3D Reconstruction Based on Underwater Video from ROV Kiel 6000 Considering Underwater Imaging Conditions

Anne Sedlazeck, Kevin Köser, and Reinhard Koch
Institute of Computer Science
Christian Albrechts University
Kiel, Germany
Email: sedlazeck@mip.informatik.uni-kiel.de

***Abstract*— This work presents a system for 3D reconstruction from underwater images or video. Aside from a camera in an underwater housing, no special equipment is required. However, if navigational data is available, it is utilized in the algorithm. The algorithm is in essence a classical structure from motion approach, which is adapted to account for the special imaging conditions. Hence, there is no need for the camera to follow a specialized trajectory. Adaptions to the underwater imaging environment include a special filtering of background and floating particles, which allows a robust estimation of the camera poses and a sparse set of 3D points. Based on the estimated camera track, dense image correspondence computation enables building a detailed 3D surface model. Once the 3D surface model is completed, the colors of the texture are corrected by a physical model for underwater light propagation, allowing to view the model without the effects of scattering and attenuation or to simulate the effects of water on light in a 3D viewer.**

I. Introduction

Underwater objects and structures like black smokers, ship wrecks, or coral reefs, which can only be observed by diving personally or operating a submarine, are difficult to study. Divers or submarines equipped with cameras provide image sequences of such objects. The approach presented in this work utilizes this information by computing 3D reconstructions based on underwater video. 3D reconstructions can for example be used for volumetric measurements, documentation, and presentation to the general public.

This project is part of the Future Ocean Excellence cluster [1] and aims to investigate feasibility and limitations of computing 3D reconstructions from videos of deep sea floor structures. The examples studied in this work are seafloor hydrothermal systems - or more precisely *black smokers*. The black smokers in question are found in the Atlantic ocean at $4°48'S$ and $12°23'W$ at a water depth of approximately 3039 m. Consequently, a remotely operated vehicle, the ROV Kiel 6000 (see figure 1) is needed to capture the video sequences. It is equipped with several cameras, one of them being a HDTV video camera. In addition, a set of rough navigation data is recorded that is used to predict the camera path and allows quantitative measurements.

[1] http://www.ozean-der-zukunft.de/

In the last years so called structure-from-motion approaches (SfM) like for example [13] or [4] have proven successfull for good imaging conditions on land, by utilizing the 3D information implicitly contained in the image sequence. One aim of this project is the adaptation to the special underwater imaging conditions, which cause a variety of difficulties that need to be considered. For example, the rigid scene assumption is violated by floating particles, fish, and other moving animals and in case of black smokers sometimes large amounts of smoke. Another challenge arises from the underwater light propagation. In contrast to light propagation in air, the single light rays are much more effected by the densly packed water molecules: they are attenuated and scattered, having a great effect on the image colors. In addition, light rays are refracted, when entering the camera housing, causing a different focal length underwater than in air and even 3D distortion effects (see [16]).

Taking these challenges into consideration, we propose a structure from motion system allowing to compute detailed 3D reconstructions from underwater objects or scenes. Figure 2 shows an overview of this method. Complementary to other comparable approaches (see [2] or [5]), only one camera is used instead of assuming a stereo rig. Other than in [2], the camera does not have to follow a predefined path, but can capture video from rigid, local seafloor structures. This is particularly different to [12], where known man-made structures are assumed.

The initial intrinsic camera calibration in figure 2 is provided by a calibration procedure as described in [15]. After that, the next steps depicted in figure 2 follow the classical approach for SfM as described in [13], but use special outlier detection, based on water/object segmentation, to filter out the usually noisy background of the images. After initialization from an image pair by exploiting the epipolar image geometry, the other frames of the sequence need to be added. However, in contrast to [11], a frame-to-frame pose estimation based on 2D-3D correspondences is computed sequentially, resulting in a complete reconstruction of the camera path. In both cases, initialization and 2D-3D pose estimation, the rough navigation data is used as a pose prediction and refined. Once

Fig. 1. ROV Kiel 6000

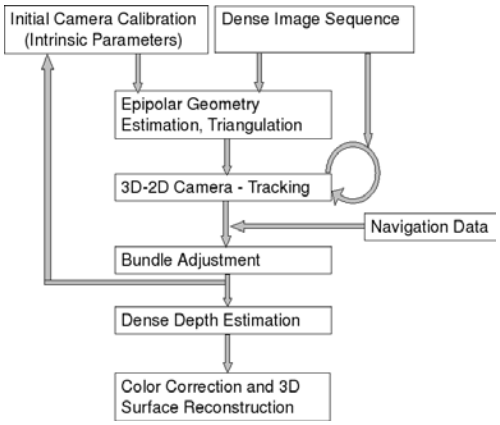

Fig. 2. Diagram of the structure from motion approach.

the structure from motion routine is completed, resulting in a sparse 3D cloud and reconstructed camera path, we compute depth for almost every pixel in each view in the camera path. This is essentially different from [11] and allows the computation of much more detailed and accurate models than linear interpolation between a sparse set of 3D points.

However, for a realistic 3D visualization the colors of the image need to be corrected. Due to wavelength dependent scattering and absorption of the different wavelengths of light in water, the image color has a blue or green hue depending on the distance between object and camera. 3D visualization includes the possibility of virtually moving around the object in question interactively, thus changing distances between the observer and the object. Therefore, the colors of the object do not have the correct hue. Consequently, a method is being developed, to correct the colors of the images based on a physical model of underwater light propagation. The physical model is similar to the ones used in [14] or [17], but unlike [17] does not require any special equipment. The model's parameters can be determined because the depth for each point is already known at this point, and it is possible to determine several 2D points for each 3D point that lie in a neighborhood with preferably constant colors. Such a color correction will allow to view the object without the effects of color attenuation due to water.

The paper is organized as follows. First, the imaging platform is introduced, then camera model and calibration are described briefly. After a section on notation and background, the algorithm will be described with special emphasis on the changes made for the underwater imaging environment. A section on implementation and results summarizes some of the tests that have been conducted. The paper is concluded with some suggestions for future work.

