step. The distribution of the task uses distributed filesystems (DFSs), in this case NFS (Network File System) but almost any modern DFS will work. This is possible because all metadata is stored as a file maintained by a single synchronizing process. This process spawns distributed children and these children need only read that metadata file and have access to the relevant source data for their section. As all data for a mission resides in this distributed file system all machines with the binaries for processing can operate on any part of the mission and store their results back onto the same directories. For the following discussion we refer to nodes that can be either threads or distributed processes. In the current implementation we multithread or distribute the pipeline using the method described as follows.

- Stereo Depth Estimation - each stereo pair is independent so each node receives a list of image pairs and stores the resulting meshes in the DFS
- Synchronization occurs and meshes are transformed into global space and divided as discussed in Section 4.2 into spatial regions for integration.
- Mesh aggregation - Each node receives a list of spatial regions to integrate

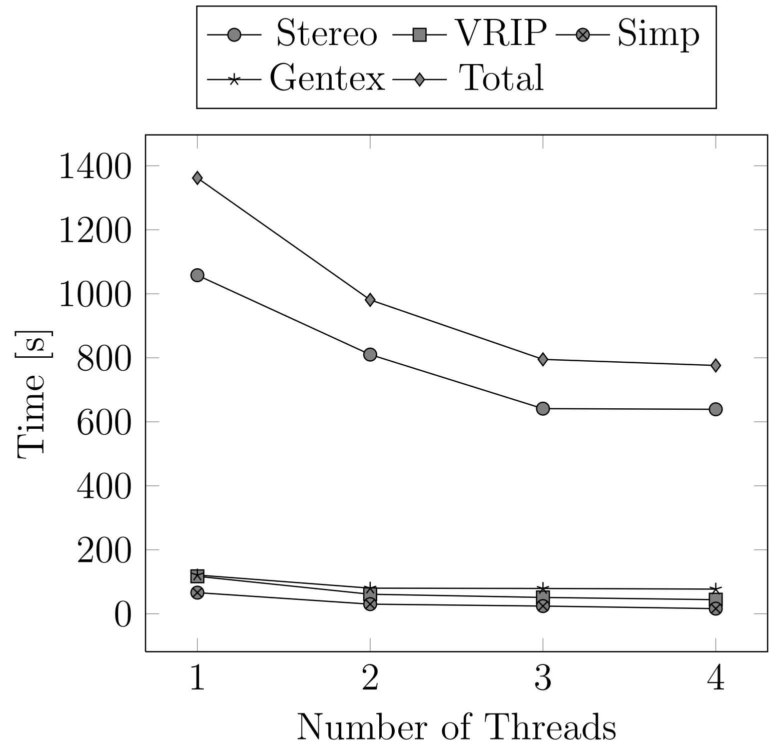


Figure 18: Timing results as a function of number of threads on a 3.2Ghz quad-core CPU processing a mission of 1623 images. Note the step between one and two threads falls short of the theoretical improvement of halving the processing time due to overhead and management costs from the thread pool implementation which is only used beyond one thread. This is compounded by a sync step which occurs when data is pushed onto the GPU.

- Synchronization occurs
- Simplification - each spatial region is simplified for LOD rendering
- Synchronization occurs
- Texturing - each node again receives a spatial region and now a list of images to process, compress, and apply to the geometry within its region.
- Finally synchronization occurs one last time and the spatial regions are tied together in LOD hierarchy for viewing.

The results of this threading can be seen in Figure 18 where timing results show the effectiveness of dividing the problem. The tests were run on a 3.2Ghz quad-core CPU. Note the diminishing returns beyond three threads. This is attributable to limited disk access speed with image loading becoming the bottleneck.

7 Results

The techniques described in this paper have been used to produce seabed reconstructions on a number of AUV missions around Australia.

Table 2: Timing results for selected missions with increasing numbers of images

num images	1623	3393	6559	19004
mission duration (s)	812	1697	3280	9502
time stereo (s)	639	1337	2863	7103
time vrip (s)	44	145	328	401
time simplification (s)	16	98	113	212
time mesh generation (s)	77	216	430	1231
Total (s)	776	1796	3734	8947

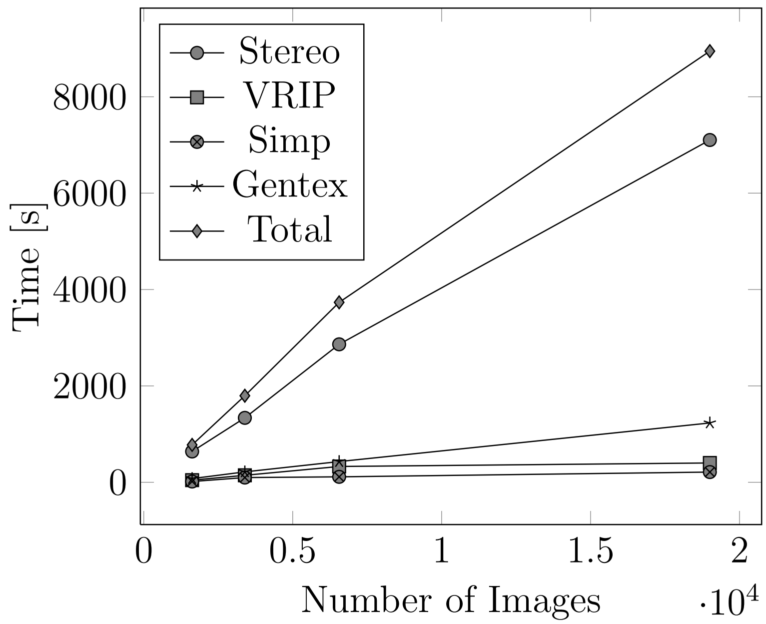


Figure 19: Timing results as a function of number of images.

