BEV-SLAM: Building a Globally-Consistent World Map Using Monocular Vision

James Ross[1] j.ross@surrey.ac.uk
Oscar Mendez[1] o.mendez@surrey.ac.uk
Avishkar Saha[1] a.saha@surrey.ac.uk
Mark Johnson[1]
Richard Bowden[1] r.bowden@surrey.ac.uk

***Abstract*—The ability to produce large-scale maps for navigation, path planning and other tasks is a crucial step for autonomous agents, but has always been challenging. In this work, we introduce BEV-SLAM, a novel type of graph-based SLAM that aligns semantically-segmented Bird's Eye View (BEV) predictions from monocular cameras. We introduce a novel form of occlusion reasoning into BEV estimation and demonstrate its importance to aid spatial aggregation of BEV predictions. The result is a versatile SLAM system that can operate across arbitrary multi-camera configurations and can be seamlessly integrated with other sensors. We show that the use of multiple cameras significantly increases performance, and achieves lower relative error than high-performance GPS.**

The resulting system is able to create large, dense, globally-consistent world maps from monocular cameras mounted around an ego vehicle. The maps are metric and correctly-scaled, making them suitable for downstream navigation tasks.

I. Introduction

Mobile autonomous agents require an information-rich representation of their environment for navigation, planning and localisation. Typically, a top-down orthographic projection (also known as a Bird's Eye View (BEV) map) is preferred, since planning tasks are simplified in orthographic space.

However, it is challenging to build this representation from sensors without prior knowledge of the environment. LiDAR is metric and allows orthographic projection, but is sparse, expensive and computationally demanding. Cameras offer an alternative, as they are cheap, already equipped on many vehicles and provide dense, high frame rate data with low throughput. This motivates a body of research aiming to produce BEV maps from monocular cameras alone. However, there is little focus on spatial aggregation of these maps, or their application in other tasks such as mapping and localisation. We present BEV-SLAM, a novel SLAM system combining state-of-the-art computer vision research with graph-based SLAM to produce complete BEV semantic maps from multiple monocular cameras mounted around an ego vehicle.

We use a CNN to map directly from images to a semantically-labelled BEV map. This approach benefits from depth reasoning implicit in the CNN, meaning that, unlike many other SLAM systems, the results are appropriately scaled and can be easily integrated with additional sensors or maps.

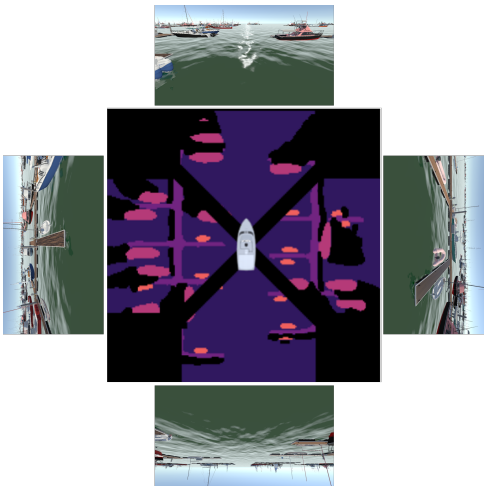

Fig. 1: Multiple occlusion-reasoned Bird's Eye View predictions from monocular cameras mounted around an ego vehicle.

We focus on the maritime domain, where the strong priors available in automotive, such as the uniformity of road structures, are unavailable. To account for the lack of available data, we develop challenging synthetic datasets for semantic BEV prediction and occlusion reasoning, which are used to optimise the system.

BEV-SLAM is built around BEV alignment: we align orthographic BEV predictions using a custom alignment framework, and use this as a motion cue in a pose graph. Loop closure candidates are identified using feature-matching, but are similarly verified using BEV alignment. Unlike the majority of visual SLAM systems, which tend to be sparse in features for feasibility, this provides a rich semantic representation of the world for downstream navigation tasks, such as path planning.

We highlight challenges associated with spatial aggregation of BEV maps due to the currently unsolved issue of temporal inconsistency, and propose explicit occlusion reasoning as a specific solution for BEV-SLAM. Following a thorough analysis of performance in simulation, we then demonstrate that BEV-SLAM is able to recreate trajectories from real-world sequences.

Our contributions are as follows:

[1]Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, UK

1) We develop graph-based BEV-SLAM, a versatile SLAM system with the ability to incorporate multiple sensors.
2) We introduce occlusion reasoning into BEV estimation, and demonstrate the importance of explicit occlusion reasoning for spatial aggregation of BEV maps for SLAM.
3) We present a maritime dataset to provide a challenging test environment for BEV-SLAM, and make it publicly available for training and validation of similar systems.

