

Geometric and Learning-Based Perception for Mobile Vision by Integrating Memory and Temporal Fusion

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Yitong Quan
aus Guangdong/China

Tübingen

2026

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der
Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:

07.08.2026

Dekan:

Prof. Dr. Thilo Stehle

1. Berichterstatter:

Prof. Dr. rer. nat. Andreas Zell

2. Berichterstatter:

Prof. Dr.-Ing. Hendrik Lensch

Abstract

Robust perception is a critical prerequisite of autonomous robotics, enabling platforms such as unmanned ground vehicles (UGVs), aerial vehicles (UAVs), and surface vehicles (USVs) to interpret, navigate, and interact with complex environments. Traditional perception pipelines often depend on multi-sensor setups, including LiDAR, radar, or multi-camera rigs, which, while effective, introduce substantial costs, complexity, and calibration requirements. In contrast, single-sensor solutions based on monocular or omnidirectional cameras offer cost-efficiency, compactness, and ease of deployment.

This dissertation investigates how geometric reasoning and learning-based methods can be combined to achieve robust object detection, localization, and distance estimation, focusing on three complementary research directions: spatial memory, learning-based methods, and lightweight temporal fusion.

First, we introduce a memory-map framework for object detection and localization, using a single 360° camera, demonstrating that spatial confidence accumulation over time improves robustness to partial occlusions and inconsistent detections. Second, we extend this concept to UAV platforms, integrating onboard metadata such as GPS, IMU, and altitude into geo-referenced confidence maps. This metadata-driven approach enhances temporal consistency in aerial video object detection and provides interpretable localization across dynamic flight paths. Third, we propose learning-based formulations for object distance estimation. On USVs, we show that extending an object detector with a regression branch enables approximate distance prediction directly from monocular imagery, offering a cost-efficient alternative to conventional range sensors. Then we broaden this concept to omnidirectional 360° images, where the method generalizes across diverse datasets and challenging conditions, outperforming geometric and alternative learning baselines without explicit calibration. Finally, we present a lightweight multi-frame integration strategy for YOLO-based video object detection. By stacking consecutive frames and supervising only the final frame, the approach leverages temporal context with minimal computational overhead, significantly improving robustness against motion blur, occlusions, and appearance changes, while preserving near single-frame throughput.

Overall, this dissertation demonstrates that single-sensor vision systems, when enhanced with spatial memory, metadata-driven reasoning, learning-based distance regression, and lightweight temporal fusion, can deliver robust, efficient, and generalizable perception across ground, aerial, and maritime platforms.

Kurzfassung

Robuste Wahrnehmung ist eine entscheidende Voraussetzung für autonome Robotik und ermöglicht es Plattformen wie unbemannten Bodenfahrzeugen (UGVs), Luftfahrzeugen (UAVs) und Oberflächenfahrzeugen (USVs), komplexe Umgebungen zu interpretieren, zu navigieren und mit ihnen zu interagieren. Herkömmliche Wahrnehmungspipelines basieren häufig auf Multisensor-Setups, darunter LiDAR, Radar oder Mehrkamera-Rigs, die zwar effektiv sind, aber erhebliche Kosten, Komplexität und Kalibrierungsanforderungen mit sich bringen. Im Gegensatz dazu bieten Einzelsensorlösungen auf Basis monokularer oder omnidirektionaler Kameras Kosteneffizienz, Kompaktheit und einfache Implementierung.

Diese Dissertation untersucht, wie geometrisches Denken und lernbasierte Methoden kombiniert werden können, um eine robuste Objekterkennung, -lokalisierung und -entfernungsschätzung zu erreichen. Der Fokus liegt dabei auf drei komplementären Forschungsrichtungen: räumliches Gedächtnis, lernbasierte Methoden und leichtgewichtige zeitliche Fusion.

Zunächst stellen wir ein Memory-Map-Framework für die Objekterkennung und -lokalisierung mit einer einzelnen 360°-Kamera vor und zeigen, dass die Akkumulation räumlicher Konfidenz im Laufe der Zeit die Robustheit gegenüber partiellen Verdeckungen und inkonsistenten Erkennungen verbessert. Zweitens erweitern wir dieses Konzept auf UAV-Plattformen und integrieren Onboard-Metadaten wie GPS, IMU und Flughöhe in georeferenzierte Konfidenzkarten. Dieser metadatenbasierte Ansatz verbessert die zeitliche Konsistenz bei der Objekterkennung per Luftbildvideo und ermöglicht eine interpretierbare Lokalisierung über dynamische Flugrouten hinweg. Drittens schlagen wir lernbasierte Formulierungen zur Objektentfernungsschätzung vor. An USVs zeigen wir, dass die Erweiterung eines Objektdetektors um einen Regressionszweig eine ungefähre Entfernungsvorhersage direkt aus monokularen Bildern ermöglicht und damit eine kostengünstige Alternative zu herkömmlichen Entfernungssensoren darstellt. Anschließend erweitern wir dieses Konzept auf omnidirektionale 360°-Bilder, wobei die Methode über verschiedene Datensätze und anspruchsvolle Bedingungen hinweg generalisierbar ist und geometrische und alternative lernbasierte Baselines ohne explizite Kalibrierung übertrifft. Abschließend präsentieren wir eine leichtgewichtige Multi-Frame-Integrationsstrategie für die YOLO-basierte Videoobjekterkennung. Durch das Stapeln aufeinanderfolgender Frames und die Überwachung nur des letzten Frames nutzt der Ansatz den zeitlichen Kontext mit minimalem Rechenaufwand. Dies verbessert die Robustheit gegenüber Bewegungsunschärfe, Okklusionen und Erscheinungsänderungen deutlich, während der Durchsatz nahezu dem einer Einzelbildverarbeitung entspricht.

Insgesamt zeigt diese Dissertation, dass Einzelsensor-Bildverarbeitungssysteme, wenn sie mit räumlichem Gedächtnis, metadatengesteuerter Argumentation, lernbasierter Distanzregression und leichter zeitlicher Fusion erweitert werden, eine robuste, effiziente und verallgemeinerbare Wahrnehmung über Boden-, Luft- und Seeplattformen hinweg liefern können.

Acknowledgments

I would like to express my sincere gratitude to Professor Andreas Zell, for giving me the opportunity to conduct my doctoral research in the Chair of Cognitive Systems, and for the guidance, trust, and support he has provided me over the years. I am also grateful to Professor Hendrik Lensch for serving as my second supervisor.

My sincere thanks go to my co-authors, collaborators, and colleagues for the discussions, feedback, and shared work that contributed to this dissertation. I also thank the administrative and technical staff for their help during my PhD.

Finally, I would like to thank my friends and family for their patience, encouragement, and support. Their trust in me has always been a source of strength that propelled me forward on this journey.

