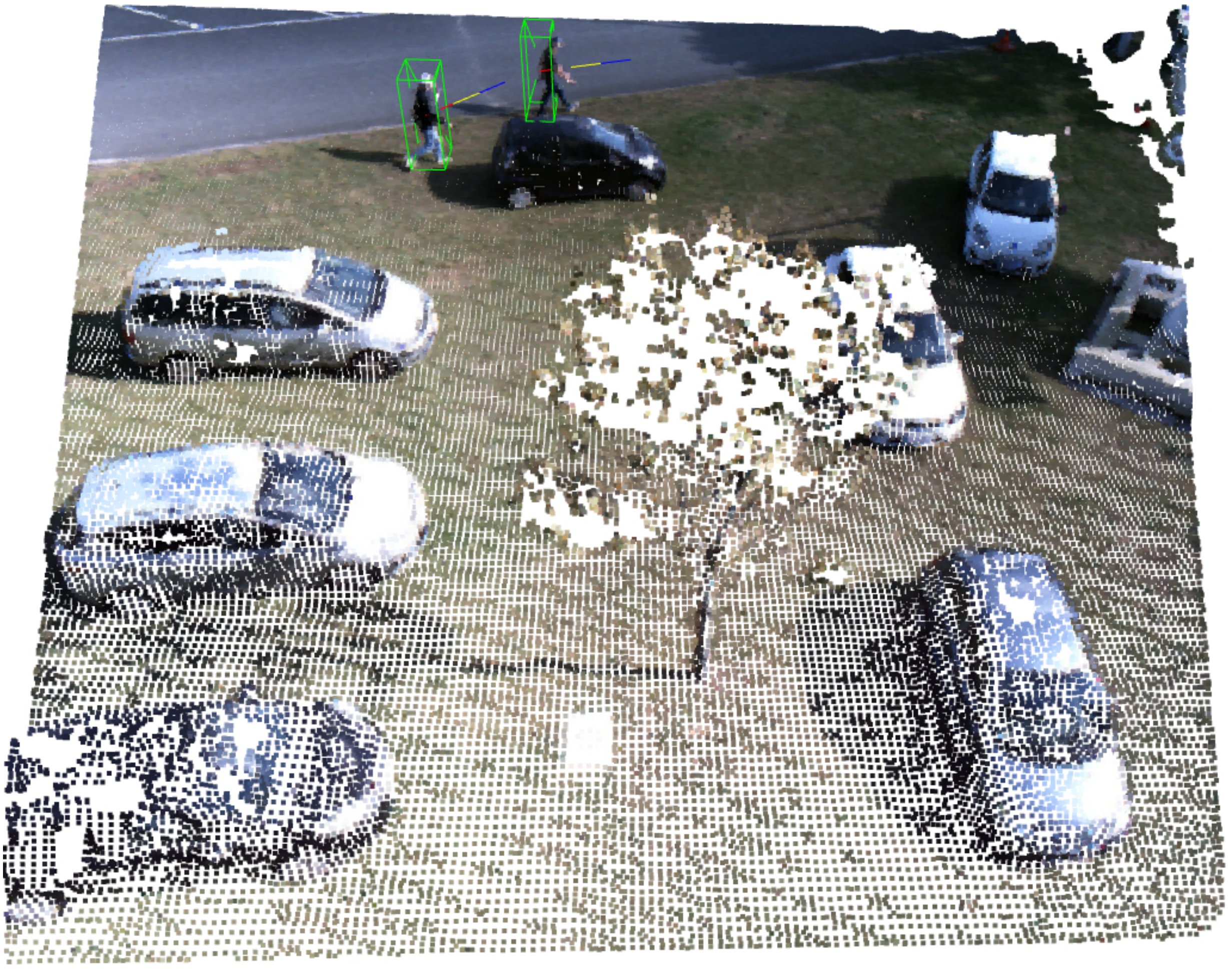
Stereo Pointclouds for Safety Monitoring of Port Environments

Femke Middelhoek



TU Delft

Stereo Pointclouds for Safety Monitoring of Port Environments

by

Femke Middelhoek

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended on September 20, 2023 at 13:00

Thesis committee:

Chair:	Dr. H. Caesar
Supervisors:	Dr. H. Caesar
	Dr. F. ter Haar
External examiner:	Dr. R. T. Rajan
	Dr. J. Kooij
Place:	Faculty of Mechanical Engineering, Delft
Project Duration:	December, 2022 - September, 2023
Student number:	4552091

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Mechanical Engineering · Delft University of Technology

TU Delft Delft University of Technology

Copyright © Femke Middelhoek, 2023
All rights reserved.

Stereo Pointclouds for Safety Monitoring of Port Environments

Femke Middelhoek *
* 4552091, f.middelhoek@student.tudelft.nl, Delft University of Technology

***Abstract*—The MOSES project develops an autonomous vessel equipped with an autonomous crane to optimise the supply chain of short-sea shipping. This study focusses on monitoring the safety of the port environment based on stereo camera data generated by sensors attached to the crane at 15m altitude, oriented 45 °downward. The objective is to detect individuals and estimate their motion. Semi Global Block Matching is implemented for stereo pointcloud generation (a pointcloud based on the disparity image and stereo camera calibration information). Voxel-averaged stereo pointcloud downsampling is performed for improved data compliance with CenterPoint. Background subtraction is implemented with Gaussian Mixture Models (GMMs). The study proposes a novel implementation to fit a GMM on per-point 3D spatial (xyz) and color information for enhanced background-foreground segmentation of the stereo pointclouds. 3D object detection and velocity prediction are based on CenterPoint, customised to take color features into account. The result is a robust detection pipeline with a top performance of 81.5% mAP, 4% Average Orientation Error and 9.4% Average Velocity Error on a simulated dense port environment dataset. Background subtraction is implemented to improve cross-environment generalisation, an important feature for MOSES considering the mobile nature of the vessel and the likelihood that it would attend unseen environments. Voxel-averaged downsampling of the stereo pointcloud advances this by creating a uniform data structure, further facilitating the transfer of learnt features to previously unobserved scenes. Including color information of the current frame reduces the impact of spatial uncertainty of the stereo pointcloud. It improves detection performance, particularly when excluding the color information of the temporal reference frames included for velocity prediction. The transferability of the pipeline developed in simulation to reality is demonstrated on a basic real-world scenario.**

***Keywords*—** Stereo Pointclouds, Gaussian Mixture Models, CenterPoint, 3D Object Detection, Cross-Environment Generalisation

I. Introduction

Staff shortage, lack of skill, or cost reduction efforts force industries to research the applicability of autonomous systems in new areas. Already found in factories and manufacturing sites, the reach of autonomous machinery is now extending to unstructured environments [1]. Such an environment is the location of the autonomous crane of the Robotic Container-Handling System created for the European Project titled: AutoMated Vessels and Supply Chain Optimization for Sustainable Short SEa Shipping (MOSES) [2, 3]. In pursuit of designing an autonomous vessel capable of docking, loading, and unloading containers by an autonomous crane while safeguarding human activity in the port environment, the design of the sensor suite is an integral part of the system.

The MOSES project is focused on small ports and requires a precise perception of the surrounding environment. Small ports exhibit distinct characteristics compared to large industrial ports, including a greater diversity of objects in the scene, relative openness of the environment, lower occupancy, and a higher likelihood of objects entering and exiting the scene. The focus of safety monitoring in this study includes the detection of individuals within the port area and the estimation of their future motion. Small objects, such as personnel and equipment, often have limited visual cues, are sensitive to noise, and can easily blend into the complex background of the port [4, 5]. Furthermore, the dynamic nature of port operations, where objects may be in motion, entering or leaving the scene, or occluded by other structures, and the variability in lighting conditions further complicate the task of detecting these objects [6].

The expected interaction between objects and environment is estimated to evaluate the safety of crane operations in relation to the predicted motion of individuals. To predict the velocity of objects, a 3D environment reconstruction is required. The sensor suite designed to achieve this objective is located at the top of the crane, looking diagonally at the scene. It features two Velodyne VLP-16 LiDAR sensors along with a stereo camera setup. The pointcloud generated by the VLP-16 sensors is too sparse for the detection of small objects (persons), a known weakness of the LiDAR sensors [7]. Consequently, the decision was made to reconstruct the environment in 3D using stereo disparity matching. The disparity image can be converted to a 3D pointcloud with stereo camera calibration information, and this pointcloud with 3D spatial and color information will be referred to as a stereo pointcloud. This image-based approach highlights high-texture regions and better retains small objects such as persons [8, 9]. However, it has weaknesses including object artefacts, depth ambiguity, and sensitivity to environmental conditions [10]. In the current study, the effect of object artefacts on object detection performance is minor due to the sparsity of the environment and the simpler scene structures minimise the chances and impact of depth ambiguity. The environmental conditions are simulated by incorporating sensor noise. This noise can replicate the uncertainties and variations encountered by real-world sensors, which, in conjunction with an experiment on real-world data, gives an indication of the robustness of the method developed in simulation to the changing and challenging conditions as encountered in real-world scenarios.

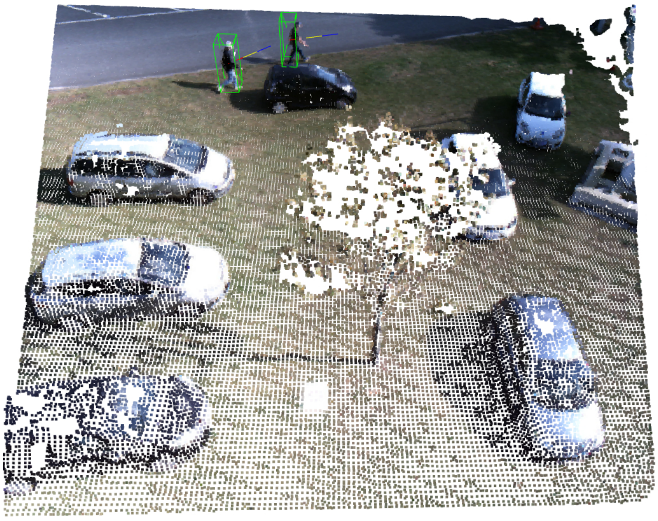


Fig. 1. Output visualisation of the MOSES pipeline. 3D stereo reconstruction of a real-world dataset. Object detections are shown in green and t+1, t+2, t+3 object motion estimates in red, yellow, and blue.

The contribution of this study to the state-of-the-art is the accurate estimation of the position and motion of 3D objects in a port environment using an end-to-end processing pipeline. A new combination of building blocks is used to achieve this; Semi-Global Block Matching [11] for stereo pointcloud generation and 3D environment reconstruction, Gaussian Mixture Models (GMM) [12–14] for background subtraction, and CenterPoint [15] with per-point color information to detect and estimate the 3D motion of objects. To emphasise the crucial elements, distinct aspects of our pipeline are explained in more detail:

- *Cross-Environment Generalisation:* As the MOSES crane attends to various ports, some of which are unseen, learning an object representation independent of the environment is desired. In addition, this would make the method more flexible. To enforce the cross-environment generalisation of learnt features, voxel-based downsampling of the stereo pointcloud and background subtraction are implemented. Voxel-based downsampling, while retaining the voxel average as a representative point, reorders the unstructured pointcloud in a structured grid format. Background subtraction enhances object isolation, which tackles the problem of small objects blending in with the background, and, simultaneously, makes the limited visual cues of objects more pronounced in the reduced pointcloud size. The small structured pointcloud enables CenterPoint to learn a more robust representation of objects of interest. The structured grid format and background subtraction combined with the facilitated learnt robust object representation were found to significantly improve performance in unseen environments, thereby enhancing the cross-environment generalisation of the MOSES pipeline and its adaptability to new scenarios.
- *3D Stereo Pointcloud based Gaussian Mixture Models:* GMMs are implemented for background subtraction. The objective of background subtraction for MOSES is to extract objects of interest from the stereo pointcloud. Here, foreground components can have appearance features similar to those of background components. In addition, objects of interest might be visible next to, behind or partially occluded by background components which have a similar associated depth. Finally, for increasing sensor noise or challenging environmental conditions, the depth value resulting from stereo matching becomes more uncertain, and a GMM fails to capture the underlying data distribution. Therefore, color alone or color and depth do not provide sufficient information for accurate segmentation. To address this challenge, Gaussian Mixture Models are fit on the stereo pointcloud with 3D spatial (xyz) and RGB color information for optimal performance.
- *Color Information for improved Object Detection Performance:* Stereo pointclouds contain noise and are relatively unstructured compared to LiDAR pointclouds. If an object of interest is close to another object in the scene, detecting object boundaries becomes challenging, which affects object localisation accuracy. This phenomenon also becomes evident when multiple frames are included with the current frame as temporal references for velocity prediction. Consequently, integrating the appearance features of the current frame and excluding the color information of the frames included for temporal reference improves the detection performance. This improvement is more pronounced in dense environments, where object boundaries are more often less distinct. Therefore, this study proposes to adapt CenterPoint, originally designed for LiDAR data, to accommodate per-point color information.

For performance evaluation, a MOSES Detection Score (MDS) is proposed. With the objective of safety monitoring based on the detection of individuals and the estimation of their future motion, accurate object localisation, orientation detection, and velocity prediction form the main performance indicators. Therefore, the MDS is a weighted average of the Bird's-Eye View (BEV) centerpoint detection distance, average velocity error, and average orientation error. Experiments reveal an MDS of 0.781 on the simulated data set of a small port environment, 0.816 on the simulated data set of a dense port environment, and an MDS of 0.664 for an environment where 10× as much camera sensor noise is introduced. The proposed end-to-end processing pipeline is robust and accurate and generalises well to unseen sceneries. Testing the pipeline on a real-world dataset shows an average precision similar to that of the simulated datasets, proving its applicability to real-world scenarios.