II. ROV Kiel 6000

Images for this project are delivered by the HDTV camera mounted on the remotely operated vehicle ROV Kiel 6000 (fig. 1). It can reach water depths up to 6000 m and is equipped with several cameras for navigation (black and white) and image capturing (color) which are listed in table I. A fiber optic cable transfers the image data to the ship operating the ROV, where the data is captured and stored as YCbCr raw data. The video data is grabbed by the HDTV camera with 25 frames per second. During capturing, the scene in question is illuminated by several lights (see table I), mounted near the ROV's top resulting in a distance of about $2m$ to the HDTV camera, which is fixed near the bottom. [6] argues that increasing distances between light source and camera improves the image quality, especially the contrast. The ROV's architecture maximizes this distance between light source and HDTV camera, thus allowing the highest possible image quality with this camera.

TABLE I
ROV Kiel 6000 Cameras and Illumination

#	Type
1	HDTV-Camera Kongsberg Maritime OE14-500
2	color video cameras (SD, standard PAL) Kongsberg Maritime OE14-366 MKII
3	black/white video OCTOPUS GmbH
1	Digital still camera , 5 Mpix, 4 x optical zoom, Kongsberg Maritime OE14-366 MKII 1-CCD
2	400 W HMI/MSR (SeaArc 2)
8	250 W Halogen (Deep Multi SeaLite)
2	70 W HID (SeaArc 5000)
1	Underwater Flashgun Kongsberg Maritime OE11-242

The navigation data is delivered by the ROV twice per second and gives information about velocity in x-, y-, and z-direction and the ROV's orientation in Euler-angles describing pitch, yaw, and roll.

III. Camera Model and Calibration

Camera calibration for underwater images is more complex than in air - when light crosses the water-air interface, as it does when entering the camera housing, light rays are refracted at the water-air interface. That is why the camera needs to be calibrated in water or the camera calibration above water needs to be converted as described in [7]. If the camera is calibrated in air or even if the camera is calibrated underwater in a tank, especially the focal length is errorneous because the index of refraction changes with pressure, temperature, and salinity.

The change in focal length is discussed in [10], concluding that the calibration can only be exact, if conducted on-site, which is often complex or impossible.

However, in this case, the initial projective 3D reconstruction can be computed with an approximative camera calibration and is updated to an Euclidean reconstruction later in the algorithm, when a bundle adjustment step [4] is applied to refine the camera calibration. So in this project, a calibration above water or off-site in a water tank is sufficiently accurate.

The camera model used is the classical pinhole camera model extended to incorporate radial distortion. The model's parameters can be grouped into intrinsic and extrinsic parameters. The intrinsic parameters describe the camera's internal characteristics, while the extrinsic parameters describe the camera's position and orientation in the world coordinate system. In order to calibrate the intrinsic parameters, a checkerboard pattern is used to take several images of a plane with known dimensions. Because the checkerboard pattern can be easily recognized by computer vision algorithms, it is possible to determine the rays connecting 2D points in the image and 3D points on the checkerboard. This allows measurement of the whole imaging space (see [15] for details) resulting in the intrinsic camera parameters described in a camera matrix $\mathbf{K}$. $\mathbf{K}$ is dependent on f, the focal length, on a, the aspect ratio of the pixels, and on (c_x, c_y), the principal point. Additionally, the coefficients describing radial distortion are estimated. Knowing the radial distortion parameters allows undistorting 2D points in images which leads to the following model of projection used throughout the algorithm.

When taking an image, a set of 3D points is projected onto 2D points. Using projective geometry, this relation can be expressed with homogenous points [4]. A 3D point $M = (X, Y, Z, 1)^T$ is projected onto a 2D point $m = (x, y, 1)^T$ by a 3×4 projection matrix $\mathbf{P}$: $m \cong \mathbf{P}M$. $\mathbf{P}$ encodes the camera's orientation $\mathbf{R}$ in a global coordinate system, its center of projection C, and intrinsic camera parameters $\mathbf{K}$ in $\mathbf{P} = \mathbf{K}\mathbf{R}^T[\mathbf{I}| - C]$.

A projection matrix $\mathbf{P}$ and the corresponding image are in the following often referred to as camera. It is meant as a short form for camera pose and the image taken from this pose. So when speaking of several cameras, we mean several images taken from different poses. The only physical camera used, is the HDTV camera mounted on the ROV.

The intrinsic parameters are now known from the calibration, the SfM algorithm is used to determine the extrinsic parameters and the bundle adjustment will refine both.

IV. Algorithm

A. Structure from Motion

When an image sequence has been recorded and the initial set of intrinsic camera parameters has been determined, the 3D reconstruction can be computed. The image sequences delivered by the HDTV camera at 25 frames per second result in a huge amount of data - a 3 minute video adding up to about 27 Gigabytes with 4500 images. For memory reasons we so far only process subsequences consisting of a few hundred images.

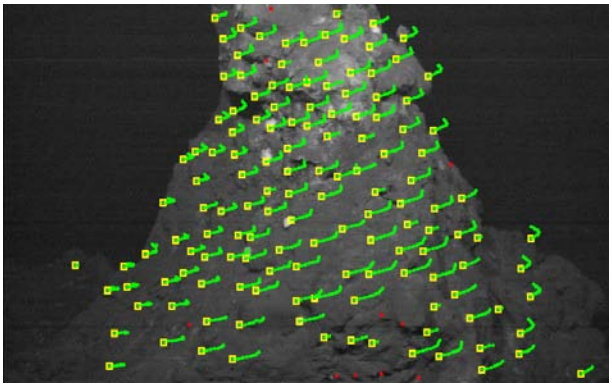

Fig. 3. Correspondences tracked over the image sequence.

When applying the reconstruction algorithm to a subsequence, the first step consists of relating the images of the sequence to each other. Therefore, features are detected in each image as a first step. The feature detector used is a corner detector based on image gradients; pixels with a neighborhood containing large gradients are detected. After the feature detection, each feature is undistorted and then normalized with the inverse camera matrix $\mathbf{K}^{-1}$. The features are tracked from image to image by the KLT tracker (see [8] for details), which registers the displacement between features. Figure 3 depicts the tracks the features have taken over the last few images. The KLT tracker is a good solution to the tracking problem in this case because the input image sequence consists of dense video: the displacement between two consecutive images is very small and the tracking can be implemented efficiently. The resulting correspondences between two images are called 2D-2D correspondences in the following.

In most cases of underwater images analyzed for this project, the images contained some structure and a lot of background, where only water was visible. However, due to floating particles and noise added by the camera, some spurious features are always detected on the background depicting floating particles in the water. One of the adaptations to underwater images made in this project consists of computing a segmentation image, which allows to segment and eliminate those erroneous 2D correspondences. In figure 4 the green correspondences are the ones kept, while the red ones have been categorized to be part of the background. The segmentation is done on the color images, based on a color sample taken from a region, which contains water in all the images. A statistical analysis of the rgb color vectors of the sample determines the range of colors considered to be water.