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Problem Setting and Scope	1
1.3	Technical Challenges	2
1.4	Research Questions	2
1.5	Approach Overview	3
1.6	Contributions and Outline	3
1.7	Dissertation Structure	6
2	Related Work	7
2.1	Monocular and Omnidirectional Vision	7
2.2	Object Detection Methods	8
2.3	Learning-Based Depth/Distance Estimation for Vision	8
2.4	Spatial Memory and Scene Representation	9
2.5	UAV Perception and Metadata-Aware Vision	9
2.6	Perception on Unmanned Surface Vehicles	10
2.7	Identified Research Gaps	10
3	Robust 360° Camera Object Detection and Localization with Memory Maps	13
3.1	Introduction	13
3.2	Related Work	15
3.3	Methodology	16
3.3.1	Dataset Generation	16
3.3.2	Lens Model for Passive Geo-localization	17
3.3.3	Confidence Calibration via Memory Mapping	20
3.4	Experimental Results	21
3.4.1	Hardware and Tools	22
3.4.2	Experimental Setup and Results	23
3.5	Conclusion and Outlook	25
4	Confidence Boosting in 3D for Object Detection on UAVs	27
4.1	Introduction	27
4.2	Related Work	29
4.3	Deriving Formulas for 3D Geometry	29
4.3.1	Relative Coordinates	30

4.3.2	Absolute Coordinates	32
4.4	Method: Temporal Memory	32
4.4.1	Map Representation	33
4.4.2	Video Object Detection	33
4.4.3	Extension to Object Tracking and Reidentification	37
4.4.4	Extension to Video Anomaly Detection	38
4.5	Results and Analysis	38
4.5.1	Video Object Detection	39
4.5.2	Object Tracking	40
4.5.3	Video Anomaly Detection	41
4.5.4	Cooperative Detection via Multiple UAVs	41
4.5.5	Accuracy and Calibration	45
4.6	Conclusion and Outlook	45
5	Approximate Supervised Object Distance Estimation on Unmanned Surface Vehicles	47
5.1	Introduction	47
5.2	Related Work	49
5.3	Methodology	51
5.3.1	Model Architecture	51
5.3.2	Distance Normalization Strategies	52
5.3.3	Training and Inference	53
5.3.4	Model Training	53
5.3.5	Dataset Collection and Annotation	53
5.3.6	Human Labeling for More Data	54
5.4	Evaluation Metrics	56
5.4.1	Weighted Distance Error Metric	56
5.5	Experiments	57
5.5.1	Object Detection Performance	57
5.5.2	Distance Estimation Accuracy	57
5.5.3	Comparison with Other Methods	58
5.5.4	Robustness Towards Camera Extrinsic	59
5.5.5	Loss Balancing and Ablation Studies	59
5.5.6	Smoothing the Distance Estimates	60
5.5.7	Evaluation of Human-Labeled Distances	61
5.6	Conclusion and Outlook	61
6	Learning-Based Distance Estimation for 360° Single-Sensor Setups	63
6.1	Introduction	63
6.2	Related Work	65
6.3	Methodology	67
6.3.1	Datasets	67

6.3.2	Model Architecture	68
6.3.3	Distance Normalization Strategies	69
6.3.4	Model Training	70
6.4	Evaluation Metrics	70
6.4.1	Absolute Distance Error	71
6.4.2	Confidence-Weighted Distance Error	71
6.5	Experiments and Results	71
6.5.1	Evaluation on LOAF Dataset	74
6.5.2	Evaluation on ULM360 and BOAT360 Datasets	77
6.5.3	Summary of Key Findings	79
6.6	Conclusion and Outlook	79
7	Lightweight Multi-Frame Integration for Robust YOLO Object Detection in Videos	81
7.1	Introduction	82
7.2	Related Work	84
7.2.1	Single-Frame Object Detection	84
7.2.2	Video Object Detection	84
7.2.3	Multi-Frame Input with Sparse Supervision	84
7.2.4	Model Interpretability in Object Detection	85
7.3	Methodology	85
7.3.1	Problem Formulation	85
7.3.2	Multi-Frame Input Representation	85
7.3.3	Network Architecture Adaptation	85
7.3.4	Training Strategy	86
7.3.5	Advantages of the Proposed Approach	87
7.4	Experiments	88
7.4.1	Datasets	88
7.4.2	Evaluation Metrics	89
7.4.3	Results	89
7.5	Conclusion	93
8	Conclusions	97
8.1	Summary	97
8.2	Contributions by Chapter	97
8.3	Synthesis of Findings	98
8.4	Limitations	99
8.5	Outlook and Future Work	99
8.6	Concluding Remarks	100
	Abbreviations	101

Bibliography	103
---------------------	------------

Chapter 1

Introduction

1.1 Background and Motivation

Perception systems play a fundamental role in enabling autonomous platforms, including unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs) and unmanned surface vehicles (USVs), to navigate complex environments, interact safely with their surroundings, and perform tasks autonomously. Robust and accurate perception is essential for overcoming the numerous real-world challenges posed by dynamic scenarios, diverse environmental conditions, occlusions, motion blur, and unpredictable object appearances. Conventional approaches often rely heavily on expensive sensor suites such as LiDAR or multi-camera systems, which, despite their advantages, incur significant costs, complexity, and computational overhead, limiting their practicality for widespread deployment in resource-constrained applications.

This thesis investigates whether single-camera-only systems can be pushed further by combining geometric reasoning with learning and explicitly leveraging temporal information through spatial memory and lightweight multi-frame fusion, while respecting embedded compute budgets.

1.2 Problem Setting and Scope

We study single-sensor mobile vision with rectilinear or 360° omnidirectional cameras, focusing on object-level tasks: detection, ground-plane localization, and distance estimation. Onboard perception on UGVs, UAVs, and USVs typically runs on battery-powered embedded GPUs. Closed-loop autonomy requires real-time outputs for control, while size, weight, power, and thermal constraints restrict compute capacity, leading to tight latency and power budgets. We avoid heavy dense depth pipelines and large recurrent models. Instead, we explore geometry-aware spatial memory, metadata-aware geo-referencing, learning-based distance estimation for monocular and panoramic imagery, and lightweight multi-frame integration that preserves the simplicity of one-shot detectors.

1.3 Technical Challenges

Single-camera perception faces several difficulties in real-world robotics:

- **Depth ambiguity.** Monocular imagery lacks direct metric scale; distance estimation is ill-posed without additional priors or supervision.
- **Projection effects.** Wide field of view and 360° equirectangular or fisheye imagery introduce severe distortions and spatially varying resolution, which standard CNNs do not model by default.
- **Temporal brittleness.** Single-image detectors degrade under occlusion, motion blur, rapid viewpoint change, and appearance shifts.
- **Embedded constraints.** Latency and memory limits restrict the use of heavy temporal modules or dense geometric reconstruction.
- **Metadata variability.** UAV GPS, IMU, and altitude information are available but noisy and asynchronous, complicating online geo-referencing.

