sensors

MDPI

Article

Real-Time Underwater StereoFusion

Matija Rossi [1,*], Petar Trslić [1], Satja Sivčev [1], James Riordan [2], Daniel Toal [1] and Gerard Dooly [1]

[1] Centre for Robotics & Intelligent Systems, University of Limerick, Limerick V94 T9PX, Ireland; petar.trslic@ul.ie (P.T.); satja.sivcev@ul.ie (S.S.); daniel.toal@ul.ie (D.T.); gerard.dooly@ul.ie (G.D.)

[2] School of Computing, Engineering, and Physical Sciences, University of the West of Scotland, Glasgow G72 0AG, UK; james.riordan@uws.ac.uk

* Correspondence: matija.rossi@ul.ie; Tel.: +353-61-213-102

Received: 14 September 2018; Accepted: 8 November 2018; Published: 14 November 2018

Abstract: Many current and future applications of underwater robotics require real-time sensing and interpretation of the environment. As the vast majority of robots are equipped with cameras, computer vision is playing an increasingly important role it this field. This paper presents the implementation and experimental results of underwater StereoFusion, an algorithm for real-time 3D dense reconstruction and camera tracking. Unlike KinectFusion on which it is based, StereoFusion relies on a stereo camera as its main sensor. The algorithm uses the depth map obtained from the stereo camera to incrementally build a volumetric 3D model of the environment, while simultaneously using the model for camera tracking. It has been successfully tested both in a lake and in the ocean, using two different state-of-the-art underwater Remotely Operated Vehicles (ROVs). Ongoing work focuses on applying the same algorithm to acoustic sensors, and on the implementation of a vision based monocular system with the same capabilities.

Keywords: stereo; underwater; ROV; GPU; real-time; 3D; fusion; camera; tracking; vision

1. Introduction

Applications of computer vision are rapidly growing across a wide spectrum of underwater operations. Vision systems are increasingly being used as the primary tool for inspection of underwater sites, in disciplines ranging from archaeology [1] and biology [2], to offshore engineering [3] and pipeline inspection [4]. This has been facilitated by the increasing industry adoption of remotely operated vehicles (ROV) and autonomous underwater vehicles (AUV) [5–7], which opens the door to many new applications for machine vision. A common task is robot navigation, for which underwater is challenging for many reasons, such as the lack of radio communications, including global navigation satellite systems (GNSS), and limited sensing technology compared to land or airborne vehicles. For this purpose, camera and acoustic sensor systems can be used to implement simultaneous localisation and mapping (SLAM) algorithms to complement inertial navigation systems (INS), which inevitably suffer from drift. If such algorithms prove sufficiently robust, vision systems may obviate the need for inertial navigation systems and replace them with image-based target referenced navigation [8].

An even more demanding task is robotic intervention, where work class ROVs equipped with underwater manipulators have traditionally been teleoperated from support vessels by human operators. Significant effort is currently being put towards the automation of such operations using computer vision [9–12]. In order for an intervention task to be carried out autonomously, it is necessary to know the structure of the scene around the target and the position of the robot relative to it. This makes it possible to then implement higher level features such as path planning, obstacle avoidance, and target identification. Additionally, even in the case of manual operations, providing an

augmented feedback could increase the ROV pilot's efficiency multiple times compared to a standard 2D camera stream, which is what is currently being used for teleoperation of manipulators. Due to offshore operations being particularly expensive, time consuming, and limited by other factors such as weather, making them more efficient is of great value [13].

The described scenarios would significantly benefit from, or even require, models of the underwater operating environment generated in real time [14]. Another common example is survey data which is usually post-processed, and is acquired without real-time feedback about its quality. This leads to situations where defects in the data, such as areas of interest not being fully covered, are discovered only after the survey.

This paper presents an underwater StereoFusion algorithm based on KinectFusion [15,16] as a reliable solution to the described requirements, capable of real-time dense 3D reconstruction and localisation using an underwater stereo camera. The main contributions of this work are the implementation of StereFusion, its application on both a custom-built and a commercial ROV, and its testing in fresh water and in the ocean, under very different visibility conditions.

Section 2 provides a brief background of previous work relevant to the development of the described system. Section 3 describes the algorithm itself, while hardware and software implementation details are described in Section 4. Section 5 presents the results from two separate underwater trials, the first one in a fresh water lake and the second one offshore, in the Atlantic ocean. The paper concludes with a summary and discussion of ongoing work in Section 6.

2. Background

Reconstruction of 3D geometry using multiple camera images is a well established area of research [17], popular for a variety of applications. Dense real-time 3D reconstruction is becoming feasible only recently, with the advent of widely available massively parallel commodity general purpose computing on graphics processing units (GPGPU). Notable steps towards real-time 3D reconstruction come from Simultaneous Localisation and Mapping (SLAM) techniques [18], such as MonoSLAM [19] for single camera visual SLAM using sparse features, and the real-time visual odometry system from [20]. Vision-based SLAM has been successfully used underwater, both using monocular [21] and stereo cameras [22].

A different approach from the filter based SLAM systems that preceded it was presented in [23], where the authors separated the camera tracking given a known map from the map update. Following the work of [24], which applied easily parallelisable convex optimisation techniques on commodity GPUs for real-time computer vision applications such as image denoising, Refs. [25,26] presented real-time dense 3D reconstruction pipelines using the feature-based [23] for tracking. The first system to use both dense tracking and mapping and capable of real-time processing was presented by [27].

The work presented in this paper is based on KinectFusion [15,16], a very popular real-time surface reconstruction and camera tracking algorithm designed for RGB + Depth (RGBD) sensors such as the Microsoft Kinect. Although some attempts have been made to use such sensors underwater [28,29], they are extremely limited under such conditions. In order to make the system robust and applicable to existing ROV equipment, this work relies on the application of the KinectFusion algorithm to a stereo camera setup instead of an active sensor, hence the name StereoFusion, as proposed by [30]. Stereo vision has been successfully used underwater for mapping and navigation by various authors [31–33].

Alternative methods using acoustic sensors instead of cameras have been explored [34,35]. Most sonars are however not suitable for high precision close range applications due to their low resolution and relatively high minimum operating range. A hybrid vision-acoustic approach has been proposed by [36] using a high-end commercial 3D sonar, which aims to provide the robustness of sonars with the colour information and accuracy of cameras. Such a sonar could be used on its own in a similar way to a depth camera, except for the absence of colour information. As part of an ongoing research project, the implementation of this type of system is currently being trialled with

promising results, but its applications are limited due to the previously mentioned limitations of currently available sonars.

3. Algorithm

The base algorithm used in this work is an implementation of KinectFusion [15], a real-time 3D mapping and tracking method developed for use with the Microsoft Kinect and similar RGBD sensors. Although there has been some research done on using range sensors in water [37,38], the results have been of very limited practical use due to difficulties in dealing with light refraction and attenuation, which means that even in good visibility the maximum achievable range is about 20 cm. To overcome this limitation, the work presented in this paper relies on two synchronised colour cameras producing a stereoscopic image pair for disparity estimation. Figure 1 shows the overall workflow of the StereoFusion algorithm.

