Constructing Point Clouds from Underwater Stereo Movies

Jesus Pulido[1], Ricardo Dutra da Silva[2], Dawn Sumner[3], Helio Pedrini[4], and Bernd Hamann[1]

[1] Institute for Data Analysis and Visualization, University of California, Davis
Davis, CA, 95616-8562, U.S.A.
[2] Informatics Department, Federal University of Technology
Curitiba, PR, 80230-901, Brazil
[3] Department of Earth and Planetary Sciences, University of California, Davis
Davis, CA 95616, U.S.A.
[4] Institute of Computing, University of Campinas
Campinas, SP, 13083-852, Brazil

Abstract. Processing images of underwater environments of Antarctic lakes is challenging due to poor lighting conditions, low saturation and noise. This paper presents a novel pipeline for dense point cloud scene reconstruction from underwater stereo images and video obtained with low-cost consumer recording hardware. Features in stereo frames are selected and matched at high density. Putative matches are triangulated to produce point clouds in 3D space. Temporal feature tracking is performed to produce and merge point clouds. We demonstrate that this framework produces dense and accurate reconstructions for several tests.

1 Introduction

Underwater environments are of great interest to geologists as they make it possible to harvest microbial organisms in extreme conditions. With the advent of low-cost and flexible video capturing hardware, geologists are interested in visualizing captured scenes in interactive virtual reality environments such as a CAVE [1]. Such a system makes it possible for geologists to better understand the origin of organisms and study the amount of microbial life growing through the computation of statistics, growth rate, and frequency of clustering in the environments. [2,3,4]. In addition, there is interest in comparing how the growth of these structures is affected by the occlusion of sunlight from large rocks based on the environment reconstruction.

We focus on the reconstruction of underwater environments using stereo video captured under harsh conditions in Antarctic Lakes [2]. Both a low-cost stereo camera system and an expensive Light Detection and Ranging (LiDAR) system were used to capture data from these environments. Unfortunately, the expensive LiDAR system produced extremely noisy scans due to the harsh underwater Antarctic environment. In one case, a single reconstructed environment required tens of hours of manual data cleaning to correct instrument noise.

Alternatively, we propose a method using visual information and image processing techniques for reconstructing these underwater environments through the use of inexpensive, GoPRO HD Hero 2 cameras in stereo configuration. Processing the images is a challenging task due to the rough conditions in which the scenes are captured. There is noise due to debris, poor illumination, uneven lighting conditions produced by lamps attached to the recording system, and a large amount of radial distortion introduced by the cameras.

The main purpose is the reconstruction of these underwater scenes with enough accuracy to measure millimeter-size structures growing in the lakes to allow further analysis by collaborating geologists. We have developed a modular scene reconstruction system that produces high accuracy and dense reconstructions. Modular components include the choice of a feature extractor, matcher, error evaluation schemes, and user-interaction components to aid a reconstruction in severe circumstances. This general system reconstructs multiple types of environments and it is robust enough for underwater reconstruction.

We have evaluated the quality of reconstructed 3D point clouds to demonstrate that the the low-cost system is accurate and an alternative to expensive systems. Several modular components have been tested to present the best combination of features and parameters for each environment.

The remaining paper is structured as follows. Section 2 presents related work and provides a background to the main image matching methods used for our system. Section 3 describes the reconstruction system in its entirety. The evaluation of the system and results are described in Section 4. We conclude with final remarks in Section 5.

2 Background

This section briefly discusses some previous works related to the reconstruction of underwater structures and image matching methods.

2.1 Underwater Reconstruction

Espiaur et al. [5] first proposed a method for extracting robustly features from underwater images for 3D reconstruction. They proposed the use of a pyramidal tracking scheme in order to match features across several time frames. Although the approach is susceptible to noise and rough lighting conditions, it serves as a starting point for our method.