1.4 Research Questions

This work is organized around four questions:

- **Spatial consistency.** How can evidence be accumulated over time in a geometry-consistent representation to stabilize single-camera detection and improve localization accuracy without large recurrent networks?
- **Metadata-aware mapping.** How can onboard metadata (GPS and IMU) be fused online to maintain an up-to-date georeferenced memory that supports temporally consistent semantics?
- **Object-level distance.** Can detectors be extended with lightweight heads to infer approximate metric object distance in rectilinear and 360° imagery without dense depth estimation?
- **Lightweight temporal fusion.** How much temporal context can be exploited through multi-frame inputs while preserving the speed and simplicity of one-shot detectors?

1.5 Approach Overview

To address the four research questions outlined above, this dissertation proposes perception frameworks that combine geometric modeling, temporal information, and learning-based inference within the constraints of embedded platforms. The approach is structured around four complementary components, each targeting a specific limitation of single-camera perception:

- **Spatial memory.** We maintain a persistent confidence map in a consistent frame such as the ground plane or a bird’s eye view. The map aggregates detections over time and suppresses false positives without relying on optical flow or recurrent models.
- **Metadata-aware geo-referencing.** For UAV applications, we project detections into a geo-referenced ground frame using GPS, IMU, and altitude when available, which yields temporally consistent semantics and interpretable trajectories.
- **Object-level distance estimation.** We attach a lightweight regression head to a detector backbone to predict object distance directly from shared features. The formulation covers rectilinear and panoramic imagery and reduces reliance on dense depth.
- **Lightweight multi-frame temporal integration.** We inject short-horizon temporal context by stacking a small number of frames at the input, with supervision applied only to the terminal frame, which improves robustness under motion blur and occlusion at minimal cost.

1.6 Contributions and Outline

In short, the presented research addresses fundamental limitations of existing monocular perception frameworks by incorporating two complementary dimensions:

- **Spatial memory**, to enable persistent and location-aware detection confidence.
- **Temporal fusion**, to leverage contextual information across multiple frames or video sequences.

The following chapters present five core contributions, each based on a peer-reviewed publication and presented as a standalone chapter. While each chapter addresses a distinct technical challenge, together they follow a coherent trajectory toward robust, efficient visual perception for mobile platforms using single-sensor setups. This section outlines the conceptual and methodological transitions between chapters, providing context for how each contribution builds on the preceding work and addresses its limitations.

Chapter 3 *Robust 360° Camera Object Detection and Localization with Memory Maps* presents a spatial memory framework for object detection and localization using a single 360° fisheye camera. By projecting detections into a continuously updated spatial map using a geometric fisheye projection model, the system aggregates observations over time and enhances detector confidence in regions with repeated detections and suppresses false positives in areas with inconsistent or absent observations. This method, validated on a real-world dataset collected from a mobile platform, serves as a cost-effective alternative to LiDAR or multi-camera systems while improving detection robustness through passive temporal fusion.

Chapter 4 *Confidence Boosting in 3D for Object Detection on UAVs* extends this concept to UAV-based perception, where camera motion, altitude variation, and dynamic viewpoints pose additional challenges. By leveraging available metadata such as GPS, IMU, and altitude measurements, we construct a geo-referenced memory map on the ground plane that accumulates object detections over time. This approach improves robustness in UAV object detection by integrating geometric reasoning with temporal continuity. Experimental results demonstrate substantial reductions in detection uncertainty and improved interpretability, particularly under challenging aerial conditions characterized by sparse features and large viewpoint variations.

Chapter 3 and 4 achieve robustness in perception by accumulating detection confidence in memory maps, relying on external localization and temporal fusion. However, when the task demands only object distance, rather than absolute position or global consistency, maintaining such spatial maps becomes unnecessary and computationally inefficient. Furthermore, reliance on pose metadata limits applicability to platforms with full localization capabilities.

Chapter 5 *Approximate Supervised Object Distance Estimation on Unmanned Surface Vehicles* addresses this limitation by proposing a learning-based approach that directly estimates the distance of each detected object from a single perspective (rectilinear) image. It augments a YOLO-style object detector with a lightweight regression head to infer real-world distance, trained end-to-end without requiring camera calibration. This shift reflects a conceptual pivot: from memory-based spatial reasoning aided by metadata, to compact, image-conditioned 3D distance inference. The method is well-suited for platforms where sensor payload and computational budgets are constrained, and where metric distance, rather than global localization, is the primary requirement.

Chapter 5 demonstrates that object-level distance estimation is feasible from monocular rectilinear images using a single-sensor setup and end-to-end learning. However, the method is evaluated on a limited field of view and a relatively homogeneous environment, dominated by flat water surfaces. As such, its robustness in environments with high geometric variability, extreme distortion, and wide fields of view remains uncertain.

Chapter 6 *Learning-Based Distance Estimation for 360° Single-Sensor Setups* tackles this challenge by generalizing the distance estimation framework to 360° omnidirectional images captured in diverse environments, including maritime and open-field scenarios. The model is adapted to fish-eye imagery as well as its equirectangular projections

and benchmarked on three datasets. This chapter extends monocular distance regression to handle wider fields of view, extreme geometric distortion, and scene diversity. It reinforces the broader research goal of developing scalable perception systems that generalize across domains without architectural complexity.

While the previous chapters focus on spatial memory and geometric estimation for improving object detection robustness, they rely primarily on per-frame reasoning or accumulated representations. While the previous chapters incorporate elements of temporal reasoning, particularly through memory accumulation about detected objects, they do not explicitly model the fine-grained temporal dynamics of visual input streams, such as motion blur, object transitions, and rapid appearance changes. These factors often degrade performance in real-world deployments when not directly addressed at the input or feature level.

Chapter 7 *Lightweight Multi-Frame Integration for Robust YOLO Object Detection in Videos* addresses this gap by proposing a lightweight multi-frame integration method for video object detection. Rather than using explicit memory maps or pose-aware fusion, it stacks consecutive input frames and applies supervision only on the final frame of each sequence. This sparse supervision strategy reduces annotation effort and avoids the complexity of recurrent or optical flow-based modules. The method complements the spatial and geometric advances of earlier chapters by introducing robustness through temporal context, without compromising speed or architectural simplicity.

These core contributions are based on the following peer-reviewed publications, which form the backbone of this dissertation:

1. Yitong Quan, Benjamin Kiefer, Martin Meßmer, Charan Ram Akupati, Rainer Graser, Andreas Zell. “Robust Single-Cam Surround View Object Detection and Localization Using Memory Maps.” In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 105-118. Cham: Springer Nature Switzerland, 2024.
2. Benjamin Kiefer, Yitong Quan, Andreas Zell. “Memory Maps for Video Object Detection and Tracking on UAVs.” In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3040-3047. IEEE, 2023.
3. Benjamin Kiefer, Yitong Quan, Andreas Zell. “Approximate Supervised Object Distance Estimation on Unmanned Surface Vehicles.” In *International Conference on Advanced Robotics (ICAR)*, pp. 585–590, IEEE, 2025.
4. Yitong Quan, Benjamin Kiefer, Martin Meßmer, Andreas Zell. “Learning-Based Distance Estimation for 360° Single-Sensor Setups.” In *European Conference on Mobile Robots (ECMR)*, pp. 1-8. IEEE, 2025.
5. Yitong Quan, Benjamin Kiefer, Martin Meßmer, Andreas Zell. “Lightweight Multi-Frame Integration for Robust YOLO Object Detection in Videos.” In *European Conference on Mobile Robots (ECMR)*, pp. 1-7. IEEE, 2025.