The structure of the paper is as follows: In Section II, related work to the problem at hand is presented: outdoor safety monitoring, stereo pointcloud generation, background subtraction, and 3D object detection and motion estimation methods. Section III discusses all aspects of the simulation including Gazebo, sensor setup, and environment design. The method comprises data generation, background subtraction, CenterPoint, and 3D motion visualisation, as detailed in Section IV. Section V presents experiments showing the performance of the proposed method and ablation studies that examine the influence of separate pipeline components on the final result. In addition, the performance of the proposed end-to-end pipeline is demonstrated on a real-world dataset. Section VI critically evaluates the presented work and motivates directions for further research.

II. Related Work

A. Unstructured Environment Safety Monitoring

Outdoor safety monitoring setups often have camera sensors (e.g., video surveillance). The sensor perspective is similar to that of MOSES, as cameras are often attached to walls or ceilings and look down at the environment. Moving objects are mostly detected in the 2D image plane [16–20]. Object tracking is done in 2D and converted to 3D results [16], directly in 3D [18–20] or in 2D and 3D concurrently and merged afterwards [17]. The cameras do not return depth information, and the third dimension is obtained using multiview methods [17, 18, 20] or by applying 3D estimation models [16, 19]. 3D estimation models often rely on environmental assumptions, such as probabilistic scan matching [16]. CraneNet [1] performs top view detection of ground workers with a camera sensor attached to a telescopic crane. Illustrating the relevance of the current research direction by stating that the ground workforce cannot be aware of their surrounding environment during crane operations at complex sites, it is noted that the objective in the article remains limited to workforce detection. As the current objective is 3D motion estimation, which requires velocity prediction and insight into the object's interaction with the environment, detection is done on a 3D environment reconstruction from stereo camera data (stereo pointclouds).

B. Stereo Pointcloud Generation

Methods for constructing stereo pointclouds from disparity images and stereo camera calibration information information can be classified as either handcrafted or based on deep learning. Common handcrafted approaches for estimating disparity are Semi Global Block Matching (SGBM) [11] and Block Matching (BM) [21]. Although these methods can cope with complex scenarios and provide interpretability, they require parameter finetuning for scenarios that exhibit substantial variation from one another. SGBM differs from BM by including the global context and is therefore of both higher accuracy and higher computational complexity. More recent studies alleviate the last concern and exploit and apply SGBM for real-time applications [22, 23]. Pseudo LiDAR++ [24] and Pyramid Stereo Matching [25] (PSM-Net) are two deep learning-based approaches that use neural networks to learn mappings between stereo images and dense pointclouds. Pseudo LiDAR ++ proposed two advances to the prior Pseudo-LiDAR framework [26]; first, to change the loss composition for the stereo depth network from disparity loss to depth loss to correct for the strong emphasis of tiny depth errors on nearby objects. Second, a depth correction of the resulting dense predicted depth map based on extremely sparse but accurate LiDAR measurements to correct the limitation of the discrete nature of disparity estimation (quantisation of depth at pixel level while the depth is continuous in the real world). PSM-Net leverages multi-scale image representations of the scene to handle different levels of detail and variations in scene depth, texture, and lightning. PSM-Net consists of two main modules; a spatial pyramid module to aggregate global context in different scales and locations to form a cost volume and a 3D Convolutional Neural Network (CNN) that learns to regularise this cost volume. While PSM-Net and Pseudo LiDAR++ excel at managing challenging scenarios, they require much labelled data and expensive computational training. The returned stereo pointcloud is dense and would still require downsampling to be compatible with the current proposed pipeline. In addition, both deep learning methods are optimised for Autonomous Driving Systems and the additional challenges of a sensor suite at altitude looking down at the scene of interest are not addressed yet. Such challenges include perspective distortion and coping with terrain height variations. Therefore, a handcrafted stereo-matching method is preferred within the current

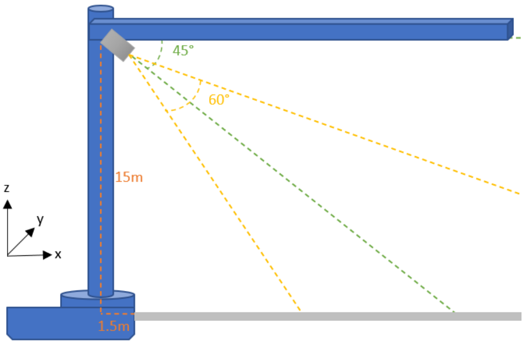


Fig. 2. Image of Sensor Setup. Dimensions not to scale. Blue: Automated vessel with Autonomous crane. Gray: Dock. Green: Sensor Orientation. Yellow: Vertical Field of View of Camera.

objective of presenting an end-to-end pipeline. SGBM emerges as a reliable, adaptable, and efficient method for creating stereo pointclouds in various practical applications such as the MOSES project.

C. Background Subtraction

Deep learning methods and the continuous enhancement of traditional approaches for background subtraction have contributed to the advances in computer vision. Background subtraction can be performed to speed up object detection and classification by constraining the search window [27]. In addition, efficiency is enforced for subsequent deep learning implementations by using the foreground as input. The challenges of background subtraction are numerous and include dealing with moving objects, changes in illumination, and dealing with shadows and reflections. Currently, no single method can tackle all challenges robustly [28]. CNNs have demonstrated remarkable capabilities in capturing complex spatial and temporal representations, leading to accurate foreground detection and background modelling [28–30]. [28] proposes CNN-SFC, a U-net encoder-decoder architecture that learns to combine the result of several background subtraction algorithms into a single output image. SubSENSE [31] detects change based on spatio-temporal binary features and color features, FTSG [32] proposes a hybrid method combining motion detection, appearance comparison and foreground-background segmentation. Finally, CwisarDH+ [33] applies weightless neural networks to each picture in the region of interest for background model learning. CNN-SFC learns to effectively use the output of the three mentioned methods for background segmentation, outperforming each method individually. In parallel, non-deep learning-based methods, such as Gaussian mixture models (GMM) and adaptive modelling techniques, have progressed significantly [34, 35]. GMM methods model the intensities of the pixels as a mixture of Gaussian distributions [12–14]. With this, they enable a flexible background representation which can be updated continuously. Adaptive background modelling algorithms can update the background model to accommodate gradual scene changes. Adaptive algorithms are GMM, Frame Differencing, and Running Average Methods. Frame differencing detects moving objects by subtracting consecutive frames [36]. Running Average methods capture slow changes in the environment by keeping e.g. an exponential moving average or a Gaussian average of the background over time [34].

A combination of traditional methods can be implemented to improve performance [37]. Deep learning methods are more robust and excel in coping with complex scenarios, but require training data for every new or changed scene; non-deep learning methods are simple, adaptable, and suitable for situations with constrained resources or gradual scene changes. They can adapt the background model over time using an incremental learning process and learn the background model when attending a new scene. As MOSES attends to various (unseen) port environments with evolving backgrounds and limited reference data, an adaptable traditional method is chosen and implemented for background subtraction (Gaussian Mixture Models).

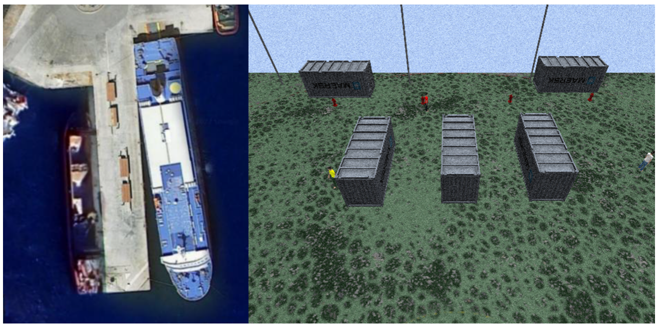


Fig. 3. Noisy Port Environment Design. On the left is the port of Mykonos (Google Earth Screenshot), which is the inspiration for the simulated world as seen on the right.

D. 3D Object Detection and Motion Estimation

For 3D motion estimation, an object detection method can be combined with a motion estimation method, or the two can be integrated. Various techniques have emerged for 3D object detection using pointclouds or stereo pointclouds. These can be classified as point-based, voxel-based, or projection-based methods. PointRCNN [38] is a point-based pointcloud method performing 3D bottom-up proposal generation. PointNet++ [39] uses the property of PointNet [40] to convert a set of local features into higher feature representations and applies this recursively to the pointcloud input to learn features at different scales. PointNet is the backbone of other object detection algorithms [41–43]. Pointcloud voxel-based methods such as the 3D backbone VoxelNet [44] segment the pointcloud into a regular 3D grid where voxel features are learned with a 3D CNN. In 2D representation-based processing, the pointcloud features are projected to e.g. BEV. In BEV, occlusions are avoided and object size is unambiguous. CenterPoint [15] uses VoxelNet as a backbone and performs projection to 2D to create features suitable for 2D CNNs. Pointpillars [45] is a 3D backbone that utilises PointNet to organise the pointcloud in vertical pillars. An encoder learns features that can be stacked to a pseudo-image. To the pseudo-image, any standard 2D convolutional detection architecture can be applied. Some 3D object detection methods exploit the combination of a 3D pointcloud and semantic image features to exploit the localisation capabilities of LiDAR methods and the classification advantages of 2D images [7, 46–48]. In the current application, the LiDAR pointcloud is too sparse to be a resource and only a few of the above-mentioned techniques may be applicable to stereo pointclouds. A promising technique such as CenterPoint, a LiDAR-based 3D object detection method that can potentially process augmented point features, is explored for our application and enhanced with colour information to improve its performance.

Within the domain of 3D pointclouds, motion estimation can be approached through both manual and learning-based methods. Manual methods for motion estimation such as Kalman filters [49] and particle filters [18, 50] rely on motion models. Kalman filters estimate motion using a recursive mathematical model based on a combination of measurements and predictions. Particle filters use probabilistic sampling to predict motion. Filter-based motion estimation methods are based on handcrafted motion rules [49]. An advantage of motion model methods is their simplicity and speed [51]. These hand-crafted methods are interpretable and suitable for particular motion patterns, but may struggle with non-linear or complex motions [52]. Most learning-based advances regarding object motion in 3D pointclouds are focused on object tracking. However, in the current context, the objective is motion estimation rather than retrospective object-track association. There is one 3D object detection and tracking method suitable to apply in the current context. CenterPoint [15] estimates motion by combining object detection and velocity prediction.

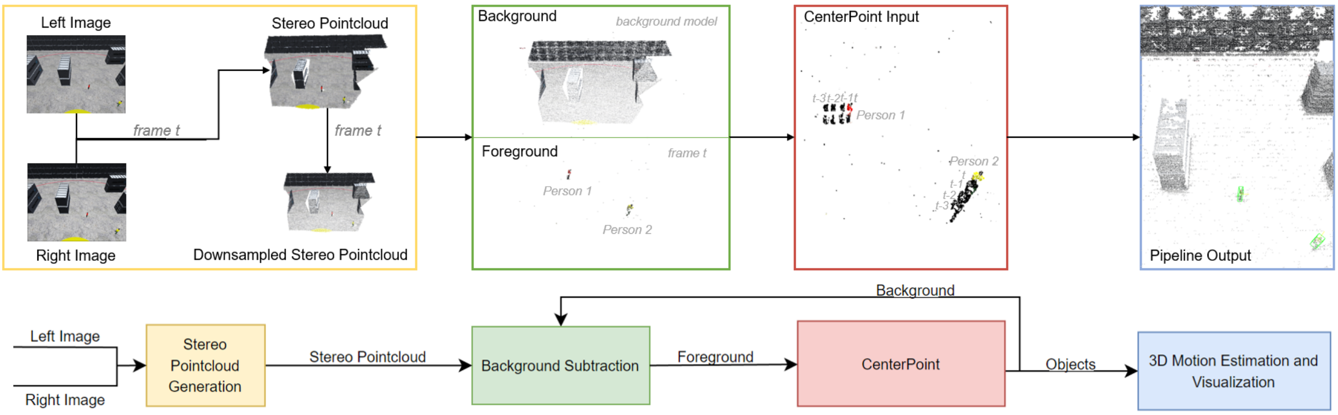


Fig. 4. MOSES Framework. Yellow: Stereo Pointcloud Generation Module. Green: Background Subtraction Module. Red: CenterPoint. Blue: 3D Motion Estimation and Visualisation.

Only requiring two additional regression outputs, the velocity in the BEV plane, the velocity prediction method is efficient and effective. This study uses CenterPoint as an enabling technology for its strength in estimating the position of objects and predicting velocity. Its center-based approach and efficient processing make it ideal for identifying individuals, and its predicted velocity output facilitates straightforward motion estimation. By tuning the stereo pointcloud representation towards CenterPoint, the network is applicable to the current situation.

III. Simulation

A. Gazebo

As no real-world model of the crane has been deployed yet, the method proposed in this paper relies on simulated data from Gazebo. Robot Operating System (ROS) handles the interaction between subcomponents. Static environments with dynamic objects are created, where the dynamic objects are persons designed with varying appearances. A recursive A* algorithm with obstacle avoidance was developed to simulate motion. Locations and realistic orientations are generated at each timestep for each person manoeuvring through the environment. A challenge in achieving synchronisation in ROS and Gazebo is the uncertain time lags that may occur in the process in any sub-component or during component interaction. The objective of synchronising actual object positions and captured camera frames revealed that some subcomponents were asynchronous. A thorough examination revealed that the main time lag in the process occurred due to the hardware our simulation was running on. The computational resources available limit the visualisation speed of model position updates in the simulator and prevent the achievement of real-time processing. A key factor in this process is the real-time factor, representing the ratio of simulation time to real-world time. In an ideal scenario, the real-time factor equals 1 and the simulation runs in real-time. For MOSES, a decision was made to limit the Real Time Factor within the simulation to 0.1 to improve synchronisation between the registration of model position updates in the Gazebo software and the visualisation speed of model position updates in the environment by the graphics engine. This facilitates the automated extraction of ground-truth annotations. ROS handles sensor output synchronisation and produces two synchronised data packages per second. Two data packages per second is a low frequency compared to the camera sensor update rate of 30 Frames Per Second (FPS). However, as the Gazebo simulation is only used for data generation to evaluate the proposed end-to-end pipeline, the mentioned values for data synchronisation speed and the real-time factor are inconsequential. An important consequence of the low data generation frequency is the potential for the trajectories that people follow to appear discontinuous, which poses challenges for the subsequent parts of the pipeline.