Sedlazeck et al. [6] performed underwater reconstruction based on video from sensory information gathered from a remotely operated vehicle. A 3D surface model is estimated based on the multiple on-board cameras and navigational data from sensors, which are not available in our setting. Despite this, our proposed method is able to estimate depth and track features over time using much lower cost hardware.

Meline et al. [7] presented recommendations for feature selection and tested different camera equipment for underwater cartography. They suggested the use

of lateral sonar, GPS localization and inertial information for their future work, however, we show that these specialized components are not required when stereo footage is used.

2.2 Image Matching Methods

The feature matching component in our proposed pipeline is modular and the most critical since the quality of the matches significantly influence the quality of the resulting point cloud.

The primary method is the pyramidal implementation of the Lukas-Kanade feature tracker (LKOpt) [8], a gradient-based algorithm that tracks features across multiple resolution levels. The algorithm is based on several assumptions: spatial coherence of features is satisfied, brightness is fairly consistent between features, and there is little distance between features for different frames.

The Scale Invariant Feature Transform (SIFT) [9] is able to extract distinctive scale- and rotation-invariant features. These features are robust enough to be matched under affine distortions, noise, and illumination changes.

A more recent feature matcher used in this pipeline combines the discrete Morse complex and graph matching (MCGM). The discrete Morse complex (DMC) relates the topology of a function with its feature points (maxima, minima and saddles) [10]. Such features are usually obtained from a derivative of the input image function [11]. Figure 1 illustrates the construction of the DMC. Given the input image, its Laplacian of the Gaussian (LoG) [12] is computed and used to obtain the DMC. Details of the computation of the DMC can be found in [13,14].

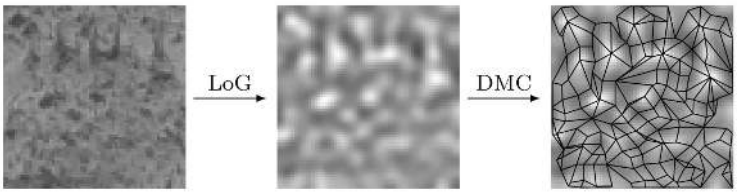


Fig. 1: Example of LoG and DMC of an image.

The DMC obtains the connections between interest points. Considering α_k and β_l as putative matches, it is possible to determine whether they are a correct match by using the arrangements of interest points around them, which are expected to be similar. The points between arrangements are corresponded with a graph matching algorithm [14]. Figure 2 shows an example of putative matches α_k and β_l and how the interest points in the arrangements around them are corresponded.

A score that counts the number of corresponding points that have similar vertical displacements relative to α_k and β_l is computed. Taking Figure 3 as an example, except from interest points identified with values 3 and 8, the displacements of five interest points are approximately the same, suggesting that the matching is probably a correct one.

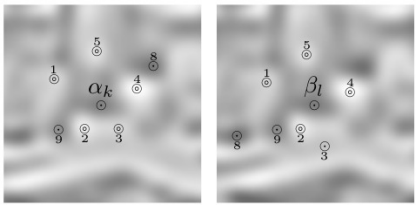


Fig. 2: Example of arrangements of two putative interest points α_k and β_l. Correspondences between points in the arrangements are represented by numbers.

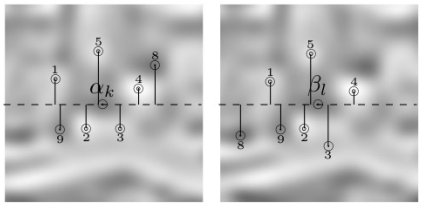


Fig. 3: Scoring putative matches. The local arrangements should be similar.

As with MCGM, both LKOpt and SIFT use score-based systems when matching features that allow us to select the highest-rated matches.

3 Methodology

Given as input a stereo video stream, features are selected and matched across stereo frames. These features are triangulated from 2D image space into 3D point clouds. The result is a single reconstructed point cloud of a time pair. Temporal feature tracking is performed to iteratively reconstruct and merge multiple frames. The reconstruction pipeline is illustrated in Figure 4.

