3D Grid Map Transmission for Underwater Mapping and Visualization under Bandwidth Constraints

Tomasz Łuczyński[1], Tobias Fromm[1], Shashank Govindaraj[2], Christian A. Mueller[1], and Andreas Birk[1]

[1]Jacobs University Bremen, Germany, Computer Science & Electrical Engineering, Robotics Group
[2]Space Applications Services NV/SA, Belgium
t.luczynski@jacobs-university.de

***Abstract*— ROV operations are expensive and there is an urge to improve on their efficiency and speed. The EU project "Effective Dexterous ROV Operations in Presence of Communications Latencies (DexROV)" proposes a solution to those needs by developing a set of hardware and software tools to support the teleoperation of ROVs over a satellite link, i.e., from remote locations including off-shore sites. A core challenge is the significant amount of data that must be transmitted to the distant Mission Control Center to allow this. To ease operation in general and especially in this application case involving latencies in the control, a 3D model of the environment that is generated during the mission by sensors on the ROV is to be transmitted. This paper focuses on using a 3D grid map for this purpose and especially its efficient transmission under significant bandwidth constraints.**

I. Introduction

The goal of the EU-project "Effective Dexterous ROV Operations in Presence of Communication Latencies (DexROV) [1]) is to develop software and hardware tools for supporting Remotely-Operated Vehicles (ROV) operations. One of the main goals of the project is to move the ROV operators from the ship to a Mission Control Center (MCC) on shore and to add partial autonomy capabilities to the ROV [1] [2].

This opens multiple opportunities for new ways in which the ROV operations can be organized. However it also creates multiple difficulties that need to be dealt with. One of the biggest limiting factors for the DexROV operations is bandwidth constraints on the satellite link between the MCC and the ship. For best results and highest comfort of work the operator should receive a map of the environment in which the ROV is working. The main focus of this paper is to efficiently and reliably transmit this 3D grid map, which is generated on the ROV to the MCC. For the representation of the 3D grid map, the popular OctoMap [3] software package is used.

II. Bandwidth Limitations

Satellite communications services for mobile offshore maritime operations are associated with high costs, bandwidth limitations (uplink and downlink), inherent delays, disruptions and require a complex stabilized satellite tracking antenna. In the context of DexROV, a maritime VSAT solution is employed from a service provider (Omniaccess) that includes a Ku band Cobham Sailor 800 tracking antenna, its controller and the related modems. The nominal data bandwidth for the uplink from the vessel is 768 kb/s and the downlink to the vessel is 256 kb/s with an inherent nominal round trip delay of 620 ms. To sustain effective remote ROV operations, multiple data flows are required such as ROV commands, video streams, pose updates, 2.5/3D environmental maps, status updates etc. between the onshore and offshore nodes. Hence, it is critical to optimize the bandwidth usage by prioritizing data flows with specific Quality of Service (QoS) information, shaping the traffic [4] to avoid overloading network and ensure data reliability with the minimal overheads. To address the described challenges, DDS OpenSplice (Prismtech) middleware [5] has been used to exchange data between the onshore and offshore nodes over the bandwidth-constrained satellite network.

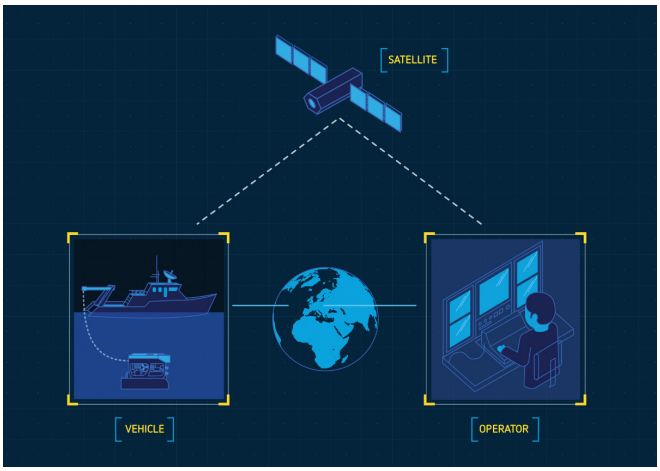

Fig. 1. In the DexROV project the ROV will be operated via satellite link from an onshore control center.