B. Sensor Setup

The configuration of the stereo camera setup of MOSES is illustrated in Figure 2. The baseline distance equals 0.7m, and the left-camera coordinates are $(1.5, 0.35, 15)$ with a positive pitch around the y-axis of 45°. With the current setup and the coordinates as mentioned, a focal length (f) of 1690.7 and image height of 2048 (h_{image}), the vertical FOV (VFOV) equals 60° (Equation 1, [53]). Accordingly, the ground plane distance in the x-direction within the FOV of the images (i.e., the depth to where objects are within camera FOV) is 5.5m to 56m. The depth within FOV of the sensor suite depends on the altitude and orientation of the stereo camera setup and is a design consideration for real-world applications.

$$VFOV = 2 \cdot atan(\frac{0.5 \cdot h_{image}}{f}) \tag{1}$$

C. Environment Design

Challenges that need to be overcome when modifying Gazebo worlds include accurate physics modelling, the inclusion of atmospheric conditions and the modelling of interactions between objects and the environment. Including an object with mass and inertia can have unforeseen effects on the simulation. In addition, detecting and resolving collisions between objects is computationally intensive and can cause unrealistic behaviour. Therefore, for each environment, the objects are set to be weightless and static at their exact location except when explicitly modified. Three environments were designed in Gazebo for data generation; a sparse environment, dense environment, and a noisy environment (Figure 20, Figure 21 and Figure 22 as visualised in Appendix A.). The sparse environment hosts containers and people and serves as a baseline for method development and evaluation. The dense environment is more chaotic and has a random spread of containers, dumpsters, trucks, cars and fire hydrants. The dense environment dataset is more complex and can serve as an indication of the adaptability of the proposed method to different small port environments. The noisy environment replicates real-world conditions more closely with increased color, lighting, and 10% camera sensor noise. Figure 3 shows the design of the noisy port environment and the inspiration for the scene's layout; a Google Earth screenshot of the small port of Mykonos in Greece.

IV. Method

The proposed end-to-end pipeline for 3D motion estimation is visualised in Figure 4. The method is divided into subparts; stereo pointcloud generation, background subtraction, CenterPoint and 3D Motion Estimation and Visualisation.

A. Stereo Pointcloud Generation

1) Stereo Matching: For MOSES, an efficient stereo pointcloud generation method able to reconstruct the environment without strict ad-

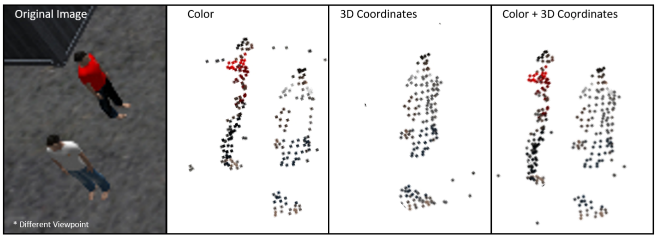


Fig. 5. The effect of including color and/or 3D spatial information in GMM Background Subtraction. Including only color risks losing foreground objects with colors similar to the background. Only 3D coordinates loses salient objects. Color + 3D coordinates as GMM input returns all objects of interest as foreground elements.

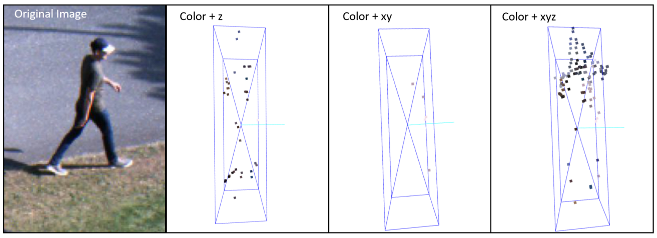


Fig. 6. The effect of each 3D coordinate on GMM Background Subtraction. Including xyz coordinates retains the most information. Blue: ground-truth bounding boxes. Light blue: ground-truth velocities.

herence to full pixel-to-pixel matching is desired. During the initial phase of the study, PSM-Net [25] was exploited for stereo matching. However, our current hardware proved inadequate to accommodate the memory demands of the framework. Therefore, Semi-Global Block Matching (SGBM) [11], a simple handcrafted approach with adjustable thresholds for matching confidence, is implemented. SGBM is a stereo-matching algorithm that uses stereo image pairs to estimate disparity maps. It divides the images into small blocks and minimises a cost function to match blocks between two images. The algorithm integrates local and global information by evaluating the cost along multiple paths. A median filter with a 5x5 window is applied to the created disparity map to reduce noise, outliers and depth artefacts and improve the robustness against illumination variations and textureless regions [22]. The configuration of the SGBM algorithm and the way the environment is perceived is sensitive to the sensor configuration. For example, the disparity range considered should be increased for sensors at low altitudes to match both far-away objects and nearby objects accurately.

2) Stereo Pointcloud Preprocessing: Stereo Preprocessing involves viewpoint reorientation and stereo pointcloud downsampling. Viewpoint reorientation transforms the stereo pointclouds from the left-camera coordinate system to the world coordinate system. The transformation provides a consistent reference frame for all crane subsystems and improves 3D object motion estimation. Aligning the stereo pointcloud with the larger spatial enables a more thorough understanding of object-environment interactions, such as collision avoidance and path planning for autonomous crane operations.

Among the methodologies employed to downsample stereo pointclouds with spatial and color information, voxel-based downsampling and reducing the image resolution are considered. Voxel-based downsampling while retaining the voxel average as a representative point ensures a uniform distribution of points in the downsampled pointcloud. Downsampling decreases memory usage and simplifies the unstructured stereo pointcloud data representation in a structured grid format. However, fine details may be lost due to the smoothing effect on the pointcloud. Additionally, depending on the strategy chosen to select representative points, it could introduce depth inaccuracies in subsequent pipeline components. Similarly, pointcloud downsampling by decreasing the image resolution can effectively reduce the computational load of subsequent parts of the pipeline. Decreasing the image resolution is not without limitations, however, as it reduces the number of pixels available for accurate disparity matching, especially for small objects such as persons. Particularly, significantly reducing image resolution before the Semi-Global Block Matching (SGBM) process carries the risk of failed matches and the loss of objects in the environment.

A hybrid method is implemented to address these considerations within the MOSES pipeline. The image resolution is reduced by 50% (2592x2048 to 1296x1024) to improve the processing speed of the SGBM while retaining detailed texture information in the image. Subsequently, voxel-based downsampling is performed, using a voxel size of 0.1 and averaging all points within the voxel to one representative point with spatial and color attributes. While this might introduce depth discrepancies, the latency advantages of the method outweigh its limitations for the current implementation. Importantly, this downsampling process successfully retains the necessary level of detailed scene information, striking a balance between computational efficiency and preserving scene intricacies.

B. Background Subtraction

Gaussian Mixture Modeling (GMM) is a probabilistic framework applied in scenarios where the data distribution can be characterised by multiple modes [13]. Most studies apply GMM to images [12, 13, 34, 35]. Here, the distribution of feature vectors (e.g. color or color and gradient) is modelled without explicitly taking into account pixel coordinates [54]. [55] uses GMMs on data generated by a RGB-D camera, expanding the feature vector of each pixel with measured depth. The depth values of a stereo pointcloud become more erroneous for increasing sensor noise or in challenging environmental conditions. Here, characterising the different components in the stereo pointcloud based on only color or color and depth becomes less reliable or even infeasible. Therefore, within the MOSES processing pipeline, GMM is leveraged for background subtraction using 3D stereo pointclouds with per-point 3D spatial (xyz) and color attributes. Figure 5 supports the inclusion of spatial features for GMM background subtraction and Figure 6 illustrates the relevance of including each of the 3 coordinates (xyz) explicitly. Figure 6 shows that $Color + xyz$ outperforms $Color + z$ (implicit inclusion of image pixel coordinates and explicit depth information similar to [55]) with respect to the preservation of information of objects of interest. $Color + xyz$ explicitly takes the color and 3D coordinates of each point in the stereo pointcloud into account, shows the best performance, and is therefore implemented in the MOSES pipeline. The combination of information improves the segmentation and makes the implementation more robust. In addition, the background model can be updated directly based on the remaining points after object detection, without the need to project the 3D points to 2D for GMM fitting. Moreover, the 3D stereo pointcloud is more sparse than the 2D images, so fitting the GMM on the background and evaluating the similarity of a new stereo pointcloud is faster. An acknowledged disadvantage of GMMs is the required prior knowledge of the environment, as the number of components used for background model fitting must be sufficient to capture the complexity of the scene. Addressing this challenge is beyond the scope of the current study.

GMMs describe data as a distribution of a linear combination of Gaussian components (Equation 2, [13]).

$$P(X) = \sum_{i=1}^{K} \pi_i \mathcal{N}(X|_{\mu_i, \sum_i}) \tag{2}$$

In Equation 2 X is the observed data, K the number of Gaussian components, π_i the weight of the i-th component (the proportion of the data assigned to the i-th component), μ_i the mean vector, and $\sum_i$ the covariance of the i-th Gaussian component of the background model. In the context of background subtraction, the objective of GMMs is to identify pixels or data points in new input data that deviate from the established distribution of background components. To achieve this, GMMs first fit a probabilistic model to the background pointcloud, allowing for the

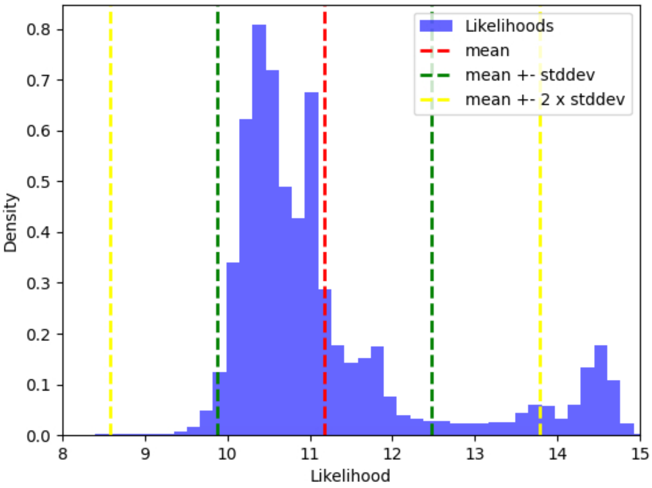


Fig. 7. Example distribution of foreground likelihoods. A likelihood closer to the 0 indicates a point more likely to be part of the background model.

derivation of individual Gaussian distributions corresponding to different components of the background scene. When a new stereo pointcloud is introduced, the log-likelihood sum is computed for each point, which is the sum of its potential association with any background component. As these log-likelihood values become more negative, they indicate a decreasing likelihood that the points associated with these likelihood scores are similar to the background distribution. The score values do not have a directly interpretable meaning in terms of physical units or probabilities. They represent the relative likelihood that a point belongs to the background model compared to other points and components in the GMM.

A preprocessing step is executed for the log-likelihood scores to refine the outcome. Each likelihood sum is transformed using: $Score = Log(-Score)$. Now, a high log-likelihood value indicates a decreasing likelihood of a point being part of the background. This recalibration enables the visualisation of the different components in the foreground, facilitating subsequent filtering and background subtraction. Figure 7 shows the transformed likelihood scores per point of a new stereo pointcloud based on a fitting of five Gaussian components to a background model of the sparse environment dataset. While five Gaussian distributions are discernible in the likelihood scores of foreground points, the background model is considered a single distribution for which a measure to determine both similarity and dissimilarity is desired.

To effectively split foreground points from the background in the new stereo pointcloud, a threshold mechanism is employed. Data points with log-likelihood values exceeding this threshold are classified as foreground elements, implying that they belong to a component which is distinct from the background distribution. On the contrary, points with log-likelihood values below the threshold are retained as background elements. The threshold value is determined as the sum of the mean and twice the standard deviation ($mean + 2 \cdot stddev$) of the current likelihood scores. This threshold helps to establish a clear division between the foreground and background points for the current observation of the environment. The result of this process is visualised in Figure 8.

In addition to background subtraction as detailed, background model initialisation and background model maintenance are essential. Initialisation and maintenance follow the same structure. Establishing a robust background model in a dynamic environment starts with gathering the residual points after object detection from a buffer of N frames. From those frames, static elements must be identified as they are likely to be (new) background parts; therefore, the buffer needs to be evaluated for temporal consistency. By applying a GMM to the buffer, it can identify clusters with narrower, taller curves that indicate stable, unchanging background points. On the contrary, broader, shorter curves form clusters of points that are likely to belong to moving objects or noise, highlighting variations over time. These clusters are likely to indicate points that do not belong to the background. The points corresponding to a taller curve are added to the background model for every N frames. The delayed updates ensure that points belonging to a false negative detection are not included in the background model and thus improves model robustness. For the MOSES-pipeline, N = 4 is implemented.

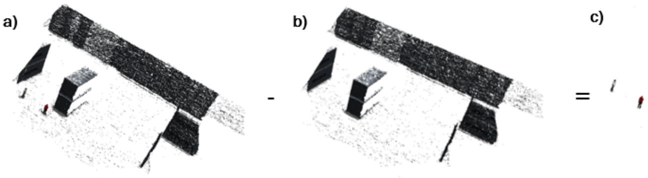


Fig. 8. Background Subtraction. a) Stereo pointcloud of the current frame. b) Retained background. c) Resulting foreground after subtraction.