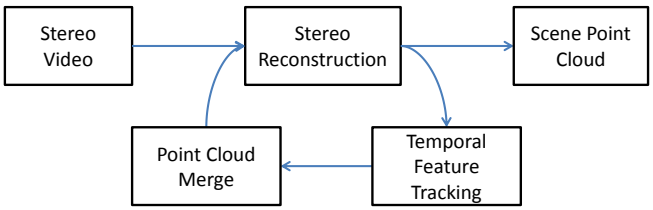


Fig. 4: Overview of the proposed reconstruction pipeline.

3.1 Stereo Images and Video

Image distortion is corrected in a preprocessing step. Zhang's method [15] is commonly used for simple distortions. During our evaluation of the camera footage, we found that Zhang's calibration method and its linear model were insufficient for removing the large amount of radial distortion from the images. Due to the wide-angle cameras with large distortion, the non-linear model presented by Claus et al. [16] was employed.

Our system can also perform calibration-less reconstruction when distortion is not present. To accomplish this, minimal extrinsic information is requested from the user such as focal length, field of view, and distance between cameras (easily found from the camera manufacturer's specification sheet).

3.2 Stereo Reconstruction: Feature Matching

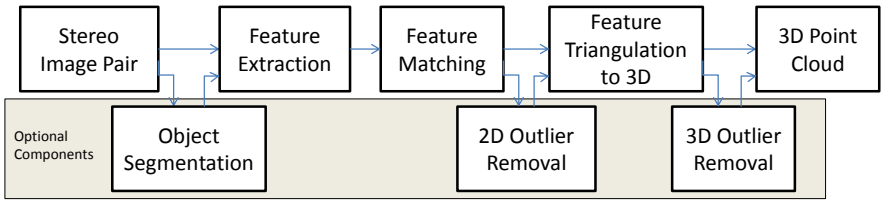


Fig. 5: Stereo image pair reconstruction pipeline.

The stereo reconstruction component of the pipeline is summarized in Figure 5. Initially, the system corresponds point features from the left-image pair to the right-image pair. A segmentation step is included so that features are matched only for regions of interest. This is important to remove a potentially background in the far distance from the region of interest.

By applying the watershed image segmentation method [17,18], we allow for automatic and user-assisted selection of regions. The automatic watershed method uses the Canny edge detector [18] as the input to the watershed method to generate regions. Regions are selected based on proximity to the center of the image and favoring those closest. The user-driven selection allows the user to manually choose objects of interest in a scene. By allowing user assistance, the system has more flexibility and direct control over the regions to be reconstructed.

We perform dense feature extraction by selecting pixels based on regions in the left stereo image. Our detection methods are based on gradients, avoiding regions with little to no intensity variance which present difficulties when matching features. We compute the variance of gray-scale pixel intensities through an odd-sized kernel that iterates throughout the entire image to select a series of pixels for stereo matching. Dealing with stereo image pairs, we allow a y-axis deviation of about ten pixels on the epipolar lines and consider matches outside of this boundary to be outliers to avoid incorrect triangulations in 3D space.

Lukas-Kanade Pyramidal Optical Flow (LKOpt) [8] is used for matching features between stereo pairs. LKOpt has been adapted to restrict the majority of the movement to the image plane's x-axis and outlier removal is performed based on movement restrictions to the y-axis.

3.3 Stereo Reconstruction: Triangulation in 3D Space

Putative matches from 2D space are triangulated using the pinhole camera model and parallel optical axis to estimate converging points in 3D space. The process is illustrated in Figure 6.

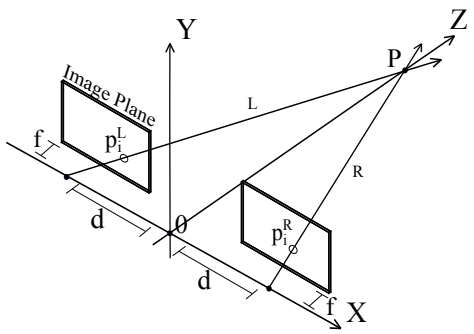


Fig. 6: We determine an intersection point P in 3D space from an identified pair of corresponding points p_i^L and p_i^R in a stereo image pair.