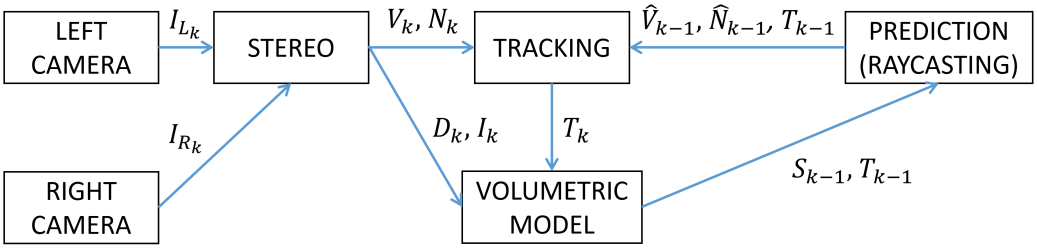


Figure 1. StereoFusion workflow for iteration k of the algorithm. Here, I_L and I_R refer to the stereo pair of RGB images; D and I are the current depth map and RGB image; V and N are the vertex map and normal map computed from the depth map; $\hat{V}$ and $\hat{N}$ are the predicted vertex and normal maps; T is the camera position; S is the Truncated Signed Distance Functions (TSDF).

3.1. Stereo

Given a pair of rectified [39] left and right images I_L and I_R, the objective is to find a disparity map [40] A_L that provides correspondences for as many pixels as possible:

$$I_L(u,v) \approx I_R(u + A_L(u,v), v). \tag{1}$$

This is achieved using a basic block matching algorithm [40]. For each pixel (u,v), a Sum of Absolute Differences (SAD) is computed between that pixel's region (the template), and a series of regions in the right image:

$$\mathrm{SAD}(u,v,a) = \sum_{i=-N_T}^{N_T} \sum_{j=-N_T}^{N_T} \left| I_L(u+i, v+j) - I_R(u+i+a, v+j) \right|, \tag{2}$$

where $2N_T + 1$ is the template size and $a \in \alpha \subset \mathbb{R}$ the disparity value for pixel (u,v) in range α. The search is limited to one dimension (u) thanks to the epipolar constraint [17]. The pixel's disparity is then determined by finding the minimum of the SAD:

$$A_L(u,v) = \min_a \{\mathrm{SAD}(u,v,a)\}. \tag{3}$$

This method has been chosen for its computational efficiency. As analysed in [40], the complexity of basic implementations of block matching algorithms is $\mathcal{O}(N_U N_A N_T)$, where N_U is the number of pixels in the image and N_A the size of the disparity search range α. By avoiding repeating redundant computations, the complexity can be reduced to $\mathcal{O}(N_U N_A)$, thus eliminating the influence of the template's size.

Once the disparity map A_L is known, a range image, or depth map D_L can be computed as:

$$D_L(u,v) = f\frac{B}{A_L(u,v)}, \tag{4}$$

where f is the camera's focal length and B is the baseline, i.e., the spacing between the optical centres of the left and the right camera.

3.2. Volumetric Model Representation

The 3D reconstruction is stored as a dense voxel volume, where each voxel contains a Signed Distance Function (SDF) describing its distance to a surface, as described in [41]. The SDF values are positive in front of the surface and negative behind it. Surfaces are therefore extracted from the volume by finding the SDF zero crossings, i.e., the set of points in which the SDF equals zero. In practice, only a truncated version $S(p)$ of the SDF (TSDF) is stored for each point $p \in \mathbb{R}^3$, such that the true SDF value is only stored within $\pm\mu$ of the measured value, thus representing the measurement's uncertainty. The TSDF is represented in each point p as:

$$S(p) = \left[F(p), W^F(p), C(p), W^C(p)\right], \tag{5}$$

where $F(p)$ is its value and $W^F(p)$ its weight, the computation of which is described in Equation (7).

Additionally, colour information is stored as $C(p)$ and its corresponding weight $W^C(p)$, in order to be able to make photometric predictions along with the geometric ones.

For a depth map D_k with a known pose T_k, the TSDF at point p is computed as:

$$\begin{aligned} F_{D_k}(p) &= \Psi\left(\lambda^{-1}\left\|t_k - p\right\| - D_k(p_c)\right), \\ \Psi(y) &= \begin{cases} \min\left(1, \frac{y}{\mu}\right), & \text{if } y \geq -\mu, \\ \text{null}, & \text{otherwise}, \end{cases} \end{aligned} \tag{6}$$

where $\lambda = \left\|K^{-1}p_c\right\|$ is a scaling factor for each pixel ray, K is the intrinsic matrix, p_c is the projection of point p onto the camera's image plane, and t_k is the translation vector from T_k.

For each new depth map D_k, the global TSDF at point p can iteratively be updated as a moving average defined by a threshold W^F_η:

$$\begin{aligned} F_k(p) &= \frac{W^F_{k-1}(p)F_{k-1}(p) + F_{D_k}(p)}{W^F_{k-1}(p) + 1}, \\ W^F_k(p) &= \min\left(W^F_{k-1}(p) + 1,\, W_\eta\right). \end{aligned} \tag{7}$$

Compared to a simple running average, this moving average method is robust to dynamic object motion in the scene. The colour TSDF components $C_k(p)$ and $W^C_k(p)$ can be updated in the same manner.

3.3. Camera Pose Estimation

This section briefly describes the two tracking methods that have been used within the scope of this work.

3.3.1. Depth Tracking

As presented in [15], depth tracking performs camera tracking exclusively based on the depth map from the stereo camera. The newly obtained depth measurements D_k are first transformed to a surface measurement composed of a vertex map V_k and a normal map N_k. A surface prediction

$(\hat{V}_{k-1}, \hat{N}_{k-1})$ is generated by raycasting from a viewpoint corresponding to the last known camera position T_{k-1}.

An ICP algorithm [42] is used in order to estimate a transformation T_k, which maps the camera coordinate frame at step k to the global frame. It begins by matching points from the surface prediction with the live surface measurement, as detailed in [16]. Given a set of corresponding points, each iteration of the ICP produces a transformation T_k minimising a point to plane objective function [43]:

$$\min_{T_k} \sum_{p_c} \left\| \left(T_k V_k(p_c) - \hat{V}_{k-1}(p_c) \right)^{\mathrm{T}} \hat{N}_{k-1}(p_c) \right\|, \quad \forall D_k(p_c) > 0. \tag{8}$$

Assuming small motion between frames, the minimisation is solved according to [44].

3.3.2. Colour Tracking

The colour tracker, as described in [45], relies on the live RGB image I_k, rather than on the depth map D_k. The first step of the tracker consists of creating a model based prediction of surface points $\hat{V}_{k-1}$ and their corresponding colour values $\hat{C}_{k-1}$. The prediction is again done from a viewpoint corresponding to the camera position at step $k-1$, like for the depth tracker described in Section 3.3.1. The cost function to be minimised in this case, however, is an ℓ^2 norm of the difference between the colour of a predicted point and the colour of the corresponding pixel in I_k:

$$\min_{T_k} \sum_{p_c} \left\| I_k \left(K T_k \hat{V}_{k-1}(p_c) \right) - \hat{C}_{k-1}(p_c) \right\|. \tag{9}$$

The minimisation is solved using the Levenberg–Marquardt algorithm [46].