7.1 Timing results

Speed was an important factor in the choices that went into developing this pipeline. One major requirement is the ability to turn source data into 3D models quickly so that this tool can be used during field deployments to rapidly perform habitat assessments and to help in planning subsequent dives. The stereo processing is by far the slowest processing step as can be seen in Figure 19. However this is the simplest step to replace and the stereo calculation is currently being cached, meaning it is typically run once per mission. The optimization efforts have been focused on the subsequent steps of the pipeline as illustrated by the performance results in Table 2. As can be seen, generating the high resolution, 3D sea-floor models takes a similar amount of time to the mission itself.

7.2 Use Cases

We present the results of two missions processed using this pipeline, and illustrate the potential use of the visualization by progressively zooming into the reconstruction.

7.2.1 Drowned Reefs in the Great Barrier Reef

The AUV was part of a three week research cruise in September 2007 aboard the R/V Southern Surveyor documenting drowned shelf edge reefs at multiple sites in four areas along the Great Barrier Reef (Webster et al., 2008; S. Williams et al., 2008). We were able to document relic reefs up to 20k years old formed during previous ice ages when sea levels were up to 70 m lower than today. The study of these structures may yield insights regarding potential future sea level changes and their potential impact on sensitive reef areas such as the Great Barrier Reef (GBR). Figure 20 contains views from a reconstruction of a dive site at Noggin Reef, one of the study sites off the coast at Cairns in far north Queensland, Australia. The dive targeted a particular reef structure in a depth of 60 m and featured multiple crossovers covering both relatively flat bottoms and high-relief reef sections. The figure illustrates how a user might examine an overview of the dive profile before zooming in on particular areas of interest.

The ways in which these reconstructions can be used as tools for science are only beginning to be fully explored. One application that has emerged based on this work is the study of spatially correlated behaviors in benthic macrofauna (Byrne et al., 2009). It was discovered using AUV imagery that subsea dunes at Hydrographers Passage on the GBR support communities of the brittle star, *Ophiopsila pantherina*. The dunes covered approximately 340 hectares of sea floor at 60-70m placing them beyond a depth at which routine SCUBA based study methods are tractable. It was thought that *O. pantherina* take refuge on the lee side of the dunes, however this is not apparent in the 2D image mosaics. The 3D reconstructions made it easy to observe that the brittle stars appeared only on the shielded slope of the dunes. This phenomena can be seen in Figure 21. In addition *O. pantherina's* appearance was quite patchy in the survey region. From the AUV navigation we were able to provide geo-referenced locations for places in the reconstruction where brittle stars had been seen. The dunes were re-sampled during a follow up research cruise to the area and the stars were found in the geo-refrenced locations predicted from the AUV dive completed twelve months earlier. Likewise, no brittle stars were found in the areas that were predicted to be free of them. This supported two important hypotheses, first that the reconstruction is capable of providing geo-refrenced observations which can be used as a guide for not only additional AUV dives but also other sampling methods (physical grabs in this case) and that biologically *O. pantherina* have locationally stable communities (Byrne et al., 2009).

7.2.2 Marine Habitats in the Tasman Peninsula

In October 2008 the AUV was used to survey several sites along the Tasman Peninsula in South Eastern Tasmania as part of the Commonwealth Environmental Research Facility (CERF) Marine Biodiversity Research Hub (S. Williams et al., 2009). The science party