Once the errorneous background correspondences have been removed, the epipolar geometry - or more precisely - the essential matrix is determined based on the correspondences. The essential matrix $\mathbf{E}$ is a relation for the 2D-2D correspondence m_1, m_2 [4]:

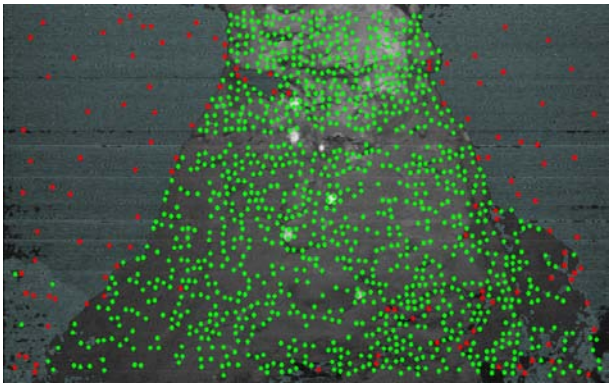

Fig. 4. Errorneous red points are eliminated based on background/foreground segmentation.

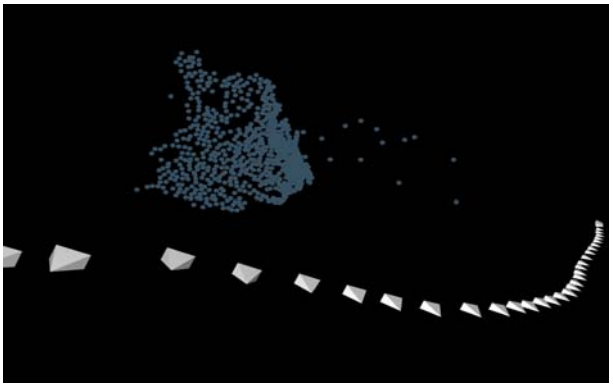

Fig. 5. Reconstructed camera path and sparse 3D point cloud.

$$m_2^T \mathbf{E} m_1 = 0. \quad (1)$$

Relating the 2D-2D correspondences in this way, the essential matrix captures the relative pose between two views.

The computation of the essential matrix is done using the RANSAC algorithm [3] in order to deal with outliers in the 2D-2D correspondences. However, in this project additional navigation data poses exist. The RANSAC algorithm receives the essential matrix compatible to these poses as a guess and they are refined with respect to the exact feature correspondences. Because of the segmentation step eliminating errorneous correspondences, this works accurately. If the refined navigation data poses provide a solution accurate enough, the RANSAC is terminated. If not, the RANSAC determines a more accurate essential matrix based solely on the 2D-2D correspondences, yielding the corresponding camera poses. That means that if no navigation data exists or if it is inaccurate, the essential matrix will be computed nonetheless.

After the poses of the first set of cameras are known, a sparse cloud of 3D points is triangulated. Subsequently, the other images are incorporated sequentially with a 2D-3D pose estimation [4]. Once again, the navigation data poses are tried as guesses for the RANSAC routine. They are refined to fit the 2D-3D correspondences and discarded if found to be too inaccurate. As the reconstruction advances along the camera trajectory, more 3D points are triangulated whenever new features are found. After the whole camera path has been computed, the sparse 3D point cloud already provides an approximation of the 3D structure to be reconstructed. Since the image sequence is dense, some camera poses and 2D points provide only little new information, but adding up to use vast amounts of memory. Therefore, only camera poses that provide enough innovation - meaning there is enough baseline between them - are kept in the camera path. This procedure is described in more detail in [1].

After all the cameras have been added to the reconstruction, a bundle adjustment step [4] is applied to optimize all rays connecting a 3D point with a camera center running through the corresponding 2D point. In addition, the intrinsic camera parameters are optimized, correcting a potential inaccuracy in the initial intrinsic parameters like focal length.

If the navigation data poses did not provide guesses accurate enough, the scale of the scene is unknown because based on images solely, it is impossible to distinguish between for example toy cars or real cars. Provided that the navigation data poses have served as successfull guesses in the RANSAC routines, the scene has approximately the correct absolute scale. However, the poses were optimized to fit to the 3D point cloud more accurately. Therefore, an extra step is needed to fix the absolute scale in both cases. After the SfM algorithm, all that is needed to fix the absolute scale is a scaling transformation for the whole scene. By considering all camera positions and comparing the computed ones with the navigation data poses, such a scaling transformation is computed. See figure 5 for a reconstruction result.

B. Dense Depth Estimation and 3D Surface Model

During the SfM algorithm, the reconstructed 3D structure only consists of a sparse 3D point cloud (see figure 5). Because of occasional outliers in the point cloud and an unknown coherence between the 3D points, a classical triangulation like the Delaunay triangulation does not always yield satisfying 3D models. That is why we propose an intermediate step of computing dense depth maps for each image that has been kept in the camera path reconstruction of the SfM part of the algorithm.

Depth maps contain the distance between the center of projection of the camera and the 3D point in the scene. Up until now, 3D points have only been computed for features, now the depth information needs to be computed for each pixel. This is possible because at this point, the camera poses of several neighboring images are known and can all be used for depth estimation.

When computing the depth map for one view (the masterview), the others are used in pairwise stereo to produce depth maps with the masterview. In order to compute the depth map, each stereo pair is rectified. Rectification [1] is the process to align an image pair such that both views have only horizontal parallax. On the rectified images, either a local sum-of-absolute-differences method (time efficient) or a

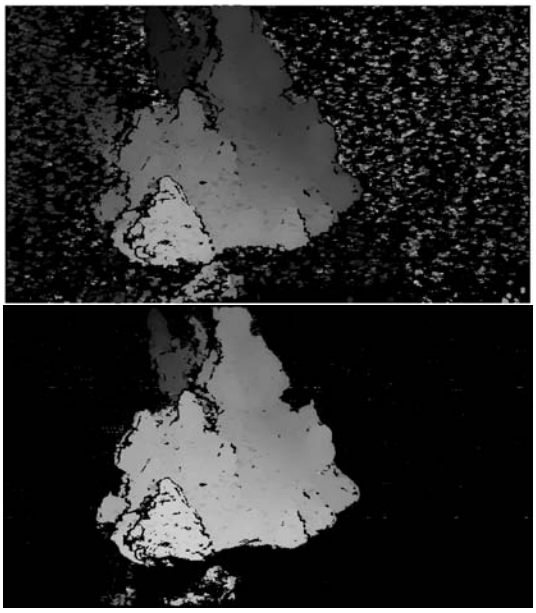

Fig. 6. Top: unfiltered depth map shows distance between 3D point and camera center (darker means further away), bottom: depth map with background filter.

