2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# Enhancing Inland Water Safety: The Lake Constance Obstacle Detection Benchmark

Dennis Griesser[1], Matthias O. Franz[1], and Georg Umlauf[1]

***Abstract*— Autonomous navigation on inland waters requires an accurate understanding of the environment in order to react to possible obstacles. Deep learning is a promising technique to detect obstacles robustly. However, supervised deep learning models require large data-sets to adjust their weights and to generalize to unseen data. Therefore, we equipped our research vessel with a laser scanner and a stereo camera to record a novel obstacle detection data-set for inland waters. We annotated 1974 stereo images and lidar point clouds with 3d bounding boxes. Furthermore, we provide an initial approach and a suitable metric[2] to compare the results on the test data-set. The data-set is publicly available[3] and seeks to make a contribution towards increasing the safety on inland waters.**

## I. INTRODUCTION

On inland waters, a wide variety of vessels share the same traffic space. Compared to road traffic, the traffic situation on inland waters is often confusing, because there are no lane markings. Maneuvers of others are often difficult to estimate due to wind and current. As a consequence, collisions occur more frequently. In the process, individuals are injured and significant financial losses and environmental pollution occur. According to a recent press release, on Lake Constance, where our data-set was recorded, there were 40 accidents caused by collisions in 2022 [6]. Collisions are the most frequent type of accident in this context. These are usually caused by inattention and inexperience of the skippers. Some of the accidents happen in good weather conditions and best visibility. Therefore, a system to assist navigation and warn about collisions aims to increase inland water traffic safety.

Various sensors can be used to implement such systems. Passive sensors like RGB cameras are suitable to locate and classify objects in 2d. Adding a second RGB camera, also distances of the objects can be measured using stereo vision. Active sensors like laser scanners or radar often do not have the resolution of cameras, but can measure distances rather accurately also in adverse weather. Thus for inland waters at close ranges, it is purposeful to combine a laser scanner and at least two cameras for object detection. For maritime scenarios and large range surveillance, also radar should be integrated to such a system. In this paper, we focus on inland waters and close range detection.

[1]Dennis Griesser, Matthias O. Franz, Georg Umlauf are with the Institute for Optical Systems, University of Applied Sciences Konstanz, Germany, {dgriesse, mfranz, umlauf}@htwg-konstanz.de

This research was funded by the Baden-Württemberg Stiftung gGmbH and BMBF (01IS19083A).

[2]https://github.com/dionysos4/lake-constance-obstacle-detection

[3]DOI: 10.48606/112

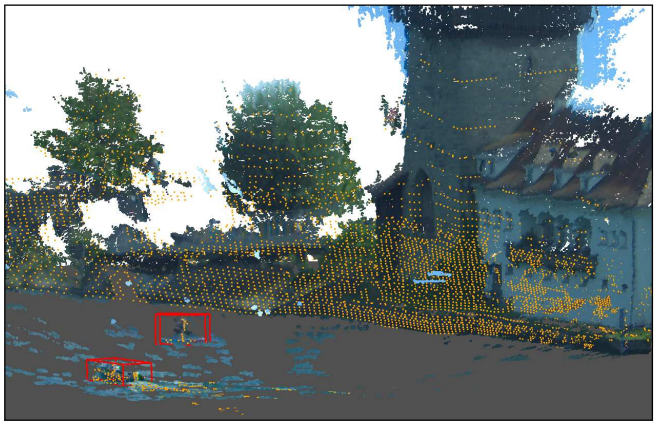

Fig. 1. One sample of our data-set. Shown are the RGB point cloud and the lidar point cloud (orange points). The approximated water surface is plotted as a gray plane. A motorboat (left) and a stand-up paddle (right) are potential collision partners and are marked with red bounding boxes.

Close range detection in inland water scenarios have different challenges than maritime scenarios. As in every other traffic scenario, static and dynamic objects must be detected and classified. However, in maritime scenarios, there are only very few static objects. These are predominantly the sky, the water surface, and a low-detailed shoreline. In inland water scenarios, there are additionally a highly detailed shoreline, stationary traffic, traffic signs, bridges, etc. as e.g. in Fig. 1. For such scenarios, we use a research vessel equipped with two RGB cameras and a lidar sensor.

The data-sets are recorded at Lake Constance on different days. We calibrated the cameras for stereo vision and the lidar to the cameras for data fusion. The data are split into a training and a test data-set. Annotated stereo images and point clouds with 3d bounding boxes are provided. We also provide an initial algorithm to detect 3d bounding boxes. For this purpose, we trained a 2d image detector and used the stereo and lidar range information to obtain the 3d bounding box. In summary, the main contributions of the paper are:

- A sensor setup for close range obstacle detection.
- A novel annotated 3d stereo and lidar data-set for inland water detection to train and test deep learning models.
- A baseline detection method for comparison purpose.