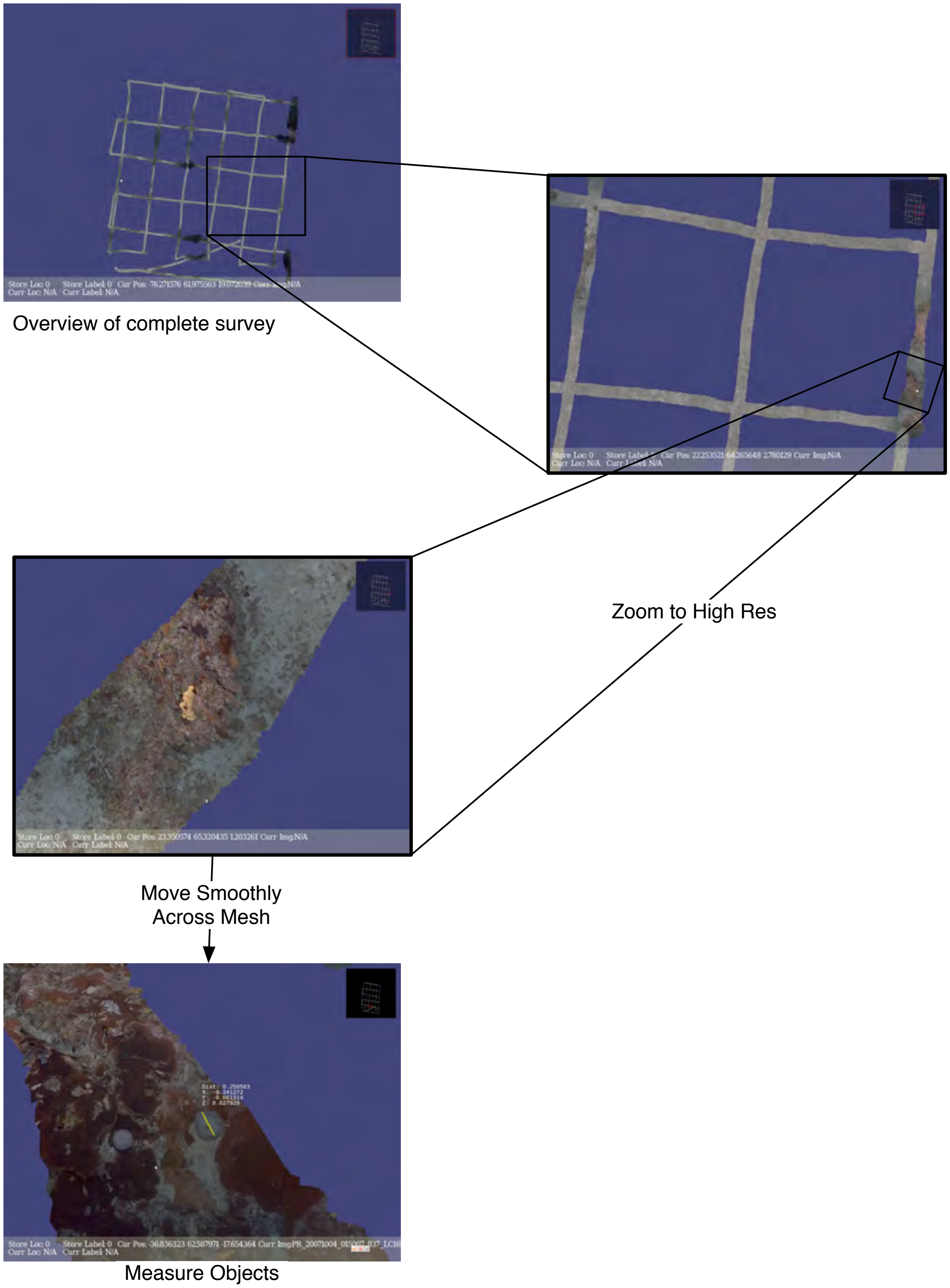


Figure 20: Usage example for a dive undertaken on the Great Barrier Reef in far north Queensland. By first examining an overview of the dive, one gets a good sense of the distribution of reefs before zooming in on particular details of the terrain structure.

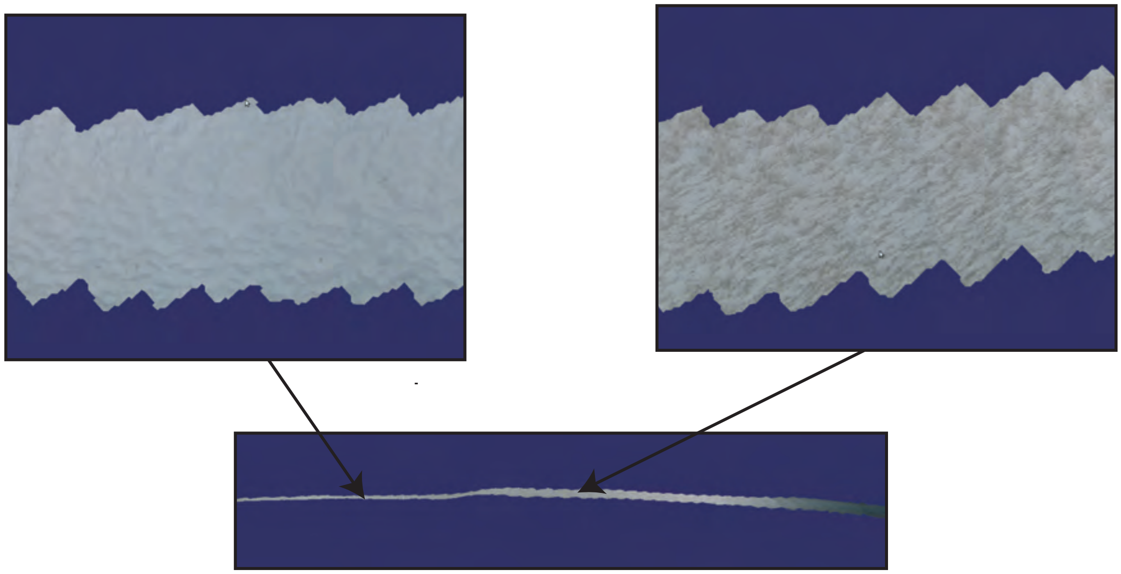


Figure 21: The reconstruction of sand dunes from a GBR AUV mission to study *Ophiopsila pantherina*, also known as brittle stars. The brittle stars appear on the right inset image but not in the left. Note that the orthographic views of the scene do not allow the observer to understand the structure of the environment. Once the 3D mesh is viewed side-on it becomes apparent where the brittle stars are living on the dunes. Additionally the geo-refrenced positions of the mesh with brittle stars was used to acquire physical samples.

used existing bathymetric maps to select sites of ecological importance beyond recreational diver depths. A total of 19 dives were completed over the course of ten days of operation, with dives lasting on the order of four to six hours. During these deployment, approximately one TB of raw data was collected. This was our first research cruise with an upgraded set of field computers and a first pass of all the post-processed data was delivered on completion of the research cruise, including geo-referenced visual SLAM navigation estimates, 3D stereo meshes and multibeam bathymetery.