III. The DexROV System

The DexROV project contains an ROV system which is deployed from a vessel in the Mediterranean sea as well as an onshore Mission Control Center used for controlling and monitoring the ROV operations (see Fig. 1).

Emerging from this setup, a collection of three different logical and spatial locations has to be considered: *onboard* the ROV, the operator *vessel* and *onshore* in the MCC. In all theses nodes, specific hardware and data has to be present to enable the respective functionality, for example to generate an environment representation with a 3D grid map:

1) stereo image pair are captured *onboard* the ROV and transmitted to the *vessel*
2) an 3D grid map is generated on the *vessel* from the stereo images and transmitted to the *onshore* MCC through the satellite link
3) the *onshore* MCC displays the map for the pilot to orient him/herself in the environment.

More examples of data to be transmitted between the spatially distant nodes are the ROV's current location (*onboard* $\rightarrow$ *vessel*), the poses of objects autonomously recognized by machine perception (*vessel* $\rightarrow$ *onshore*) or motion commands (*onshore* $\rightarrow$ *onboard*).

The onshore MCC and the ROV control and perception framework embed command and data interfaces in the Robot Operating System (ROS) [6] through asynchronous publish-subscribe mechanisms over named and type-specific topics. A ROS2DDS bridge has been developed which can be configured to interface with existing ROS topics in a system. For each ROS topic, the ROS2DDS bridge automatically creates a corresponding DDS data reader, data writer and named topics across the distributed nodes with associated QoS policies, as per the Object Management Group's DDS QoS specification.

IV. Map Representation

The ROV generates a 3D map based on an underwater stereo system with accurate calibration using the Pinax model [7] and with the help of navigation sensors. The mapping task may be performed in multiple ways from odometry using simple sensors to advanced self-localization and mapping techniques, allowing to trade map quality with required computation power and update rates. However this lies beyond the scope of this paper and will not be discussed here.

Once the map has been built, an efficient representation is required to store the 3D grid map on the ROV for further navigation and as basis for transmission to the MCC. The octree data-structure [8] is well known to be a very efficient basis for a 3D grid map representation [9] and the OctoMap [3] software is a widely used implementation of this data-structure, which is also used here. The octree provides several beneficial properties, which are exploited in this work for the efficient data transmission under bandwidth constraints. The details of this modifications and the whole process will be discussed in the following sections.

V. Color Representation

A. Underwater Light Conditions

Underwater images suffer from exceptionally bad light conditions. Not only is there often a low amount of natural light in the scene, but the present light is heavily attenuated and scattered. Attenuation is responsible for absorbing light with the distance from the source. This absorption depends on the wavelength causing change in perceived color of the observed scene. Scattering may be further divided into two phenomena: forward and back scattering. The underlying physics for both are the same but the source of the scattered signal differs, therefore having different influence on the image. Forward scattering occurs when light reflected from the scene scatters on its way to the camera resulting in slight blur in the image. Backscattering, on the other hand, occurs when ambient light is being scattered and reflected back to the camera, adding a new signal component to the image. The further the scene from the camera, the stronger and more visible the backscattering component gets. It can be compared to observing the environment in the fog.

Underwater image formation model and image improvement methods are described, e.g., in [10], [11]. From the perspective of our mapping task, the colors of each part of the map change depending on the distance between the camera and the scene during registration.

B. Color-optimized OctoMap

Color deterioration may be partially reduced when using an OctoMap. Each voxel, which is the smallest data unit used in an OctoMap, accumulates multiple observations of the same region. These measurements may differ in perceived color. This gives the possibility of selecting the most accurate color to create a map with overall more accurate colors than those in the registered images.