global stereo dynamic programming method [1] (high quality depth maps) is used to calculate the corresponding disparity map. The time efficient local method allows the computation of previews of the surface model by transferring the dense depth estimation to the GPU. Because of the rectification, the matches are found on the horizontal parallaxing lines resulting in a one-dimensional search problem, which is then solved by either one of the methods. If the model needs to be detailed and of high quality, the depth estimation is done by the global approach, which yields far denser and smoother depth maps than the local approach on the GPU. The set of depth maps produced by either one the methods, is fused with the others, so that a detailed, robust depth map is computed for each masterview.

In the underwater case a lot of depth values are estimated in parts of the images that contain only water and are therefore erroneous depths. In order to eliminate these errorneous depths, the same segmentation approach described in the section above is applied to the depth maps. Results of this segmentation can be seen in figure 6

Once a depth map is computed for each image, 3D surface models can easily be generated from the depth maps and then combinded into one model. For a better visualization, the colors need to be mapped onto the model.

C. Color Correction

Colors of underwater images are dominated by a strong green or blue hue. This is due to the strong wavelength dependent attenuation of the different colors. In order to provide an accurate visualization of a scene, it is necessary to remove the green or blue hue and try to reconstruct the real colors of the object in question. Color correction is done by using a physical model for underwater light propagation. A light ray from the object travels through a water body to the camera (along the line of sight, short: LOS). On its way it can be absorbed or scattered on water molecules or other matter in the water, depending on its wavelength. In addition, other rays can be scattered into the line of sight adding to the amount of light coming from the object (for a more detailed discussion refer to [9]).

That suggests that the light reaching a pixel of the camera's CCD sensor consists of two parts: one coming from the object directly called signal and one being scattered into the LOS, called backscatter. Similar models for light propagation are commonly used in the literature, see for example [17] and [14].

The signal S_λ^{ij} is the attenuated value of the object color $L_{obj_\lambda}^i$:

$$S_\lambda^{ij} = L_{obj_\lambda}^i e^{-2z^{ij}\eta_\lambda}, \tag{2}$$

where the 3D point i is observed from camera j. λ depicts the wavelength, but is modeled in three discrete color channels (rgb). z^{ij} is the distance between camera center j and 3D point i. η_λ^{ij} is the attenuation coefficient for the wavelength λ of that particular water body.

The backscatter is amplified with increasing distance between camera center and object as there is a higher chance of light being scattered into the line of sight, the longer the distance is:

$$B_\lambda^{ij} = B_{\infty_\lambda}(1 - e^{-\eta_\lambda z^{ij}}), \tag{3}$$

where B_{∞_λ} is the color value of channel λ of the so called veiling light as defined in [17]. It is the color that can be observed at the background, where no structure is visible, just water.

Both components are added to define the value of the pixel measured:

$$I_\lambda^{ij} = \alpha_\lambda(S_\lambda^{ij} + B_\lambda^{ij}), \tag{4}$$

where α_λ is a camera parameter that is applied to each channel separately. α_λ depicts the white balance, being a multiplier to each channel, which is applied by the camera to account for the color of light.

Figure 7 shows how the light of different wave lengths on the different channels is attenuated or scattered into the LOS respectively. One characteristic of water is that the red colors are attenuated stronger than green and blue. This characteristic can be modeled through η, which has been set to $\eta_r = 0.4$ in the example, while $\eta_g = 0.1$ and $\eta_b = 0.05$. The veiling light color has been set to $B_{\infty_r} = 0.1$, $B_{\infty_g} = 0.2$, and $B_{\infty_b} = 0.5$. The real color of the object considered was $L_{obj} = 1.0$ (white) for all three channels. As can be seen, the red part of the light is covered by the backscatter component after traveling through water for about $3m$. $\eta_r = 0.4$ is a fairly high value for a real water body, but it clarifies the challenge of recovering the colors. It becomes clear that the recovery of the red colors is limited to a few meters, which should be sufficient to cover

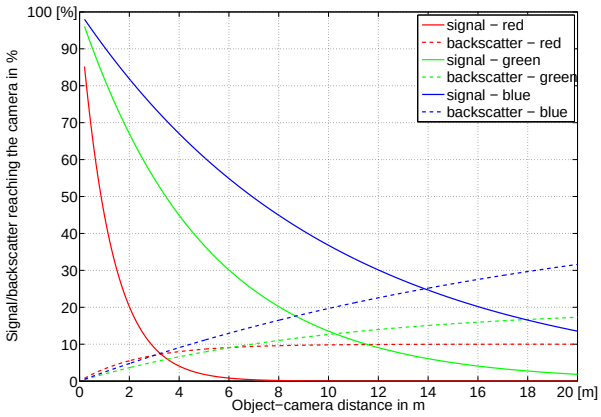

Fig. 7. Attenuation and backscattering on the different color channels depending on the distance traveled through water to reach camera. Solid lines depict attenuation, dashed lines backscatter.

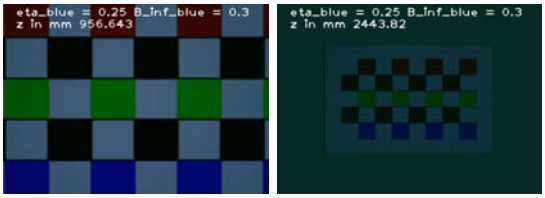

Fig. 8. Simulation of viewing a colored checkerboard underwater from different distances.

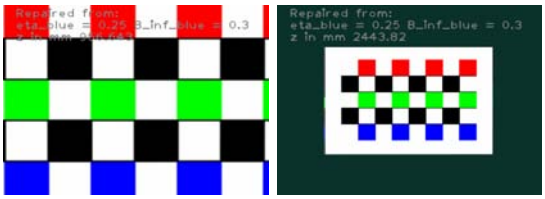

Fig. 9. Colore correction applied to synthetic underwater checkerboard images from figure 8.

interesting close up frames of black smokers, ship wrecks, etc. If the scene in question is further away from the camera, the color correction task quickly becomes impossible. Figure 8 shows a simulation of a colored checkerboard underwater at two distances from the camera. One can observe that the red color at the top is far more strongly attenuated, then the blue and green. Figure 9 shows the same images as in figure 8, but with corrected colors.