II. Related Work

A. Bird's Eye View Maps

Mobile autonomous systems require a compact, semantically-rich, spatially-meaningful representation of their environment which captures both the useful aspects of geometry and the overall layout of the scene. The importance of spatial reasoning motivates a body of research that aims to produce overhead views from ground-level inputs, and vice-versa (often called a world map, top-down map, or Bird's Eye View (BEV) map).

There are multiple routes to produce a complete BEV map. The simplest are geometric approaches, using inverse perspective mapping to perform the transformation from the camera perspective space to the orthographic BEV space [1], [2]. However, this results in "shadows" in the occluded regions behind objects that lie above the ground plane, which can only be resolved if the objects are visible from multiple angles. The same issue also affects orthographic projections from LiDAR. One solution is to use a Generative Adversarial Network (GAN) [3], [4], trained with aerial imagery to produce realistic overhead views, but it is challenging to ensure that geometry is preserved [5].

To combat this issue and introduce spatial reasoning, learning approaches have been adopted to transform directly to BEV. These can be divided by whether they require the camera geometry to be provided explicitly [6]–[8], or can implicitly learn the transformation [9]–[11]. BEV predictors have been used as an intermediate representation for the estimation of, for example, 3D bounding boxes [7]. However, building on advances in 2D scene understanding, recent research [11]–[13] has focused on extending semantic segmentation to BEV such that a complete BEV map of the environment is a useful endpoint in itself. Usually, the goal is to produce a semantic occupancy grid, due to its suitability for sensor fusion and deep learning [9]. Some examples use a two-step process [1], [11], performing semantic segmentation in the image plane before performing the transformation to BEV, while others adopt a more streamlined end-to-end learning approach [8], [9].

One advantage of learning approaches is that they are able to reason about objects that are wholly or partially occluded; occlusion reasoning is recognised as a major stepping stone for 3D scene understanding [6], [14], [15], but is challenging using exclusively monocular cameras.

Roddick et al. [8] substantially improved upon previous end-to-end deep learning approaches through the use of pyramid occupancy networks with a multi-scale pyramid to transform between the perspective camera and orthographic BEV coordinate systems, outperforming the previous state-of-the-art [11]. Saha et al. [16] improved upon this and addressed temporal aggregation of BEV maps from monocular cameras. However, temporal consistency remains a concern; current approaches rely upon the uniformity of the automotive environment to aid temporal understanding, which does not extend to other domains. Hu et al. [17] focused on future prediction using instance segmentation, but, to our knowledge, no research currently explores BEV-based mapping or localisation.

B. Visual SLAM

Since Simultaneous Localisation and Mapping (SLAM) is considered a prerequisite for truly autonomous mobile systems, a substantial body of research exists to realize more general solutions and improve mapping systems [18]. Theoretically, SLAM is a solved problem [19]: Leonard et al. [20] demonstrated in the early 1990s that SLAM solutions exist in the infinite data limit. However, issues remain with building information-dense maps for practical applications, and realising hardware-tractable solutions. This motivates the use of deep-learning-based sensors in a SLAM pipeline.

Monocular SLAM is a particularly challenging problem due to the lack of three-dimensional cues and loss of scale. The first solution was MonoSLAM [21], [22], using an Extended Kalman Filter with points tracked between images. Kalman-filter-based approaches build upon this technique, often focusing on improving mid-term data association to retrieve structure from motion, such as in [23].

Global optimisation is required to improve upon these methods; keyframe-based approaches estimate a world map using only a few frames and use bundle adjustment to enforce consistency. Alternatively, graph-based SLAM approaches, where poses are represented as vertices and odometry measurements as edges in a pose graph, perform periodic non-linear optimisation to ensure a consistent world map.

ORB-SLAM [24], and its direct descendants ORB-SLAM2 [25] and ORB-SLAM3 [26], are perhaps the most complete indirect visual SLAM implementations, and remain popular because they can be configured with multiple sensor types, such as monocular, stereo and RGB-D cameras. Here, ORB features are tracked between keyframes, and long-term data association is achieved using a visual bag-of-words. Though ORB-SLAM estimates an output trajectory, it produces a very sparse map that does not retain information useful for downstream navigation tasks.

In contrast, LSD-SLAM [28] is a direct method which constructs a pose graph based on keyframes with an inverse depth map, and directly uses pixel intensities for tracking between images. However, LSD-SLAM struggles with occlusion and moving objects, and is complicated by the transformation from the photometric alignment to world space. LSD-SLAM densifies the map by using more images, but neither ORB-SLAM nor LSD-SLAM produce a complete dense world representation. Later works from the TUM community, such as Direct Sparse Mapping (DSM) [29] and Direct Sparse

Odometry (DSO) [30], which can include loop closure (LDSO) [31], can predict accurate poses where point-detection-based approaches fail, but still produce sparse maps. All of these techniques also produce unscaled maps without the addition of inertial sensors, stereo cameras or learning-based measurements, making integration with other sensors challenging.