Our physical camera system contains fixed stereo cameras that restricts stereo image pairs to the horizontal plane. To triangulate stereo matches to a point P, we first define

$$\partial_L = \begin{pmatrix} 0 \\ 0 \\ -d \end{pmatrix} \quad \partial_R = \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} \quad p_i^L = \begin{pmatrix} x^L \\ y^L \\ f \end{pmatrix} \quad p_i^R = \begin{pmatrix} x^R \\ y^R \\ f \end{pmatrix} \tag{1}$$

where d is the distance between the point of origin 0 and the center of either camera ∂_R or ∂_L, x^L / x^R and y^L / y^R are the position of a feature in image space for its respective camera p_i^L / p_i^R in 3D, and f is the focal length. We can create a relationship u and w to calculate the point P

$$L : P(u) = \partial_L + u \cdot p_i^L \qquad R : P(w) = \partial_R + w \cdot p_i^R \tag{2}$$

$$p_i^L \cdot u - p_i^R \cdot w = \partial_R - \partial_L \tag{3}$$

$$M \cdot u' = r' \tag{4}$$

where

$$M = \left[p_i^L\right] \left[-p_i^R\right], u' = \begin{pmatrix} u \\ w \end{pmatrix}, r' = \begin{pmatrix} 2d \\ 0 \\ 0 \end{pmatrix}. \tag{5}$$

We create a linear system of equations

$$M^t M \cdot u' = M^t r' \tag{6}$$

$$P = \text{midpoint}(u') \tag{7}$$

and solve a least-square problem in order to triangulate where both image plane features intersect at P in the depth axis. We improve our solution by using iterative refinement on the least-squares approximation of u' for maximum precision.

After processing each feature pair, we obtain a dense point cloud. An additional outlier removal step is performed by creating a bounding cube surrounding the point cloud removing points outside five standard deviations.

3.4 Temporal Feature Tracking in Images

The reconstructed 3D points and their stereo image features are used for temporal tracking and merging of multiple time frames. A new stereo reconstruction is first performed for the next frame, and features are temporally matched through the left stereo images.

Temporal matches are no longer restricted to the horizontal axis since images can present varying perspectives. In that case, LKOpt can be sensitive and produce low-quality temporal matches especially when two image pairs have vastly different perspectives. In this situation, features can be more carefully tracked by using more video frames between changes.

The system is modular and allows using methods that produce more accurate matches at higher computational cost. When incorrect temporal matches occur, the reconstructed point cloud will appear to drift non-linearly in 3D space and any newly reconstructed frames will be merged incorrectly. To improve overall quality for temporal matches, we use SIFT [9] and MCGM (Section 2.2). Figure 7 shows the distribution, quantity and characteristics of temporal features matched by both SIFT and MCGM.

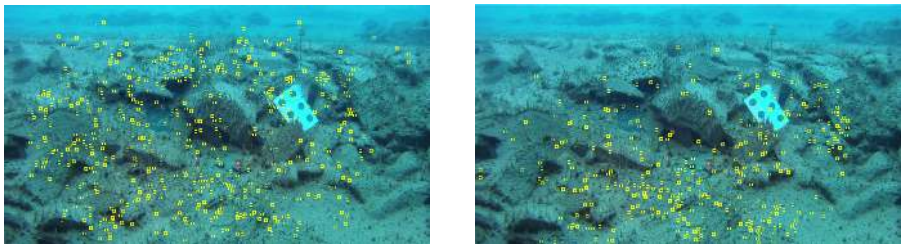


Fig. 7: A comparison showing the quantity, type and distribution of correctly registered temporal matches by MCGM (left) and SIFT (right).