The model leads to the following assumption: a 3D point i of an object has a constant color, however, viewed from different points of view, the color changes with the distance between 3D point and camera due to the water body in between. Hence, different camera positions with respect to the 3D point yield different color measurements S_λ^{ij}, which depend on the model. Knowing which points are white - see lime parts on black smoker or checkerboard - means that one knows the color of $L^i_{obj_\lambda}$. A set of white points seen from several images and different distances can be used to estimate all three parameters for each channel. This can be done by optimizing the following error function:

$$\epsilon_\lambda = argmin_{\alpha_\lambda,\eta_\lambda,B_{\infty_\lambda}} \left(\sum_i \sum_j I_\lambda^{ij} - \alpha_\lambda (1^i_\lambda e^{-2z^{ij}\eta_\lambda} + B_{\infty_\lambda}(1 - e^{-\eta_\lambda z^{ij}})) \right). \quad (5)$$

Each channel is optimized separately. The initialization for the parameters is fairly straight forward: B_∞ is sampled from the water background of the images directly. α is set to 1 for each channel and η is set to 0.1 for fairly clear water. Usually the red channel is attenuated stronger then the other two, however we found that conditions of the water are very different, owing the special composition of floating particles in the water body present, so it is difficult to find a better guess.

Once the texture colors have been corrected, the 3D model needs to be visualized by a 3D viewer. We use an OpenGL-based viewer that allows the user to view the reconstructed scene interactively. Since the texture colors have been repaired, the object can be viewed without the attenuation or - if wished, the effects can be simulated explicitly, which means that if the user is close to the object, the colors are less dominated by green and blue than if he moves further away, providing a far more realistic visualization of the underwater scene, than the non-color corrected object alone.

Having explained the algorithm, the next section exhibits some results.

V. Results

In this section some results generated with the proposed system are presented.

A. Color Correction on Synthetic Data

The color correction algorithm has been applied to synthetic data in order to test its performance and robustness. A random set of 3D points has been generated and their color has been set to white. Then the 3D points were viewed from different distances, with the effect of the water being simulated. The optimization routine was then used to retrieve the parameters for η, B_∞, and α.

Figures 10 and 11 show the results. Two testing scenarios are depicted. In 10 the number of points has been changed, testing the robustness, when only very few white points have been used. For this test noise with a magnitude of 1% has been added to the colors. The top image shows the results of estimating η, the bottom the results of estimating the veiling light color. As can be seen, the estimation has been fairly stable as long as there where more than 100 points.

In the second test case (figure 11), the distance deviation between the closest and farthest point has been varied. In general the closest point has always been $3m$ away from the camera centers. The maximum distance has been varied between $3.5m$ and $12m$ causing a difference between $0.5m$ and $9m$ as depicted in the figures. Figure 11 shows that

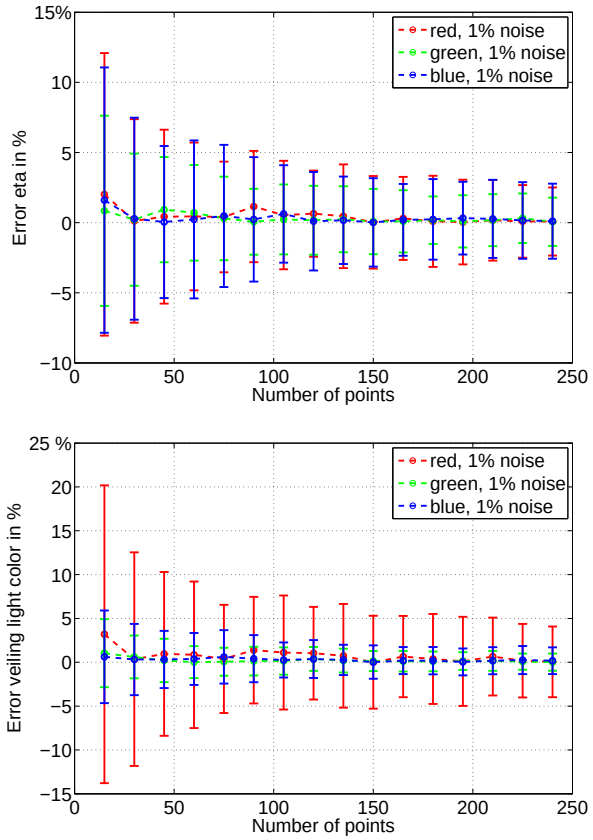

Fig. 10. Estimation of parameters depending on the number of points. Top: error when estimating η, bottom: error when estimating the veiling light color.

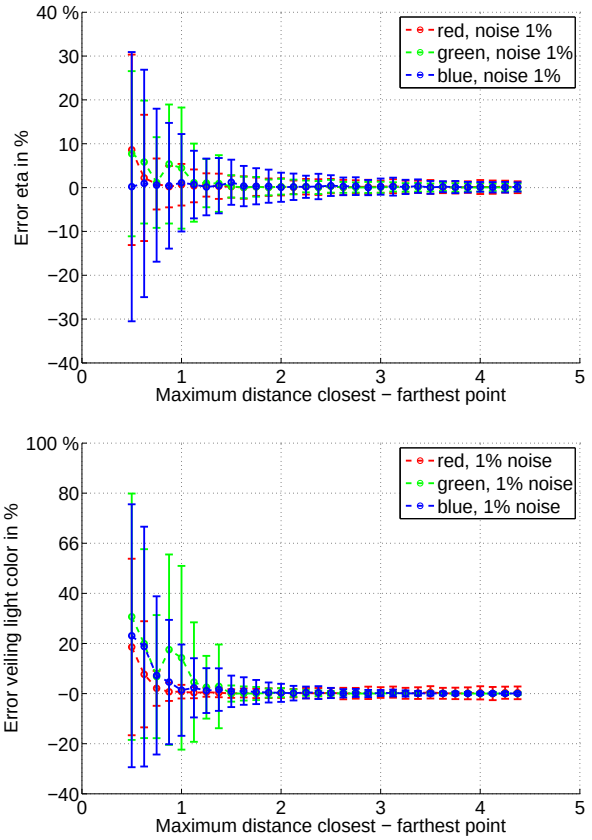

Fig. 11. Estimation of parameters depending on distance between closest and farthest point. Top: error when estimating η, bottom: error when estimating the veiling light color.

the estimation of η (top image) and the estimation of the veiling light color (bottom image) generally become more stable with growing distance between the points considered. This is not surprising, since increasing distance differences lead to increasing differences in color, therefore resulting in a more stable estimation of the physical parameters involved.