More recently, there has been a drive to create flexible frameworks that allow multiple sensors and techniques to be seamlessly combined. Plug-and-Play SLAM [32] and maplab [33] are examples of multi-modal SLAM systems that give the ability to combine different sensor configurations. Plug-and-Play SLAM, for example, has been demonstrated with wheel odometry, inertial sensors, LiDAR and RGB-D cameras. These frameworks attempt to standardise SLAM solutions and demonstrate the importance of modularity and customizability, characteristics that we wish to build into BEV-SLAM.

III. Methodology

BEV-SLAM is developed using the common recursive Bayesian SLAM formulation, where ego pose and map estimates are represented by the probability distribution

$$P(\boldsymbol{x}_k, \boldsymbol{m} \mid \boldsymbol{Z}_{0:k}, \boldsymbol{U}_{0:k}, \boldsymbol{x}_0). \tag{1}$$

In this case, $\boldsymbol{x}$ is the ego vehicle pose, $\boldsymbol{m}$ the BEV map, $\boldsymbol{Z}$ landmark observations in orthographic BEV space, and $\boldsymbol{U}$ alignment between subsequent BEV maps for timestep k. This can be separated into a sensor model

$$P(\boldsymbol{z}_k \mid \boldsymbol{x}_k, \boldsymbol{m}) \tag{2}$$

and motion model

$$P(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{u}_k) \tag{3}$$

in accordance with Bayes' Theorem and the Markov assumption.

We build a pose graph using the alignment of BEV map predictions from monocular images to obtain an initial estimate for the current pose from (3). Landmarks observed in the BEV plane, and alignment between the corresponding maps, can then be used to add additional edges to the graph. We perform a pose graph optimisation to obtain pose estimates $\boldsymbol{x}_{0:k}$, and combine the original BEV maps accordingly to produce a top-down semantic map $\boldsymbol{m} \in \mathbf{R}^{W \times H \times C}$ for C classes (semantic labels). Alternatively, BEV-SLAM can be deployed in an online fashion, where (2) and (3) are computed iteratively, and pose optimisation is performed periodically.

A system overview is presented in Fig. 2.

A. BEV Prediction

Pyramid occupancy networks have been shown to be the state-of-the-art for BEV estimation [8], [16]. Since our goal is to create globally consistent maps, we use a similar approach, with key modifications for occlusion. The input image is passed through a ResNet and feature pyramid at multiple scales, then collapsed along the y-axis and expanded in the Z direction. This is based on the observation that object height in the image plane is a key indicator of distance. A multi-scale transformer layer is applied to remove perspective distortion (taking into account the focal length of the camera), and the output decoded.

For training, we use a multi-scale dice loss across C classes and N scales:

$$\mathscr{L}_{dice} = 1 - \frac{1}{|C|} \sum_{c=1}^{C} \frac{2 \sum_i^N \hat{t_i}^c t_i^c}{\sum_i^N \hat{t_i}^c + t_i^c + \epsilon} \tag{4}$$

where $\hat{t_i}^c$ is the ground truth occupancy and t_i^c is the network prediction (ϵ is a small constant to prevent division by zero).

B. Occlusion Reasoning

Though deep learning approaches to BEV prediction enable more effective spatial reasoning than pure geometric methods, consistency through time remains unsolved and is an active area of research. This is especially true in occluded regions in the input image, where predictions are inherently unpredictable. This creates a problem for spatial aggregation of BEV maps, as naïve alignment will consistently produce inaccurate odometry measurements.

We therefore propose a novel solution: an extra output layer, enabling the network to predict an occlusion mask in addition to the semantic map. The predicted occlusion mask can then be used for the downstream alignment task, to improve the accuracy and temporal consistency of odometry measurements.

We introduce specific rules to encourage temporal consistency: smaller classes are assumed to be entirely visible, even if they are partially occluded, and we assume that we are able to accurately predict the depth of larger classes. This is because the network can reasonably infer shape and size provided part of a small object is visible, and prior work has shown that BEV predictors have sufficient spatial reasoning to accurately predict depth.

Occlusion masks can be rendered directly in simulation using ray-casting; in the real world, similar maps can be obtained through the use of LiDAR.

Example BEV predictions with occlusion reasoning for multiple cameras mounted around a vehicle are shown in Fig. 1 (black indicates an area that is predicted as unknown or occluded).