C. CenterPoint

CenterPoint is chosen for its integration of object detection and object velocity prediction [15]. While the predicted velocity is most often applied for tracking [7], in MOSES, the velocity is used to estimate the motion of objects. CenterPoint uses VoxelNet [44] as a 3D backbone to handle the stereo pointcloud data. VoxelNet divides the 3D space into a grid where each cell contains information about all points within the voxel bounds. Each voxel is projected to BEV to enable the use of 2D convolutional networks, ideal for MOSES where ground plane movement is of predominant interest. Subsequently, the 2D CenterNet architecture [56] is applied, which uses heatmaps to predict the centerpoints of the object based on the BEV data, where high probabilities indicate a high likelihood of an object being present. Decoding these heatmaps for each object attribute produces precise 3D bounding box predictions by identifying peak points as potential object centers. The velocity estimation module matches detected 3D centerpoints of objects across frames and computes their motion vectors, allowing for the prediction of their speed and direction of movement.

1) Temporal Information: CenterPoint requires temporal information for velocity prediction. For this, each point is augmented with a timestamp in an additional feature channel. The current frame t is assigned a timestamp of 0, while preceding frames receive positive time increments relative to frame t. The MOSES pipeline uses frames t, t-1, t-2, and t-3 as input, providing three temporal references for accurate velocity estimation.

2) Color Information: Stereo pointclouds generated based on images allow us to exploit the per-point color information. Therefore, the colors can be learned as discriminative features to improve detection performance [24, 46, 57]. In addition, stereo pointclouds are unstructured pointclouds where the spatial structure of the points representing the objects of interest is not always a robust representation of the object features. Therefore, the decision was made to include per-point color information as input into the CenterPoint network. This required customisation of the network architecture. The number of per-point input features was changed from 5 (x, y, z, intensity, time) to 7 (x, y, z, r, g, b, time), since a stereo pointcloud has no intensity attributes but does have color information. The color information of the current frame t is included, and the color information of frames t-1, t-2 and t-3 is excluded. This improves detection performance and helps the network discriminate between the points in the current frame for object detection and the points included as temporal information for velocity prediction.

D. 3D Motion Estimation and Visualization

3D Motion Estimation is based on the desired prediction horizon for the current application. The velocity of the object is assumed to be constant over the prediction horizon and is modelled from its centerpoint. The

Dataset	**mAP** ↑	**Recall** ↑	**AOE** (rad) ↓	**AVE** (m/s) ↓	**IoU 0.25** ↑	**MDS** ↑
Sparse Environment	0.749	0.768	0.128	0.248	0.666	0.781
Dense Environment	**0.815**	**0.794**	**0.127**	**0.240**	**0.777**	**0.816**
Noisy Environment	0.550	0.625	0.153	0.290	0.407	0.664

TABLE I
PERFORMANCE EVALUATION OF MOSES PIPELINE. BOLD HIGHLIGHTS THE BEST ENTRY PER COLUMN.

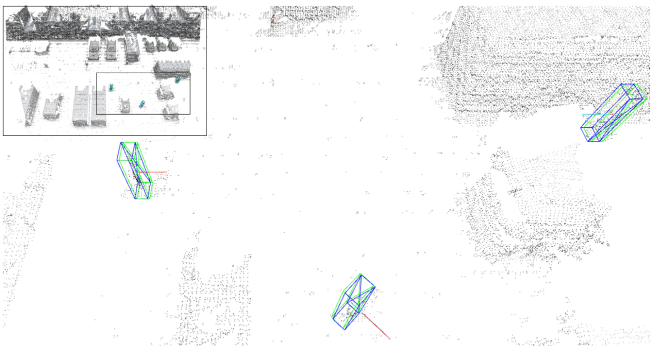


Fig. 9. Qualitative results of the MOSES pipeline on the Dense Environment validation set. Ground-truth bounding boxes are shown in blue, ground-truth velocities are shown in light blue, detected bounding boxes are shown in green, and predicted velocities are shown in red.

prediction horizon is long-term (1.5s) [58]. The visualisation takes the network output and displays object detection and motion prediction on the downsampled stereo pointcloud to facilitate a practical representation for operational implementations.

V. EXPERIMENTS

A. Settings

1) Implementation Details: The implementation of CenterPoint in the OpenPCDet Python library is used. For MOSES, the loss of all outputs of the network is equally weighted (x, y, z, dx, dy, dz, yaw, vx, vy). Training is done for 20 epochs with a batch size of 4. Inference times are measured on a GeForce GTX 970 GPU and an Intel Core i7 CPU. The times presented serve to compare and evaluate design decisions for the MOSES pipeline.

2) Datasets: Datasets used during the experiments are simulated datasets of a sparse environment, a dense environment, and a noisy environment. The sparse environment dataset consists of 1611 frames, the dense dataset of 1811 frames and the noisy dataset of 1745 frames. The distribution of locations and orientations included in the datasets, object velocities and other specifics can be found in Appendix A. For each dataset, the voxel size is set to (0.125, 0.15, 0.25) with a detection range of x [0,50], y [-30,30], and z [0,10] corresponding to the stereo pointcloud dimensions. The x-axis points towards the scene, the y-axis from right to left, and the z-axis points upward. The number of voxels for training and testing is set to 80000. The training dataset is augmented with 5% random scaling, random rotation by 45 °, and random flipping along the x-axis similar to [59]. 70% of the data is used for training and 30% for validation.

3) Evaluation Metrics: Evaluation metrics to examine the performance of the MOSES pipeline are mAP [60], Recall, Intersection over Union (IoU) at 0.25 overlap [61], and the true positive metrics Average Orientation Error (AOE), and Average Velocity Error (AVE) [62]. The mAP is based on a BEV centerpoint distance of 0.1, 0.3, and 0.5m. AOE represents the smallest difference in yaw angle between the predicted and ground-truth orientations in radians. AVE is the absolute velocity prediction error in m/s. Recall, AVE and AOE are reported for an object centerpoint distance of 0.3m. The metrics are consolidated in a downscaled nuScenes Detection Score [62], the MOSES Detection Score (MDS). The true positive metrics (AVE, AOE) are converted as Error$_{score}$ = 1-Error. A weight of 2 is assigned to mAP and a weight of 1 to each error. The MDS is divided by 4 to calculate the normalised sum. Notably, while IoU @ 0.25 currently lacks distinctiveness in evaluating person detection performance, its significance could potentially increase with the inclusion of more object categories in the detection network. Therefore, it is presented for reference but not incorporated into the MDS calculation, as person localisation is valued over object dimension detection for MOSES.

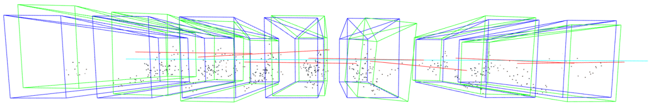


Fig. 10. 7 frame trajectory of a single person. Ground-truth bounding boxes are shown in blue, ground-truth velocities are shown in light blue, detected bounding boxes are shown in green, and predicted velocities are shown in red.

Processing Step	SGBM	Downsampling	BS	3D Detection
Time	131.6 ms	77.9 ms	70.7 ms	15.6 ms

TABLE II
DETAILED TIMING ANALYSIS OF MOSES.

B. Experiment Results

1) Performance evaluation: Figure 9 presents a visualisation of the 3D object detection and motion estimation output of the MOSES pipeline. Figure 10 shows a segment of an object trajectory. The ground-truth velocity is based on a single previous frame and, therefore, is sensitive to sudden changes in object location. The predicted velocity takes the previous three frames into account and changes more gradually as a result. Quantitative results of the proposed pipeline on the validation set of the sparse, dense and noisy environment can be seen in Table I. It is noted that the mAP score for the dense environment dataset is higher than for the sparse environment dataset. This can be attributed to a difference in dataset composition (e.g. different number of frames, number of partially occluded objects). For the dense environment, an orientation error of 0.127 rad equals 7.3°, an error of 4% at a maximum of 180°difference. The velocity error of 0.240 m/s at a maximum object velocity of 2.55 m/s gives an error of 9.4%. The higher velocity error can be influenced by two key factors; manual preprocessing of the dataset for compatibility, and sudden changes in object location (and consequently velocity) due to inconsistencies in the data generation frequency of the simulation. In addition, stereo pointclouds come with a particular depth uncertainty, possibly increased by the downsampling of the stereo pointcloud. This is expected to affect the localisation accuracy and, as a consequence, the velocity prediction accuracy of the MOSES pipeline. Orientation detection is less affected by these factors. As the noisy environment has more sensor noise introduced to the camera sensors, the performance of the pipeline on the dataset is expected to degrade, which can be seen in Table I. Although performance is satisfactory, there is room for improvement. With more sensor noise, the stereo pointcloud is less accurate in depth direction, and learning robust features in a stereo pointcloud with higher spatial uncertainty requires more reference data. A next step would be to increase the size of the dataset used for the experiment to evaluate this hypothesis.

Dataset	**Color**	**mAP ↑**	**Recall ↑**	**AOE** (rad) ↓	**AVE** (m/s) ↓	**IoU 0.25 ↑**	**MDS ↑**
Sparse Environment	frame t	**0.749**	**0.768**	**0.128**	**0.248**	**0.666**	**0.781**
	frame t, t-1, t-2, t-3	0.632	0.691	0.180	0.276	0.498	0.702
Dense Environment	frame t	**0.815**	**0.794**	**0.127**	**0.240**	**0.777**	**0.816**
	frame t, t-1, t-2, t-3	0.616	0.682	0.218	0.303	0.517	0.678

TABLE III

THE EFFECT OF INCLUDING COLOR INFORMATION ON MOSES PIPELINE PERFORMANCE. BOLD HIGHLIGHTS THE BEST ENTRY FOR EACH DATASET.

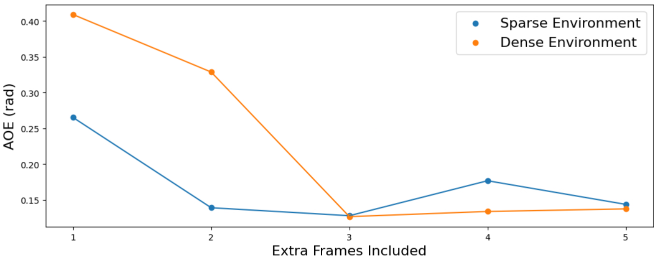


Fig. 11. AOE versus Temporal Information.

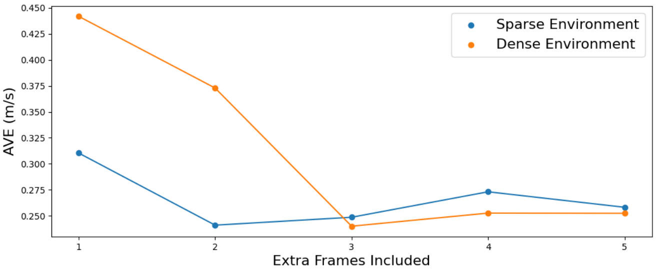


Fig. 12. AVE versus Temporal Information.

2) Latency Assessment: As object detection and velocity prediction in port environments will only be of added value if the pipeline has real-time or near real-time performance, the latency of the MOSES pipeline is evaluated. The definition of real-time performance depends on the application. For some applications, $\sim 11FPS$ equals real-time performance [59] and for others $\sim 30FPS$ or higher is required [22]. For MOSES, the processing frequency of other components (e.g. the container detection and localisation pipeline) equals 10 FPS, which serves as a real-time processing reference. Table II illustrates the latency breakdown of the MOSES pipeline proposed in this study for a single set of stereo images. The total processing time (with our current hardware) is equal to $\sim 3FPS$. Nevertheless, the computer is not considered high performance, which significantly affects the latency. High-performance hardware and pipeline optimisation by means of process parallelisation is expected to increase the processing speed to (near) real-time performance. Furthermore, the breakdown of the processing speed of $\sim 3FPS$ can serve as an indication of directions for further research. From Table II, it can be deduced that the bottlenecks in the current processing pipeline are SGBM for pointcloud creation and the subsequent voxel-based downsampling of the pointcloud. The optimisation of these building blocks is a potential research direction to further decrease the latency of the end-to-end method in place.

3) Temporal Information: The influence of number of temporal reference frames provided as input to CenterPoint on the performance of the MOSES pipeline is seen in Figure 11 and Figure 12. It can be seen that increasing the extra frames included improves the AOE and AVE at 0.3m centerpoint distance. The performance improvement indicates that the temporal correlation among sequential frames plays a role in refining object detection and localisation. The improvement stagnates at the inclusion of frames t-1, t-2, and t-3 as temporal information, which suggests that the network can only effectively use information from a limited temporal window. Therefore, an additional three frames are provided as input to the network for temporal reference to balance enhanced performance and increased model complexity. Increasing the number of input frames increases the computational requirements of the network, including memory and processing power. In real-time applications, this could introduce latency concerns. However, for MOSES with pointclouds of $\sim$1000 points (for the current sensor noise setting) after background subtraction, the increase in computational requirements is negligible.

4) Color Information: The color information experiment is performed to understand the effects of including color data from the current frame (frame t) while retaining the temporal context provided by frames t-1, t-2, and t-3. The effect of including color information for the current frame (frame t) or for all frames (frame t, t-1, t-2, and t-3) on the performance of the proposed MOSES pipeline is visualised in Table III. The results demonstrate that introducing color information exclusively for the current frame leads to improvements in the performance evaluation metrics. Augmenting 3D points with color information for the current timestamp can help the network better differentiate object boundaries and other characteristics. This is particularly beneficial when objects have complex shapes or textures. The decrease in mAP for the inclusion of color information from all frames could indicate the struggle of the network to distinguish the boundaries of the objects from the temporal window provided when each frame is augmented with appearance features. The performance improvement caused by adding color information is more pronounced for the dense port environment ($+32.3\%$ mAP for the dense environment versus $+18.51\%$ mAP for the sparse environment). An explanation is that in an occupied environment the persons are more often in the proximity of background objects, making it harder to distinguish object boundaries. Therefore, the performance increases more significantly when color information is included.