Figure 22 illustrates a reconstruction from a dive undertaken on this research cruise. This particular dive featured a 2 km cross-shelf transect, starting on the reef in a depth of 35 m and transiting out to a flat bottom at a depth of approximately 75 m. The figure demonstrates how the entire dive transect can be viewed, zooming in to examine particular detail of the sea-floor relief. Multibeam data gathered by the AUV is included here to add context to the visual swath. There is good correspondence between the structure generated by our system and that recovered using the multibeam swaths, once again demonstrating the quality of the stereo reconstructions.

We have also found the tool useful in visualizing and debugging the performance of the vehicle. One example of this arises when the vehicle traverses complex terrain while bottom following. It uses a forward looking obstacle avoidance sonar to identify obstacles in its path and rises to avoid these. The present trajectory generator of the vehicle leads it to remain high over the terrain if it travels off a ledge, taking time to return to the programmed altitude instead of dropping back down immediately. The effect of this, depicted in Figure 23, is a

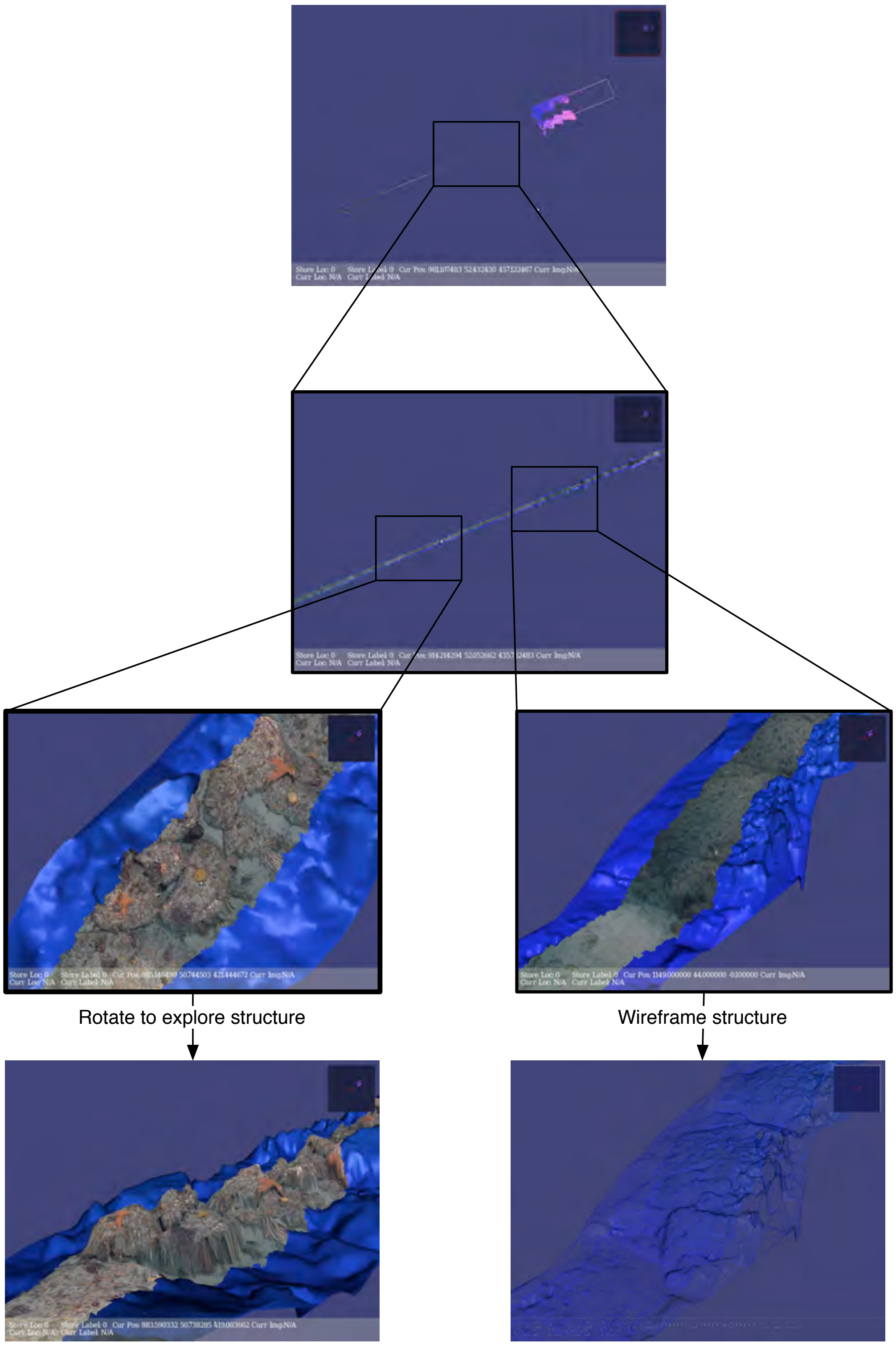


Figure 22: Usage example for a dive completed on the Tasman Peninsula demonstrating how the system allows a user to look at an entire dive transect, in this case over 2 km long, before zooming in to examine details. The imagery is shown here alongside the multibeam data collected by the vehicle and a user is able to interactively turn off the texturing to examine the underlying mesh.