C. BEV Alignment

Given a pair of predicted semantic maps and occlusion masks, we aim to find the optimal alignment and transform it to an ego vehicle odometry measurement. Since BEV maps are already in orthographic space, and we assume mounted camera height to remain unchanged, the alignment problem reduces to a Euclidian transform. Pitch and roll changes, which are common in maritime, introduce distortions in BEV predictions, but in practice can be normalised out if the images are rotated to ensure a constant level horizon, using, for example, horizon detection. We can therefore use the enhanced correlation coefficient [34] to find a suitable alignment in the BEV plane. The predicted occlusion reasoning is used

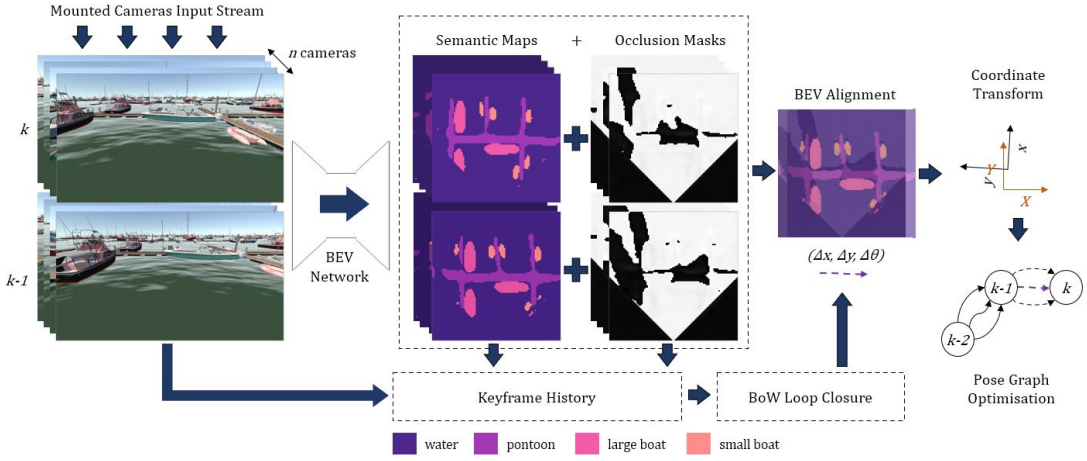

Fig. 2: Overview of BEV-SLAM system. Semantic maps best viewed in colour.

as a mask when calculating this metric, to ensure that only temporally-consistent regions are used for alignment. The goal is therefore to find the optimal alignment $\Delta x^*, \Delta y^*, \Delta\theta^*$ with

$$\underset{\Delta x, \Delta y, \Delta\theta}{\arg\min} \left\| \left[\frac{\boldsymbol{i}_r}{\|\boldsymbol{i}_r\|} - \frac{\boldsymbol{i}_w}{\|\boldsymbol{i}_w\|} \right] \odot \boldsymbol{M}_{occ_r} \odot \boldsymbol{M}_{occ_w} \right\|^2 \tag{5}$$

where $\boldsymbol{i}_r$ and $\boldsymbol{M}_{occ_r}$ represent a zero-mean version of the reference BEV map and the corresponding binary occlusion mask respectively, and $\boldsymbol{i}_w$ and $\boldsymbol{M}_{occ_w}$ are the warped maps, transformed by $\Delta x, \Delta y$ and $\Delta\theta$. $\|\cdot\|$ denotes the L2 norm and $\odot$ element-wise multiplication. Constant motion is used as an initial estimate for the optimisation algorithm.

D. Multiple Cameras

Alignment can be ineffective when there is little geometry visible from a single camera's viewpoint. However, motion cues can also come from additional cameras mounted around the ego vehicle. To incorporate multiple sensors, the optimal alignment (5) must be transformed into an SE2 $(\Delta X, \Delta Y, \Delta\Theta)$ odometry measurement in the ego vehicle coordinate frame. The appropriate transformation matrix can be obtained using the camera extrinsic matrix relative to the ego vehicle, applying a correction to account for the centre of rotation being at the ego vehicle centre:

$$\boldsymbol{M} = \boldsymbol{G} \cdot \left[\begin{array}{c|c} \boldsymbol{R} & \boldsymbol{T} - \boldsymbol{R}^{\boldsymbol{\theta}} \boldsymbol{X} \end{array} \right] \tag{6}$$

where $\boldsymbol{R}^{\boldsymbol{\theta}} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $\boldsymbol{X} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix}$.

$\boldsymbol{G}$ is the camera extrinsic matrix, $\boldsymbol{R}$ and $\boldsymbol{T}$ are the BEV alignment rotation and translation matrices respectively, θ is the alignment angle, and $(x_0,\ y_0)$ is the ego vehicle centre of rotation in the BEV pixel coordinate frame.

This correction gives a transformation matrix $\boldsymbol{M}$ representing $(\Delta X, \Delta Y, \Delta\Theta)$ in the ego vehicle frame, which is then added as an edge in a pose graph.