Theoretically, when a constant ambient light illuminates the whole scene equally, colors from the shortest distance should be taken. However there is a more efficient way of achieving the same result: when multiple points fall into the same voxel the brightest color is selected as the voxel color. With this simple measure we guarantee that the map colors are taken from points when the camera was close to the surface and/or the additional lights (if available) were in the optimal position to properly illuminate the given patch of the surface.

C. Color Assignment Evaluation

In order to present the results of the proposed method for assigning colors to the voxels, the following experiment has been conducted, producing the results of Fig. 2: The same data was accumulated in the OctoMap twice. The first run was performed using the method proposed here and the second time with a naive implementation – a voxel takes the color of the last point falling into the given voxel. The data used here was recorded in Biograd na Moru in Croatia in 2015 within the EU project "Marine robotic system of self-organizing, logically linked physical nodes (MORPH)". The AUV was moving near the surface recording the seabed. It is easy to observe that there are sharp lines on the edges between the subsequent stereo scans, whereas in the color-optimized OctoMap these differences are not visible (compare Fig. 2).

VI. OctoMap Transmission

A. Compression Method

As discussed earlier, representing the environment using an octree has multiple advantages. On the other hand, this also creates some challenges for the DexROV application scenario. After each registration of a new point cloud from

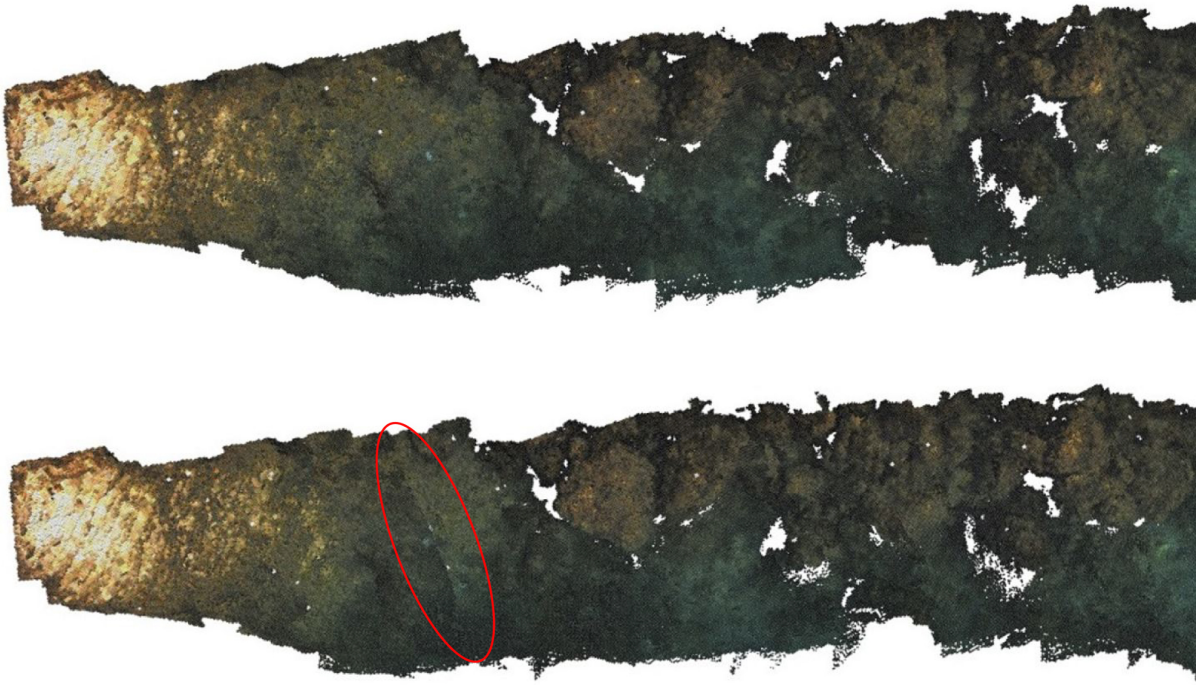

Fig. 2. Top: OctoMap colored with our approach. Bottom: naive implementation, last registered point decides about the color of the given voxel. Please note that our method eliminates the sharp color changes on the edges of registered scans (e.g. marked with a red loop). It also improves the overall appearance and color accuracy when recording inconsistently illuminated environment.