## II. RELATED WORK

For autonomous driving many object detection data-sets with 3d object annotations exist. The most popular ones are KITTI [11] and nuScenes [7]. However, for waterborne scenarios there are few annotated 3d data-sets.

For maritime applications 2d detection benchmarks are more common. In the marine data-sets of Modd [13] and Modd2 [3], the obstacles are annotated with 2d bounding boxes. A benchmark for deep learning based object detection in maritime scenarios is presented in [15]. The benchmark consists of RGB and near infrared images from the Singapore Maritime Data-set [17] with annotated 2d bounding boxes.

In the field of waterborne 3d object detection, there is the Roboat project which is an autonomous surface vehicle for urban waterways [21][22]. The sensor system uses lidar, camera, and IMU. Objects are detected in the lidar point cloud with a predefined region of interest, a height filter, and a density based clustering algorithm. In [16] a similar technique is used, based on a point cloud reconstructed by RGB stereo cameras. However, both data-sets do not provide annotated 3d bounding boxes.

The USV inland multi-sensor data-set and benchmark [8] utilize a sensor setup similar to ours, but its benchmark tackles different problems like simultaneous localization and mapping, stereo matching, and water segmentation. The most similar benchmark to ours is the Reeds robot perception benchmark [1]. The sensor setup combines two lidars, six cameras, radar, global navigation satellite system (GNSS), and inertial measurement unit (IMU). The authors mention that they will also provide 3d bounding boxes with different object categories. However, at the time of submission, we were unable to find any public data from the data-set. As far as we know, our proposed approach is the first publicly available multi-modal data-set with oriented 3d bounding box annotations for inland waters.

## III. SENSOR SETUP

As experimental platform, we use the research vessel "Solgenia" equipped with the sensor system shown in Fig. 2. An aluminum profile serves as base element on which the sensors are mounted. The aluminum profile itself is mounted on the roof of the vessel.

To capture RGB images, two Basler ace acA1920-40gc cameras with Kowa LM12HC lenses with constant focal length of 12.5mm are used. This configuration yields a horizontal and vertical aperture angle of $\sim 49°$ and $\sim 32°$. To obtain an IP66 protection level, we protect the cameras against bad weather conditions and splash water with protective housings. Due to the rigid aluminum profile, the cameras are almost in stereo normal case. The baseline of the cameras is about 1.6m. The cameras provide 12bit raw images with a resolution of $1920 \times 1200$ pixels.

A Velodyne VLS-128 lidar is mounted between the cameras. This laser scanner provides 128 channels with a measurement range of 245m. The horizontal and vertical fields of view are $360°$ and $40°$. The range accuracy is specified by the manufacturer by $\pm$3cm. The laser scanner generates a point cloud, where each point measurement consists of 3d point coordinates, channel information, and intensity value.

Additionally, a Xsens-MTi-G-710 (IMU) is mounted to measure the vessel's motion and a Trimble BX992 (GNSS) for synchronization.


Fig. 2. The research vessel "Solgenia" equipped with two RBG cameras, a laser scanner, and an IMU.

### A. Synchronization

Sensor fusion requires highly synchronized data. Therefore, we use a network time protocol (NTP) server that is running on the GNSS device. This time information is also transmitted via a NMEA sentence in combination with a pulse per second to the lidar. Since the lidar is rotating, it continuously yields point cloud data. Therefore, it serves as the time clock for triggering the cameras. The cameras are triggered when the laser beams point forward (forward orientation of the vessel). To reconstruct 3d-information with a stereo system, the two cameras are also synchronized. Since the cameras support the precision time protocol (PTP), a synchronization of the cameras up to nanoseconds is possible using a PTP-grandmaster, that gets the correct time from the NTP-server. With this synchronization, the lidar and the cameras get the same time-stamp in their data headers.

### B. Sensor parameters

The different sensors and other entities in the setup have local coordinate systems. The rigid body transformations between these coordinate systems are defined as

$$\mathbf{T}^{\text{to}}_{\text{from}} = \begin{bmatrix} \mathbf{R}^{\text{to}}_{\text{from}} & \mathbf{t}^{\text{to}}_{\text{from}} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4},$$

with rotation $\mathbf{R}^{\text{to}}_{\text{from}} \in \mathbb{R}^{3\times3}$ and translation $\mathbf{t}^{\text{to}}_{\text{from}} \in \mathbb{R}^{3}$. The sub-script "from" can be from $\{\text{cam}_l, \text{cam}_r, \text{lidar}, \text{chs}, \text{pln}, \text{obj}\}$ for the left and right camera, the lidar, the calibration chessboard, the water surface plane, or the detected object. The super-script "to" is restricted to $\{\text{cam}_l, \text{cam}_r, \text{pln}\}$.