In addition to these core contributions, the following peer-reviewed works were published during the doctoral studies but are not directly related to the contributions presented in this dissertation and are therefore not discussed here:

1. Benjamin Kiefer, . . . , Yitong Quan, et al. “1st Workshop on Maritime Computer Vision (MaCVi) 2023: *Challenge Results*.” In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 265–302. IEEE, 2023.
2. Benjamin Kiefer, . . . , Yitong Quan, et al. “3rd Workshop on Maritime Computer Vision (MaCVi) 2025: *Challenge Results*.” In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 1457–1484. IEEE, 2025.

1.7 Dissertation Structure

The overarching contribution of this dissertation lies in advancing single-sensor perception on mobile platforms through a series of complementary methods that incorporate geometric reasoning, temporal fusion, and learning-based techniques. Together, these contributions improve robustness and practicality under real-world constraints without requiring expensive multi-sensor setups. Through extensive empirical validation across diverse datasets and operational scenarios, this work demonstrates that integrating spatial memory and temporal fusion leads to reliable and efficient object detection and localization, especially under limited computational budgets.

The remainder of this dissertation is structured as follows: Chapter 2 provides a brief review of related work and identifies critical research gaps addressed by our methodologies. Chapters 3 through 7 detail the individual contributions in turn. The final chapter synthesizes key findings and discusses future research directions.

Chapter 2

Related Work

This chapter presents a brief review of relevant literature that underpins the technical development of this dissertation. It synthesizes prior work across key domains including monocular and omnidirectional perception, visual object detection, spatiotemporal integration, spatial memory representation, and metadata-informed UAV perception. It identifies unresolved challenges in the field and motivates the integrated strategies advanced throughout this dissertation. A more comprehensive discussion of related work is provided in the subsequent chapters, where each contribution is examined in depth within the context of its specific research domain.

2.1 Monocular and Omnidirectional Vision

Monocular vision is attractive for mobile robotics due to its low cost, compactness, and ease of deployment. Classical pipelines infer scene geometry through multi-view constraints and structure-from-motion, but the underlying assumptions of rich texture, static scenes, and accurate pose estimation are often violated in real-world settings. Nevertheless, camera-only monocular pipelines have enabled onboard state estimation and autonomous flight on micro aerial vehicles (Yang *et al.*, 2013, 2014).

Omnidirectional cameras extend the field of view dramatically, trading simple coverage for stronger geometric distortions that require dedicated camera models. Their wide field of view has been exploited for camera-only robot localization using panoramic imagery (Tamimi *et al.*, 2006). Canonical formulations include the unifying theory of central catadioptric systems (Geyer and Daniilidis, 2001) and generic fisheye parameterizations (Kannala and Brandt, 2006b; Mei and Rives, 2007), which remain standard for calibration and metric reasoning. Related work has also addressed feature processing and navigation directly in omnidirectional imagery, including local feature matching improvements for omnidirectional camera systems (Resch *et al.*, 2014) and efficient visual homing based on panoramic views (Stürzl and Mallot, 2006). Such models enable single-sensor panoramic perception, simplifying mechanical design and synchronization compared with multi-camera rigs. Nevertheless, purely geometric reconstruction pipelines struggle under dynamic motion, exposure variation, and rolling-shutter artifacts, motivating learning-based estimation strategies discussed in the next section.

2.2 Object Detection Methods

Modern detectors fall broadly into two families. Two-stage approaches such as Faster R-CNN (Ren *et al.*, 2015a) decouple region proposal from classification/regression and achieve high accuracy, while one-stage methods (YOLO/RetinaNet) directly regress dense predictions for efficiency (Redmon *et al.*, 2016a; Redmon and Farhadi, 2018; Lin *et al.*, 2017). Recent YOLO variants improve speed/accuracy trade-offs and deployment practicality on embedded platforms; YOLOv7 is a representative strong baseline (Wang *et al.*, 2023). Transformer-based detectors (DETR (Carion *et al.*, 2020) and Deformable DETR (Zhu *et al.*, 2021)) replace hand-crafted assignment with set prediction and sparse multi-scale attention, offering an alternative paradigm with robust optimization improvements. A related multi-view extension is SpatialDETR, which performs object-centric 3D detection with geometry-aware cross-sensor attention (Doll *et al.*, 2022).

Single-frame detectors, however, are brittle under motion blur, occlusion, and abrupt appearance changes. Video object detection (VID) addresses this by exploiting temporal redundancy: feature warping and aggregation (Zhu *et al.*, 2017a,c), sequence-level semantic aggregation (SELSA (Wu *et al.*, 2019a)), and memory-enhanced global reasoning (MEGA (Chen *et al.*, 2020)). While effective, these frameworks often increase latency and are memory-challenging for real-time, power-constrained robots. Beyond supervised VID, object-centric video representation learning has also shown that temporal feature similarities can serve as a motion bias for discovering coherent object structure across frames (Zadaianchuk *et al.*, 2023). This motivates lightweight temporal integration strategies that preserve single-shot simplicity while capitalizing on multi-frame context.

2.3 Learning-Based Depth/Distance Estimation for Vision

Learning-based monocular depth estimation has advanced rapidly since the seminal multi-scale network of Eigen *et al.* (2014). Self-supervised methods leveraging photometric consistency (Godard *et al.*, 2017, 2019) eliminated the need for dense ground truth and enabled training directly from video. Related reconstruction work has explored cross-view consistency for multi-view depth estimation (Wei *et al.*, 2014) and self-supervised adaptation of learned stereo models to new domains (Mallick *et al.*, 2020), both of which are relevant to the generalization challenges faced by camera-based distance estimation.

For omnidirectional imagery, architectures often accommodate spherical geometry and the severe distortions of equirectangular projection. Dedicated models such as OmniDepth (Zioulis *et al.*, 2018), BiFuse (Wang *et al.*, 2020), and UniFuse (Jin *et al.*, 2020) integrate panoramic encodings and geometry-aware fusion to reduce projection artifacts. Rotationally equivariant representations (Coors *et al.*, 2018; Esteves *et al.*, 2018) further enhance consistency across viewpoints.

Despite their accuracy, these dense predictors are typically large and computationally intensive, which limits their suitability for embedded robotic platforms and makes them sensitive to domain shift and sparse panoramic supervision. These constraints motivate the development of lightweight, object-centric distance estimation tightly coupled with detection backbones (see Sec. 2.2), enabling efficient, geometry-aware reasoning within unified perception frameworks.