E. Loop Closure and Multiple Sensors

Potential loop closures are identified using a visual bag of words with SIFT features on the input images, implemented as a tree for efficient storage. When the likelihood exceeds a set threshold, the loop closure is verified using alignment in the BEV space from the algorithm described in III-C. BEV maps are considered to be unrelated if the alignment algorithm does not converge. Otherwise, a new edge is added to the pose graph, and pose graph optimisation is performed.

Using the camera extrinsic matrices and the rotational correction (6), loop closures can be identified across multiple cameras: for example, a landmark can be observed from one camera on one pass, and from the other side on the return journey, and the two keyframes associated correctly.

A significant advantage of employing graph-based SLAM is that additional sensors can be seamlessly integrated into the system before pose graph optimisation. GPS measurements (in the SE2 state space), for example, can be added as initial node estimates. In our implementation, we used g2o [35] for pose graph optimisation.

IV. Experiments and Results

We first evaluate BEV-SLAM in a maritime domain using synthetic data and compare to SOTA. We ablate our system and demonstrate the efficacy of our proposed occlusion reasoning method. We then demonstrate BEV-SLAM running on a real-world docking sequence, and on a publicly-available automotive dataset for further comparison with SOTA.

A. Dataset Generation

A particular challenge for BEV network training is obtaining suitable ground truth data (both semantic maps and occlusion masks). Existing research typically uses additional sensors such as LiDAR to generate BEV ground truth, but this is time-consuming, expensive to collect on a large scale, and cannot always provide strong supervision.

Since equivalent datasets do not exist for maritime, a maritime simulator was developed specifically for BEV-SLAM research and dataset generation, rendered in Unity [36]. A

TABLE I: Mean Absolute and Relative Pose Errors (APE and RPE respectively) of output trajectories for BEV-SLAM with and without multiple cameras, and comparison to GPS

	Trajectory					
	A (165.7 m)		**B (242.6 m)**		**C (322.9 m)**	
	***APE* [m]**	***RPE* [m]**	***APE* [m]**	***RPE* [m]**	***APE* [m]**	***RPE* [m]**
Low-cost GPS	3.598	4.719	3.880	5.272	3.835	4.777
High-cost GPS	1.265	1.807	1.237	1.867	1.227	1.750
Single-camera BEV-SLAM	2.831	0.161	3.764	0.125	4.280	0.720
Multi-camera BEV-SLAM	1.480	**0.040**	1.863	**0.066**	1.954	**0.171**
Multi-camera BEV-SLAM + Low-cost GPS	1.323	0.069	1.375	0.078	1.919	1.998
Multi-camera BEV-SLAM + High-cost GPS	**1.064**	0.093	**1.130**	0.102	**1.223**	0.784

TABLE II: Maritime semantic classes

Name	**Description**
Navigable space	Any area where the ego vehicle could manoeuvre
Large boat	Yachts, cruisers, fishing vessels
Small boat	Rowboats, small sailboats
Other boat	Covered boats, boats on land
Pontoon	Marina elements, walkways and attached ropes
Obstacle	Any other object that intersects the water plane

dataset is provided and made publicly available for the benefit of the research community, encouraging the development of BEV systems in the maritime domain. The dataset contains five 10-minute mounted camera sequences from around an ego vehicle with corresponding BEV semantic maps and occlusion masks in three pseudo-random procedurally-generated environments. Ground truth poses for each camera and the ego vehicle are also available, allowing the reconstruction of ground-truth trajectories. In addition, training sets are available, which consist of 200,000 snapshots (images, semantic maps and occlusion masks) taken from over 100 environments to prevent overfitting to specific sequences or arrangements. The top-down semantic classes provided in the dataset are shown in Table II.

Three validation sequences have been collected from unseen environments, each increasing in length. These sequences are used for evaluation of BEV-SLAM in the maritime domain, with the mean Absolute Pose Error (APE) and mean Relative Pose Error (RPE) [37] of the predicted trajectories used as the primary evaluation metrics. Ground-truth trajectories for the three sequences are shown in Fig. 3.

B. Ablation Study

We test the system by removing elements to investigate the effect of running the system with a single- or multi-camera setup, and with and without the presence of GPS. Pose errors for the three simulation trajectories are shown in Table I, after applying detected loop closures.

Here, multi-camera BEV-SLAM uses four cameras mounted around the ego vehicle in the configuration shown in Fig. 1. Single-camera BEV-SLAM results are the mean errors across all four cameras, tested individually.