Both cameras in the sensor setup have their individual intrinsic $\mathbf{K}_{\text{cam}_l}, \mathbf{K}_{\text{cam}_r} \in \mathbb{R}^{3\times3}$, distortion parameters, and extrinsic camera parameters. Since reconstruction from stereo cameras is simpler from rectified cameras, we rectify the cameras using [4]. This approach yields rectified projection matrices $\overline{\mathbf{P}}_{\text{cam}_l}, \overline{\mathbf{P}}_{\text{cam}_r} \in \mathbb{R}^{3\times4}$ and the rotation matrices from the un-rectified to the rectified camera systems. The difference between $\overline{\mathbf{P}}_{\text{cam}_l}$ and $\overline{\mathbf{P}}_{\text{cam}_r}$ is then a translation

along the baseline $[-B_x, 0, 0]^T$. Note that variables with an over-line, refer to rectified cameras.

In the sequel, we use the coordinate system of the rectified left camera as global coordinate system for all subsequent steps. In particular, $\mathbf{t}^{\text{pln}}_{\text{cam}_l}$ is chosen to be parallel to the water surface normal. Objects on the water surface are given with respect to the water surface and can only be rotated around the water surface's normal. Thus, the object's bounding boxes will be aligned with the water surface.

### C. Calibration

To calibrate the cameras and the stereo system, we used a flat chessboard pattern with $9 \times 6$ corners which have a distance of 10cm. So, there are $54$ corresponding image and world points per view.

During calibration, the chessboard pattern is moved in front of the camera to get images of the pattern from different perspectives. The chessboard corners are detected with sub-pixel accuracy using [9]. The intrinsics $\mathbf{K}_{\text{cam}_l}, \mathbf{K}_{\text{cam}_r}$ of the cameras are estimated with Zhang's method [23]. In cases where the chessboard was slightly bent, we used the method of Strobl/Hirzinger [20]. The radial distortion parameters are estimated to compensate lens distortion.

For the stereo calibration, we used the same chessboard pattern captured by both cameras simultaneously. With the known intrinsics, the relative rigid body motions $\mathbf{T}^{\text{cam}_l}_{\text{chs}}$ and $\mathbf{T}^{\text{cam}_r}_{\text{chs}}$ are estimated. As an initial guess for the final least square minimization of the re-projection error we use

$$\begin{aligned} \mathbf{R}^{\text{cam}_r}_{\text{cam}_l} &= \mathbf{R}^{\text{cam}_r}_{\text{chs}} \cdot (\mathbf{R}^{\text{cam}_l}_{\text{chs}})^T \quad \text{and} \\ \mathbf{t}^{\text{cam}_r}_{\text{cam}_l} &= \mathbf{t}^{\text{cam}_r}_{\text{chs}} - \mathbf{R}^{\text{cam}_r}_{\text{cam}_l} \cdot \mathbf{t}^{\text{cam}_l}_{\text{chs}}. \end{aligned}$$

For the subsequent rectification [4] we take as input the intrinsics, the distortion parameters, and $\mathbf{T}^{\text{cam}_r}_{\text{cam}_l}$ and yields $\overline{\mathbf{P}}_{\text{cam}_l}, \overline{\mathbf{P}}_{\text{cam}_r}$. In summary, the camera calibration yields the following parameters for each camera:

- un-rectified intrinsic camera parameters,
- radial distortion parameters,
- rotation from un-rectified to rectified coordinates,
- rectified projection matrix.

The sensor fusion of cameras to lidar requires the rigid body motion $\overline{\mathbf{T}}^{\text{cam}_l}_{\text{lidar}}$. To this end, the calibrated stereo system is used to reconstruct an RBG point cloud $\mathcal{P}_{\text{RGB}}$. On the other hand, the lidar also generates a point cloud $\mathcal{P}_{\text{lidar}}$. For the sensor fusion, these two point clouds are registered. For the coarse registration, we ignored the rotational component and the remaining translation of the left camera to the lidar was measured manually. For the fine registration, we use the iterative closest point (ICP) algorithm from [2] to compute $\overline{\mathbf{T}}^{\text{cam}_l}_{\text{lidar}}$. This allows to project a homogeneous lidar point $\mathbf{p} \in \mathcal{P}_{\text{lidar}}$ into the rectified image via

$$\begin{bmatrix} u & v & w \end{bmatrix}^T = \overline{\mathbf{P}}_{\text{cam}_l} \cdot \overline{\mathbf{T}}^{\text{cam}_l}_{\text{lidar}} \cdot \mathbf{p}. \tag{1}$$

Furthermore, we provide the filtered IMU data, although we do not use it in our setup.