2.4 Spatial Memory and Scene Representation

Persistent representations allow agents to reason beyond a single frame. Classical occupancy grids (Elfes, 1989; Thrun *et al.*, 2005) and SLAM-based maps form the backbone of geometric memory, whereas modern perception systems increasingly integrate semantics within mapping, e.g., SemanticFusion (McCormac *et al.*, 2017). Learning-based memory modules support long-horizon reasoning (Neural Map (Parisotto and Salakhutdinov, 2018)) and temporal association via heatmap-style cues (CenterTrack (Zhou *et al.*, 2020)). For camera-only platforms, BEV lifting from images (Phillion and Fidler, 2020) offers a geometry-consistent canvas to fuse detections/spatial evidence over time with minimal instrumentation. Complementary cognitive and biomimetic perspectives model navigation in terms of view-based place representations, graph-like spatial memory, and metric embeddings derived from vision and odometry (Mallot and Basten, 2009; Hübner and Mallot, 2007; Gillner *et al.*, 2008).

Despite these advances, many memory-enhanced systems assume multi-sensor setups or recurrent networks with sizable compute footprints. A practical gap remains for single-camera, geometry-aware confidence accumulation that *robustly* stabilizes detections at low cost—precisely the niche addressed by the memory-map formulations developed in this dissertation.

2.5 UAV Perception and Metadata-Aware Vision

Unmanned aerial vehicles (UAVs) face perception challenges arising from large and rapid viewpoint changes, strong parallax, variable scene scales, and often limited texture. Community benchmarks (VisDrone (Zhu *et al.*, 2018), UAVDT (Du *et al.*, 2018)) make these difficulties explicit for detection and tracking from aerial viewpoints. On the state-estimation side, fusing inertial sensing resolves scale and improves robustness (VINS-Mono (Qin *et al.*, 2018), ORB-SLAM3 (Campos *et al.*, 2021)), yet such pipelines are primarily optimized for pose rather than semantics. While some mapping frameworks incorporate geolocation, they rarely integrate detection and localization within a unified memory space.

To bridge this gap, we fuse onboard metadata (GPS/IMU/barometer) with the estimated camera pose to project detector outputs into a geo-referenced frame. The resulting

metadata-aware memory map enables persistent, interpretable confidence accumulation across flight sequences, improving reliability under sparse appearance cues and supporting semantic consistency over large areas. This tight, online coupling of geospatial fusion with object detection remains comparatively underexplored in aerial robotics and motivates our approach.

2.6 Perception on Unmanned Surface Vehicles

Perception on unmanned surface vehicles (USVs) poses unique challenges compared to ground or aerial platforms due to dynamic water backgrounds, the absence of a stable ground plane, and continuous platform motion induced by waves. Early approaches showed that exploiting a simple maritime scene structure (sky–water–obstacle) enables robust obstacle detection from onboard video (Kristan *et al.*, 2016). Public datasets such as MODD/MODD2 and MaSTr1325 provided USV imagery with semantic labels, and benchmarks like MODS established evaluation protocols for detection and segmentation (Bovcon *et al.*, 2019, 2021). More recent models, such as WaSR, refine water segmentation to reduce false positives, and LaRS extends evaluation to diverse aquatic environments (Bovcon and Kristan, 2022; Žust *et al.*, 2023). Despite these advances, existing methods remain largely segmentation-based and rely on active sensing or careful calibration for distance estimation. This highlights the need for lightweight, learning-based approaches that infer object distance directly from monocular imagery and remain robust under platform motion, a direction explored in subsequent chapters.

2.7 Identified Research Gaps

Despite the significant progress in monocular, panoramic, and mobile perception systems, several key limitations remain unresolved:

- Lightweight image detectors often operate on single frames and lack mechanisms to exploit temporal context for improved robustness under motion, blur, or occlusion.
- Spatial memory mechanisms are underexplored for single-camera setups with limited computational capacity, leading to unstable detections over time.
- Pixel-wise depth estimation networks remain resource-demanding, which hinders their integration with lightweight object detection pipelines on embedded platforms.
- UAV perception pipelines seldom exploit onboard metadata (e.g., GPS, IMU, altitude) for maintaining geo-referenced semantic consistency.

- USV perception remains dominated by segmentation-based methods and lacks lightweight monocular distance estimation suited for embedded deployment.

To bridge these gaps, this dissertation develops three complementary research directions:

- 1. Spatial reasoning through memory and metadata.** A memory-enhanced detection scheme accumulates confidence over time in a consistent spatial frame, improving stability without recurrent architectures. For UAVs, onboard metadata (GPS, IMU, altitude) is leveraged to construct geo-referenced detection maps, enabling temporally consistent semantic perception.
- 2. Temporal integration for lightweight robustness.** A multi-frame fusion strategy introduces short-term temporal context directly at the input level, improving robustness to blur, occlusion, and motion while keeping computation low enough for embedded real-time operation.
- 3. Object-level distance estimation.** Distance regression heads are integrated into YOLO-like detectors for monocular and 360° imagery, extending the approach to USV scenarios where active range sensors are unavailable. This formulation enables metric distance estimation without dense depth supervision or heavy geometric modeling.

Collectively, these contributions advance the state of real-time single-sensor perception for autonomous ground, aerial, and surface platforms operating under tight sensing and computational constraints.

Chapter 3

Robust 360° Camera Object Detection and Localization with Memory Maps

This chapter presents a cost-effective approach for object localization and location-aware object detection using a single 360° fisheye camera lens on mobile platforms such as street cleaning vehicles. We propose a system that captures a comprehensive view of the surroundings and accurately detects people and objects. Using the camera's geometry, the system infers distances to objects on the ground and projects them into global coordinates, creating a temporal spatial map. This 'memory map' is continuously updated, allowing for the accumulation of detection predictions over time. This approach significantly enhances the robustness and accuracy of object detection in dynamic environments. Our experiments demonstrate the system's efficacy, making it a strong candidate for implementation in various real-world applications requiring enhanced situational awareness and autonomous decision-making capabilities.

3.1 Introduction

Accurately perceiving and interpreting the surrounding environment is crucial, especially for autonomous vehicles operating in dynamic settings. Traditional object detection systems frequently encounter difficulties due to their restricted field of view, limiting their effectiveness in monitoring dynamic changes across entire environments. This chapter introduces a new approach that addresses these challenges by integrating a single 360° camera with advanced object detectors on mobile platforms, such as street cleaning vehicles.

Our motivation stems from the critical need for enhanced situational awareness in these vehicles. In agricultural and urban maintenance applications, the ability to detect people and objects reliably in real-time is not just a matter of efficiency, but of safety and operational effectiveness. The dynamic nature of these environments, coupled with the diverse range of dynamic obstacles encountered, poses a unique set of challenges. Current methods often fall short in providing a comprehensive and accurate detection necessary for these applications (Miller *et al.*, 2022; Wang *et al.*, 2022).