To simulate GPS, we use ground-truth poses with added zero-mean Gaussian noise. For low-cost GPS, standard deviation $\sigma = 3$ m. For high-cost GPS, we use standard deviation

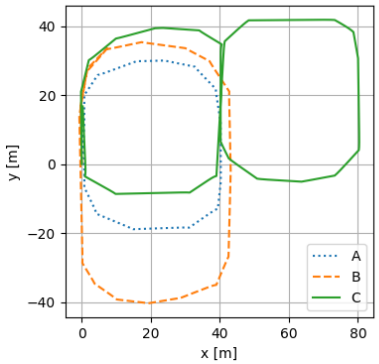

Fig. 3: Simulation trajectories (ground truth)

$\sigma = 1$ m, which represents a strong GPS signal with a high-performance receiver (for example, the GA150 GPS Antenna).

The first observation is that multi-camera BEV-SLAM with this camera configuration significantly outperforms the single-camera equivalent, and both BEV-SLAM systems outperform low-cost GPS.

High-cost GPS is able to outperform BEV-SLAM in terms of absolute error, since there is no drift over time, but its relative error is higher due to the addition of Gaussian noise. A combination of multi-camera BEV-SLAM and high-cost GPS achieves the lowest absolute pose error across all three sequences, and results in significantly reduced relative error compared to GPS alone.

C. Benchmarks

We compare BEV-SLAM to state-of-the-art ORB-SLAM2, ORB-SLAM3 and LDSO. Since these methods use monocular footage, we compare each with single camera BEV-SLAM for four cameras mounted around the ego vehicle over simulated sequence B. We then compare to multi-camera BEV-SLAM, to demonstrate the advantage of the use of multiple cameras.

Initially, ORB-SLAM2 failed on three of the four cameras (key points could not be found for a suitable initialisation). Closer inspection revealed that this is because ORB-SLAM2 chooses features on the moving water surface, which cannot later be used for re-localisation. The one camera sequence for which ORB-SLAM2 was successful was from the starboard camera, though the output trajectory was poor. Empirically, this is an issue on real sequences too.

TABLE III: Mean relative pose error [m] comparison on maritime simulation sequence B

	Camera			
	Port	**Bow**	**Starboard**	**Stern**
ORB-SLAM2	-	-	0.709	-
ORB-SLAM2 (no water)	0.310	0.281	0.235	0.669
ORB-SLAM3	0.398	0.304	0.237	0.410
ORB-SLAM3 (no water)	0.290	0.277	0.186	0.110
LDSO	0.188	0.254	0.163	0.101
Single-camera BEV-SLAM	**0.100**	**0.187**	**0.117**	**0.094**
Multi-camera BEV-SLAM	**0.066**			

TABLE IV: Absolute and relative pose errors [m] using different occlusion reasoning methods

	Trajectory					
	A (165.7 m)		**B (242.6 m)**		**C (322.9 m)**	
	APE	***RPE***	***APE***	***RPE***	***APE***	***RPE***
No reasoning	2.346	0.075	2.699	0.120	2.362	0.279
Implicit	2.218	0.071	2.638	0.111	2.281	0.270
Explicit	**1.480**	**0.040**	**1.863**	**0.066**	**1.954**	**0.171**

To offer a fair comparison, we recreated a simulation sequence, rendering the water surface black to encourage ORB-SLAM2 to select features on the surrounding geometry. This simulates perfect segmentation of the input image and is artificially favourable towards feature-based approaches. ORB-SLAM2 was then able to successfully run on all four cameras: the mean RPEs of the output trajectories are shown in Table III. ORB-SLAM3 was able to run with water rendered, but results are shown for ORB-SLAM3 without water for completeness. A full map comparison cannot be performed, since ORB-SLAM uses sparse features and does not produce a dense map.

We can observe from these results that BEV-SLAM outperforms SOTA in all cases, in addition to being able to produce a dense map. Using multiple cameras reduces the relative pose error of BEV-SLAM even further, because, in areas where one camera may not be able to observe sufficient geometry for alignment, another camera can provide a motion cue.

D. Occlusion

We demonstrate the effectiveness of our occlusion reasoning approach for the BEV-SLAM application by comparing our method to BEV-SLAM using a BEV network trained with no occlusion reasoning, and one trained using implicit reasoning [8]. We judge each method based on the mean APE and RPE of recorded sequences after alignment and pose graph optimisation for three simulation sequences; results are shown in Table IV. The results show a small improvement in trajectory prediction using implicit reasoning, and a significant benefit using our reasoning. This is because the unoccluded regions in the input images produce more temporally consistent BEV predictions. Using only these areas results in more accurate BEV alignment and lower relative pose error, which leads to lower absolute error overall. We therefore can conclude that our occlusion reasoning approach is the most suitable for SLAM applications.