The methods score the quality of the matches allowing us to choose a small subset of high-precision matches. To further improve our selection, the image homography between temporal image features is estimated and Random Sample Consensus (RANSAC) [19] is used to select inliers that loosely, but not strictly, adhere to the model. When inliers strictly adhere to the model, a degenerate case cannot be avoided resulting in a loss of representation of an axis in 3D space, as explained in the next section.

3.5 Merging Point Cloud Data

When a stereo pair of images is reconstructed, the resulting point cloud exists in its own local 3D system. We create a global 3D coordinate system into which all local systems are merged. In order to successfully merge multiple point clouds, we use the image-space temporal matches to correlate their 3D point counterparts. Shinji Umeyama's refined least-squares approach [20] is used to estimate an affine transformation between the local and global systems through the use of four or more correlated 3D points. In order to have a robust and accurate computation,

it is required to have multiple points lying on each spatial dimension, therefore, it is recommended that source footage, when possible, should avoid extremely localized features.

We evaluate the quality of a transformation matrix by comparing 3D points in the current system to those in the global system. RANSAC is used to select the matrix minimizing the total L_1-norm error.

Once the best transformation is found, we perform stereo reconstruction for a new stereo pair as discussed in Section 3.2. The dense local point cloud is transformed and merged in the global coordinate system through the new transformation. Remaining frames are then considered, following the cycle shown in Figure 4.

3.6 Scene Point Cloud and Post-processing Steps

Further outlier removal is done when all time steps have been reconstructed and merged into a single global system. We perform a k-nearest-neighbor (KNN) search for every point and remove those points that lack neighbors within a user-defined Euclidean distance. Point refinement also performed but limited due to the unpredictability of non-overlapping footage from underwater scenes. The remaining points are rendered using surface elements [21].

4 Experimental Results

4.1 Datasets and Test Cases

We evaluate the effectiveness of our pipeline by considering three test environments: two synthetic scenes emulating the pinhole camera model, two non-underwater scenes of a natural environment and an underwater scene.

The synthetic scenes are controlled test environments constructed and used to benchmark the algorithms and measure the quality of reconstructions. One scene depicts a unit cylinder and another a unit sphere. A floor and a spherical background are added to the scenes. Unique textures from natural environments are applied to the elements composing the scenes. Both scenes are shown in Figure 8.

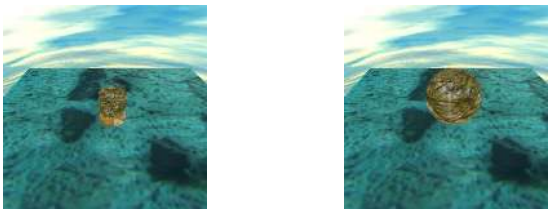


Fig. 8: Synthetic scenes: cylinder and sphere.

Two real-world scenes form more controlled tests consisting of footage resulting from recording two rocks with the same hardware used to record the

underwater footage (Figure 9). The rocks contain similar features when compared to the synthetic scenes, however, they present the challenge of concavities in the rock surfaces. The ground truth was obtained with a LiDAR scanner that captured 360-degree views of the rocks.

Fig. 9: Sample rock images.

This test presents several challenges due to the wide field of view of the cameras and the auto light balancing and color correction done by the cameras. The wide field of view introduces a large amount of barrel distortion that is mostly removed by our calibration techniques but not completely.

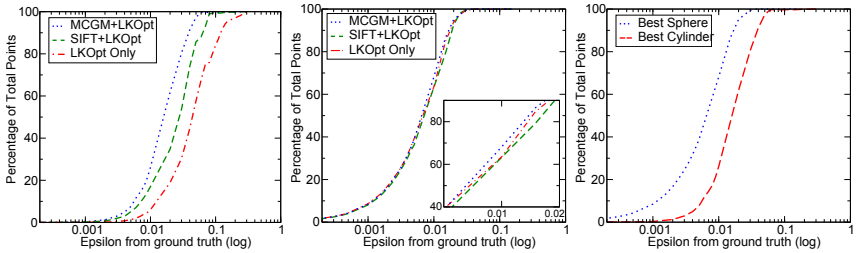
The underwater scene is a complex and uncontrolled test. It consists of footage taken from the floor of an Antarctic lake, containing visible particles flowing around making outlier removal more difficult. The poor lighting conditions further complicate the processing of this data. Although we have no ground truth data, based on the results from the controlled tests, we can provide qualitative results.