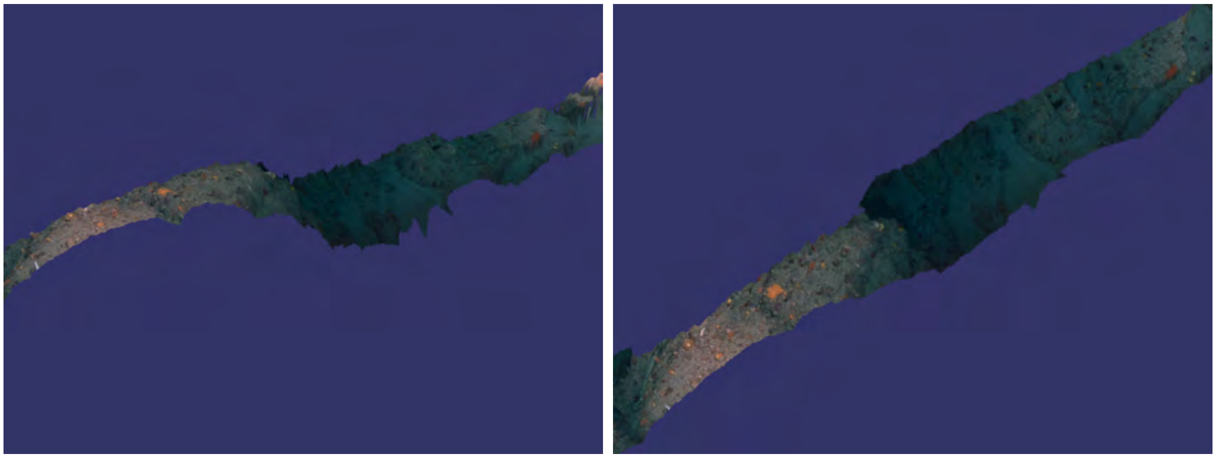


Figure 23: The effect of conservative bottom following controller. Notice the increase in stereo footprint as the vehicle is coming off the slope and the bottom drops off. The lighting correction begins to break down as the assumptions about terrain height are violated, something we discovered upon visualizing the reconstruction.

much wider stereo footprint and darker images. This consistent pattern, and its dependence on the magnitude of the drop-off, is readily visible in the 3D reconstructions while harder to observe solely from image sequences. The simple lighting correction approach adopted here breaks down in these high altitude situations and explicitly incorporating range-dependent corrections has become a direction for future work.

8 Conclusion and Future Work

Our system has performed reliably over multiple dives covering a broad range of marine habitats. The emphasis to this point has been in establishing a working pipeline capable of processing and rendering tens of thousands of images with sufficient speed to allow end user interaction with the reconstruction in time to inform further data gathering missions. The timing results show that the system can deliver the full reconstruction in time comparable to the mission duration. These results also show that our stereo depth estimation stage accounts for most of the time. It is straightforward to try other stereo processing modules and future work will evaluate recent promising approaches (Marks et al., 2008; Seitz et al., 2006). It is reasonable to assume that both system memory and processing power of CPUs and GPUs will continue to increase. This will allow us to revisit the use of dense stereo by a level or two of higher resolution LOD.

Lighting correction and band-limited blending give the end-user the impression of interacting with a visually consistent reconstruction. The underlying structure is only approximate and so far we have only partially characterized the accuracy of the individual stereo reconstructions. Future work will include using ground truth surfaces and objects at small to medium scales (requiring one to tens of stereo meshes) in water. One possible approach is to laser scan the scene in a tank and then fill it with water (Pizarro et al., 2004).

The current pipeline does not explicitly account for a dynamic world. In practice this has not been a significant drawback for most surveys for several reasons: the AUV is used mostly beyond routine scientific diving depths which implies very little motion from surface waves, so that any sessile organisms that can 'sway' in the current are essentially static when imaged over a few seconds. Cases that present problems are motile organisms such as fish, though in general it seems that they get out of the way of the AUV and are rarely captured in the imagery. Another problem occurs when loop closing after minutes or hours and slow moving organisms such as starfish have visibly moved, or if the tide has changed and anchored organisms are leaning in another direction. When they occur, moving objects are generally sparse in an image frame and robust matching still establishes correct loop closures. Even with these deficiencies, we argue that the representation allows the end user to increase their understanding of the environment at multiple scales. However, one could envision a system in which dynamic objects were detected and removed or possibly even modeled and tracked. In possible future science applications this might be a beneficial or even necessary extension of this work.

Mesh aggregation using VRIP requires partitioning the bounding volume into sections that VRIP can handle effectively with available system memory. This bounds the use of memory and imposes a computational cost approximately linear with number of images. The disadvantage with this approach is that inconsistencies are possible at partition boundaries. We are currently investigating other forms of mesh aggregation that scale to large volumes.

Being able to interact with a textured 3D reconstruction is only the first step in effectively using data acquired by robotic platforms. We are currently augmenting our system to allow it to incorporate multiple layers of textures that can represent overlays of quantitative information such as surface rugosity or classification of marine habitats. This should allow marine scientist to observe relevant spatial patterns at multiple scales, hopefully facilitating their ability to identify the underlying processes that generate the patterns, as well as helping create testable hypothesis that will ultimately lead to an improved understanding of our oceans.

Acknowledgments

This work is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government, and the Integrated Marine Observing System (IMOS) through the DIISR National Collaborative Research Infrastructure Scheme. A special thanks to the captain and crew of the R/V Southern Surveyor and R/V Challenger; without their determined assistance we could not have gathered the data used in this paper. We also acknowledge the help of all those who have contributed to the development and operation of the AUV, including Duncan Mercer, George Powell, Stephen Barkby, Ritesh Lal, Paul Rigby, Jeremy Randle, Bruce Crundwell and the late Alan Trinder.

References

Agrawal, M., Konolige, K., & Bolles, R. C. (2007, February). Localization and mapping for autonomous navigation in outdoor terrains : A stereo vision approach. In *IEEE Workshop on Applications of Computer Vision* (pp. 7–13). Austin, TX,.