## IV. DATA-SET

Deep learning models require a large amount of data to adjust their weights. Additionally, supervised learning algorithms need annotations, whose generation is laborious and time consuming. Nevertheless, we recorded a data-set with the sensor setup proposed in Section III and annotated interesting sequences by human annotators. The name of the data-set is, Lake Constance Obstacle Detection Data-set.

### A. Acquisition

The sensor system on the "Solgenia" is controlled by a notebook on the vessel, which allows to monitor the sensor data. For the communication with the sensors the robot operating system (ROS) is used. We recorded data at five different days to generate variation in the data. On each recording day, we captured data of the chessboard pattern so that a later offline calibration was possible. The lidar and the cameras were operated at 10Hz. The sensor data was saved into rosbag files for offline replay of the sensor data.

The data was recorded at Seerhein, which is a section of the Rhine River and on Lake Constance. The visibility was clear on all days and the weather was good. Obviously, scenarios in bad weather would have been interesting as well. However in these conditions, almost no other traffic was on the lake and the recording process was too dangerous.

### B. Annotation

Because we saved the sensor data in rosbag files, we are able to use the rosbag API to read the data. Since there is a header with a time-stamp for each date, synchronized data can be read easily. To annotate the point clouds with bounding boxes we used ROS and RViz, which is a 3d visualization tool for ROS. To annotate a point cloud, we predefined time-stamps where the data contain an interesting scenario. During data loading the ROS stereo processing node rectifies the camera images and computes $\mathcal{P}_{\text{RGB}}$.

Due to current and waves, the distance of the sensors to the water surface is not constant and might be rotated. So, first the water surface is approximated with a water plane. For objects in close range, the earth's curvature can be neglected, such that all objects lie on a local water plane. This water plane is estimated for every time-step.

To estimate the water plane, the annotator had to mark the water surface in the left camera image with a 2d bounding box. A random sample consensus (RANSAC) [10] is then used to fit a plane to the points of $\mathcal{P}_{\text{RGB}}$ inside the marked image patch. The RANSAC returns a normalized normal $\mathbf{n}$ and a support point $\mathbf{s}$. Thus, $\overline{\mathbf{T}}^{\text{pln}}_{\text{cam}_l}$ given by (cf. Fig. 3)

$$\overline{\mathbf{R}}^{\text{pln}}_{\text{cam}_l} = \begin{bmatrix} \mathbf{x} & \mathbf{n} & \mathbf{z} \end{bmatrix}^T \quad , \quad \overline{\mathbf{t}}^{\text{pln}}_{\text{cam}_t} = \begin{bmatrix} 0 & -\mathbf{n} \cdot \mathbf{s} & 0 \end{bmatrix}^T,$$

with $\mathbf{z} = [1, 0, 0]^T \times \mathbf{n}$ and $\mathbf{x} = \mathbf{n} \times \mathbf{z}$. A homogeneous lidar point $\mathbf{p} \in \mathcal{P}_{\text{lidar}}$ is mapped to water plane coordinates $\mathbf{q}$ via

$$\mathbf{q} = \overline{\mathbf{T}}^{\text{pln}}_{\text{cam}_l} \cdot \overline{\mathbf{T}}^{\text{cam}_l}_{\text{lidar}} \cdot \mathbf{p}.$$

Compared to the usage of the IMU data, the advantages of this approach are, that the sensor height can be estimated as well and there is no need to calibrate the IMU to the camera.

This allows you to directly annotate bounding boxes for the individual objects in the plane coordinate system. A bounding box can only be placed and moved along this plane. Since $\mathcal{P}_{\text{RGB}}$ is noisy due to depth errors, we assume that $\mathcal{P}_{\text{lidar}}$ is more accurate and is used as 3d ground truth. Thus, the annotator translates (in the water plane), rotates (around the planes normal), and scales the bounding box such that the object points from $\mathcal{P}_{\text{lidar}}$ are enclosed by the bounding box.

For objects that are far away, this might be ambiguous. Then, there are very few lidar measurements, and for the annotator, it is difficult to estimate the size and orientation of the object correctly. Therefore, the bounding box was projected back to the rectified image to support the annotator.

The bounding box is oriented according to the orientation of the vessel in the range $[\pi, -\pi]$. This means that an object which has a heading of $0$ is aligned along the $z$-axis of the plane. A bounding box is defined by the position of the center, its extent (width, height, length), and the rotation around the $y$-axis of the plane. With this information, the rigid body motion $\overline{\mathbf{T}}^{\text{pln}}_{\text{obj}}$ is well defined, see Fig. 3.