4.2 Results for Synthetic Data

Figure 10(a) compares the percentage of points, out of one million 3D points, that lie closest to the ground truth. Quality is measured as the distance between the unit objects and each reconstructed point. Visually, all methods produce very similar-quality results, but those by LKOpt only produce numerically inferior results. When stereo LKOpt is complemented by SIFT for matching features over time, a significant increase in accuracy is gained. A slightly better increase in accuracy is achieved with the usage of MCGM. LKOpt alone is unable to track temporal features around rigid objects such as the cylinder. With such combinations, obtain high-density stereo matches and accurate time matches.

Considering Figure 10(b), using only LKOpt for both stereo and time-pairs performs better than substituting SIFT for the time pair registration. This is a result of LKOpt being able to track temporal features better in areas where larger surfaces are present. When combined with SIFT, LKOpt has difficulty tracking SIFT's time features because SIFT uses a more complex descriptor than LKOpt. The time matches provided by MCGM are better received by LKOpt and provided the best reconstruction results. Because MCGM extracts time features that are more distributed than SIFT, LKOpt has a broader-range subset of features to stereo match.

Overall, Figure 10(c) plots the best of both synthetic methods to show the ability to reconstruct with very high accuracy, where over 65% of points fall



(a) Cylinder accuracy (b) Sphere accuracy (c) Best of methods

Fig. 10: Reconstruction results for the synthetic dataset.

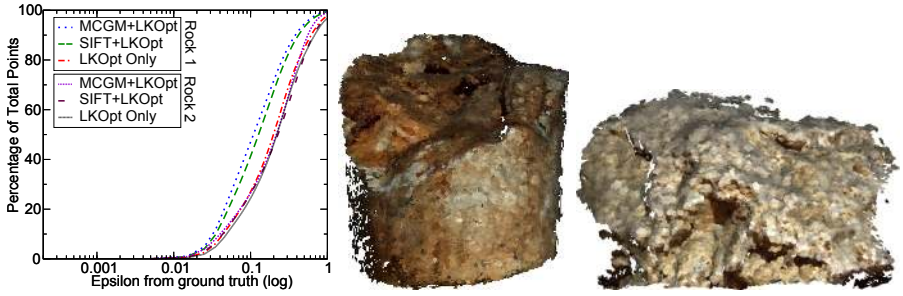
within 1% error for the sphere and 25% of points within 1% for the cylinder. Both reconstructed point clouds are illustrated in Figure 11

(a) Synthetic cylinder cloud (b) Synthetic sphere cloud

Fig. 11: Reconstructed point clouds of cylinder and sphere.

4.3 Results for Real-world Data

In Figure 12, the real-world rocks are reconstructed and accuracy is evaluated by performing a closest-point estimation to the ground truth dataset. Similar to the synthetic cylinder results, all methods exhibit the same reconstruction quality. The same difficulties seen in the synthetic cylinder are observed in the first rock that is of cylinder-like shape.



(a) Accuracy of Rock 1 and 2 (b) Rock 1 cloud (c) Rock 2 cloud

Fig. 12: Real-world rock reconstruction results.

Considering the results achieved in the previous tests, the types of structures in the underwater scenes and the goal of achieving the most accurate and dense

reconstruction possible, we combine MCGM for the time matches and LKOpt for the stereo matches. LKOpt provides dense, high quality stereo matches and MCGM the best spatially distributed and accurate temporal matches.

4.4 Underwater Results

The underwater environments can be reconstructed with high fidelity and density. Figure 13(b) shows a close-up of a reconstructed underwater point cloud. We are able to capture the intricate features of the environment, such as concave and convex features on the rocks.