The long term goal being a fully automatic estimation of the parameters, up until now some assumptions need to be used in the optimization process - the most important example are the known white parts of the black smoker.

B. Real Data

Real image sequences captured by the ROV's HDTV camera from a black smoker have also been used to test the system. Two subsequences were fed into the algorithm. They both show the same black smoker, one sequence the upper part, the other the bottom. Figures 12 and 14 show input image, exemplary depth map, and screenshot from 3D surface model for the upper part and the bottom part respectively. As can be seen on the surface model, the global depth estimation method delivers detailed and accurate depth maps, which can be used for triangulation directly.

Figures 13 and 15 show the resulting 3D models with color corrected texture. The color correction parameters have been estimated based on the images of the top part (figure 13) containing large areas of lime, thus fullfilling the assumption of known white color. The 3D surface of the model in figure 13 is based on the local depth estimation using a shiftable window implemented on the GPU. This very fast method delivers results not as accurate as the dynamic programming approach in figure 15, but still good enough to give an accurate impression of the scene.

Figure 16 shows the 3D reconstruction of a shipwreck that is based not on video, but single photos. Imaging conditions were difficult in this case - visibility was less than $1m$. Hence, the ship wreck of several meters could not be observed in one photograph. So in this case, the 3D reconstruction provides a crucial visualization for divers and non-divers alike. There was no data containing known white structures, so the automatic parameterization for the light propagation model could not be done. Therefore, the color correction has been done with a manual tool in this case (see figure 16 bottom).

VI. Conclusion and Future Work

We have presented a system for 3D reconstruction based on underwater video that requires no special equipment. 3D models are computed complete with color correction and can be interactively visualized in a 3D viewer. In the future, we are expecting new sequences to analyze and the system is going

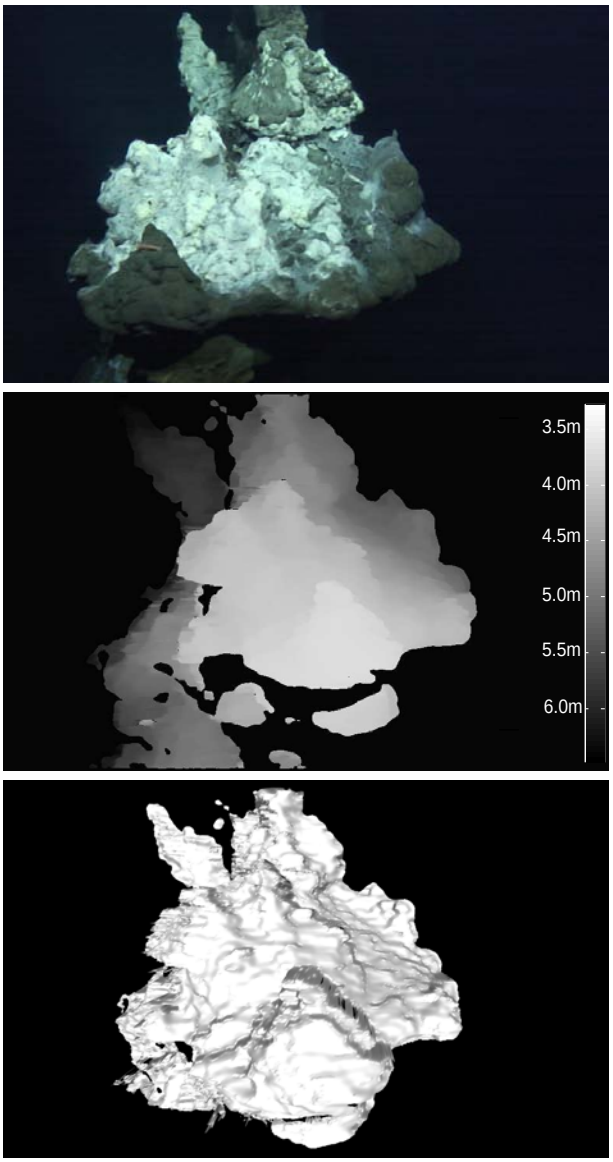

Fig. 12. Top: one of the input images showing top part of a black smoker at $4^\circ 48' S$ and $12^\circ 23' W$ (Atlantic Ocean) at depth $3039m$, middle: dense depth map for one image (shows distance from camera to 3D point for each pixel: darker means further away), bottom: untextured 3D surface model.

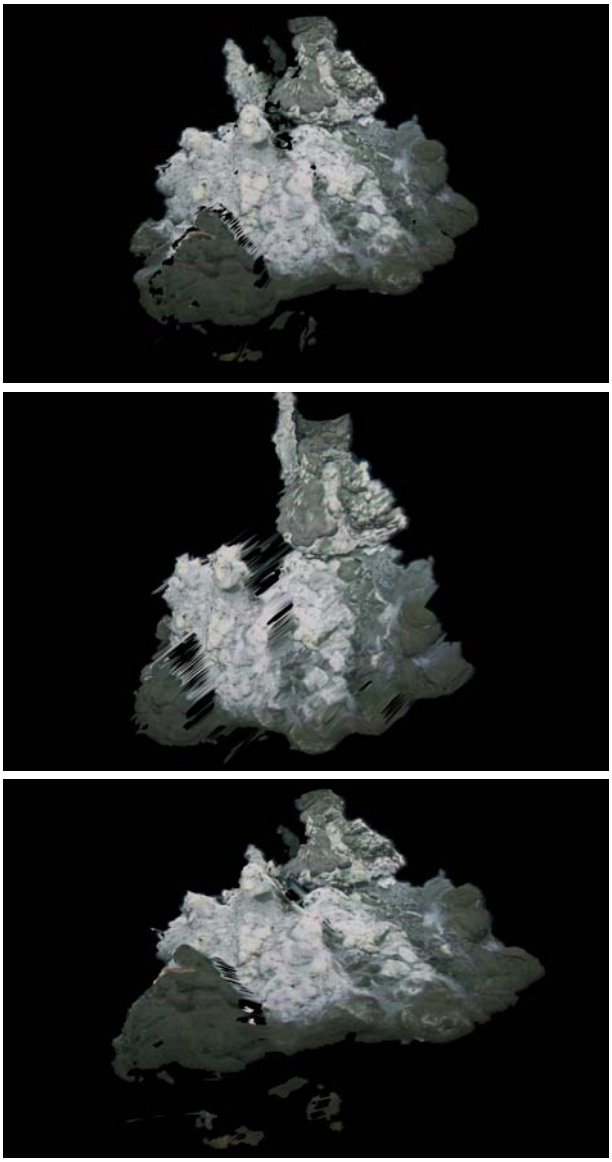

Fig. 13. 3D model with corrected color from 3 different points of view.