stereo, an updated map would have to be transmitted to the Mission Control Center to give the operator the best knowledge about the environment. Of course this is not acceptable from the perspective of bandwidth limitations. Very often, once the map has been built, new pointclouds typically only change a small part of the 3D map.

It is important to notice at this point that from the perspective of the MCC the OctoMap structure per se is not important either. ROV operators make the decisions based on the visualization, but the navigation happens internally on board of the vehicle. Therefore sending the full OctoMap after registering each new pointcloud is not only impossible from the bandwidth limitation perspective but also unnecessary. Furthermore, for the majority of the map, higher-level voxels (with a low resolution) can just be transmitted as in many regions high resolution is not needed. Only in certain regions specified by the operator the maximum resolution is necessary.

This allows for the following operation scheme:

- The OctoMap is constructed with the maximum possible resolution (with respect to the on-board computer computation capabilities).
- Every time a new pointcloud is registered, the OctoMap is locally updated.
- When integrating the point cloud into the map, it is checked if any voxel has changed its state, i.e., if a previously unknown region is now identified as free/occupied, a voxel that was occupied is now free or the free voxel is now occupied.
- The user has an option to specify the bounding box for a region of interest. Voxels in this region will be transmitted with highest available resolution and with color, voxels outside this bounding box will be transmitted with lower resolution and without color. Following this rule, voxels that have changed their state are added to the list of voxels that will be published with adequate color information and resolution.
- When all the points from the given pointcloud are integrated into the OctoMap, four lists of voxels are being published: newly-occupied voxels with coarse resolution, newly-occupied voxels within the bounding box using fine resolution and the respective lists of newly-unoccupied voxels.

This way, in the beginning of the mission every new region is transmitted to the operator with low resolution. After this first stage of the mission, when the ROV flies over already known terrain, no data is transmitted unless high resolution for some region is needed or the environment has changed. At the same time the log odds in the OctoMap are constantly updated on board of the vehicle. Of course the MCC may request the transmission of the full OctoMap with highest possible resolution. This operation will take more time but is only triggered when online visualization is not needed, e.g. after completing the mission, for the documentation, or to (re)plan the mission.

B. Satellite Link Configuration

Communication between the mapping node on the vessel and the MCC happens over the satellite link and is supported by our ROS2DDS bridge. Configurations to setup the

```
"RovOctomap Ros Publisher":
{
        "enable": "true",
        "dataType": "Octomap",
    "streamType": "RosPublisher",
        "topics":
             {
         "ros": "/rov/octomap",
                 "dds": "octomap"
         },
    "qos":
    {
        "reliability": {
                "kind": "RELIABLE_RELIABILITY_QOS",
                "max_blocking_time.sec": "0",
                "max_blocking_time.nanosec": "0",
                "synchronous": "true"
                },
        "durability":   {
                "kind": "TRANSIENT_LOCAL_DURABILITY_QOS"
                },
        "durability_service": {
                "history_depth": "1",
                "history_kind": "KEEP_LAST_HISTORY_QOS",
                "max_instances": "-1",
                    "max_samples": "-1",
                    "max_samples_per_instance": "-1"
                },
        "latency_budget": {
                "duration.sec": "0",
                    "duration.nanosec": "0"
                },
        "history":      {
                "kind": "KEEP_ALL_HISTORY_QOS",
                "depth":"1"
                },
        "resource_limits": {
                "max_instances": "-1",
                "max_samples": "-1",
                "max_samples_per_instance": "-1"
                },
        "transport_priority": {
                "value": "96"
                },
        "lifespan":     {
                "duration.sec": "2147483647",
                "duration.nanosec": "2147483647"
                },
        "destination_order": {
                "kind": "BY_RECEPTION_TIMESTAMP_DESTINATIONORDER_QOS"
                }
    }

},
```


Fig. 3. ROS2DDS configuration with QoS for transmitting an OctoMap.