Amenta, N., & Bern, M. (1999). Surface reconstruction by voronoi filtering. *Discrete and Computational Geometry*, *22*(4), 481–504.

Armstrong, R., Singh, H., Torres, J., Nemeth, R., Can, A., Roman, C., et al. (2006). Characterizing the deep insular shelf coral reef habitat of the Hind Bank marine conservation district (US Virgin Islands) using the Seabed autonomous underwater vehicle. *Continental Shelf Research*, *26*, 194–205.

Bajracharya, M., Maimone, M., & Helmick, D. (2008). Autonomy for mars rovers: Past, present, and future. *IEEE Computer*, *41*(12), 44–50.

Ballard, R., McCann, A., Yoerger, D., Whitcomb, L., Mindell, D., Oleson, J., et al. (2000). The discovery of ancient history in the deep sea using advanced deep submergence technology. *Deep-Sea Research Part I*, *47*(9), 1591–1620.

Barnard, K., Cardei, V., & Funt, B. (2002). A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data. *Image Processing, IEEE Transactions on*, *11*, 972 – 984.

Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359.

Boissonnat, J.-D. (1984). Geometric structures for three-dimensional shape representation. *Transactions on Graphics (TOG)*, *3*, 266–286.

Burghouts, G. J., & Geusebroek, J.-M. (2009). Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, *113*, 48–62.

Burt, P., & Adelson, E. (1983). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, *2*, 217—236.

Byrne, M., Woolsey, E., Williams, S., Pizarro, O., Bridge, T., Thornborough, K., et al. (2009). Flashing stars light up subaqueous dunes off the GBR. *Ecos*(July).

Campbell, R., & Flynn, P. (2001). *A survey of free-form object representation and recognition techniques* (Vol. 81).

Catmull, E. E. (1974). *A subdivision algorithm for computer display of curved surfaces.* Unpublished doctoral dissertation, The University of Utah.

Clark, J. (1976). Hierarchical geometric models for visible-surface algorithms. *Communications of the ACM*, *19*, 547 – 554.

Cornelis, N., & Gool, L. V. (2008). Fast scale invariant feature detection and matching on programmable graphics hardware. *Computer Vision and Pattern Recognition Workshops, 2008. CVPR Workshops 2008. IEEE Computer Society Conference on*, 1 – 8.

Curless, B., & Levoy, M. (1996). A volumetric method for building complex models from range images. In *ACM SIGGRAPH: Computer Graphics and Interactive Techniques* (pp. 303–312). New York, NY, USA: ACM.

Debevec, P. E., & Malik, J. (1997). Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH: Computer Graphics and Interactive Techniques* (pp. 369–378). New York, NY, USA: ACM.

Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE*

Transactions on Robotics and Automation, *17*(3), 229–241.

Duntley, S. (1963). Light in the sea. *Journal of the Optical Society of America*, *53*(2), 214–233.

Durrant-Whyte, H. (1996). An autonomous guided vehicle for cargo handling applications. *The International Journal of Robotics Research*, *15*(5), 407 – 440.

Durrant-whyte, H., & Bailey, T. (2006). Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, *13*, 99-110.

Eustice, R., Singh, H., Leonard, J., & Walter, M. (2006). Visually mapping the RMS Titanic: conservative covariance estimates for SLAM information filters. *The International Journal of Robotics Research*, *25*(12), 1223-1242.

Fitzgibbon, A., & Zisserman, A. (1998, June). Automatic Camera Recovery for Closed or Open Image Sequences. In *Proceedings of the 5th European Conference on Computer Vision* (pp. 311–326). Freiburg, Germany: Springer-Verlag.

Fleischer, S., Wang, H., Rock, S., & Lee, M. (1996, June). Video mosaicking along arbitrary vehicle paths. In *Proceedings of the 1996 symposium on autonomous underwater vehicle technology, 1996* (p. 293-299). Monterey, CA.

Fowler, R., & Little, J. (1979). Automatic extraction of irregular network digital terrain models. *ACM SIGGRAPH: Computer Graphics and Interactive Techniques*, 199 – 207.

Fruh, C., & Zakhor, A. (2004). An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, *60*, 5–24.

Garcia, R., Nicosevici, T., & Cufi, X. (2002). On the way to solve lighting problems in underwater imaging. In *MTS/IEEE Oceans* (Vol. 2, pp. 1018 – 1024).

Garland, M., & Heckbert, P. (1997). Surface simplification using quadric error metrics. *ACM SIGGRAPH: Computer Graphics and Interactive Techniques*, 209 – 216.

Garland, M., & Heckbert, P. (1998). Simplifying surfaces with color and texture using quadric error metrics. *Visualization '98. Proceedings*, 263–269.

German, C. R., Yoerger, D. R., Jakuba, M., Shank, T. M., Langmuir, C. H., & Nakamura, K. ichi. (2008). Hydrothermal exploration with the autonomous benthic explorer. *Deep-Sea Res Pt I*, *55*, 203–219.

Gracias, N., & Santos-Victor, J. (2001). Underwater mosaicing and trajectory reconstruction using globalalignment. In *MTS/IEEE Oceans* (Vol. 4, p. 2557-2563).

Harris, C., & Stephens, M. (1988). A combined corner and edge detection. *Proceedings of the Fourth Alvey Vision Conference*, 147–151.

Hartley, R., & Zisserman, A. (2000). *Multiple view geometry in computer vision.* Cambridge University Press.