5) Cross-Environment Generalisation: Cross-environment generalisation is examined to validate the hypothesis that voxel-based downsampling and background subtraction aid in enforcing a robust object representation within a neural network. This representation, in turn, should facilitate accurate object detection in diverse and previously unseen environments, provided that voxel-based downsampling is performed and the background is also subtracted in those environments. The effect of voxel-based downsampling and background subtraction are evaluated separately for clarification. To assess the effect of background subtraction, two datasets were selected to train the network: one consisting of the sparse port environment with background subtraction and the other featuring the same environment without background subtraction. The generalisation of the learned features across environments was assessed by testing the network on the noisy environment and dense environment with and without background subtraction according to the respective training dataset. Table IV shows the results of the experiment. The best performance for cross-environment generalisation was achieved when the model was trained on the sparse environment dataset with background subtraction, as shown by the test results on unseen datasets with background subtraction. The findings support the hypothesis of the advantage of background subtraction for cross-environment generalisation, of value for a mobile sensor setup such as MOSES which is expected to attend to new port environments by autonomous vessel continuously. The conducted experiment highlights the potential benefits of background subtraction in the context of cross-environment generalisation for object detection within small port environments. While the results are promising,

Training Data	**Test Data**	**mAP** ↑	**Recall** ↑	**AOE** (rad) ↓	**AVE** (m/s) ↓	**IoU 0.25** ↑	**MDS** ↑
Sparse Environment (BGS)	Dense Environment (BGS)	**0.612**	**0.746**	**0.141**	**0.255**	**0.641**	**0.707**
	Noisy Environment (BGS)	**0.422**	**0.405**	0.226	0.311	**0.288**	**0.577**
Sparse Environment	Dense Environment	0.487	0.541	0.257	0.311	0.326	0.536
	Noisy Environment	0.013	0.011	**0.019**	**0.013**	0.0083	0.49

TABLE IV
CROSS-ENVIRONMENT GENERALISATION EXPERIMENT OF MOSES-PIPELINE. BGS STANDS FOR BACKGROUND SUBTRACTION. BOLD HIGHLIGHTS THE BEST ENTRY FOR EACH TEST DATASET.

Training Data	Downsampling	**Test Data**	**mAP** ↑	**Recall** ↑	**AOE** (rad) ↓	**AVE** (m/s) ↓	**IoU 0.25** ↑	**MDS** ↑
Sparse Environment	YES	Dense Environment	**0.612**	**0.746**	**0.141**	**0.255**	**0.641**	**0.707**
		Noisy Environment	**0.422**	**0.405**	0.226	0.311	**0.288**	**0.577**
Sparse Environment	NO	Dense Environment	0.378	0.599	0.1864	0.2627	0.308	0.577
		Noisy Environment	0.169	0.238	**0.189**	**0.236**	0.0796	0.478

TABLE V
EFFECT OF VOXEL DOWNSAMPLING ON CROSS-ENVIRONMENT GENERALISATION. BOLD HIGHLIGHTS THE BEST ENTRY FOR EACH TEST DATASET.

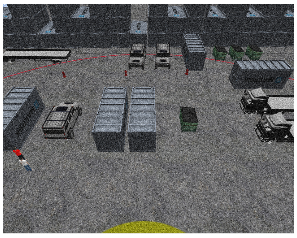


Fig. 13. Dense Port Environment with 20% Sensor Noise.

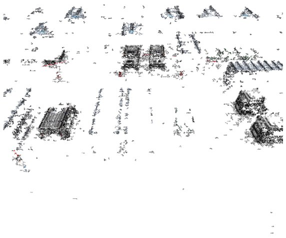


Fig. 14. Effect of noise on SGBM matching result.

Sensor Noise (stddev)	**SGBM (points)**	**Downsampling (points)**
0.007	1,312,487	90,142
0.01	1,265,180	87,989
0.05	520,408	94,285
0.1	204,082	19,591
0.2	47,551	2,857
0.3	10,204	612
0.4	714	143
0.5	204	41

TABLE VI
EFFECT OF INCREASING NOISE ON POINTCLOUD SIZE FOR THE DENSE PORT ENVIRONMENT.

there are challenges and limitations associated with this approach. It is essential to note that the network is trained with the same settings for both datasets, with and without background subtraction. The observed performance variations can be therefore be partly attributed to the limited time that the network is given to learn the increased input space. Table IV also shows the weakness of MDS. The bottom row shows that an extremely low mAP corresponds to low AOE and low AVE. This occurs when only a few true positives are detected, and those detected are very accurate. The resulting MDS of 0.499, while it seems to demonstrate good performance, is meaningless. The identical sensitivity flaw can be identified in the widely recognised nuScenes Detection Score, making it acceptable to keep the MDS calculation as is and always report it in combination with mAP, AVE and AOE.

The effect of voxel-based downsampling on cross-environment generalisation is evaluated with two baseline datasets, one without downsampling and one with voxel-based downsampling, but both with background subtraction. Table V shows the results of the experiment performed. It can be seen that voxel-based downsampling outperforms a method where no downsampling would be applied for both test datasets. It should be noted that without downsampling, the AOE and AVE in the noisy environment test dataset are lower than the errors when testing is performed on the noisy dataset with voxel-based downsampling. The reason for this is similar to that previously mentioned. AOE and AVE are true positive metrics, which are biased toward the accuracy of orientation detection and velocity prediction compared to the number of object detections.

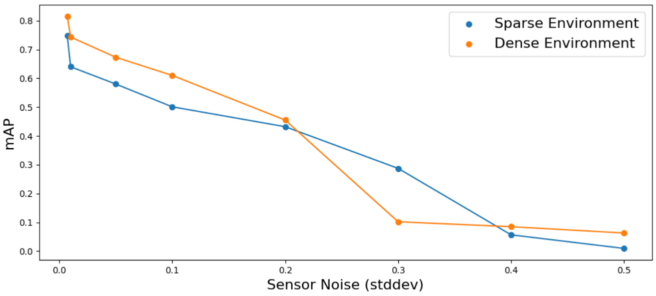


Fig. 15. Effect of increasing sensor noise on mAP on the validation set of the Sparse and Dense Environment

6) Sensitivity to Noise: In Gazebo, camera sensor noise is modeled by independently adding a Gaussian-sampled disturbance to each pixel, with adjustable mean and standard deviation. The default standard deviation (stddev) is set to 0.007. To achieve a realistic simulation of real-world environments, standard deviations in the range of 0.01 to 0.5 are considered, with lower values applicable to daylight conditions and higher values to simulate adverse weather conditions. Figure 13 visualises the effect of 0.2 stddev noise on the 3D information that can be retrieved by the SGBM algorithm. While the reduction in matched pixels is significant, it can be seen that high-texture regions (such as persons) are retained, a valuable finding for the subsequent evaluation of the impact of noise on the performance of the MOSES-pipeline.

The impact of elevated sensor noise on the stereo pointcloud size generated by the SGBM and number of points that remain after downsampling is visualised in Table VI. The results show the pointcloud size as sensor noise is increased for the simulated dense port environment. The observations seem to indicate that up to a 0.2 standard deviation increase in sensor noise, an adequate amount of information is retrieved. However, augmenting the noise beyond this threshold leaves too few points for accurate background subtraction and subsequent object detection.

The final part of this experiment evaluates the MOSES pipeline performance for 0.01, 0.05, 0.1, 0.2, 0.3, 0.4 and 0.5 standard deviation Gaussian noise introduced to the camera sensors to evaluate low, moderate and high noise conditions. Datasets are created for each noise level for both the sparse and the dense environment. Figure 15 shows the effect of increasing sensor noise on the mAP performance indicator. For the sparse environment dataset, a significant degradation in performance is visible for noise greater than 0.3 stddev. At 0.4 standard deviation the mAP nears zero, indicating that learning becomes almost impossible.

Dataset	**Downsampling**	**Latency**	**mAP** ↑	**Recall** ↑	**AOE** (rad) ↓	**AVE** (m/s) ↓	**IoU 0.25** ↑	**MDS** ↑
Sparse Environment	Voxel (Averaged Center)	**77.9 ms**	**0.749**	**0.768**	**0.128**	**0.248**	**0.666**	**0.781**
	Voxel (Nearest Centroid)	26357.9 ms	0.643	0.764	0.311	0.355	0.494	0.698
	Angular Downsampling	100 ms	0.678	0.720	0.201	0.289	0.559	0.716

TABLE VII
ABLATION STUDY FOR STEREO POINTCLOUD DOWNSAMPLING. BOLD HIGHLIGHTS THE BEST ENTRY PER COLUMN.

Dataset	**BGS**	**Training Time**	**mAP** ↑	**Recall** ↑	**AOE (rad)** ↓	**AVE (m/s)** ↓	**IoU 0.25**↑	**MDS** ↑
Sparse Environment	YES	00:06:15	**0.749**	**0.768**	**0.128**	**0.248**	**0.666**	**0.781**
	NO	02:55:10	0.601	0.643	0.250	0.319	0.549	0.659
Dense Environment	YES	00:06:55	**0.815**	**0.794**	**0.127**	**0.240**	**0.777**	**0.816**
	NO	03:15:10	0.768	0.792	0.250	0.348	0.680	0.736

TABLE VIII
ABLATION STUDY FOR BACKGROUND SUBTRACTION (BGS). BOLD HIGHLIGHTS THE BEST ENTRY PER DATASET.

Dataset	**Color Information**	**mAP** ↑	**Recall** ↑	**AOE (rad)** ↓	**AVE (m/s)** ↓	**IoU 0.25** ↑	**MDS** ↑
Sparse Environment	RGB	0.749	**0.768**	**0.128**	**0.248**	**0.666**	**0.781**
	Gray	**0.760**	**0.768**	0.159	0.271	0.649	0.772
	None	0.707	0.738	0.160	0.275	0.648	0.745
Dense Environment	RGB	**0.815**	**0.794**	**0.127**	**0.240**	**0.777**	**0.816**
	Gray	0.795	0.797	0.174	0.260	0.691	0.789
	None	0.738	0.763	0.223	0.293	0.674	0.740

TABLE IX
ABLATION STUDY FOR COLOR INFORMATION. BOLD HIGHLIGHTS THE BEST ENTRY PER DATASET.

On the dense environment the pipeline experiences a steep decrease in performance from 0.2 to 0.3 stddev sensor noise. Object detection in sparse environments is more robust to sensor noise compared to a more occupied environment as the reduced occlusion and surrounding objects facilitate more reliable object detection. Therefore, the network is expected to learn predictions accurately up to a Gaussian sensor noise with 0.2 standard deviation for diverse port environments (when presented with adequate data). After that, the MOSES pipeline fails to learn and perform.

C. Ablation Studies

1) Stereo Pointcloud Downsampling: It is acknowledged that the current implementation of voxel-based downsampling could introduce depth discrepancies in the subsequent parts of the pipeline. Therefore, an ablation study is performed considering other downsampling approaches. As the image resolution has already been decreased, further reduction of this is left out of scope to avoid the risk of losing fine details in the environment during stereo matching. The downsampling methods considered along with voxel-based downsampling while retaining the averaged center and color information as a representative point (Voxel (Average Center) in Table VII) are voxel-based downsampling while retaining the point nearest to the voxel centroid as a representative point (Voxel (Nearest Centroid)) and Angular Downsampling [24]. Angular Downsampling returns spatial and color information of points at specific angles similar to the data generated by a LiDAR sensor, which might improve the performance of CenterPoint (originally built for LiDAR data) on the generated stereo pointcloud data. However, angular downsampling may lead to an uneven point distribution, potentially resulting in underrepresentation of the environment and loss of fine details (small objects). Voxel (Nearest Centroid) might alleviate the concern of depth discrepancies introduced by point averaging, as it retains a point that was part of the original stereo pointcloud. Nevertheless, the resulting point distribution is irregular and influenced by noise and outliers.

Table VII shows the performance of the MOSES pipeline on the sparse environment dataset with different stereo pointcloud downsampling methods. The Voxel (Averaged Center) approach introduces the lowest latency into the pipeline while achieving high performance. The Voxel (Nearest Centroid) method has similar precision but increased AOE, AVE and latency. It should be noted that the implementation of Voxel (Nearest Centroid) downsampling might not be optimised yet, but the current low latency limits its broader application. Voxel (Nearest Centroid) was evaluated for its possible improvement with regards to localisation accuracy. However, it seems that the irregularity of a pointcloud based on voxel centroids compared to a stereo pointcloud based on Voxel (Averaged Center) downsampling results in an increase in detection and prediction errors. Angular Downsampling strikes a balance between latency and performance metrics, showing timely and reasonably accurate object detection. In practical implementations, the selection of a downsampling method should be in line with real-world requirements. For MOSES the Voxel Averaged Center method outperforms all other downsampling methods on the current evaluation metrics, therefore, the choice of downsampling method is validated.