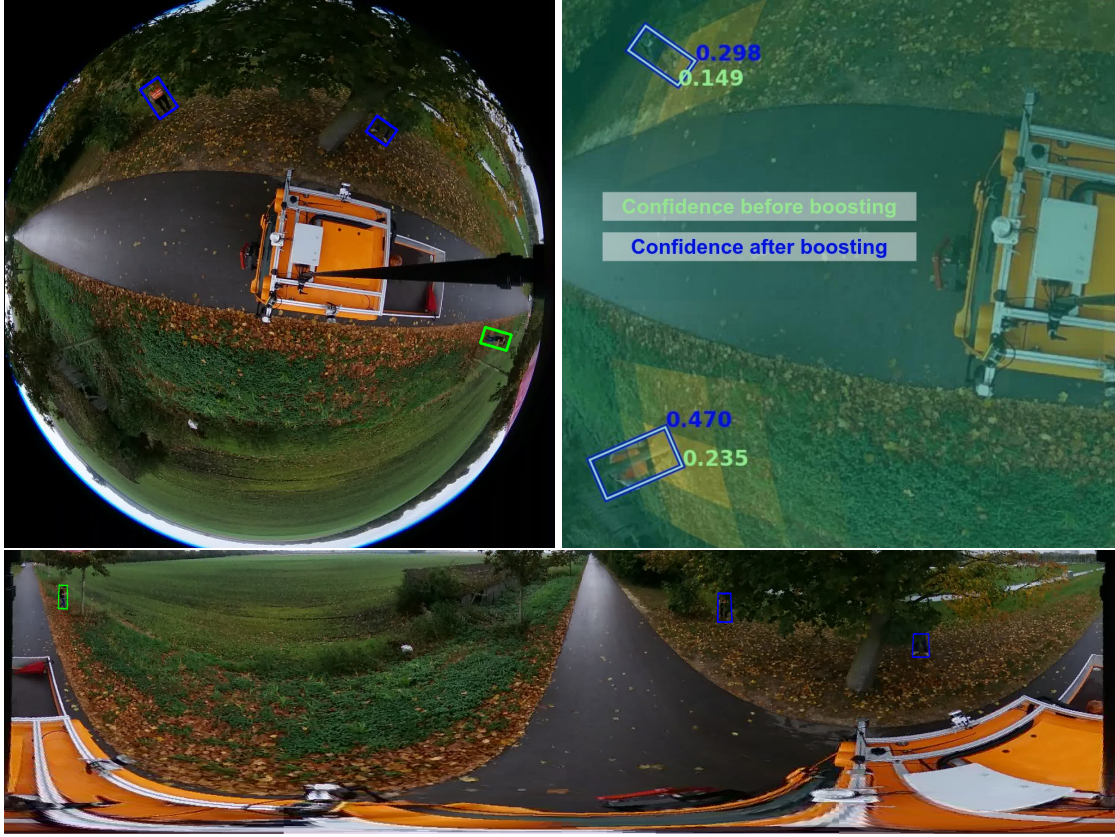


Figure 3.1: With just a single camera, we obtain a full surround view. Top left: fisheye view, bottom: equirectangular view. We propose a triangulation-based distance estimation and downstream ‘memory map’ in world space to improve object detection and alleviate missing detections caused by (partial) occlusions. With the ‘memory map’, we boost the confidence of our object detector in regions where we’ve seen more detections before (top right). A standard object detector only detects the green box, whereas the memory map is able to detect the blue boxes as well (top left).

While LiDAR has been a solution for many autonomous systems, offering precise distance measurements and object detection (Venugopala, 2022; Li and Ibanez-Guzman, 2020), it comes with notable downsides: LiDAR systems are often expensive, adding significant cost to the deployment of autonomous technologies. Additionally, LiDAR data requires intensive computational resources to process, which can introduce latency in real-time decision-making scenarios. Furthermore, LiDAR sensors often only provide sparse point clouds that may miss objects and are insufficient for obtaining a comprehensive semantic understanding of the scene (Wang *et al.*, 2016).

While multi-camera systems enhance situational awareness by offering a panoramic view, they also present challenges. Calibration and stitching complexity, increased data processing demands, and susceptibility to environmental conditions such as varying

lighting and weather disturbances are notable disadvantages (Rampinelli *et al.*, 2014). These systems require sophisticated setup and high-capacity computing resources, leading to higher costs and potential limitations in scalability and adaptability (Losada *et al.*, 2010).

To address this, we propose a simple and inexpensive system that leverages the expansive field of vision offered by a 360° camera, combined with an object detection system running in real-time (≈ 20 FPS). This setup not only captures a complete view of the surrounding environment but also uses the camera’s geometric properties to infer the distance to detected objects. These objects are then projected into global coordinates, forming a temporal spatial map. A key innovation of our approach is the development of a ”memory map”, an averaged aggregation of detection predictions over time, enhancing the robustness and accuracy of the object detection process.

Our main contributions are:

- **Dataset:** We collected a dataset comprising omnidirectional images with accompanying ground truth object detection labels and point cloud data on a mobile platform and make it publicly available¹. This dataset features a novel setup that requires only a single camera without the need for stitching.
- **3D Localization:** We explore an application by merging omnidirectional images with point cloud data for 3D localization, where the point cloud serves as a ground truth for evaluating our image-based localization method. This approach seeks to refine the precision and extend the application of localization techniques in varied outdoor settings.
- **Memory Map:** Our method enhances the object detection process by integrating geometric projections and 3D spatial information into a memory map. This enhancement aims to improve both the accuracy and robustness of detecting objects in complex outdoor environments, using spatial data to navigate the intricacies of such scenes effectively.

3.2 Related Work

Traditional object detection systems often are constrained by the narrow field of view (FOV) of perspective cameras, facing challenges in dynamic environments. Most of the works on surround view systems have been done on multi-camera systems (Varlashin *et al.*, 2020; Kumar *et al.*, 2021a), requiring extensive installation and calibration procedures alongside heavy compute demands.

The integration of a single 360° camera offers a broader FOV, mitigating data synchronization and stitching issues and enhancing environmental perception. Heinemann *et al.*

¹ <https://cloud.cs.uni-tuebingen.de/index.php/s/SPcSPDsigYboy7S>

(2004) represents a seminal approach in fast and accurate environment modeling using omnidirectional vision. Their methodology emphasizes the importance of a panoramic view for complete environmental analysis. However, while their approach contributes significantly to indoor environment modeling, our method extends these concepts to address the dynamic nature of outdoor settings.