Heckbert, P. (1986). Survey of texture mapping. *Computer Graphics and Applications, IEEE*, *6*, 56 – 67.

Ho, N., & Jarvis, R. (2007). Large scale 3D environmental modelling for stereoscopic walk-through visualisation. In *3DTV conference* (pp. 1 – 4).

Howland, J. (1999, December). *Digital Data Logging and Processing, Derbyshire Survey, 1997* (Tech. Rep.). Woods Hole Oceanographic Institution.

Hu, J., You, S., & Neumann, U. (2003). Approaches to large-scale urban modeling. *Computer Graphics and Applications, IEEE*, *23*, 62 – 69.

Huber, P. J. (1981). *Robust statistics.* Wiley.

Jaffe, J. (1990). Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, *15*(2), 101 – 111.

Jenkin, M., Hogue, A., German, A., Gill, S., Topol, A., & Wilson, S. (2008). Modeling underwater structures. *IJCINI*, *2*(4), 1–14.

Johnson-Roberson, M., Pizarro, O., & Willams, S. (2009, May). Large scale optical and acoustic sensor integration for visualization. In *MTS/IEEE Oceans.*

Kazhdan, M. (2005). Reconstruction of solid models from oriented point sets. *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*.

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, 61–70.

Kelley, D. S., Karson, J. A., Früh-Green, G. L., Yoerger, D. R., Shank, T. M., Butterfield, D. A., et al. (2005). A serpentinite-hosted ecosystem: The lost city hydrothermal field. *Science*, *307*, 1428.

Kim, J., & Sukkarieh, S. (2004). Autonomous airborne navigation in unknown terrain environments. *Aerospace and Electronic Systems, IEEE Transactions on*, *40*, 1031 – 1045.

Lemaire, T., Berger, C., Jung, I.-K., & Lacroix, S. (2007). Vision-based slam: Stereo and monocular approaches. *International Journal of Computer Vision*, *74*, 343–364.

Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., et al. (2000). The Digital Michelangelo project: 3D scanning of large statues. *ACM SIGGRAPH: Computer Graphics and Interactive Techniques*, 131 – 144.

Liou, D., Huang, Y., & Reynolds, N. (1990). A new microcomputer based imaging system with c3 technique. *TENCON 90. 1990 IEEE Region 10 Conference on Computer and Communication Systems*, 555 – 559 vol.2.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*.

Lucas, B., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)* (p. 674-679).

Ma, X., Wu, J., & Shen, X. (2007). A new progressive mesh with adaptive subdivision for lod models. *LECTURE NOTES IN COMPUTER SCIENCE*.

Mahon, I. (2008). *Vision-based navigation for autonomous underwater vehicles.* Unpublished doctoral dissertation, University of Sydney.

Mahon, I., Williams, S., Pizarro, O., & Johnson-Roberson, M. (2008). Efficient view-based slam using visual loop closures. *Robotics, IEEE Transactions on*, *24*, 1002 – 1014.

Marks, T., Howard, A., Bajracharya, M., Cottrell, G., & Matthies, L. (2008). Gamma-slam: Using stereo vision and variance grid maps for slam in unstructured environments. *International Conference on Robotics and Automation*, 3717 – 3724.

Matthies, L., & Shafer, S. (1987). Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, *3*(3), 239–248.

Negahdaripour, S., & Firoozfam, P. (2006). An ROV stereovision system for ship-hull inspection. *IEEE Journal of Oceanic Engineering*, *31*(3), 551–564.

Negahdaripour, S., & Madjidi, H. (2003, October). Stereovision imaging on submersible platforms for 3-d mapping of benthic habitats and sea-floor structures. *IEEE Journal of Oceanic Engineering*, *28*(4), 625–650.

Negahdaripour, S., Xu, X., & Jin, L. (1999, July). Direct estimation of motion from sea floor images for automatic station-keeping of submersible platforms. *IEEE Journal of Oceanic Engineering*, *24*(3), 370-382.

Negahdaripour, S., & Xun, X. (2002, January). Mosaic-Based Positioning and Improved Motion-Estimation Methods for Automatic Navigation of Submersible Vehicles. *IEEE Journal of Oceanic Engineering*, *27*(1), 79–99.

Nicosevici, T., & Garcia, R. (2008). Online robust 3D mapping using structure from motion cues. In *MTS/IEEE Oceans* (pp. 1 – 7).

NVIDIA, C. (2008). *NVIDIA texture tools 2.* (http://developer.nvidia.com/object/texture_tools.html)

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., & Seidel, H.-P. (2003). Multi-level partition of unity implicits. *ACM Transactions on Graphics*, *22*(3), 463–470.

Oka, M., Tsutsui, K., Ohba, A., Kurauchi, Y., & Tago, T. (1987). Real-time manipulation of texture-mapped surfaces. In *ACM SIGGRAPH: Computer Graphics and Interactive Techniques* (Vol. 21, pp. 181–188). New York, NY, USA: ACM.

Openscenegraph. (n.d.). `http://www.openscenegraph.org/projects/osg`.

Pizarro, O., Eustice, R., & Singh, H. (2004). Large area 3D reconstructions from underwater surveys. In *MTS/IEEE Oceans* (Vol. 2, pp. 678 – 687).

Pizarro, O., & Singh, H. (2003). Toward large-area mosaicing for underwater scientific applications. *IEEE Journal of Oceanic Engineering*, *28*(4), 651–672.

Pollefeys, M., Koch, R., Vergauwen, M., & Gool, L. V. (2000). Automated reconstruction of 3d scenes from sequences of images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *55*, 251–267.