ROS2DDS bridge can be scripted within a JSON file, which reflects the mapping between the ROS and DDS data flows. Some of the main QoS policies that are of interest in the context of DexROV are (i) reliability – best effort and reliable types defined by the DDS wire protocol, (ii) durability, (iii) latency budget, (iv) resource limits, (v) lifespan (vi), destination order. A sample of the configuration for publishing OctoMaps from the offshore ROS2DDS bridge is shown in Fig. 3 with the corresponding QoS policies.

The current architecture of the bridge is scalable to deploy multiple nodes with dynamic discovery of distributed ROS2DDS entities. The maximum burst sizes indicate the amount in bytes to be sent at maximum every "resolution" milliseconds. With "reliable" reliability QoS (guaranteed data delivery), the maximum burst values are typically set just below the maximum bandwidth available for the uplink from the offshore side. For example, in the presence of a 768 kb/s satellite uplink, the maximum burst size of 650 kb/s (around 85% of the bandwidth) was found to be efficient. The remaining bandwidth is made available for retransmissions of data packets. In the case of "best effort" reliability QoS (not necessary to re-send or acknowledge any received packets), the maximum burst size can be set to the available bandwidth. During the marine trials, the satellite link is shared between multiple data flows that are assigned different priorities and the maximum burst sizes are adapted proportionately.

There are two options to configure the satellite network for communication between the onshore and offshore nodes using global static IP addresses on both sides – (i) a VPN tunnel can be established between the nodes with built-in encryption using VPN enabled routers or (ii) ROS2DDS can utilize the encryption methods that are configured within the OpenSplice DDS framework to ensure point-to-point communication between the nodes. The solution of using a lightweight encryption method was found to be more efficient than a VPN as the VPN link management headers consume additional data in the already constrained bandwidth.

VII. Effectiveness Validation in Simulation

A. Scene Modeling

Within the DexROV framework, a simulation environment has been developed and utilized in a continuous integration manner [12]. The system includes a simulated ROV equipped with sensors as well as simulated textured environment.

In order to allow for the operator to quickly perceive the current vehicle view, occupied areas, and autonomously recognized objects, we use a simulated depth camera beneath a simulated RGB camera (Fig. 4(c) & (e)) to autonomously recognize and localize objects in the scene (Fig. 4(f)) using different texture and shape-based machine recognition methods [13] [14]. The same RGBD input is used to generate the OctoMap, which is used for autonomous collision avoidance and coarse-grained scene representation (Fig. 4(d)).

With the help of this testbed, we can evaluate our methods with ground truth data. Additionally, stress testing is facilitated with respect to noise level, bandwidth and computational limitations. This allows for extensive testing and improvement cycles before field trials.

B. Bandwidth Constraint Modeling

In order to simulate the bandwidth constraints imposed on the scenario as described in Section II, we deploy our software components in one Docker [15] container each for the ROV, the operator vessel and the MCC stations. These containers are interfaced with a network simulator [16] based