2) Background Subtraction: In the framework of the MOSES pipeline, the background subtraction (BGS) module holds significant importance. In early stages of the study, other background subtraction methods were considered. Spatial Color Thresholding was implemented, where points in the current frame were segmented into foreground and background points based on their nearest neighbour distance (both based on color and 3D location) to the background model. For optimal performance, the method required a different threshold for each environment. As the aim of the MOSES-pipeline is to generalise to diverse environments and changing circumstances, this method is unsuitable. Therefore, the current ablation study focusses on the inclusion or exclusion of background subtraction in the pipeline, and not on the effect of different background subtraction approaches. Table VIII emphasises the advantages of background subtraction in the proposed MOSES pipeline, demonstrating improvements in performance metrics such as increased mAP, improved recall, reduced AOE, and higher IoU scores in sparse and dense environments. It is important to note that the network is trained with identical settings for both dataset types (with and without background subtraction). Part of the observed performance discrepancy can therefore be attributed to the difference in input variables and the insufficient training time for the network to acquire a robust feature representation of the input when no background subtraction is performed. An advantage of BGS is its computational efficiency and simplification of the learning process, resulting in faster convergence and improved cross-environment generalisation. It is important to consider the potential limitations of background subtraction. The main concern is the risk of removing relevant points due to the implementation of an aggressive background subtraction method. Within the current pipeline, efforts are

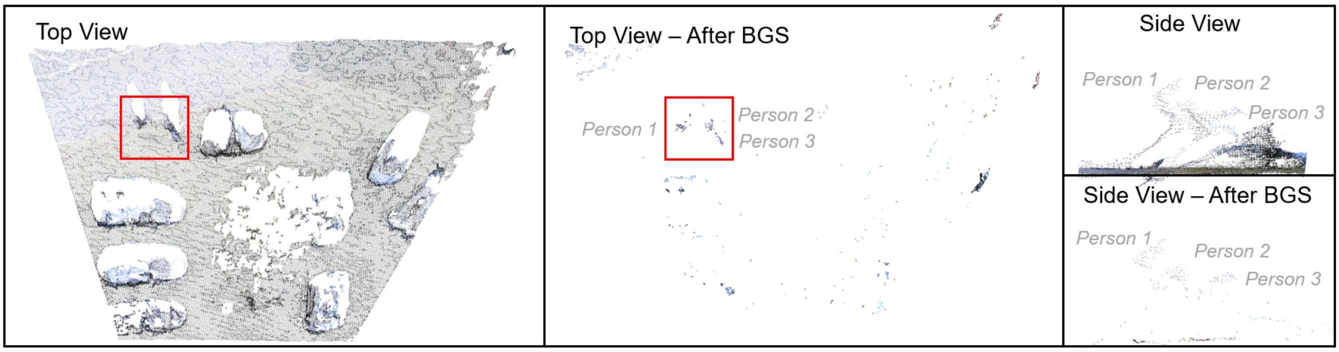


Fig. 16. Background Subtraction (BGS) result on the real-world dataset. The two images for top and side view show the stereo pointcloud before and after BGS based on a 5 component GMM is performed.

made to alleviate this concern by means of the N-frame buffer before background model initialisation and model updates are performed.

3) Color Information: The integration of color information increases the computational complexity of the pipeline due to the augmentation of stereo pointcloud features. An ablation study is therefore essential to validate the design decision. The upcoming ablation study examines the effects of the inclusion of color information, and illustrates the justification for selecting RGB over grayscale representation. RGB color information has been shown to lead to better discrimination, resulting in improved object detection performance, as indicated in Table IX. The data demonstrates that incorporating RGB information yields slight performance enhancements for the sparse port environment dataset, and more significant enhancement for the dense port environment. In the sparse environment, including an appearance cue of either grayscale or RGB yields an approximately similar performance improvement. For the dense environment, with more diverse and more complex objects, RGB information is demonstrated to be a more discriminative feature. With closely spaced objects and more frequent occlusions, the inclusion of multi-scale RGB color information disambiguates objects more effectively compared to grayscale information. The data in Table IX show that incorporating RGB color information improves detection performance when looking at all environments, with higher mAP, recall, reduced AOE, and AVE, along with increased IoU scores. Although the ablation study supports the design decision and current implementation, it is essential to recognise potential downsides. Integrating RGB information introduces an additional processing burden due to increased input dimensionality. Additionally, color variations arising from challenging lighting conditions and changing object appearances were not addressed in the simulation, but could reduce or eliminate the performance improvement used as motivation for the design decision.

D. Real Dataset

After concluding a thorough examination of the proposed method by means of experiments, and a validation of the pipeline design decisions by ablation of its main components, a final experiment is conducted on a real-world dataset. The dataset is generated at 12.5 m altitude with downward oriented stereo cameras at an angle of 45°. The environment has a static tree close to the sensors, static cars, and three persons walking along almost horizontal lines. The real-world dataset is visualised in Figure 17. The sensor setup is similar to that of the virtual setup in Gazebo, and therefore the hypothesis is that the optimised MOSES pipeline and the conclusions of the ablation study based on simulated data can be transferred to the real-world dataset and the applicability of the MOSES pipeline to real-world scenarios can be proved. The real dataset consists of 189 frames for which the ground truth boxes had to be manually labelled. To facilitate the annotation process, we used existing 2D labels on the images of the left camera. A 5x5 window around the center of the 2D bounding boxes is projected on the depth image and used as a prior for the 3D object centerpoint location. After automated generation of the 3D boxes, their locations and dimensions were manually refined. Ground-truth velocities are established by performing a greedy Euler distance association of annotations for each frame t and t-1. Velocities in the x and y directions are based on the centerpoint distance between the annotations in the two consecutive frames. For an Euler distance exceeding 1m, the velocity is generalised to $1m/s$ in the negative or positive y direction, depending on the displacement of the person between the two frames. Manual annotation of 3D bounding boxes proved challenging, as it relied on visual inspection of stereo pointclouds as the only reference. Hence, inaccuracies in object position, potentially up to decimeter-scale, and consequently velocity inaccuracies are present. Furthermore, ground-truth orientation was not annotated. Two experiments are performed with the real-world dataset; a general performance evaluation and a cross-environment generalisation assessment from data generated in simulation to real-world data.

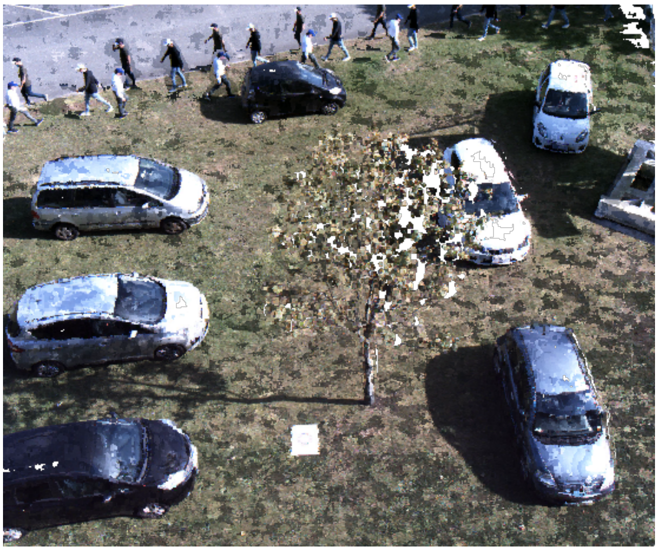


Fig. 17. Real-world Dataset. Screencapture of an overlay of 9 stereo pointclouds. Static cars, the central tree and trajectories followed by the three walking persons can be identified.

1) Real Dataset Performance: At first, the performance of the pipeline on the real-world dataset is assessed. For this, the dataset is divided into 142 frames for training and 47 frames for validation. The first building blocks of the pipeline are voxel-based downsampling and GMM based background subtraction. Figure 16 shows the output of this process on the real-world dataset. The leftmost top view shows a voxel-based downsampled stereo pointcloud where the data is reordered in

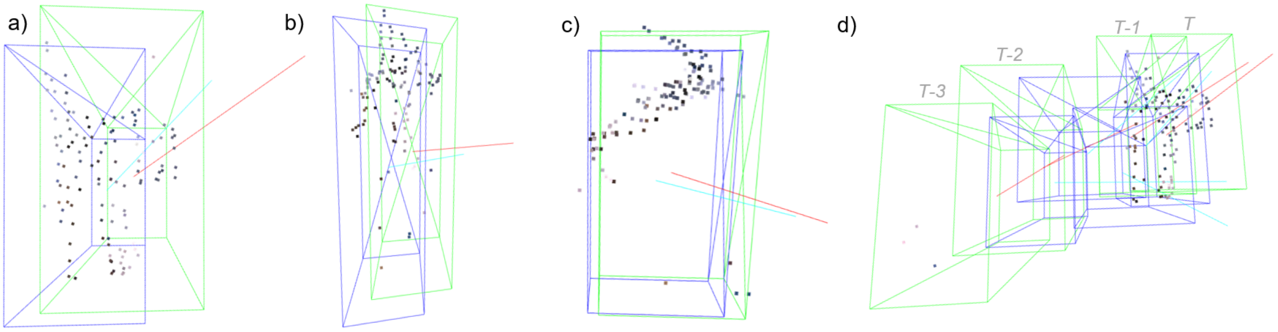


Fig. 18. MOSES Pipeline output on the real-world dataset on a single frame zoomed in on one bounding box. Blue: ground-truth bounding box. Light blue: ground-truth velocity. Green: predicted bounding box. Red: predicted velocity. a) Top view. b) Front view. c) Side view. d) Trajectory visualisation.

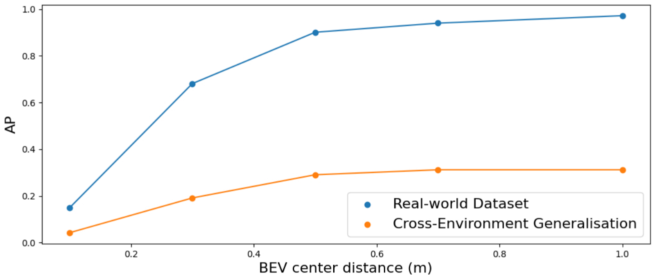


Fig. 19. Performance of MOSES pipeline on real-world dataset and Cross-Environment Generalisation with training on dense dataset environment, testing on real-world data. AP versus BEV center distance between ground truth and predicted object location.

a structured grid format. Both images to the right of Figure 16 (top and side view) show the remaining points before and after BGS with a 5-component GMM is performed. The CenterPoint output on a single frame zoomed in to one bounding box is shown in Figure 18. Figures a), b) and c) illustrate that the ground-truth annotations and the detections each capture a different part of the points representing the object and thereby visualises the inaccuracies in the ground-truth annotations. As the ground-truth annotations have an inherent object location deviation, the evaluation metrics cannot become completely accurate. However, the qualitative illustration of object detection and prediction of object velocity shows reasonable performance. Figure 18 d) illustrates the trajectory of one person and the associated predicted and ground-truth velocities. Here, it is important to illustrate that the created ground-truth velocities do not nearly represent the trajectory of the object well. However, the end-to-end pipeline is capable of predicting the trajectory of the person over time very accurately.

Figure 19 presents the increasing precision of the MOSES-pipeline for an increase in allowed distance between ground truth object centers and predicted object centerpoints. While this might seem a trivial statement, its purpose here is to be able to assess the performance on the real-world data while taking into consideration the inaccuracy of the manual annotations. It can be seen that for an increase in allowed center distance, the precision quickly increases to above 90% accuracy (at 0.5m), and the improvement in performance stagnates afterward. Therefore, it is expected that in the case of more precise labels, the mAP as considered for the simulated data (averaged over 0.1, 0.3, 0.5 m center distance) will illustrate highly accurate performance on the real-world dataset. Hence, the hypothesis, as mentioned above, that the MOSES pipeline can be applied to real-world data is adequately verified. Evaluation of AOE and AVE is left out of scope, as the ground-truth velocities are inaccurate (Figure 18 d)) and no ground-truth orientations were annotated.

2) Cross-Environment Generalisation (Simulation to Real-World): Cross-environment generalisation of the proposed pipeline was evaluated for the datasets generated in the simulation. We assessed how well the features learnt on a sparse port environment dataset adapt to datasets of dense and noisy port environments. A deeper level of understanding of cross-environment generalisation of the proposed end-to-end MOSES pipeline can be gained when an evaluation is done where the training dataset is simulated and the test is performed on real-world data. For this experiment, the dense environment dataset is chosen as a training dataset, as it is expected that the appearances of persons are more challenging and diverse compared to the sparse environment and the representations learnt will be more robust to transfer to real-world data. Figure 19 illustrates the precision for increasing BEV center distance range for a network trained on the dense environment and tested on the real-world data. Considering the difference in environmental conditions, the coarse manual labelling of the real-world data and the small dataset size, an AP of 0.291 at 0.5m centerpoint distance is promising. Similar to Figure 19, the performance improvement stagnates after 0.5m distance. Future works can be pursued in the direction of training on a large simulated dataset and finetuning the network parameters on a small real-world dataset to improve the performance of the end-to-end pipeline in real-world conditions.

VI. Discussion

The main objective of this study is to address the challenges associated with person detection and velocity prediction in small port environments using a sensor suite located on an autonomous crane. The focus is on designing a robust processing pipeline that incorporates stereo camera information for 3D object motion estimation and safety monitoring. The main components of the proposed pipeline are the stereo matching method, background subtraction based on Gaussian Mixture Models and the usage of CenterPoint on stereo pointclouds. Experiments are performed on simulated data as no real-world model of the crane has been deployed yet. The results of the experiment provide clear insight into both the strengths and limitations of the approach, which, together with an experiment on a basic real-world scenario, are crucial for assessing its practicality in the real world.