Each bounding box has some meta-data. We defined nine categories and have assigned a category to each bounding box. The categories are motorboat, sailboat, sailboat under bare poles, stand-up paddle, catamaran (aka speedboat), car ferry, pedal boat, motor vessel, and pile. One example of each category is shown in Fig. 4. Other attributes that we have assigned to a bounding box are occlusion and visibility. Occlusion means how much an object is occluded by another object. Visibility means the percentage of an object that is not visible in the image (out of image boundary). We define four levels for each attribute: $0\%$, $25\%$, $50\%$, $75\%$. In case of occlusion or low visibility, the bounding boxes are scaled to match the original size of the object, even if there is only one lidar point. Fig. 7 shows ten examples from the data-set.

There are further restrictions for the annotations. For a bounding box, at least one lidar point must be captured. This also means that only objects within $245$m were annotated. In addition, only objects that are in the field of view of the cameras were annotated. Due to privacy policy, we had to pixelate visible faces and vessel identifiers. Sometimes short sequences with intervals of only $100$ms were annotated. This

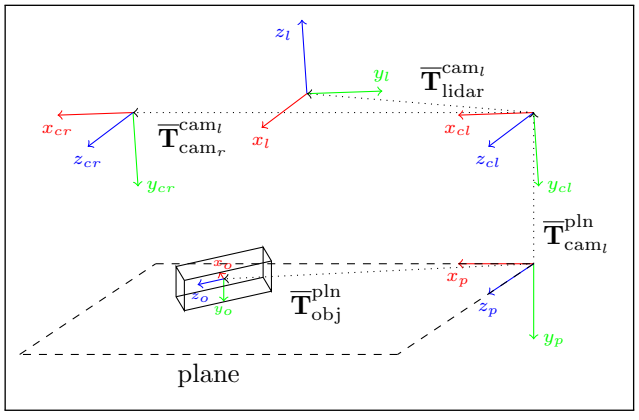


Fig. 3. Illustration of the coordinate systems used in our data-set.

Fig. 4. Categories present in the data-set in the order from top left to bottom right: motorboat, sailboat, sailboat under bare poles, stand-up paddle, catamaran, car ferry, pedal boat, motor vessel, pile.

leads to small changes in the image but provides also many small orientation changes.

Finally, the sensor data, calibration data, and annotations are saved in a hierarchical data format (hdf5). Each file is named after the time-stamp when the data was recorded. The training data-set has $1664$ annotated frames with $2600$ annotated objects. The test data-set has $310$ annotated frames with $389$ annotated objects. The distribution of the individual categories and the number of objects per image are shown in Fig. 5. The distance between consecutive annotations per sequence is shown in Fig. 6. The test data-set is independent from the training data-set. It was recorded at the same locations but on two days when no training data was recorded.

## V. Benchmark

To provide an initial baseline, we combine some approaches presented in Section II. In [21] Euclidean clustering is applied to the lidar point cloud to obtain candidate obstacles. However, in our baseline, we replace the Euclidean clustering with a neural network in image space.

### A. 2d object detection

First, we fine-tune a 2d detection network to find interesting regions, where objects could be located in the image. This is used to cluster the individual objects. One advantage of the neural approach is that the objects can also be classified. In order to train a 2d object detector, ground truth 2d bounding boxes in the image plane are required. Since the data-set contains the vertices $\{\mathbf{v}_0, \ldots, \mathbf{v}_7\} =: \mathcal{V}$ of the 3d bounding boxes in water plane coordinates, we first map them to the

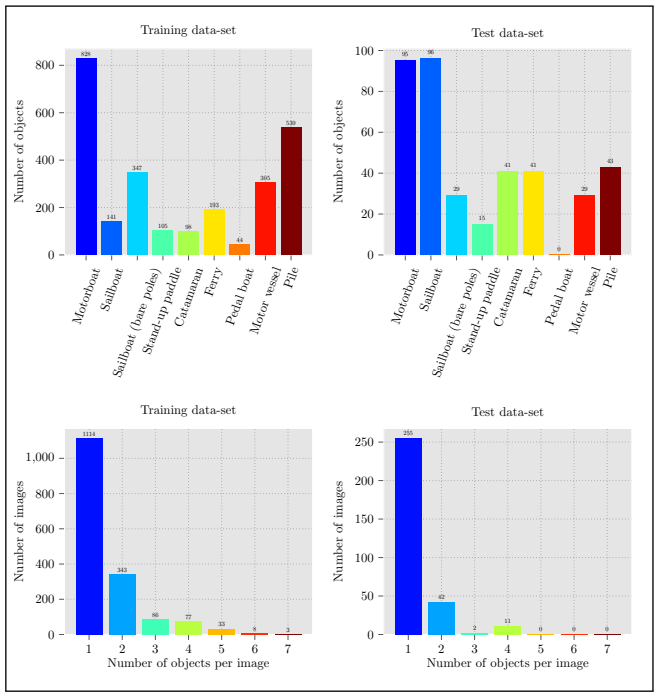


Fig. 5. This figure shows the distribution of the categories in the training and test data-set (top) and the distribution, of how many objects occur in an image (bottom).