3.4. Raycasting

Raycasting [47] is used to obtain surface predictions both for camera tracking, as seen in Section 3.3, and for visualisation of the model. The process renders the zero level set $F_k(p) = 0$ of the current TSDF into a viewpoint with position T_r.

A virtual ray $T_r K^{-1} p_c$ is generated for each pixel p_c of the image being rendered. The algorithm steps through the volume along each ray, looking for a change in the sign of the TSDF values. If the TSDF values have changed from positive to negative, a surface interface has successfully been detected, which provides the data for the pixel being rendered.

4. Implementation

4.1. Software

The software implementation is based on the InfiniTAM [48] framework, adapted to work with the Robot Operating System (ROS) [49] and a stereo camera as the input device. All steps of the KinectFusion algorithm are very well suited for parallel execution, as described in [16]. For this reason, all computation is performed on a Graphics Processing Unit (GPU). Modern consumer grade GPUs typically consist of several hundreds or thousands of processing units. From a general purpose programming point of view, GPUs can be considered SIMD (single instruction, multiple data) devices. This means that optimal performance is achieved in cases when the same computation needs to be performed on a large number of inputs. Every StereoFusion step can be parallelised in such manner, performing the same operation either for each image pixel or for each voxel in the model volume, making it perfectly suited for this type of hardware. For the purpose of this work, everything has been implemented using Nvidia CUDA. The test GPU is an Nvidia GTX 980 Ti, which has 2816 CUDA cores and 6 GB of global memory. Other relevant components of the computer used are an Intel i7-4930k processor and 16 GB of RAM. The software has been running on Ubuntu 16.04 with ROS Kinetic Kame.

4.2. Stereo Rig

The algorithm described in Section 3 has been tested using a stereo RGB camera rig. It consists of a pair of FLIR (formerly Point Grey) Blackfly GigE Vision cameras (BFLY-PGE-23S6C-C), with global shutter Sony IMX249 1/1.2" CMOS sensors. These are standard industrial machine vision cameras enclosed in underwater housings rated to a water depth of 1000 m. The camera and the housing can be seen in Figure 2. The stereo rig was mounted on the front of two ROVs, as shown in Figures 3 and 4. The shutters are synchronised through the cameras' GPIO (General-Purpose Input/Output) pins as described in [50], where one camera emits a signal that triggers the second one, without the need of an external signal generator. This is necessary in order to accurately estimate disparities from a stereo image when motion is involved.

Figure 2. Camera and underwater housing.

All the processing is performed on the surface, using a dedicated computer, described in Section 4.1, located in the ROV control cabin. In order to communicate with the surface PC through the two kilometres long ROV umbilical, the Gigabit Ethernet connections are transformed from copper to optical fibre connections and back to copper using TP-Link Gigabit SFP Media Converters. Both cameras rely on Power over Ethernet (PoE), which is injected to the Gigabit Ethernet lines on-board the ROV.

Two Kowa LM6HC wide angle lenses are used in order to have the widest possible field of view (FOV). Despite the lenses, the FOV would be reduced due to light refraction if the cameras were behind flat ports, for this reason both enclosures use dome ports. Having a wide FOV allows a relatively large stereo baseline (40 cm in this case) without sacrificing close range measurements. It is also important for guaranteeing overlap between consecutive images, and aiding the vision-based tracking algorithm by reducing the probability of completely losing sight of the observed scene.

5. Results

The algorithm discussed in Section 3 has been tested on two ROV systems under different conditions. In both cases, the cameras produced RGB images with a resolution of 960×600 pixels at 22 frames per second. The frames are processed in real time and for all the presented results the 3D models are being built and updated on-line. The voxel size used for all the presented models is 5 millimetres.

5.1. Good Visibility, Fresh Water

The main results presented in this work have been obtained during trials with an inspection class ROV in a flooded quarry near Portroe, County Tipperary, Ireland. This quarry is normally used as a scuba diving centre. Various items have been placed underwater for the entertainment of divers (e.g., a van, a boat, a car, a bar, computers, etc.). These are interesting targets to test the described algorithms, thanks to their complex geometry and texture. Additionally, using familiar targets facilitates qualitative analysis of the 3D models.

The vehicle used, called ROV Áed and shown in Figure 3, is a custom system built in-house at the Centre for Robotics and Intelligent Systems (CRIS) at the University of Limerick (UL). It is intended to be a lightweight highly manoeuvrable inspection ROV capable of operating in strong currents and other challenging environments.

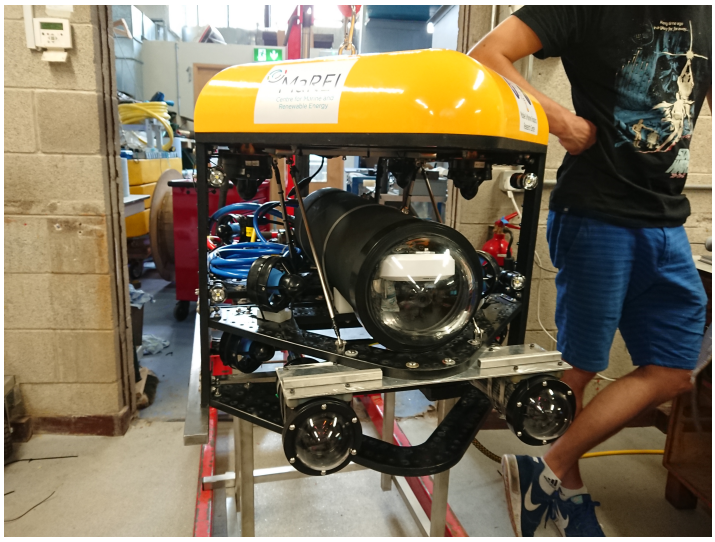


Figure 3. University of Limerick ROV Áed with the stereo camera setup mounted below the main piloting camera.