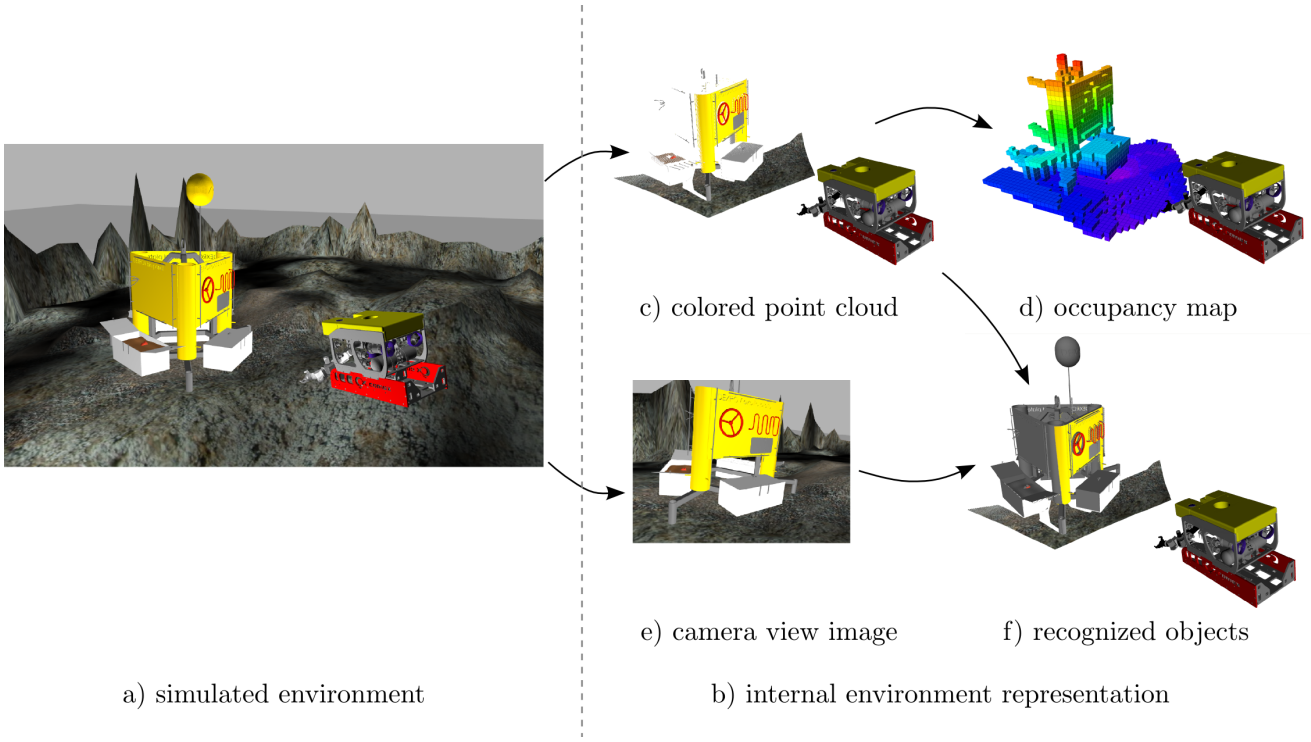

Fig. 4. Scene Modeling Overview [12] – find a video showing our ROV modeling a simulated scene on `https://youtu.be/pcYAgYt65Bc`

on netem [17] which allows for dynamic changes of several parameters:

- available bandwidth
- delay (with variance)
- package loss percentage
- package corruptness percentage.

Setting these parameters on realistic limitations as given in Section II reflects the actual system in a realistic way.

C. Results

Experiments were conducted as follows: stereo point-clouds and navigation data are recorded in the simulator. The ROV is teleoperated to first explore a small region of the seabed and then move around this already known environment. This scenario allows to evaluate the bandwidth usage when building a map and during update of an existing one. Furthermore, as mentioned in Section VI, it is possible to use non-colored and low-resolution voxels in some regions of the map; during this experiment, however, the whole map was treated as a priority region and transmitted with the highest possible resolution and with color. The bandwidth used for this transmission is compared to transmitting the same map as a full OctoMap – each time a new scan is integrated the full OctoMap is transmitted.