to be extended to adapt even more to the underwater imaging conditions. That includes a self-calibration approach, merging of models based on subsequences, a more robust estimation of the color correction parameters without the assumption of having known white color in the image, and special treatment of known moving objects like animals for example fish and crabs, floating particle, and smoke.

ACKNOWLEDGMENT

This work has been funded by the German Science Foundation (DFG within the Future Ocean Excellence Cluster). The authors would like to thank the team operating the ROV Kiel 6000 for capturing the image sequences shown here and taking their time to calibrate the camera in both: water and air. We would also like to thank Florian Huber and the Institute of Prehistoric and Protohistoric Archaeology of the Kiel university for the shipwreck images.

REFERENCES

[1] B. Bartczak, K. Koeser, and R. Koch. Extraction of 3d freeform surfaces as visual landmarks for real-time tracking. *Journal of Real Time Image Processing*, 2:81–101, 2007.

[2] V. Brandou, A.G. Allais, M. Perrier, E. Malis, P. Rives, J. Sarrazin, and P.M. Sarradin. 3d reconstruction of natural underwater scenes using the stereovision system iris. In *Proc. OCEANS 2007 - Europe*, pages 1–6, 2007.

[3] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[5] A. Hogue, A. German, J. Zacher, and M. Jenkin. Underwater 3d mapping: Experiences and lessons learned. *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, pages 24–24, 2006.

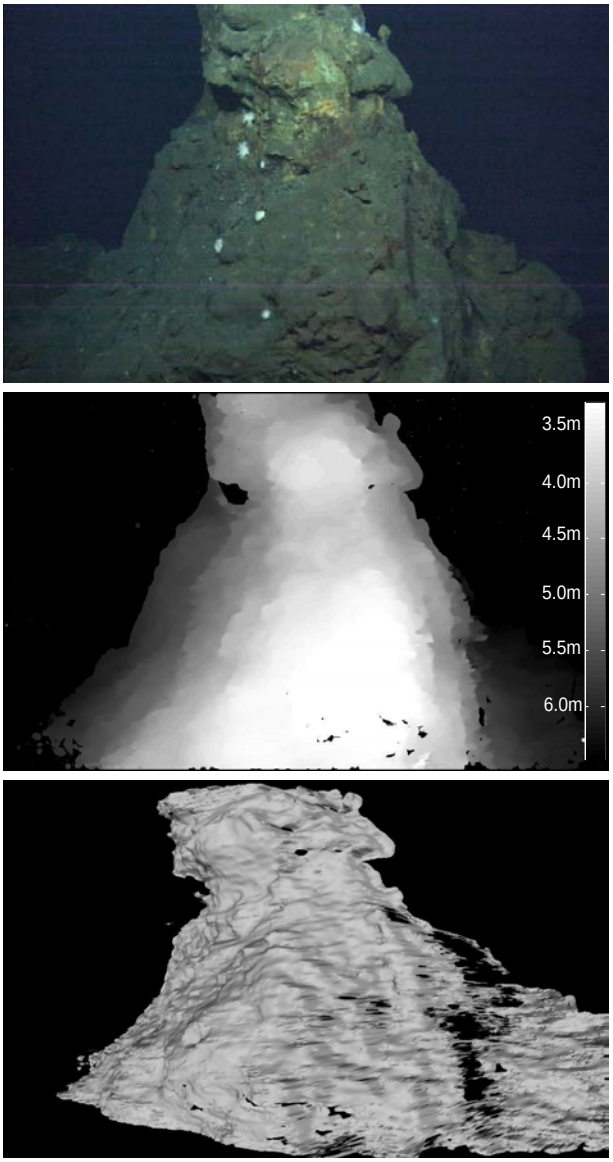

Fig. 14. Top: one of the input images showing bottom part of a black smoker at $4^\circ 48' S$ and $12^\circ 23' W$ (Atlantic Ocean) at depth $3039m$, middle: dense depth map for one image (shows distance from camera to 3D point for each pixel: darker means further away), bottom: untextured 3D surface model.

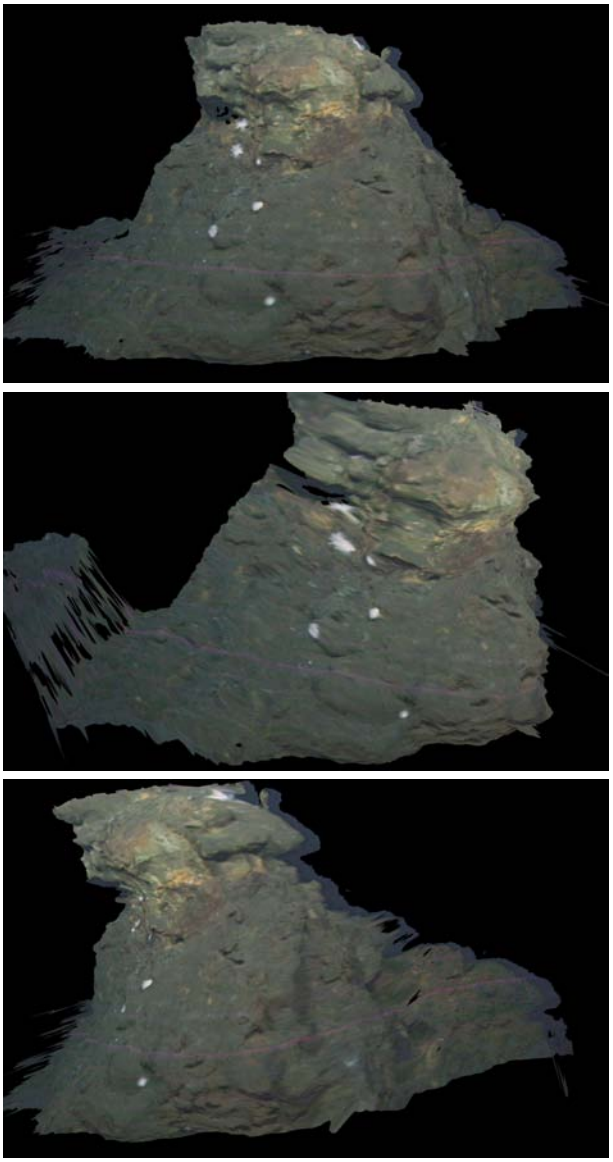

Fig. 15. 3D model with corrected color from 3 different points of view.