rectified left camera image

$$\begin{bmatrix} x & y & w \end{bmatrix}^T = \overline{\mathbf{P}}_{\text{cam}_l} \cdot \left( \overline{\mathbf{T}}_{\text{cam}_l}^{\text{pln}} \right)^{-1} \cdot \mathbf{v}, \quad \mathbf{v} \in \mathcal{V}.$$

The 2d axis-aligned bounding box is defined with the top left $\mathbf{a}$ and bottom right point $\mathbf{b}$ which are obtained by

$$\mathbf{a} = \left[ \min\{x/w | \mathbf{v} \in \mathcal{V}\}, \min\{y/w | \mathbf{v} \in \mathcal{V}\} \right]^T,$$
$$\mathbf{b} = \left[ \max\{x/w | \mathbf{v} \in \mathcal{V}\}, \max\{y/w | \mathbf{v} \in \mathcal{V}\} \right]^T.$$

This ensures that an object is completely included in its 2d bounding box. It will be used to tune the 2d object detector.

For the baseline, we used a Faster R-CNN [19] with a ResNet-50 [12] backbone. The neural network was trained on the training data-set and we measured the generalization performance on the test data-set with the mean average precision (mAP), mean average precision at the intersection over union (IoU) level 50 ($\text{mAP}^{50}$) and 75 ($\text{mAP}^{75}$). The final model achieved a mAP of 0.282, $\text{mAP}^{50}$ of 0.570, and $\text{mAP}^{75}$ of 0.244. The model is able to detect objects that have not been annotated due to the distance. However, this influences the metric. It also partially misclassifies motor vessels, catamaran, and ferry. This is probably because the objects are very similar.

For fine-tuning, the model was initialized with pre-trained weights and we replaced the last classification layer that our new classes are learned. The neural network models were fine-tuned (updated all weights) on an NVIDIA A100 GPU with an initial learning rate of $5 \times 10^{-5}$, a weight decay of $10^{-4}$ with an ADAM optimizer, and batch size 6. We reduce the learning rate by $10^{-1}$ if the loss does not decrease for three epochs on validation data, until a learning rate of $10^{-9}$ is reached. We used early stopping and reduced the image resolution for network input to $928 \times 576$ pixels. During fine-tuning, we split the training data-set into 80% training and 20% validation data to prevent overfitting. The images are augmented in each batch by a horizontal flip with a probability of 0.5.

### B. 3d bounding box estimation

With the detected objects in the image, the corresponding 3d bounding boxes must be computed. To obtain 3d information of an object, we use the lidar sensor.

We first projected $\mathcal{P}_{\text{lidar}}$ into the image (1) to determine the lidar point cluster. All points that fall into a detected 2d bounding box belong to one cluster. This results in a 3d point cluster for each object in the scene. To determine 3d bounding boxes on the water plane, we need an estimate of the water plane. Assuming that the last 500 rows in each stereo image pair capture the water surface, the stereo point cloud is reconstructed and RANSAC is used to estimate the water plane, see Sec. IV-B. This allows to map the reconstructed 3d points of each cluster to the water plane coordinates. To obtain the 3d bounding box for each cluster, we project the point clusters to the 2d plane by removing the $y$-component (height). In [16] the main axes of the bounding box are determined with principal component analysis (PCA). We estimate the oriented bounding box with the PCA of the convex hull. Since we set the $y$-component to 0 only the 2d bounding box in the $xz$-plane is estimated. To get the bounding box height, we take the minimum $y$-value in the point cluster.

### C. Evaluation

For evaluation, we used the mAP metrics from the COCO benchmark [14]. This metric is designed only for 2d detectors with axis-aligned bounding boxes. For the evaluation of

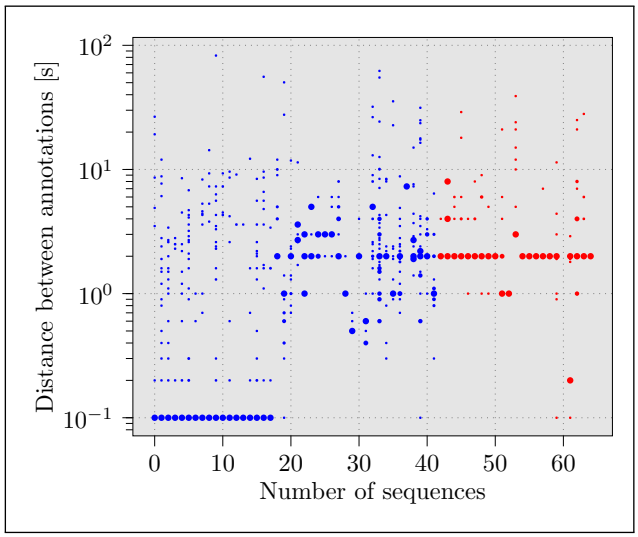


Fig. 6. Distance between annotations per recorded sequence (blue: training data-set, red: test data-set). The dot size indicates the density of the occurring differences in a sequence.