This experiment was conducted twice on the same dataset: once with the OctoMap working on 10 cm resolution and the second time on a 15 cm resolution. Fig. 5 shows how the bandwidth usage is changing over time, when transmitting the full OctoMap for both resolutions. Fig. 6 shows a comparison of the same scenarios, only this time using our proposed method. Finally Fig. 7 presents a direct comparison of our method and a transmission of the full OctoMap using 15 cm resolution.

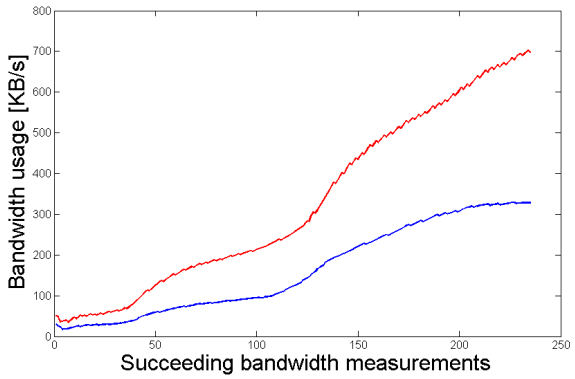

Fig. 5. Comparison of bandwidth usage for 10 (red) and 15 (blue) cm resolution OctoMap. In both cases the full OctoMap was transmitted after each scan (classic transmission scheme).

The incremental update obviously clearly outperforms the continous retransmission of the map. More, importantly, our overall implementation is suited to be used in the context of remote ROV control under high bandwidth constraints.

VIII. Conclusions

We presented work to support remote ROV control from a on-shore control center. To ease mission planning and control, a 3D grid map is transmitted from the ROV to the operation center. The 3D grid map is based on an

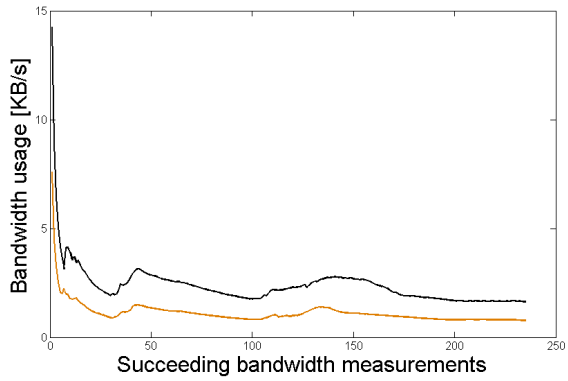

Fig. 6. Comparison of bandwidth usage for 10 (black) and 15 (orange) cm resolution OctoMap. In both cases the our transmission scheme was used.

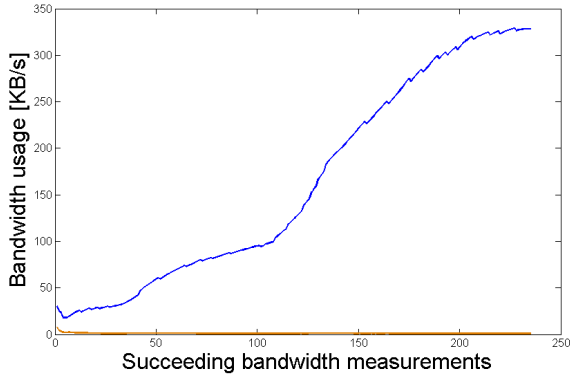

Fig. 7. Direct comparison of our method (orange) and transmission of the full OctoMap (blue) using 15cm resolution.

octree data-structure implemented in the OctoMap software. The OctoMap is augmented with color information by us, for which a simple but efficient strategy for getting the best color value is used, namely employing the brightest - and hence most likely nearest - color value for a voxel if multiple data is available. A strategy for incremental update of the 3D grid map at the mission center was developed, using among others support for a region of interest, and implemented including among others with the development of a ROS2DDS bridge between the Robot Operating System (ROS) and the DDS OpenSplice (Prismtech) middleware for the satellite communication.