(a) Original image (b) Reconstructed scene

Fig. 13: Results for underwater data.

5 Conclusions

We have described a pipeline that is able to perform high-accuracy, high-density reconstruction of stereo video, including underwater footage under difficult conditions obtained using low-cost hardware. To achieve the most accurate reconstructions, we use a combination of LKOpt and MCGM for temporal matches. For more rapidly generated, slight less accurate reconstructions, we use LKOpt and SIFT. We intend to explore parallel computation of the components to allow real-time dense reconstruction. Currently, an octree structure is computed in a post-processing step. One could consider generating this structure in one of the earlier processing steps and update it following the addition of new frames.

Acknowledgements

The authors thank FAPESP, CNPq and CAPES for the financial support. We also thank our colleagues from the Institute for Data Analysis and Visualization (IDAV) at UC Davis, as they have provided several helpful comments and suggestions.

References

1. Kreylos, O.: Environment-Independent VR Development. In: Advances in Visual Computing. Volume 5358 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2008) 901–912 1

2. Stevens, E.W., Sumner, D.Y.: Identifying Key Structural Features and Spatial Relationships in Archean Microbialites Using 2D and 3D Visualization Methods. AGU Fall Meeting Abstracts (2009) A318 1
3. Hawes, I., Sumner, D., Andersen, D., Mackey, T.: Legacies of Recent Environmental Change in the Benthic Communities of Lake Joyce, a Perennially Ice-Covered Antarctic Lake. Geobiology **9** (2011) 394–410 1
4. Hawes, I., Sumner, D., Andersen, D., Jungblut, A., Mackey, T.: Timescales of Growth Response of Microbial Mats to Environmental Change in an Ice-Covered Antarctic Lake. Biology **1** (2013) 151–176 1
5. Espiau, F.X., Rives, P.: Extracting Robust Features and 3D Reconstruction in Underwater Images. In: OCEANS, 2001. Volume 4. (2001) 2564–2569 2
6. Sedlazeck, A., Koser, K., Koch, R.: 3D Reconstruction based on Underwater Video from ROV Kiel 6000 Considering Underwater Imaging Conditions. In: OCEANS 2009. (2009) 1–10 2
7. Meline, A., Triboulet, J., Jouvencel, B.: A Camcorder for 3D Underwater Reconstruction of Archeological Objects. In: OCEANS 2010. (2010) 1–9 2
8. Bouguet, J.Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker. Intel Corporation, Microprocessor Research Labs (2000) 3, 5
9. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **60** (2004) 91–110 3, 7
10. Forman, R.: Morse Theory for Cell Complexes. Advances in Mathematics **134** (1998) 90–145 3
11. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A Comparison of Affine Region Detectors. International Journal of Computer Vision **65** (2005) 43–72 3
12. Szeliski, R.: Computer Vision: Algorithms and Applications. 1st edn., New York, NY, USA (2010) 3
13. Robins, V., Wood, P.J., Sheppard, A.P.: Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images. IEEE Transaction on Pattern Analysis and Machine Intelligence **33** (2011) 1646–1658 3
14. Silva, R.D.: Discrete Morse Complex of Images: Algorithms, Modeling and Applications. PhD thesis, University of Campinas, Campinas, Brazil (2013) 3
15. Zhang, Z.: A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (1998) 1330–1334 4
16. Claus, D., Fitzgibbon, A.: A Rational Function Lens Distortion Model for General Cameras. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2005) 213–219 4
17. Roerdink, J., Meijster, A.: The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. Fundamenta Informaticae **41** (2000) 187–228 5
18. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2006) 5
19. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24** (1981) 381–395 7
20. Umeyama, S.: Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **13** (1991) 376–380 7
21. Pfister, H., Zwicker, M., van Baar, J., Gross, M.: Surfels: Surface Elements As Rendering Primitives. In: 27th Annual Conference on Computer Graphics and Interactive Techniques. (2000) 335–342 8