A general performance evaluation reveals an mAP of 81.5%, an orientation error of 4% and a velocity error of 9.4%. The metrics show a satisfactory level of accuracy but simultaneously expose areas for refinement. The main sources of error are the depth uncertainty inherent to the disparity matching method (due to e.g. image noise and the discrete nature of disparity levels) and the downsampling of stereo pointclouds. The vulnerability of the velocity prediction to these factors is evident, as velocity depends on localisation accuracy. The experiments illustrated that 3 temporal reference frames ensure optimal velocity prediction and orientation detection. A downside to using three time steps is the delay in the prediction of velocity, which makes an object of interest less safe. In addition, for more complex real-world trajectories, the motivation behind 3 frames for temporal reference might not endure. Therefore, for real-world applications, using one frame as a temporal reference might be more suitable. Moreover, if movement toward the crane can be detected, the object of interest will be visible in subsequent frames, and the velocity prediction can be refined afterwards. An evaluation of end-to-end processing of simulated data revealed a processing frequency $\sim 3fps$. The FPS is expected to increase by using more advanced hardware. A latency breakdown revealed that the bottlenecks in the current pipeline are SGBM and voxel-based downsampling. Therefore, the trade-off between a large image required for small object disparity matching and a concise stereo pointcloud for efficient background subtraction and subsequent 3D object detection and velocity prediction must be investigated. Augmenting points with color information as input into CenterPoint was found to improve the performance of the MOSES pipeline, particularly in more occupied environments where the detection of object boundaries is challenging. Aspects not addressed in the simulation are possible variations in object color caused by challenging lighting conditions or changing object appearances. These could diminish the illustrated performance gains or even counteract the proposed method, removing the motivation for the design decision. The final part of the pipeline to consider is the subtraction of the background. A novel method for background subtraction with Gaussian Mixture Models is proposed, where the model is fit on a stereo pointcloud with per-point color and 3D spatial (xyz) information. This has superior performance on both simulated and real-world data compared to other implementations. Background subtraction was included in the pipeline to foster cross-environment generalisation by establishing a robust object representation that generalises to unseen environments. The pipeline trained with background-subtracted data outperforms that trained on data without background subtraction. This finding underlines the significance of background subtraction in enhancing general features within the network, improving performance in unseen environments - a valuable finding for MOSES, given its mobile sensor setup. The sensitivity of background subtraction to changes in lighting and challenging environmental factors remains to be evaluated.

A final test was performed on a real-world dataset with a sensor configuration similar to the stereo cameras on the autonomous crane. The dataset had to be manually annotated, which introduced position and velocity inaccuracies in the ground-truth labels and led to the exclusion of object orientation information. A performance evaluation revealed an accuracy of >90% on an allowed center distance of 0.5m. This is a good indication that the pipeline developed on simulated data is robust and applicable to the real world. It is expected that similar performance can be achieved at a smaller allowed center distance if the ground-truth annotations are refined. However, the current evaluation is limited to a basic real-world scenario, and more complex scenarios remain to be assessed. A cross-environment generalisation experiment revealed that the motivation for background subtraction and voxel-based downsampling to create a uniform object representation also applies to the transition of simulated data to real-world data. Training on the dense environment dataset and testing on real-world data resulted in an AP of 0.291 on 0.5m center distance. This preliminary assessment of transfer learning is promising.

Future research recommendations are to evaluate pipeline latency on high-performing hardware with respect to the desired real-time performance (10 fps) and to perform experiments on more and larger real-world datasets. More experiments on real-world data should provide insight into the effect of challenging lighting conditions and changing object appearances on the performance of the MOSES pipeline. Additionally, realistic real-world scenarios might present other aspects that should be considered, but were not reproduced in simulation, such as object-object interactions and complex object trajectories. Also, it is expected that e.g. terrain height differences significantly affect the result of the stereo-matching module and with this all subsequent parts of the pipeline. Taking these real-world scenarios into account will increase the overall robustness of the proposed MOSES pipeline.

VII. Conclusion

This paper presents an end-to-end pipeline for stereo point cloud-based 3D object detection and motion estimation in small port environments for safety monitoring where stereo cameras are attached to an autonomous crane on an autonomous vessel. Main pipeline components are stereo pointcloud generation, background subtraction, and CenterPoint for object detection and velocity prediction. The voxel-based downsampling of the stereo point cloud and subsequent Gaussian Mixture Modelling-based background subtraction create a structured downsized pointcloud representation improving the robustness of learned features and consequently the cross-environment generalisation of the pipeline. The novel method in which a Gaussian Mixture Model is fit on the stereopoint cloud with per-point color and 3D spatial (xyz) information shows excellent performance both in simulation and reality. The inclusion of color information enhances the performance of CenterPoint. Experiments on a real-world dataset prove that the Moses pipeline designed and tuned in simulation also works for a real-world scenario. Future work recommendations focus on latency evaluation on high-performing hardware to reveal the bottlenecks for real-time performance and experimenting on real-world data of more complex scenarios to evaluate the effect of noise, changing lighting conditions and challenging object trajectories on the performance of the proposed pipeline.

References

[1] Gelayol Golcarenarenji, Ignacio Martinez-Alpiste, Qi Wang, and Jose Maria Alcaraz-Calero. Machine-learning-based top-view safety monitoring of ground workforce on complex industrial sites. *Neural Computing and Applications*, 34(6):4207–4220, 2022.

[2] Moses: Automated vessels and supply chain optimization for short sea shipping. https://moses-h2020.eu/, 2020. Accessed: 18-01-2023.

[3] Hans van den Broek and Jasper Waa. Intelligent operator support concepts for shore control centers, 02 2022.

[4] Yang Liu, Peng Sun, Nickolas Wergeles, and Yi Shang. A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 172:114602, 2021.

[5] Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 2020.

[6] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-Fast-RCNN: hard positive generation via adversary for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2606–2615, 2017.

[7] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022.

[8] Rui Qian, Xin Lai, and Xirong Li. 3d object detection for autonomous driving: a survey. *Pattern Recognition*, page 108796, 2022.

[9] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3): 1341–1360, 2020.

[10] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

[11] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814. IEEE, 2005.

[12] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 28–31. IEEE, 2004.

[13] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[14] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, volume 2, pages 246–252. IEEE, 1999.

[15] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[16] Sven Fleck, Florian Busch, Peter Biber, and Wolfgang Straber. 3d surveillance a distributed network of smart cameras for real-time tracking and its visualization in 3d. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 118–118. IEEE, 2006.

[17] James Black, Tim Ellis, and Paul Rosin. Multi view image surveillance and tracking. In *Workshop on Motion and Video Computing, 2002. Proceedings.*, pages 169–174. IEEE, 2002.

[18] Jian Yao and Jean-Marc Odobez. Multi-camera 3d person tracking with particle filter in a surveillance environment. In *2008 16th European Signal Processing Conference*, pages 1–5. IEEE, 2008.

[19] Jin-hyung Park, Seungmin Rho, and Chang-sung Jeong. Real-time robust 3d object tracking and estimation for surveillance system. *Security and Communication Networks*, 7(10):1599–1611, 2014.

[20] Osman Topçu, A Aydin Alatan, and Ali Ozer Ercan. Occlusion-aware 3d multiple object tracker with two cameras for visual surveillance. In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 56–61. IEEE, 2014.

[21] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[22] Christian Banz, Sebastian Hesselbarth, Holger Flatt, Holger Blume, and Peter Pirsch. Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation. In *2010 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, pages 93–101. IEEE, 2010.

[23] Stefan K Gehrig, Felix Eberli, and Thomas Meyer. A real-time low-power stereo vision engine using semi-global matching. In *International Conference on Computer Vision Systems*, pages 134–143. Springer, 2009.

[24] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019.

[25] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5418, 2018.

[26] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[27] Marco Cristani, Michela Farenzena, Domenico Bloisi, and Vittorio Murino. Background subtraction for automated multisensor surveillance: a comprehensive review. *EURASIP Journal on Advances in signal Processing*, 2010:1–24, 2010.

[28] Dongdong Zeng, Ming Zhu, and Arjan Kuijper. Combining background subtraction algorithms with convolutional neural network. *Journal of Electronic Imaging*, 28(1):013011–013011, 2019.

[29] Jhony H Giraldo and Thierry Bouwmans. Semi-supervised background subtraction of unseen videos: Minimization of the total variation of graph signals. In *2020 IEEE international conference on image processing (ICIP)*, pages 3224–3228. IEEE, 2020.

[30] Wenbo Zheng, Kunfeng Wang, and Fei-Yue Wang. A novel background subtraction algorithm based on parallel vision and bayesian gans. *Neurocomputing*, 394:178–200, 2020.

[31] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1): 359–373, 2014.

[32] Rui Wang, Filiz Bunyak, Guna Seetharaman, and Kannappan Palaniappan. Static and moving object detection using flux tensor with split gaussian models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 414–418, 2014.

[33] Massimo De Gregorio and Maurizio Giordano. Cwisardh: Background detection in rgbd videos by learning of weightless neural networks. In *International Conference on Image Analysis and Processing*, pages 242–253. Springer, 2017.

[34] Massimo Piccardi. Background subtraction techniques: a review. In *2004 IEEE international conference on systems, man and cybernetics (IEEE Cat. No. 04CH37583)*, volume 4, pages 3099–3104. IEEE, 2004.

[35] Rudrika Kalsotra and Sakshi Arora. Background subtraction for moving object detection: explorations of recent developments and challenges. *The Visual Computer*, 38(12):4151–4178, 2022.

[36] Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile, Hélène Laurent, and Christophe Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[37] Zheng Yi and Fan Liangzhong. Moving object detection based on running average background and temporal difference. In *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, pages 270–272. IEEE, 2010.

[38] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019.

[39] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[40] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[41] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[42] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[43] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6851–6860, 2019.

[44] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[45] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[46] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceed-*

ings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4604–4612, 2020.

[47] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Zhou Bin, and Liangjun Zhang. Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3047–3054. IEEE, 2021.

[48] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse fuse dense: Towards high quality 3d detection with depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5418–5427, 2022.

[49] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366. IEEE, 2020.

[50] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Fast multiple object tracking via a hierarchical particle filter. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 212–219. IEEE, 2005.

[51] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 941–951, 2019.

[52] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[53] Anders Brahme. *Comprehensive biomedical physics*. Newnes, 2014.

[54] Kalpana Goyal and Jyoti Singhai. Review of background subtraction methods using gaussian mixture model for video surveillance systems. *Artificial Intelligence Review*, 50:241–259, 2018.

[55] Benjamin Langmann, Seyed E Ghobadi, Klaus Hartmann, and Otmar Loffeld. Multi-modal background subtraction using gaussian mixture models. In *ISPRS Symposium on Photogrammetry Computer Vision and Image Analysis*, pages 61–66, 2010.

[56] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019.

[57] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[58] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2891–2900, 2017.

[59] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[60] Petru Soviany and Radu Tudor Ionescu. Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. In *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 209–214. IEEE, 2018.

[61] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*, pages 237–242. IEEE, 2020.

[62] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

APPENDIX A
DATASET ANALYSIS

This appendix presents a detailed overview of the characteristic of each dataset simulated for the current study. Figure 20 shows the sparse environment design, Figure 21 the dense environment design and Figure 22 the noisy environment. The main function of the sparse environment is to serve as a baseline for method development. Applying the method to the dense environment can give indications about the behaviour of the pipeline when the input data gets more cluttered. Finally, the noisy environment aims to present a first step towards evaluating the performance of the MOSES-pipeline in more realistic real-world scenarios.

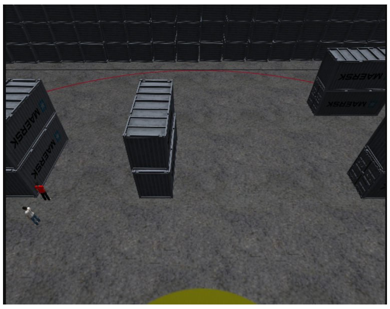


Fig. 20. Sparse Port Environment

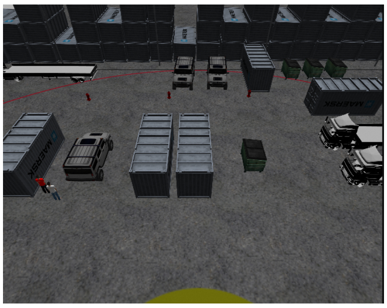


Fig. 21. Dense Port Environment

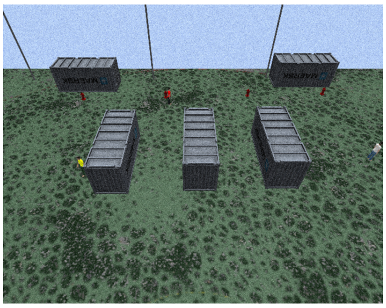


Fig. 22. Noisy Port Environment

Table X shows the numeric characteristics of the different datasets. Each environment was simulated to generate 2000 samples, after which the first preprocessing step entailed getting rid of all frames without objects. Since the trajectories for the dynamic objects are randomly generated, this resulted in a reduced dataset of different size for each environment. 70% of each dataset is used for training and 30% for validation.

Dataset	**Frames**	**Training Frames**	**Validation Frames**	**3D Boxes**	**Object Velocity (mean/min/max)** (m/s)
Sparse Port Environment	1611	1119	492	3533	0.977/0.200/2.55
Dense Port Environment	1811	1256	555	2842	0.971/0.200/2.55
Noisy Port Environment	1785	1300	485	3278	1.00/0.200/2.26

TABLE X
DETAILED DATASET ANALYSIS.

Object attributes of interest to the MOSES pipeline are object location, object orientation and object velocity. Object velocity per dataset is showed in Table X. Figure 23 illustrates the object orientations per dataset. It is evident that within the simulation trajectories from left to right or right to left are preferred, corresponding with an object orientation of 0 or π.