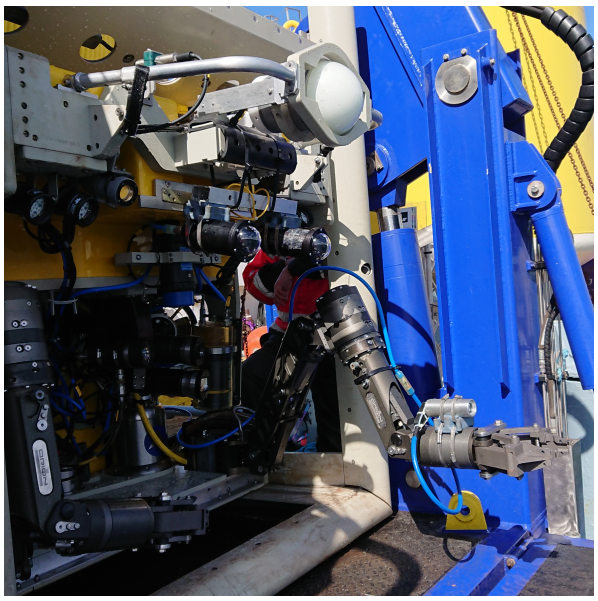
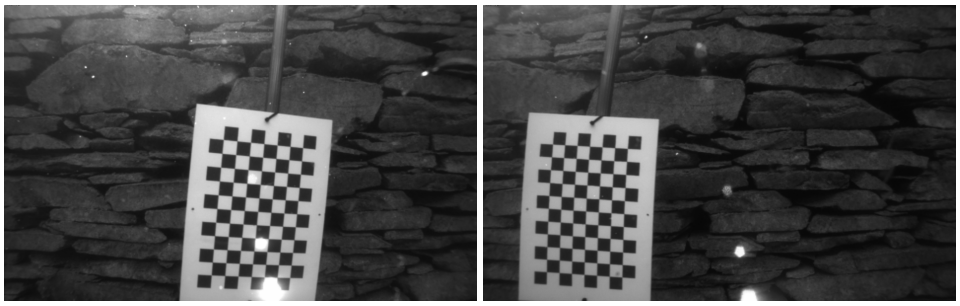


Figure 4. University of Limerick ROV Étaín with the stereo camera setup mounted above the manipulators.

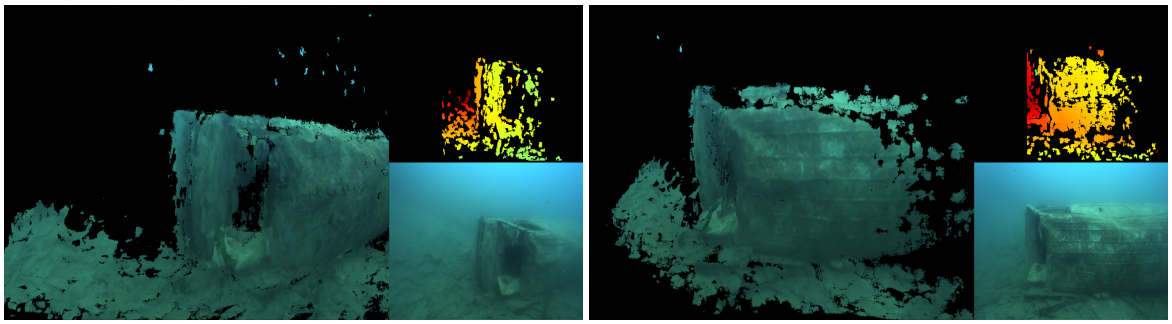
The cameras have been calibrated underwater on-site, using a 7×11 chessboard with the sides of each square measuring 280 mm, as shown in Figure 5. The chessboard has been printed on an A3 sized (297×420 mm) PVC board, which was then attached on a pole in order to submerge it and move it manually from the side of a pier. The cameras have been calibrated using a pinhole camera model, and the distortion corrected using the plumb bob model (radial polynomial + thin prism model) [51].



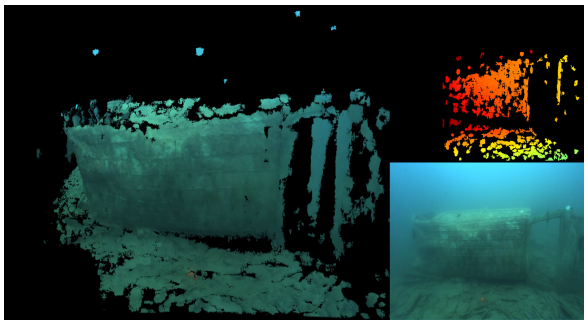
(**a**) (**b**)

Figure 5. Example of a stereo pair used for camera calibration. (**a**) left camera; (**b**) right camera.

Figure 6 shows a series of images from a survey of a submerged van, about 4 m in length. It displays the 3D model, the current frame from the left camera of the stereo pair, and the depth map calculated in the left camera's reference frame. The model has been built incrementally, as the ROV was manually piloted around the van. The camera position was continuously estimated using the depth tracking method described in Section 3.3.1.



(**a**) Van survey using depth tracking (**b**) Van survey using depth tracking



(**c**) Van survey using depth tracking

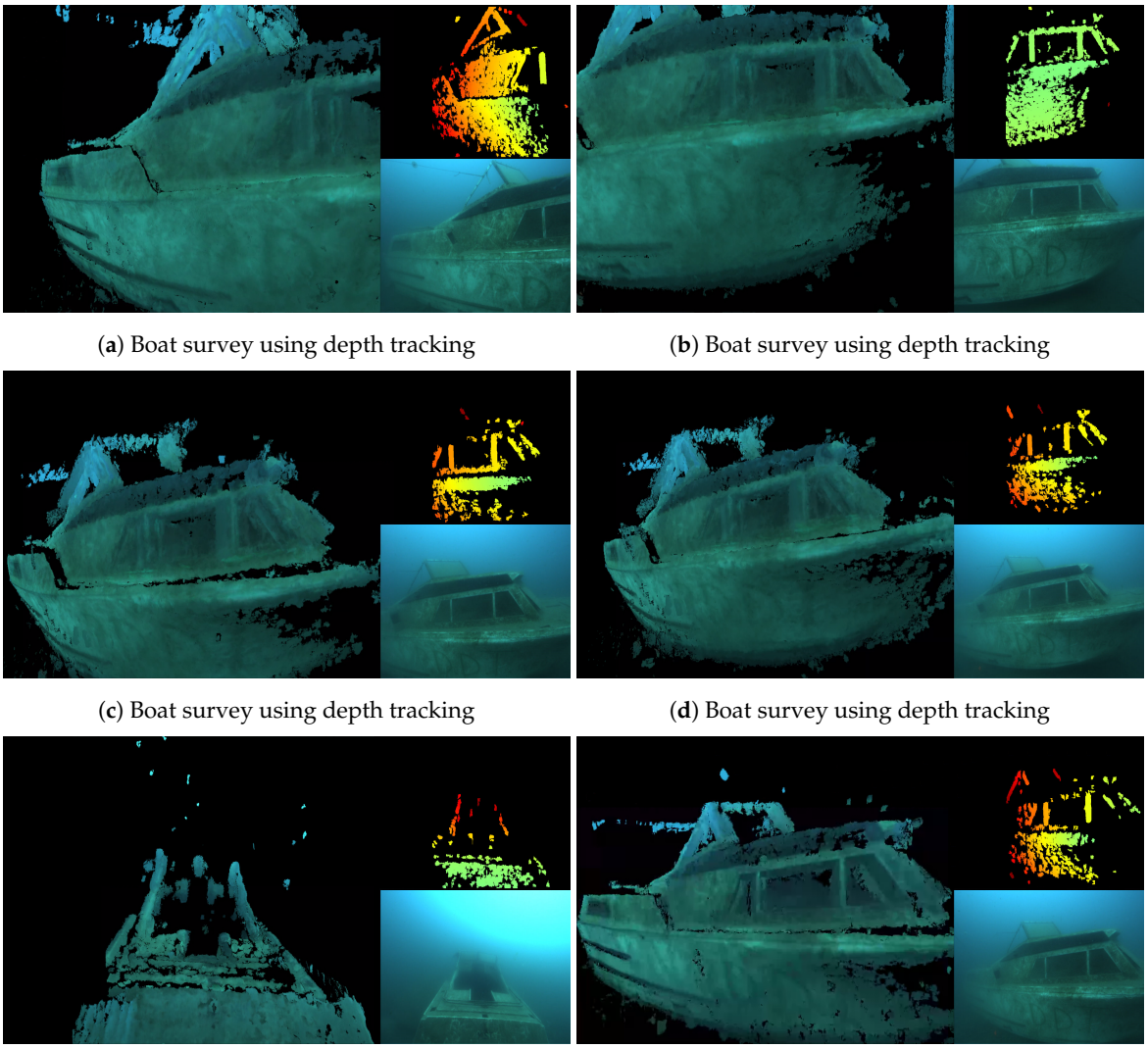
Figure 6. Van reconstruction in good visibility. The main panel shows the 3D model, the top right shows the depth maps, and the bottom right shows the original colour image from the left camera.