TABLE V: Mean Relative Pose Errors (RPE) [m] of output trajectories for Nuscenes sequences at different framerates for different manoeuvres

	2 Hz		12 Hz	
	Straight	Turn	Straight	Turn
ORB-SLAM3	0.230	-	0.166	-
LDSO	0.108	0.131	0.076	0.096
BEV-SLAM	**0.055**	**0.064**	**0.054**	**0.063**

E. Automotive Evaluation

Although our main focus is to demonstrate feasibility in the maritime domain, we train BEV-SLAM on Nuscenes data and run on automotive Nuscenes sequences to easily compare with state-of-the-art on a public dataset. We show results using both keyframes (2 Hz) and all available frames (12 Hz) to demonstrate that BEV-SLAM is better able to handle low framerates than the competition. It was observed that existing approaches often lose tracking when performing turning manoeuvres, so we show results separately for straight segments and turns. Mean Relative Pose Errors (RPEs) of BEV-SLAM and its competitors are shown in Table V.

F. Qualitative Results

Fig. 4 shows the BEV-SLAM world map prediction (right) and corresponding ground truth semantic map (left) for sequence B (an unseen environment).

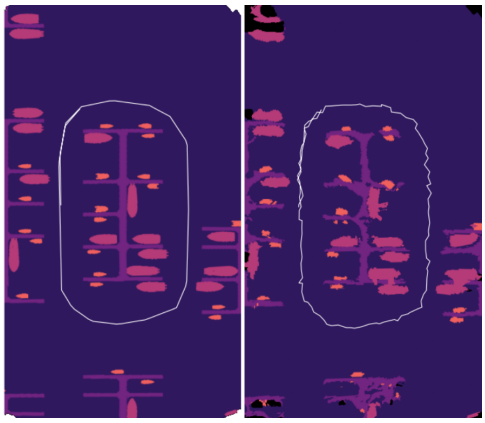

Fig. 4: Ground truth semantic map for sequence B (left) and BEV-SLAM prediction (right) with trajectory shown.

Due to the lack of ground-truth maritime data with suitable trajectories, we are unable to quantitatively evaluate BEV-SLAM in the real-world maritime domain. However, we are able to provide a qualitative example for a docking sequence. The ego boat was configured with three mounted cameras (port, starboard and stern), and the SLAM system was trained using ground truth data captured from a drone. The resulting generated semantic map is shown in Fig. 5 with the predicted trajectory shown.

V. Conclusions

We have introduced a novel approach to graph-based SLAM using semantically-segmented Bird's Eye View (BEV) predictions to create a dense world map from monocular cameras

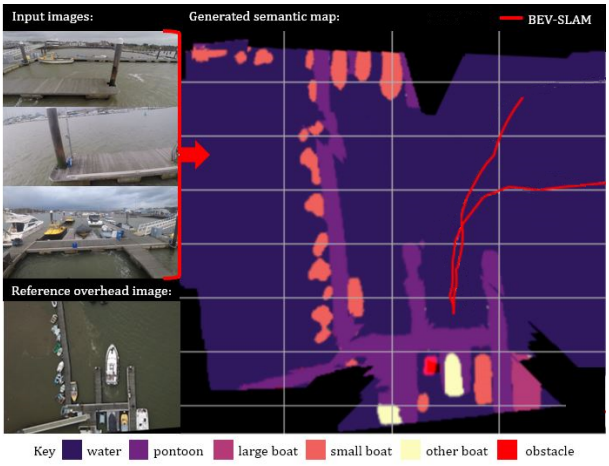

Fig. 5: Generated semantic map for real-world maritime docking sequence. Top-left: Example input images. Right: output semantic map. Bottom-left: reference drone image.

mounted around an ego vehicle, and demonstrated its feasibility in the maritime domain. Unlike other SLAM systems (which tend to use sparse features), it can produce dense, correctly-scaled semantic world maps ideal for navigation tasks or as a visual aide for challenging manoeuvres. Future work could focus on better integration of other sensor types into the SLAM system, accounting for their uncertainties when optimising the pose graph, leveraging additional information they can provide, and dealing with moving classes.

REFERENCES

[1] S. Sengupta, P. Sturgess, L. Ladický and P. H. S. Torr, "Automatic dense visual semantic mapping from street-level imagery," International Conference on Intelligent Robots and Systems, 2012, pp. 857-862

[2] S. A. Abbas and A. Zisserman, "A Geometric Approach to Obtain a Bird's Eye View From an Image," 2019 IEEE/CVF International Conference on Computer Vision Workshop, 2019, pp. 4095-4104

[3] K. Regmi and A. Borji, "Cross-View Image Synthesis Using Conditional GANs," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3501-3510

[4] Xinge Zhu et al. "Generative adversarial frontal view to bird view synthesis." 2018 International Conference on 3D Vision, 2018

[5] Tom Bruls et al. "The right (angled) perspective: improving the understanding of road scenes using boosted inverse perspective mapping," 2019 IEEE Intelligent Vehicles Symposium, 2019