Pollefeys, M., Nister, D., Frahm, J. M., Akbarzadeh, A., Mordohai, P., Clipp, B., et al. (2008). Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, *78*, 143–167.

Ramos, F., Chover, M., Ripolles, O., & Granell, C. (2006). Continuous level of detail on graphics hardware. *Lecture notes in computer science*, *4245*, 460 – 469.

Rekleitis, I., Bedwani, J., & Dupuis, E. (2007). Over-the-horizon, autonomous navigation for planetary exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2248–2255).

Roman, C., & Singh, H. (2007). A self-consistent bathymetric mapping algorithm. *Journal of Field Robotics*, *24*(1-2), 23-50.

Saez, J., Hogue, A., Escolano, F., & Jenkin, M. (2006). Underwater 3D SLAM through entropy minimization. In *International Conference on Robotics and Automation* (pp. 3562–3567).

Sawhney, H., Hsu, S., & Kumar, R. (1998). Robust video mosaicing through topology inference and local to global alignment. In *European conference on computer vision* (p. 103-119). Freiburg, Germany.

Sawhney, H., & Kumar, R. (1999, March). True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(3), 235–243.

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*, 7–42.

Segal, M., & Akeley, K. (2003). The OpenGL Graphics System: A Specification (Version 1.5). *URL: http://www.opengl.org/documentation/specs/version1.5/glspec15.pdf*.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, *1*, 519 – 528.

Shlyakhter, I., Rozenoer, M., Dorsey, J., & Teller, S. (2001). Reconstructing 3d tree models from instrumented photographs. *Computer Graphics and Applications, IEEE*, *21*, 53–61.

Singh, H., Can, A., Eustice, R., Lerner, S., McPhee, N., Pizarro, O., et al. (2004). SeaBED AUV offers new platform for high-resolution imaging. *Eos, Transactions, American Geophysical Union*, *85*(31), 289,294-295.

Singh, H., Eustice, R., Roman, C., Pizarro, O., Armstrong, R., Gilbes, F., et al. (2004, January). Imaging coral I: Imaging coral habitats with the SeaBED AUV. *Subsurface Sensing Technologies and Applications*, *5*(1), 25–42.

Singh, H., Howland, J., & Pizarro, O. (2004). Advances in large-area photomosaicking underwater. *IEEE Journal of Oceanic Engineering*, *29*(3), 872–886.

Singh, H., Roman, C., Pizarro, O., Eustice, R., & Can, A. (2007). Towards high-resolution imaging from underwater vehicles. *The International Journal of Robotics Research*, *26*(1), 55.

Sinha, S., Frahm, J.-M., Pollefeys, M., & Genc, Y. (2007). Feature tracking and matching in video using programmable graphics hardware. *Machine Vision and Applications*.

Smits, B. (1999). Efficiency issues for ray tracing. *Journal of Graphics Tools*, *6*, 1–14.

Steder, B., Grisetti, G., Grzonka, S., Stachniss, C., Rottmann, A., & Burgard, W. (2007). Learning maps in 3D using attitude and noisy vision sensors. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 644 – 649.

Strengert, M., Kraus, M., & Ertl, T. (2006). Pyramid methods in gpu-based image processing. *Proceedings of the Vision, Modeling, and Visualization*, 169–176.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., et al. (2006). Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, *23*(9), 661–692.

Thrun, S., Thayer, S., Whittaker, W., Baker, C., Burgard, W., Ferguson, D., et al. (2004). Autonomous exploration and mapping of abandoned mines. *Robotics & Automation Magazine, IEEE*, *11*, 79 – 91.

Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography - a factorization method. *International Journal of Computer Vision*, *9*, 137–154.

Turk, G., & Levoy, M. (1994). Zippered polygon meshes from range images. In *ACM SIGGRAPH: Computer Graphics and Interactive Techniques* (pp. 311–318).

Uyttendaele, M., Eden, A., & Skeliski, R. (2001). Eliminating ghosting and exposure artifacts in image mosaics. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, *2*, II-509 - II–516 vol.2.

Walter, M., Hover, F., & Leonard, J. (2008). SLAM for ship hull inspection using exactly sparse extended information filters. In *International Conference on Robotics and Automation* (pp. 1463 – 1470).

Webster, J. M., Beaman, R. J., Bridge, T., Davies, P. J., Byrne, M., Williams, S. B., et al. (2008). From corals to canyons: The Great Barrier Reef margin. *Eos, Transactions, American Geophysical Union*, *89*, 217.

Williams, L. (1983). Pyramidal parametrics. *ACM SIGGRAPH Computer Graphics*, *17*, 1 – 11.

Williams, S., & Mahon, I. (2004). Simultaneous Localisation and Mapping on the Great Barrier Reef. In *International Conference on Robotics and Automation* (Vol. 2, p. 1771-

1776).

Williams, S., Pizarro, O., Jakuba, M., & Barrett, N. (2009). AUV benthic habitat mapping in south eastern Tasmania. In *International Conference on Field and Service Robotics.* inpress.

Williams, S., Pizarro, O., Webster, J., Beaman, R., Johnson-Roberson, M., Mahon, I., et al. (2008). Auv-assisted surveying of relic reef sites. In *MTS/IEEE Oceans.*

Yoerger, D. R., Jakuba, M., Bradley, A. M., & Bingham, B. (2007). Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *International Journal of Robotics Research*, *26*, 41–54.

Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. *Academic Press Graphics Gems IV*, 478–485.