Figure 7a–e show images of the algorithm running during a survey of a submerged boat, about 6 m in length. Figure 7f is from a different survey of the same boat, but this time using the colour tracker from Section 3.3.2. In the quarry, both trackers performed equally well, producing good reconstructions and providing reliable camera tracking.

5.2. Bad Visibility, Sea Water

Testing has been carried out at sea in Galway Bay, Ireland, with the stereo rig mounted on the UL work class ROV Étaín. This ROV is a commercial Sub-Atlantic Comanche system, it was launched from the Irish Lights Vessel (ILV) Granuaile, shown in Figure 8. The target surveyed in this scenario is a 2 metre tall metal frame with panels that simulate valves, which is used to test manipulator control

algorithms. Visibility in this case was low, with a lot of particles being moved around by the strong tidal current, as can be seen in Figure 9.



(**a**) Boat survey using depth tracking (**b**) Boat survey using depth tracking

(**c**) Boat survey using depth tracking (**d**) Boat survey using depth tracking

(**e**) Boat survey using depth tracking (**f**) Boat survey using colour tracking

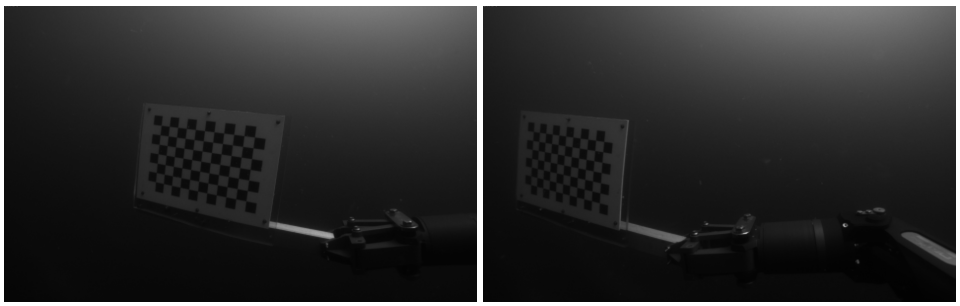
Figure 7. Boat reconstruction in good visibility.

Figure 8. ROV Étaín inside its Tether Management System, being deployed from the ILV Granuaile.

Figure 9. The stereo rig (red) on the ROV in bad visibility conditions.

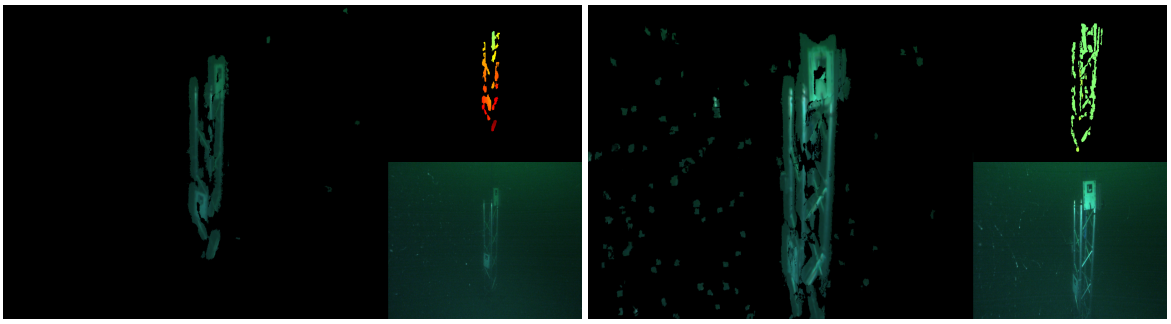
The stereo cameras have been calibrated underwater using the same chessboard and models as described in Section 5.1. In this case, however, due to physical constraints and safety considerations, the chessboard could not be moved manually from the side of the ship, therefore it had to be operated by one of the two Schilling Orion 7P hydraulic manipulators that are on board the ROV, as shown in Figure 10. Obtaining the necessary images, this way proved to be very challenging and time consuming for the pilot, mostly due to the manipulator's inaccurate control and limited motion capabilities. A consideration for future operations would be to automate the calibration procedure by predefining the manipulator's trajectory using the Cartesian control presented in [10].

Figure 11 shows snapshots of the survey, which was performed by manually piloting the ROV in a circle around the target. Unlike in the scenario described in Section 5.1, in this case, the colour tracker proved to have a significant advantage over the depth tracker. This is due to the target's geometry, i.e., the beams from which the metal frame is composed are relatively thin and do not always appear clearly in the depth map.

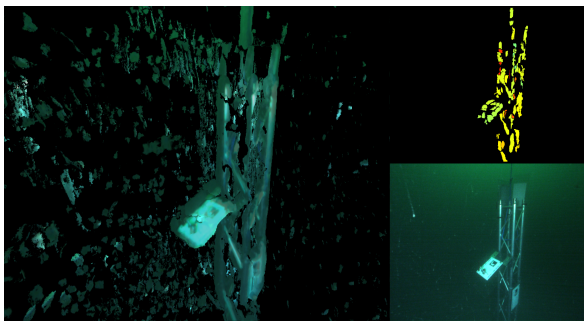


(**a**) (**b**)

Figure 10. Example of a stereo pair used for camera calibration with the manipulator. (**a**) left camera; (**b**) right camera.