Acknowledgments

The research leading to the presented results was supported in part by the European Community's Horizon2020 Programme under grant agreement n. 635491 "Dexterous ROV: effective dexterous ROV operations in presence of communication latencies (DexROV)".

References

[1] J. Gancet, P. Weiss, G. Antonelli, M. Pfingsthorn, S. Calinon, A. Turetta, C. Walen, D. Urbina, S. Govindaraj, P. Letier, X. Martinez, J. Salini, B. Chemisky, G. Indiveri, G. Casalino, P. di Lillo, E. Simetti, D. de Palma, A. Birk, T. Fromm, C. Mueller, A. Tanwani, I. Havoutis, A. Caffaz, and L. Guilpain, "Dexterous Undersea Interventions with Far Distance Onshore Supervision: the DexROV Project," in *IFAC Conference on Control Applications in Marine Systems*, 2016.

[2] J. Gancet, D. Urbina, P. Letier, M. Ilzkovitz, P. Weiss, F. Gauch, B. Chemisky, G. Antonelli, G. Casalino, G. Indiveri, A. Birk, M. F. Pfingsthorn, S. Calinon, A. Turetta, C. Walen, and L. Guilpain, "DexROV: Enabling effective dexterous ROV operations in presence of communication latency," in *OCEANS 2015 - Genova*, May 2015, pp. 1–6.

[3] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at http://octomap.github.com. [Online]. Available: http://octomap.github.com

[4] M. Karaliopoulos, R. Tafazolli, and B. G. Evans, "Providing differentiated service to tcp flows over bandwidth on demand geostationary satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 2, pp. 333–347, Feb 2004.

[5] "Limited-Bandwidth Plug-ins for DDS - Integrating Applications over Low Bandwidth, Unreliable And Constrained Networks using RTI Data Distribution Service," Real-Time Innovations - RTI. [Online]. Available: https://www.rti.com/hubfs/docs/DDS_Over_Low_Bandwidth.pdf

[6] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[7] T. Łuczyński, M. Pfingsthorn, and A. Birk, "The pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings," *Ocean Engineering*, vol. 133, pp. 9 – 22, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0029801817300434

[8] D. Meagher, "Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer," Rensselaer Polytechnic Institute, Tech. Rep., 1980.

[9] ——, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, p. 129–147, 1982.

[10] J. S. Jaffe, "Computer modeling and the design of optimal underwater imaging systems," *IEEE Journal of Oceanic Engineering*, vol. 15, no. 2, pp. 101–111, Apr 1990.

[11] R. Schettini and S. Corchs, "Underwater Image Processing: State of the Art of Restoration and Image Enhancement Methods," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 746052, 2010. [Online]. Available: http://dx.doi.org/10.1155/2010/746052

[12] T. Fromm, C. A. Mueller, M. Pfingsthorn, A. Birk, and P. di Lillo, "Efficient Continuous System Integration and Validation for Deep-Sea Robotics Applications," in *OCEANS 2017 - Aberdeen*, 2017.

[13] H. Bülow and A. Birk, "Spectral 6-DOF Registration of Noisy 3D Range Data with Partial Overlap," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 954–969, 2013.

[14] C. A. Mueller and A. Birk, "Hierarchical Graph-Based Discovery of Non-Primitive-Shaped Objects in Unstructured Environments," in *International Conference on Robotics and Automation*, May 2016.

[15] D. Merkel, "Docker: Lightweight Linux Containers for Consistent Development and Deployment," *Linux Journal*, vol. 2014, no. 239, Mar. 2014.

[16] M. Pfingsthorn, *Docker Networking Simulation*, Jacobs University, 2016, https://github.com/maxpfingsthorn/mini-network-simulator.

[17] "Netem," The Linux Foundation. [Online]. Available: https://wiki.linuxfoundation.org/networking/netem