The LOAF dataset (Yang *et al.*, 2023a) employs a fisheye camera for large-scale person detection and localization, focusing on fisheye-specific distortion through novel network training techniques. However, the method employed for evaluating localization accuracy, measuring the pixel-distance relationship with a static camera placement and a single instance measurement using a measuring tape on the floor, poses limitations in terms of dynamic accuracy and real-world applicability. In contrast, our work advances this approach by leveraging LiDAR-recorded point cloud data as a ground truth. This method not only provides a more robust and accurate basis for evaluating localization but also enables the exploration of camera performance on dynamic platforms, offering a significant enhancement over the static and singular measurement technique employed previously.

Complementing this, Kiefer *et al.* (2023e) leverages UAV metadata to create robust dynamic maps for improved object tracking and localization, showcasing the potential of integrating metadata for enhanced UAV surveillance capabilities. DAB-DETR (Liu *et al.*, 2022b) integrates dynamic anchor boxes with the DETR model, significantly accelerating training convergence and enhances detection precision, showcasing the power of combining explicit positional priors with transformer architectures. Our work benefits from this advancement, applying DAB-DETR within the context of 360° environmental understanding, pushing the boundaries of detection accuracy in dynamic settings.

The class of (occupancy) grid maps (Nuss *et al.*, 2018) is related to the simple memory map we discuss in our work. We are not concerned with creating static maps and occupancies, but rather with high-likelihood areas to make object detection temporally more robust. There are related concepts of dynamic maps, e.g. Chung and Huang (2010), but our focus to leverage these is on the improvement of object detection in surround view object detection via employing a simple memory map.

3.3 Methodology

3.3.1 Dataset Generation

Our dataset comprises of omnidirectional images and point cloud data, collected from a mobile platform equipped with a single-lens Ricoh Theta Z1 camera and three Ouster OS-1 sensors. The downward-facing camera was mounted atop the platform via a camera pole mounted at varying lengths above the vehicle, aligning the lens' central axis perpendicular to the platform. This setup gives us a *full 360° horizontal field of view* and a vertical view up to the horizon line. It mitigates the need for image stitching, simplifying the data processing pipeline considerably. Figure 3.2 shows the setup of our sensor



Figure 3.2: Sensor setup on the data collection vehicle. Ouster lidar sensors are positioned at the front, right, and left (B, C, D). The 360° camera is mounted at A.

platform.

We focus on person detection and manually annotated each visible individual with radius-aligned rotated bounding boxes on the collected omnidirectional images, using our own newly designed annotation tool. This annotation approach significantly enhances the dataset’s accuracy by facilitating exact person positioning on the ground and enabling precise 3D geolocation from 2D pixel coordinates. In contrast, classical axis-aligned bounding boxes do not allow for precisely geolocating objects as their ground surface point is not as exact, as shown in Figure 3.3. Moreover, it establishes a ground truth for evaluating our 3D geo-location-aware detection model against traditional detectors that do not use 3D geo-location information.

To capture accurate 3D positions of objects surrounding the platform, we used three Ouster OS-1 LiDAR sensors, oriented towards the front, left, and right. The integration of these sensors with the camera enabled the synchronization of omnidirectional imagery with dense point cloud data for precise 3D localization of detected objects. The transformations among the camera and lidar sensors are pre-defined. The camera is located inside the field of view of at least one of the ouster sensors. See an overview of the dataset in Table 3.1.

3.3.2 Lens Model for Passive Geo-localization

Figure 3.4 illustrates the fisheye lens model. Through radial distortion, a point P on the ground plane is mapped to a point p on the image plane. The figure delineates how the distortion caused by the fisheye lens alters the projection compared to that of a traditional perspective lens, represented by point p' .

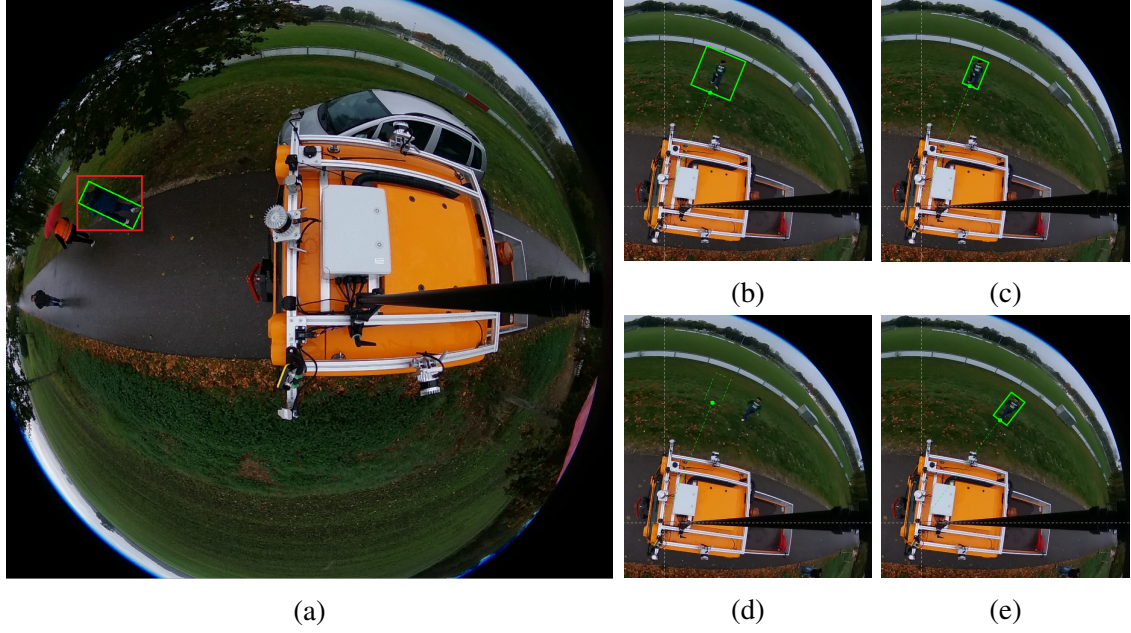


Figure 3.3: (a) The axis-aligned box (red) exhibits worse properties for downstream geolocalization as the surface contact point is not as precise as for the radially corrected bounding box (green). (b)-(e) illustrate the labeling process: A rotated bounding box of default size is initially placed using mouse cursor input (b), the box size is adjusted via mouse wheel (c). We copy the annotation to the next frame (d), from where its position needs to be adjusted (e).

As stated in Kannala and Brandt (2006a), a general mapping between the radial distance on the image plane r and the incidence angle θ measured with respect to the optical axis can be described as following, where k_i are the distortion coefficients of the lens model.

$$r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^7 + k_5\theta^9 + \dots \quad (3.1)$$

Rather than calculating the coefficients k_i for the general lens model as expressed in Eq. 3.1, we establish the correlation between θ and r through linear interpolation between the calibration data supplied by Ricoh, as shown in Figure 3.4. This approach not only bypasses the need for higher-order polynomial computations and their inverses but also enhances the inference speed.