(**a**) Frame survey using colour tracking (**b**) Frame survey using colour tracking



(**c**) Frame survey using colour tracking

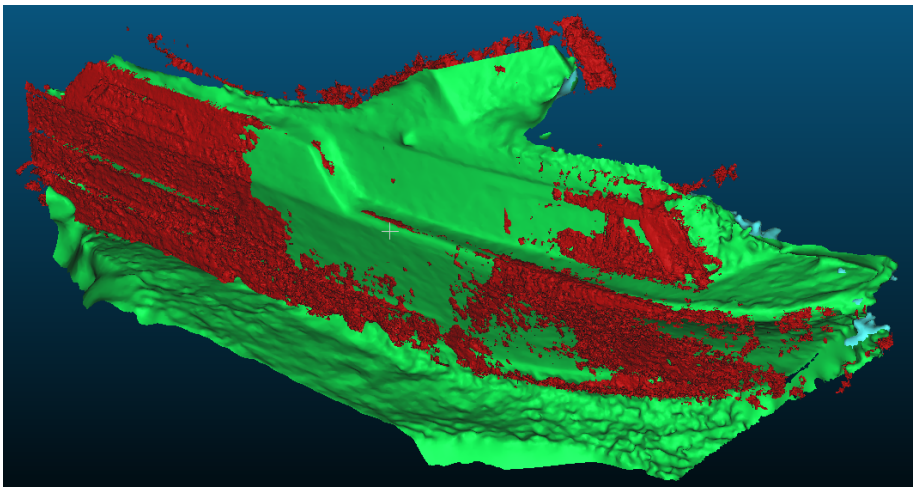
Figure 11. Metal frame reconstruction in bad visibility. The main panels show the 3D model, the top right show the range images, and the bottom right show the original colour image from the left camera. (**a**) approaching the target; (**b**) target approached; (**c**) after moving 90° clockwise around the target.

The effect of the floating particles is visible in the reconstruction, especially in Figure 11c, which is filled with tiny artefacts. Apart from the effects on the 3D model itself, this poses a serious challenge for camera tracking, which in this case was pushed to its limits. Due to frequent tracking failure, the survey had to be performed several times before a full circle around the target was successfully accomplished. Although a model has been obtained, it is clear that a purely vision based system is not robust enough for reliable operation under such challenging conditions.

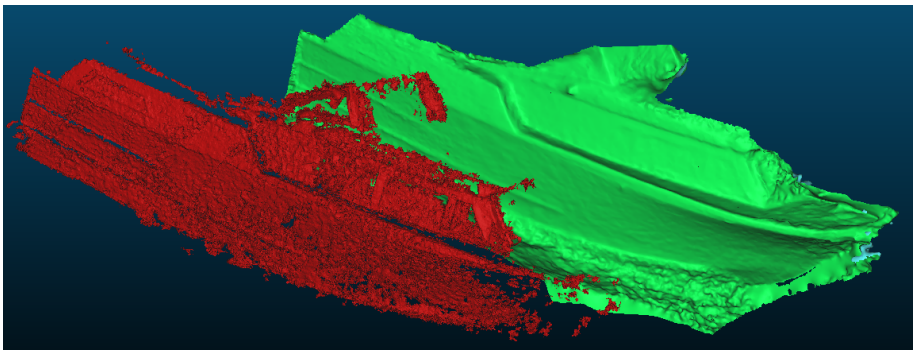
5.3. Qualitative Comparison to Post-Processed Photogrammetry

Although StereoFusion does not aim to challenge post-processing techniques in terms of reconstruction quality, but rather produce usable models in real time, a comparison is nonetheless useful in order to estimate its results. Figure 12 shows a qualitative comparison of the 3D model of the boat from Figure 7 with one obtained using the same image sequence with a post-processing photogrammetry technique. The post-processed model, displayed in Figure 12a, has been generated using Agisoft PhotoScan [52], a widely used commercial photogrammetry tool. Figure 12b presents a comparison of the two models aligned with each other, where the green one is the result of post-processing while the red one is produced in real-time by StereoFusion. For better understanding, both models are shown in Figure 12c side-by-side. From this comparison, it is clear that the geometry produced by StereoFusion matches the one obtained in post-processing, although not perfectly. It is important to note that the post-processed model is only used for qualitative comparison, as it does not represent ground truth. The exact geometry of the sunken boat used in this sequence is unknown. However, the targets presented in this work have been intentionally chosen because they are familiar objects, thus making a qualitative analysis of the presented results more intuitive to the reader.

(**a**)



(**b**)



(**c**)

Figure 12. Qualitative comparison between StereoFusion and post-processed photogrammetry. The software used for photogrammetry is Agisoft PhotoScan. (**a**) textured model obtained from the boat sequence using Agisoft PhotoScan; (**b**) overlapping models: the green one is built using Agisoft PhotoScan and the red one using StereoFusion; (**c**) models side-by-side: the green one is built using Agisoft PhotoScan and the red one using StereoFusion.

From these comparative results, StereoFusion can be considered applicable to the underwater domain. The presented real-time data could be useful in applications such as obstacle avoidance, path planning, target referenced localisation, object detection, and augmented reality for enhancing the pilot's perception.

6. Conclusions and Future Work

This paper presented underwater StereoFusion, an adaptation of KinectFusion which instead of a Kinect uses a stereo camera. The software has been tested in two underwater scenarios, using different vehicles. In good visibility, the system proved capable of reliable operation at video frame rate. Because it provides real-time dense 3D reconstruction of targets and relative camera tracking, it opens a variety of new possibilities in underwater robotics. Such a system can be used for applications such as online verification of survey data, navigation relative to a target, semi and fully autonomous manipulation, path planning, obstacle avoidance, etc.

As with most underwater vision systems, its main problem is operation in low visibility environments, as shown in Section 5.2. Although still capable of producing a 3D model, the quality of both the reconstruction and the tracking significantly decreases. One way to tackle this will be to aid the camera tracking by exploiting navigation data from the onboard inertial navigation system. Fusing inertial with vision-based navigation will improve both the estimation quality and the system's robustness. Additionally, the system can be made more robust to low visibility by using a 3D sonar, either instead of the stereo camera, or by fusing the acoustic with vision data in a manner similar to that described by [36]. Ongoing testing will provide information on the quality of a 3D sonar used as the only sensor, using the same algorithm described in this paper.

Further ongoing development of the vision-based system is focused on a monocular implementation, relying on the work of [27,53]. This is particularly relevant for underwater manipulation automation for multiple reasons. Existing manipulators in the global work class ROV fleet are typically equipped with a single camera mounted near the gripper (Figure 4), and are not suited to host a stereo rig due to its size. In addition to mechanical considerations, a fixed baseline stereo system is not well suited for operation at significantly different distances from the target. As discussed by the authors in [9], in order to automate manipulation tasks, it is necessary to reconstruct the scene from a distance while also being able to continuously keep track of the camera as the manipulator approaches its target. These limitations of stereo methods can be solved by monocular systems.

Author Contributions: M.R. developed and implemented the algorithm; M.R., P.T., S.S., J.R., and G.D. worked on hardware integration; M.R., P.T., S.S., G.D., J.R., and D.T. organised and contributed to the experimental trials; G.D. and D.T. contributed to research programme scope development, supervision, and funding acquisition; M.R. wrote the paper.