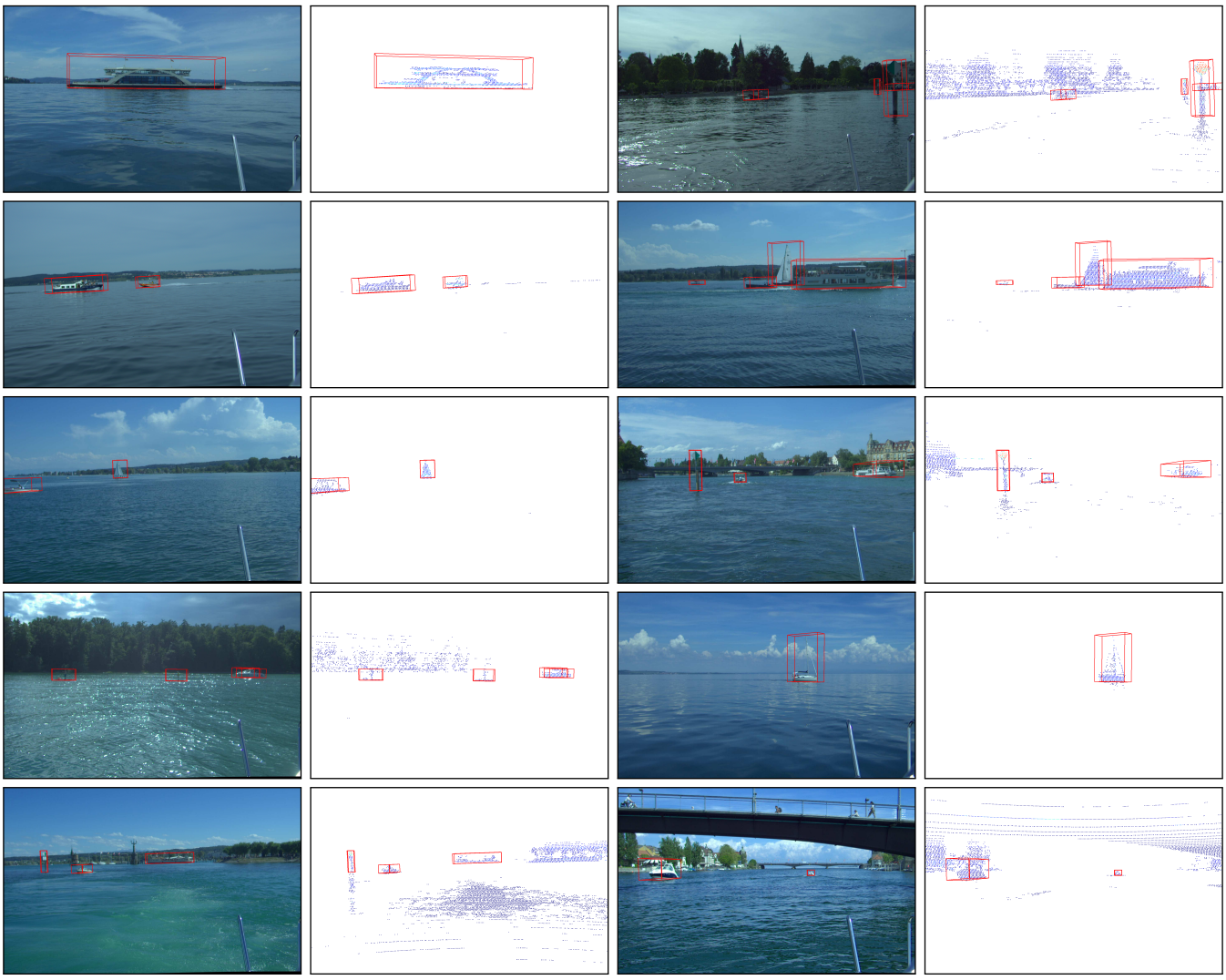

Fig. 7. Ten annotated left camera and point cloud samples of the data-set. The first three rows show samples of the training data-set. The fourth and the fifth rows show samples of the test data-set.