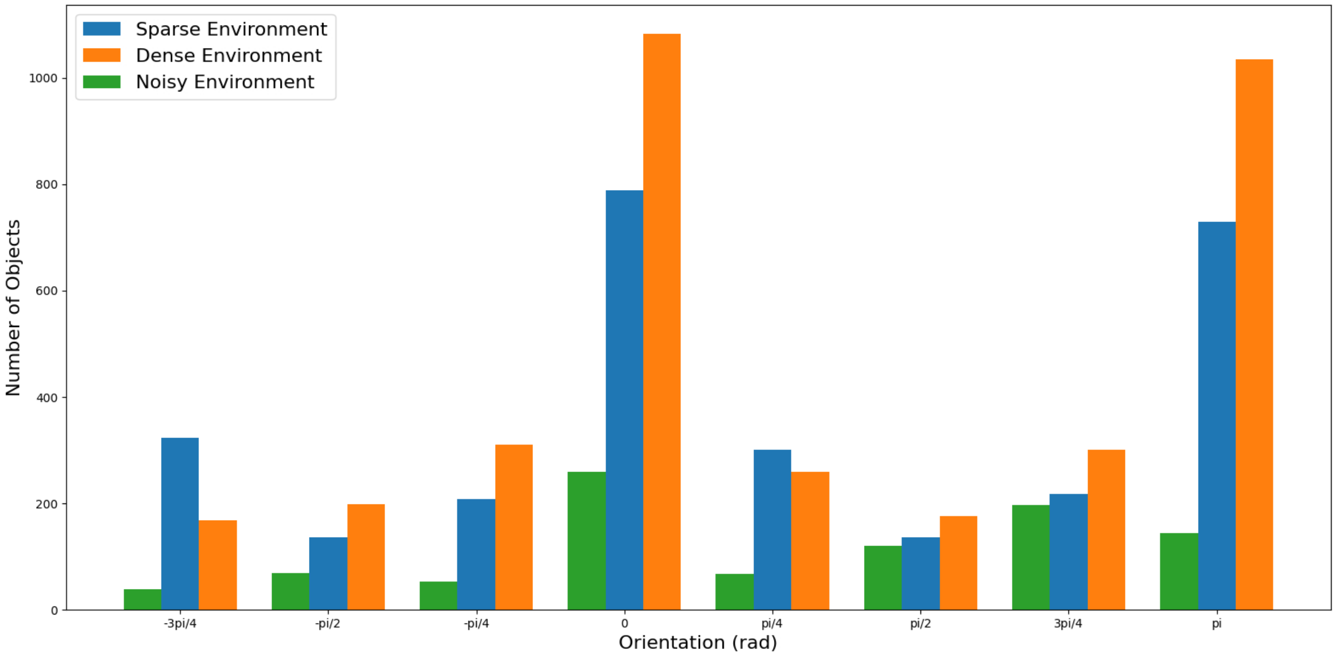


Fig. 23. Distribution of Object Orientations per Dataset.

The distribution of object x,y locations over the environment is displayed in Figure 24. The main takeaway from the figure is that no area is omitted in the trajectory generation. The open spaces that are visible are the objects present in the environment. Therefore, the conclusion can be drawn that the dataset generated is representative for object localization throughout the simulated environments.

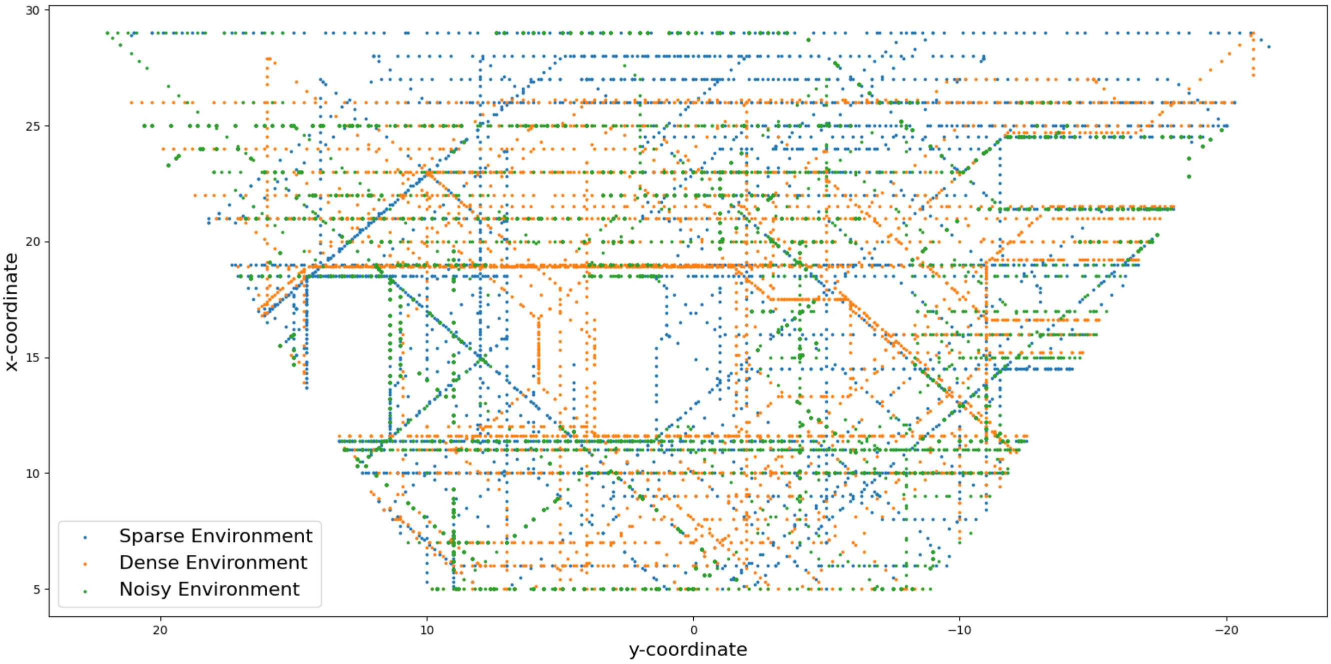


Fig. 24. Variation of Object Locations per Dataset.

Figure 25 and Figure 26 show an example ground truth annotation and respective output of the MOSES pipeline for a single frame in the sparse environment dataset. Figure 27 and Figure 28 present the same data for the dense environment. Finally, Figure 29 and Figure 30 depict identical information but for the noisy environment dataset. The figures give an intuition of the data format used in the MOSES pipeline and visualise the annotations and output of the method per dataset.

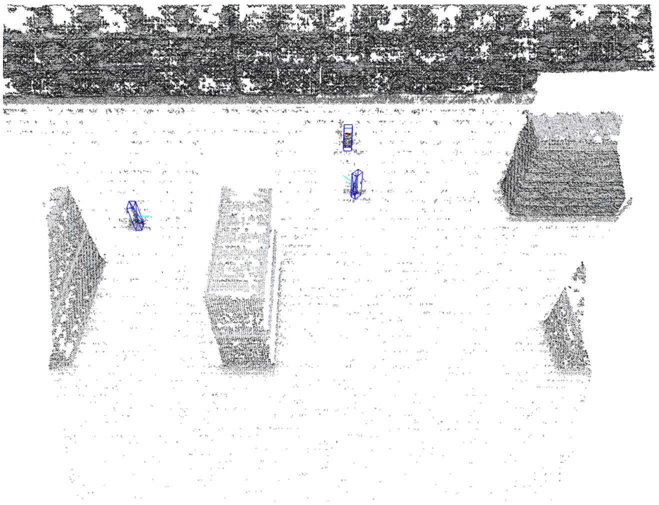


Fig. 25. Ground-truth annotation for a single frame in the sparse environment dataset. Blue: ground-truth bounding box. Light blue: ground-truth velocity.

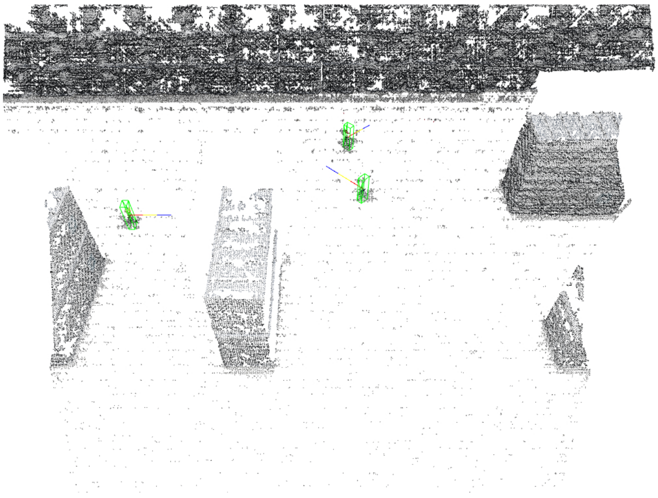


Fig. 26. Output visualisation of the MOSES pipeline on the sparse environment. Object detections are shown in green and t+1, t+2, t+3 object motion estimates are shown in in red, yellow, and blue.

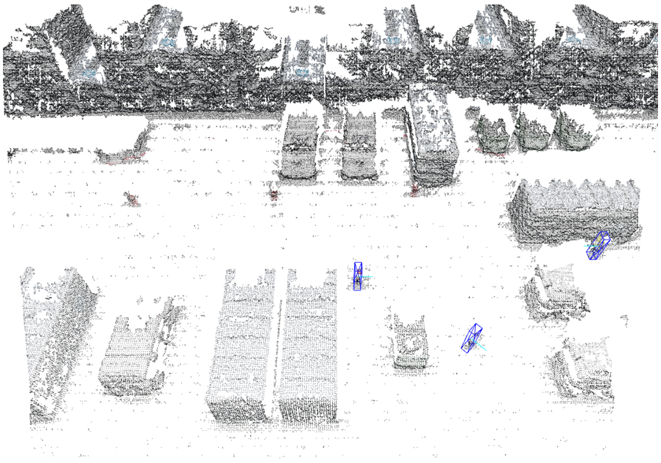


Fig. 27. Ground-truth annotation for a single frame in the dense environment dataset. Blue: ground-truth bounding box. Light blue: ground-truth velocity.

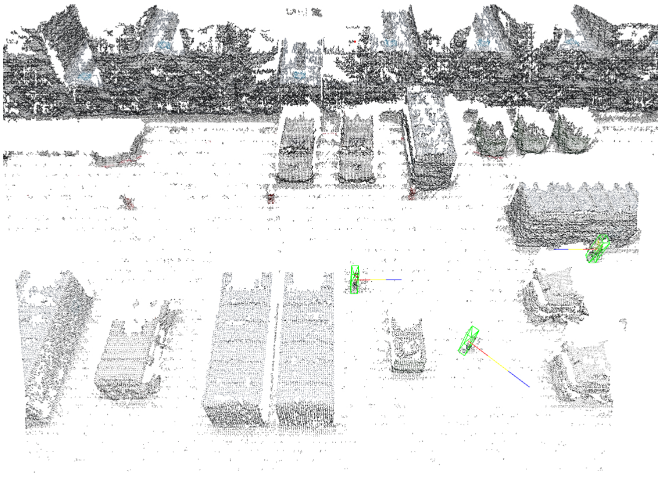


Fig. 28. Output visualisation of the MOSES pipeline on the dense environment. Object detections are shown in green and t+1, t+2, t+3 object motion estimates are shown in red, yellow, and blue

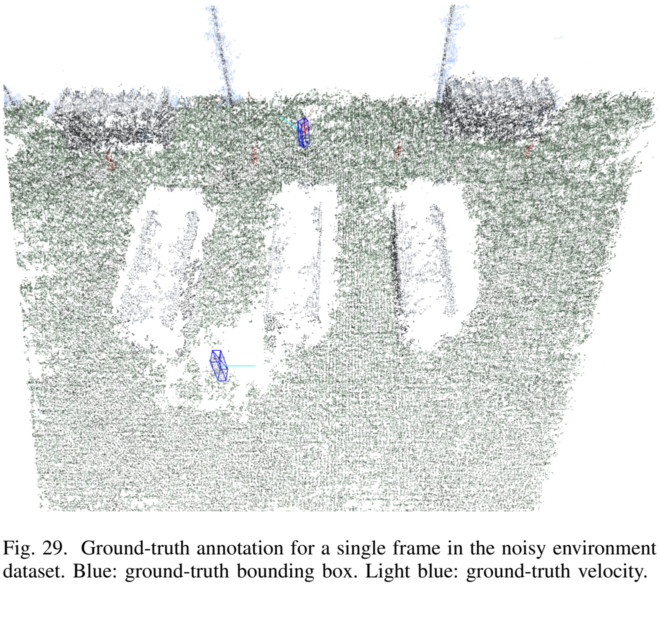


Fig. 29. Ground-truth annotation for a single frame in the noisy environment dataset. Blue: ground-truth bounding box. Light blue: ground-truth velocity.

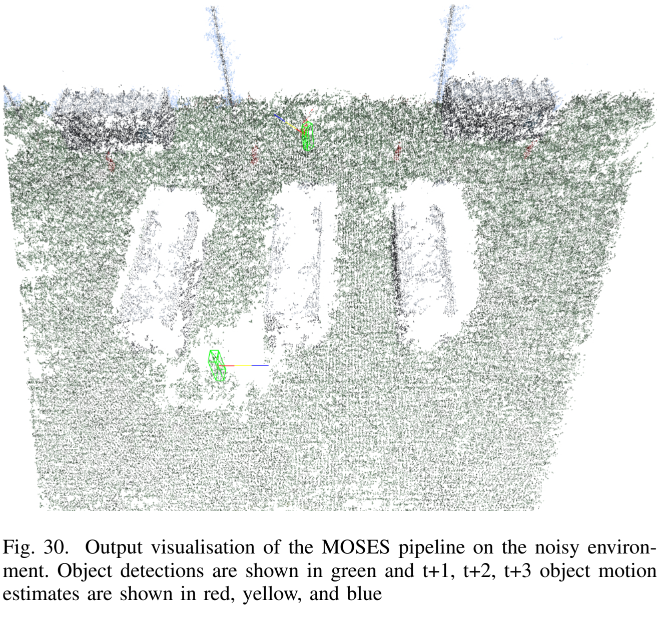


Fig. 30. Output visualisation of the MOSES pipeline on the noisy environment. Object detections are shown in green and t+1, t+2, t+3 object motion estimates are shown in red, yellow, and blue