Funding: This publication has emanated from research supported by the Science Foundation Ireland under the MaREI Centre research programme (Grant No. SFI/12/RC/2302 and SFI/14/SP/2740). The MaREI project is also supported by the following industrial partners: Teledyne Reson, Teledyne BlueView, SonarSim, Resolve Marine Group, Shannon Foynes Port Company, and Commissioners of Irish Lights.

Conflicts of Interest: The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ROV	Remotely operated vehicle
AUV	Autonomous underwater vehicle
SLAM	Simultaneous localisation and mapping
GPU	Graphics processing unit
GPGPU	General purpose GPU
ICP	Iterative closest point
SDF	Signed distance function
TSDF	Truncated SDF

FOV	Field of view
RGB	Red, green, blue
RGBD	RGB + depth
CMOS	Complementary Metal-Oxide-Semiconductor

References

1. Chapman, P.; Bale, K.; Drap, P. We All Live in a Virtual Submarine. *IEEE Comput. Graph. Appl.* **2010**, *30*, 85–89, doi:10.1109/MCG.2010.20. [CrossRef] [PubMed]
2. Cocito, S.; Sgorbini, S.; Peirano, A.; Valle, M. 3-D reconstruction of biological objects using underwater video technique and image processing. *J. Exp. Mar. Biol. Ecol.* **2003**, *297*, 57–70, doi:10.1016/S0022-0981(03)00369-1. [CrossRef]
3. Negahdaripour, S.; Firoozfam, P. An ROV Stereovision System for Ship-Hull Inspection. *IEEE J. Ocean. Eng.* **2006**, *31*, 551–564, doi:10.1109/JOE.2005.851391. [CrossRef]
4. Ledezma, F.D.; Amer, A.; Abdellatif, F.; Outa, A.; Trigui, H.; Patel, S.; Binyahib, R. A Market Survey of Offshore Underwater Robotic Inspection Technologies for the Oil and Gas Industry. *Soc. Pet. Eng.* **2015**, doi:10.2118/177989-MS. [CrossRef]
5. Antonelli, G. *Underwater Robots*; Springer Tracts in Advanced Robotics; Springer: Berlin, Germany, 2014; Volume 96, doi:10.1007/978-3-319-02877-4.
6. Elvander, J.; Hawkes, G. ROVs and AUVs in support of marine renewable technologies. In Proceedings of the 2012 Oceans, Hampton Roads, VA, USA, 14–19 October 2012; pp. 1–6, doi:10.1109/OCEANS.2012.6405139. [CrossRef]
7. Allotta, B.; Conti, R.; Costanzi, R.; Fanelli, F.; Gelli, J.; Meli, E.; Monni, N.; Ridolfi, A.; Rindi, A. A Low Cost Autonomous Underwater Vehicle for Patrolling and Monitoring. *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.* **2017**, *231*, 740–749, doi:10.1177/1475090216681354. [CrossRef]
8. Ferreira, F.; Veruggio, G.; Caccia, M.; Bruzzone, G. A Survey on Real-Time Motion Estimation Techniques for Underwater Robots. *J. Real-Time Image Process.* **2016**, *11*, 693–711, doi:10.1007/s11554-014-0416-z. [CrossRef]
9. Sivčev, S.; Rossi, M.; Coleman, J.; Dooly, G.; Omerdić, E.; Toal, D. Fully Automatic Visual Servoing Control for Work-Class Marine Intervention ROVs. *Control Eng. Pract.* **2018**, *74*, 153–167, doi:10.1016/j.conengprac.2018.03.005. [CrossRef]
10. Sivčev, S.; Coleman, J.; Adley, D.; Dooly, G.; Omerdić, E.; Toal, D. Closing the Gap between Industrial Robots and Underwater Manipulators. In Proceedings of the OCEANS 2015-MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–7, doi:10.23919/OCEANS.2015.7404563. [CrossRef]
11. Cieslak, P.; Ridao, P.; Giergiel, M. Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 529–536, doi:10.1109/ICRA.2015.7139230. [CrossRef]
12. Ribas, D.; Ridao, P.; Turetta, A.; Melchiorri, C.; Palli, G.; Fernandez, J.; Sanz, P. I-AUV Mechatronics Integration for the TRIDENT FP7 Project. *IEEE/ASME Trans. Mechatron.* **2015**, *PP*, 1–10, doi:10.1109/TMECH.2015.2395413. [CrossRef]
13. Omerdic, E.; Toal, D. OceanRINGS: System concept & applications. In Proceedings of the 2012 20th Mediterranean Conference on Control Automation (MED), Barcelona, Spain, 3–6 July 2012; pp. 1391–1396, doi:10.1109/MED.2012.6265833. [CrossRef]
14. Rossi, M.; Scaradozzi, D.; Drap, P.; Recanatini, P.; Dooly, G.; Omerdić, E.; Toal, D. Real-Time Reconstruction of Underwater Environments: From 2D to 3D. In Proceedings of the OCEANS 2015-MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–6, doi:10.23919/OCEANS.2015.7404506. [CrossRef]
15. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Basel, Switzerland, 26–29 October 2011; pp. 127–136, doi:10.1109/ISMAR.2011.6092378. [CrossRef]
16. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology