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

[7] Jean-Marc Lavest, Gérard Rives, and Jean-Thierry Lapresté. Underwater camera calibration. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 654–668, 2000.

[8] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, 1981.

[9] Curtis D. Mobley. *Light and Water: Radiative Transfer in Natural Waters*. Academic Press, 1994.

[10] N. Pessel, Jan Opderbecke, and Marie-Jos Aldon. Camera self-calibration in underwater environment. In *WSCG*, 2003.

[11] O. Pizarro, R. Eustice, and H. Singh. Large area 3d reconstructions from underwater surveys. In *Proc. OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, volume 2, pages 678–687 Vol.2, 2004.

[12] K. Plakas, E. Trucco, and A. Fusiello. Uncalibrated vision for 3-d underwater applications. In *Proc. OCEANS '98*, volume 1, pages 272–276 vol.1, 1998.

[13] M. Pollefeys, L. van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[14] J. Queiroz-Neto, R. Carceroni, W. Barros, and M. Campos. Underwater stereo. *Computer Graphics and Image Processing, Brazilian Symposium on*, 0:170–177, 2004.

[15] Ingo Schiller, Christian Beder, and Reinhard Koch. Calibration of a pmd camera using a planar calibration object together with a multi-camera setup. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume Vol. XXXVII. Part B3a, pages 297–302, Beijing, China, 2008. XXI. ISPRS Congress.

[16] T. Treibitz, Y. Y. Schechner, and H. Singh. Flat refractive geometry. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008*, pages 1–8, 2008.

[17] N. Karpel Y. Y. Schechner. Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of Oceanic Engineering*, 30(3):570–587, 2005.

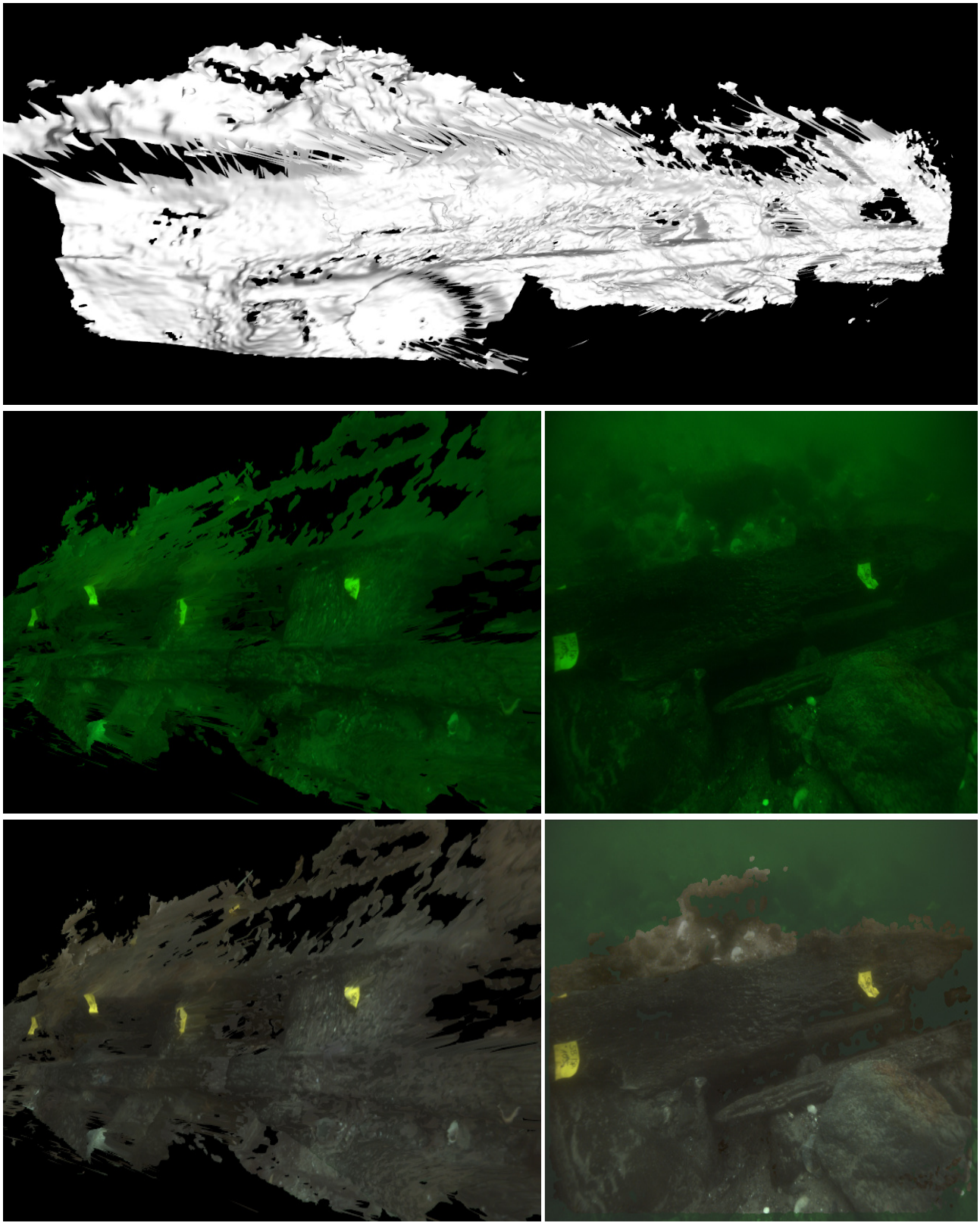

Fig. 16. Reconstruction from single images of a shipwreck found in the Kiel fjord. Top: 3D surface model of the shipwreck, middle: textured model of the shipwreck, bottom: 3D model with corrected color of the shipwreck.