[6] Samuel Schulter et al. "Learning to look around objects for top-view representations of outdoor scenes," CoRR abs/1803.10870, 2018

[7] Thomas Roddick, Alex Kendall, and Roberto Cipolla. "Orthographic feature transform for monocular 3D object detection". CoRR abs/1811.08188, 2018

[8] T. Roddick and R. Cipolla, "Predicting Semantic Map Representations From Images Using Pyramid Occupancy Networks," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[9] Chenyang Lu, Marinus Jacobus Gerardus van de Molengraft, and Gijs Dubbelman. "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks," In: IEEE Robotics and Automation Letters 4.2, 2019, pp. 445–452

[10] Jonah Philion and Sanja Fidler. "Lift, splat, shoot: encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," CoRR abs/2008.05711, 2020

[11] Bowen Pan et al. "Cross-view semantic segmentation for sensing surroundings," IEEE Robotics and Automation Letters 5(3):4867–4873, 2020

[12] Menghua Zhai et al. "Predicting ground-level scene layout from aerial imagery," CoRR abs/1612.02709, 2016

[13] Kaustubh Mani et al. "MonoLayout: Amodal scene layout from a single image," CoRR abs/2002.08394, 2020

[14] Ruiqi Guo and Derek Hoiem. "Beyond the line of sight: labeling the underlying surfaces," English (US). Computer Vision, ECCV 2012 - 12th European Conference on Computer Vision, Proceedings. 012, pp. 761–774.

[15] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. "Scene parsing with object instances and occlusion ordering". In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014, pp. 3748–3755.

[16] A. Saha, O. Mendez, C. Russell and R. Bowden, "Enabling spatio-temporal aggregation in Birds-Eye-View Vehicle Estimation," IEEE International Conference on Robotics and Automation (ICRA), 2021

[17] Anthony Hu et al. "FIERY: future instance prediction in bird's-eye view from surround monocular cameras," IEEE International Conference on Computer Vision, 2021

[18] T. Bailey and H. Durrant-Whyte. "Simultaneous localization and mapping: part II," IEEE Robotics Automation Magazine 13.3, 2006.

[19] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping: part I". IEEE Robotics Automation Magazine 13.2, 2006.

[20] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems, 1991, pp. 1442-1447.

[21] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," Proc. IEEE Int. Conf. Computer Vision (ICCV), 2003, pp. 1403–1410

[22] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1052–1067, 2007.

[23] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," Journal of field robotics, vol. 27, no. 5, pp. 609–631, 2010.

[24] R. Mur-Artal, J. M. M. Montiel, and J. D. Tard os. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics, 31(5):1147–1163, 2015.

[25] R. Mur-Artal and J. D. Tard os. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics, 33(5):1255–1262, 2017.

[26] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Moniel, and J. D. Tard os. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. IEEE Transactions on Robotics, 37(6):1874-1890, 2021.

[27] N. Ragot, R. Khemmar, A. Pokala, R. Rossi and J. Ertaud, "Benchmark of visual slam algorithms: ORB-SLAM2 vs RTAB-Map*," 2019 Eighth International Conference on Emerging Security Technologies (EST), 2019, pp. 1-6.

[28] J. Engel, T. Schops, D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM" European Conference on Computer Vision, 2014

[29] J. Zubizarreta, I. Aguinaga and J. M. M. Montiel, "Direct Sparse Mapping," IEEE Transactions on Robotics, 36(4):1363-1370, 2020.

[30] J. Engel, V. Koltun and D. Cremers, "Direct Sparse Odometry," IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(3):611-625, 2018.

[31] X. Gao, R. Wang, N. Demmel and D. Cremers, "LDSO: Direct Sparse Odometry with Loop Closure," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2018, 2198-2204, 2018.

[32] M. Colosi et al., "Plug-and-Play SLAM: A Unified SLAM Architecture for Modularity and Ease of Use," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 5051-5057, 2020.

[33] T. Schneider et al., "Maplab: An Open Framework for Research in Visual-Inertial Mapping and Localization," IEEE Robotics and Automation Letters, 3(3):1418-1425, 2018.

[34] Georgios Evangelidis, Emmanouil Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. IEEE Transactions on Pattern Analysis and Machine Intelligence, Institute of Electrical and Electronics Engineers, 2008, 30 (10), pp.1858-1865.

[35] R. Kümmerle et ak., "G2o: A general framework for graph optimization," IEEE International Conference on Robotics and Automation, pp. 3607-3613, 2011.

[36] Unity Technologies, "Unity Real-Time Development Platform". https://unity.com

[37] Michael Grupp. evo: Python package for the evaluation of odometry and SLAM. https://github.com/MichaelGrupp/evo. 2017.