(UIST '11), Santa Barbara, CA, USA, 16–19 October 2011; ACM: New York, NY, USA, 2011; pp. 559–568, doi:10.1145/2047196.2047270. [CrossRef]
17. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: New York, NY, USA, 2003.
18. Thrun, S. Robotic Mapping: A Survey. In *Exploring Artificial Intelligence in the New Millenium*; Morgan Kaufmann: Burlington, MA, USA, 2002.
19. Davison, A.J. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV '03), Nice, France, 13–16 October 2003; IEEE Computer Society: Washington, DC, USA, 2003; Volume 2, p. 1403.
20. Nister, D.; Naroditsky, O.; Bergen, J. Visual Odometry. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004; Volume 1, p. I; doi:10.1109/CVPR.2004.1315094.
21. Chaves, S.M.; Galceran, E.; Ozog, P.; Walls, J.M.; Eustice, R.M. Pose-Graph SLAM for Underwater Navigation. In *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*; Fossen, T.I., Pettersen, K.Y., Nijmeijer, H., Eds.; Lecture Notes in Control and Information Sciences; Springer International Publishing: Cham, Switzerland, 2017; pp. 143–160, doi:10.1007/978-3-319-55372-6_7.
22. Bonin-Font, F.; Cosic, A.; Negre, P.L.; Solbach, M.; Oliver, G. Stereo SLAM for Robust Dense 3D Reconstruction of Underwater Environments. In Proceedings of the OCEANS 2015-Genova, Genoa, Italy, 18–21 May 2015; pp. 1–6, doi:10.1109/OCEANS-Genova.2015.7271333. [CrossRef]
23. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '07), Nara, Japan, 13–16 November 2007; IEEE Computer Society: Washington, DC, USA, 2007; pp. 1–10, doi:10.1109/ISMAR.2007.4538852. [CrossRef]
24. Chambolle, A.; Pock, T. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *J. Math. Imag. Vis.* **2010**, *40*, 120–145, doi:10.1007/s10851-010-0251-1. [CrossRef]
25. Newcombe, R.A.; Davison, A.J. Live Dense Reconstruction with a Single Moving Camera. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 1498–1505, doi:10.1109/CVPR.2010.5539794. [CrossRef]
26. Graber, G.; Pock, T.; Bischof, H. Online 3D Reconstruction Using Convex Optimization. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 708–711, doi:10.1109/ICCVW.2011.6130318. [CrossRef]
27. Newcombe, R.A.; Lovegrove, S.; Davison, A. DTAM: Dense Tracking and Mapping in Real-Time. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2320–2327, doi:10.1109/ICCV.2011.6126513. [CrossRef]
28. Anwer, A.; Ali, S.S.A.; Khan, A.; Mériaudeau, F. Underwater 3-D Scene Reconstruction Using Kinect v2 Based on Physical Models for Refraction and Time of Flight Correction. *IEEE Access* **2017**, *5*, 15960–15970, doi:10.1109/ACCESS.2017.2733003. [CrossRef]
29. Lu, H.; Zhang, Y.; Li, Y.; Zhou, Q.; Tadoh, R.; Uemura, T.; Kim, H.; Serikawa, S. Depth Map Reconstruction for Underwater Kinect Camera Using Inpainting and Local Image Mode Filtering. *IEEE Access* **2017**, *5*, 7115–7122, doi:10.1109/ACCESS.2017.2690455. [CrossRef]
30. Yilmaz, O.; Karakus, F. Stereo and Kinect Fusion for Continuous 3D Reconstruction and Visual Odometry. In Proceedings of the 2013 International Conference on Electronics, Computer and Computation (ICECCO), Ankara, Turkey, 7–9 November 2013; pp. 115–118, doi:10.1109/ICECCO.2013.6718242. [CrossRef]
31. Hogue, A.; German, A.; Jenkin, M. Underwater Environment Reconstruction Using Stereo and Inertial Data. In Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, 7–10 October 2007; pp. 2372–2377, doi:10.1109/ICSMC.2007.4413666. [CrossRef]
32. Servos, J.; Smart, M.; Waslander, S.L. Underwater Stereo SLAM with Refraction Correction. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 3350–3355, doi:10.1109/IROS.2013.6696833. [CrossRef]
33. Wu, Y.; Nian, R.; He, B. 3D Reconstruction Model of Underwater Environment in Stereo Vision System. In Proceedings of the 2013 OCEANS-San Diego, San Diego, CA, USA, 23–27 September 2013; pp. 1–4, doi:10.23919/OCEANS.2013.6741275. [CrossRef]

34. Hurtós, N.; Nagappa, S.; Palomeras, N.; Salvi, J. Real-Time Mosaicing with Two-Dimensional Forward-Looking Sonar. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 601–606, doi:10.1109/ICRA.2014.6906916. [CrossRef]
35. Li, H.; Dong, Y.; He, X.; Xie, S.; Luo, J. A Sonar Image Mosaicing Algorithm Based on Improved SIFT for USV. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014; pp. 1839–1843, doi:10.1109/ICMA.2014.6885981. [CrossRef]
36. Lagudi, A.; Bianco, G.; Muzzupappa, M.; Bruno, F. An Alignment Method for the Integration of Underwater 3D Data Captured by a Stereovision System and an Acoustic Camera. *Sensors* **2016**, *16*, 536, doi:10.3390/s16040536. [CrossRef] [PubMed]
37. Digumarti, S.T.; Chaurasia, G.; Taneja, A.; Siegwart, R.; Thomas, A.; Beardsley, P. Underwater 3D Capture Using a Low-Cost Commercial Depth Camera. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–10 March 2016; pp. 1–9, doi:10.1109/WACV.2016.7477644. [CrossRef]
38. Dancu, A.; Fourgeaud, M.; Franjcic, Z.; Avetisyan, R. Underwater Reconstruction Using Depth Sensors. In Proceedings of the SIGGRAPH Asia 2014 Technical Briefs (SA '14), Shenzhen, China, 3–6 December 2014. doi:10.1145/2669024.2669042.
39. Fusiello, A.; Trucco, E.; Verri, A. A Compact Algorithm for Rectification of Stereo Pairs. *Mach. Vis. Appl.* **2000**, *12*, 16–22, doi:10.1007/s001380050120. [CrossRef]
40. Brown, M.Z.; Burschka, D.; Hager, G.D. Advances in Computational Stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 993–1008, doi:10.1109/TPAMI.2003.1217603. [CrossRef]
41. Curless, B.; Levoy, M. A Volumetric Method for Building Complex Models from Range Images. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 4–9 August 1996; pp. 303–312.
42. Rusinkiewicz, S.; Levoy, M. Efficient Variants of the ICP Algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152, doi:10.1109/IM.2001.924423. [CrossRef]
43. Chen, Y.; Medioni, G. Object Modeling by Registration of Multiple Range Images. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; Volume 3, pp. 2724–2729, doi:10.1109/ROBOT.1991.132043. [CrossRef]
44. Low, K.L. *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration*; Technical Report; University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 2004.
45. Prisacariu, V.A.; Kähler, O.; Cheng, M.M.; Ren, C.Y.; Valentin, J.; Torr, P.H.S.; Reid, I.D.; Murray, D.W. A Framework for the Volumetric Integration of Depth Images. *arXiv* **2014**, arXiv:cs/1410.0925.
46. Marquardt, D. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441, doi:10.1137/0111030. [CrossRef]
47. Parker, S.; Shirley, P.; Livnat, Y.; Hansen, C.; Sloan, P. Interactive Ray Tracing for Isosurface Rendering. In Proceedings of the Visualization '98 Research Triangle Park, NC, USA, 18–23 October 1998; pp. 233–238, doi:10.1109/VISUAL.1998.745713. [CrossRef]
48. Kähler, O.; Prisacariu, V.A.; Ren, C.Y.; Sun, X.; Torr, P.; Murray, D. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices. *IEEE Trans. Vis. Comput. Graph.* **2015**, *21*, 1241–1250, doi:10.1109/TVCG.2015.2459891. [CrossRef] [PubMed]
49. Ravi. Fork of the Voxel Hashing Based Volumetric Integration of Depth Images, InfiniTAM, That Enables ROS as an Input Source. 2017. Available online: https://github.com/ravich2-7183/InfiniTAM (accessed on 7 October 2018).
50. Technical Application Notes. Available online: https://www.ptgrey.com/tan/11052 (accessed on 22 May 2018).
51. Brown, D.C. Decentering Distortion of Lenses. *Photogramm. Eng. Remote Sens.* **1966**, *32*, 444–462.
52. Agisoft PhotoScan. Available online: http://www.agisoft.com/ (accessed on 26 October 2018).
53. Pradeep, V.; Rhemann, C.; Izadi, S.; Zach, C.; Bleyer, M.; Bathiche, S. MonoFusion: Real-Time 3D Reconstruction of Small Scenes with a Single Web Camera. In Proceedings of the 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, Australia, 1–4 October 2013; pp. 83–88, doi:10.1109/ISMAR.2013.6671767. [CrossRef]