arbitrarily oriented bounding boxes, the IoU module was replaced with an algorithm for IoU computation of oriented 3d bounding boxes [18, 5]. As in [5], the metric is called $\text{mAP}_{3D}$. We measured for the proposed method a $\text{mAP}_{3D}$ of $0.069$, $\text{mAP}^{50}_{3D}$ of $0.194$, and $\text{mAP}^{75}_{3D}$ of $0.035$. For evaluation, the ground truth and the estimated 3d bounding box are back-projected to the camera coordinate system. It is clear that the 3d detection can be highest equal to the previous 2d detection. With the image-lidar approach, the estimated 3d bounding boxes are usually too small. This is because the lidar sometimes only measures one side of the object and therefore with PCA it is not possible to measure the whole extent. Too small bounding boxes can also occur when there are not enough lidar measurements, for example, objects are too far away. We compute with our method too large bounding boxes when two objects are behind each other. Then the points of both objects are in one cluster and PCA calculates the bounding box as too large.

## VI. CONCLUSION

In this paper, we have introduced a novel multi-modal annotated 3d data-set for inland waters. The data-set contains 2989 annotated 3d bounding boxes with category, occlusion, and visibility information. We provide a training data-set and an independent test data-set. We propose a sensor setup for close range obstacle detection on inland waters and a method for image-lidar detection.

The purpose of this data-set is to push the development of collision warning systems and to contribute to the development of deep learning models for inland water safety.


## ACKNOWLEDGMENT

We thank J. Reuter for providing the lidar sensor and the research vessel. We also thank M. Albrecht and T. Baur, who helped with the sensor setup and the data acquisition. Finally, we thank P. Sutter, F. Mayer, and T. Schaub for their support with the annotation tool and labeling.

## References


[1] O. Benderius, C. Berger, and K. Blanch. "Are we ready for beyond-application high-volume data? The Reeds robot perception benchmark dataset". In: (2021). arXiv: 2109.08250 [cs.CV].

[2] P. J. Besl and Neil D. McKay. "A method for registration of 3-D shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256.

[3] B. Bovcon, R. Mandeljc, J. Perš, and M. Kristan. "Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation". In: *Robotics and Autonomous Systems* 104 (2018), pp. 1–13.

[4] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[5] G. Brazil, A. Kumar, J. Straub, N. Ravi, J. Johnson, and G. Gkioxari. "Omni3D: A Large Benchmark and Model for 3D Object Detection in the Wild". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 13154–13164.

[6] Polizei BW. *Internationale Unfallstatistik für den Bodensee 2022*. 2023. URL: https://www.presseportal.de/blaulicht/pm/110981/5454916 (visited on 09/14/2023).

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. "nuScenes: A Multimodal Dataset for Autonomous Driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11621–11631.

[8] Y. Cheng, M. Jiang, J. Zhu, and Y. Liu. "Are We Ready for Unmanned Surface Vehicles in Inland Waterways? The USVInland Multisensor Dataset and Benchmark". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3964–3970.

[9] A. Duda and U. Frese. "Accurate Detection and Localization of Checkerboard Corners for Calibration". In: *British Machine Vision Conference*. 2018.

[10] M. A. Fischler and R. C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Association for Computing Machinery*. 1981.

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets robotics: The kitti dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[12] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[13] M. Kristan, V. Sulić Kenk, S. Kovačič, and J. Perš. "Fast Image-Based Obstacle Detection From Unmanned Surface Vehicles". In: *IEEE Transactions on Cybernetics* 46.3 (2016), pp. 641–654.

[14] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV*. 2014, pp. 740–755.

[15] S. Moosbauer, D. König, J. Jäkel, and M. Teutsch. "A Benchmark for Deep Learning Based Object Detection in Maritime Environments". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 916–925.

[16] J. Muhovič, R. Mandeljc, B. Bovcon, M. Kristan, and J. Perš. "Obstacle Tracking for Unmanned Surface Vessels Using 3-D Point Cloud". In: *IEEE Journal of Oceanic Engineering* 45.3 (2020), pp. 786–798.

[17] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek. "Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 18.8 (2017), pp. 1993–2016.

[18] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W. Lo, J. Johnson, and G. Gkioxari. "Accelerating 3D Deep Learning with PyTorch3D". In: *arXiv:2007.08501* (2020).

[19] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. Vol. 28. 2015.

[20] K. H. Strobl and G. Hirzinger. "More accurate pinhole camera calibration with imperfect planar target". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 2011, pp. 1068–1075.

[21] W. Wang, B. Gheneti, L. A. Mateos, F. Duarte, C. Ratti, and D. Rus. "Roboat: An autonomous surface vehicle for urban waterways". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 6340–6347.

[22] W. Wang, T. Shan, P. Leoni, D. Fernández-Gutiérrez, D. Meyers, C. Ratti, and D. Rus. "Roboat ii: A novel autonomous surface vessel for urban environments". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 1740–1747.

[23] Z. Zhang. "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334.