Once the mapping between θ and r has been set, the location of P on the ground plane can be determined from geometric calculations, knowing the projected pixel location p' on image, and the camera focus length f , the camera height Z . In the following formulae, (u, v) represent the pixel coordinates of point p , (u_c, v_c) denote the central pixel of the image. The function $\theta(r)$ relates the radial distance r to the angle θ , where θ is the angle of incidence. The angle ϕ is calculated between the vector pointing to the pixel and the

Table 3.1: Overview of Recorded Dataset. The arrow $\downarrow$ indicates down-sampling.

Data Type	#Frames	FPS	Resolution	#Instances
Total Image Frames	5770	3 ($\downarrow$ 30)	1920x1920	17275
Total Point Cloud Frames	1921	1 ($\downarrow$ 10)	1024x128	5536
Image Frames - Occlusion	859	3 ($\downarrow$ 30)	1920x1920	1875
Point Cloud Frames - Occlusion	286	1 ($\downarrow$ 10)	1024x128	725

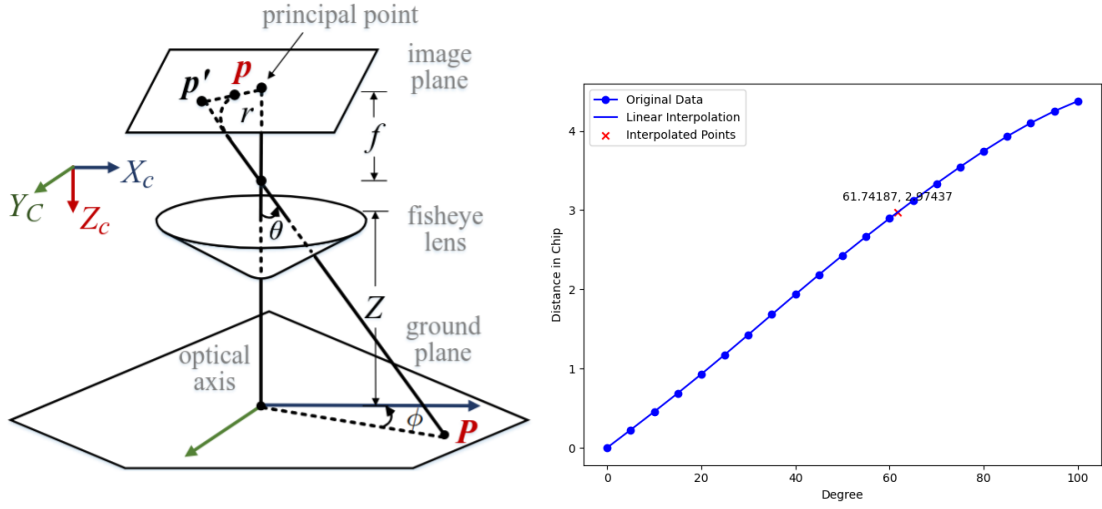


Figure 3.4: Fisheye lens model for 3d localization, illustrated from Yang *et al.* (2023a) (left). Interpolation between θ in degrees measured with respect to the optical axis and r in millimeters on the chip. (right).

image's horizontal axis. I_{res} represents the image resolution in pixels, and S_{size} is the sensor size in millimeters.

$$x = Z \cdot \tan(\theta(r)) \cdot \cos(\phi) \quad (3.2)$$

$$y = Z \cdot \tan(\theta(r)) \cdot \sin(\phi) \quad (3.3)$$

$$z = 0 \quad (3.4)$$

$$\phi = \arctan 2(v - v_c, u - u_c) \quad (3.5)$$

$$r = \frac{\sqrt{(u - u_c)^2 + (v - v_c)^2}}{I_{res}} \cdot S_{size} \quad (3.6)$$

Figure 3.5 presents a visual exemplification of our lens model’s capabilities in geolocating objects from image data.

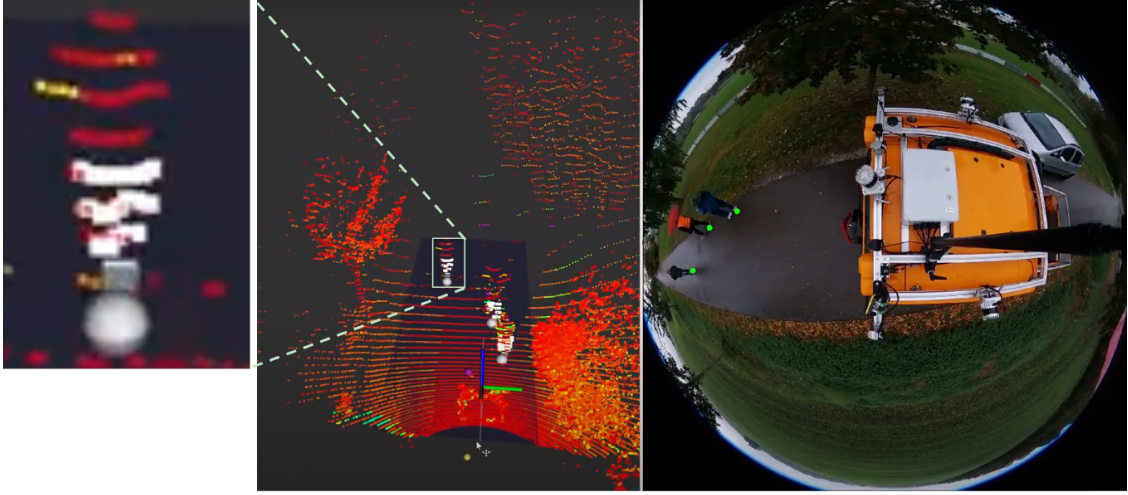


Figure 3.5: Visual representation of geo-localization results using our lens model. The right panel shows an omnidirectional image captured by the Ricoh, with the green dots representing the person detections. In the middle, the point cloud data from the LiDAR sensor is displayed, where white points indicate the LiDAR’s person-classified points. The centroids of these white point clusters are projected onto the ground, depicted as gray cubes, representing the actual standing locations of persons in 3D space. The gray spheres correspond to the projected locations of the green dots from the omnidirectional image, transformed into the 3D coordinate system using the fisheye lens model.

3.3.3 Confidence Calibration via Memory Mapping

The dynamic nature of the observed scene, both from the sensor platform’s mobility and the subjects’ movements, prompts a shift in pixel coordinates within the image space. However, the physical constraints on a person’s movement speed in the real world imply a relative constancy in 3D space. Leveraging this disparity, we propose a method that employs a temporal-spatial memory map to calibrate detection confidence levels, particularly in the presence of partial occlusions.

Further developing the work from Kiefer *et al.* (2023e), we construct a 3D memory map that evolves over time, informed by the projection of detected object locations and their confidence scores. This probabilistic-like approach extrapolates the change of object presence in future frames based on accumulated historical data. For the next frame, we project the newly detected bounding boxes into this 3D memory map to retrieve the corresponding memory values, which then inform the calibration of the confidence scores of the newly detected bounding boxes. This method not only fortifies detection

reliability in areas with a history of presence but also reduces false positives in areas without prior detections.

The memory map M spans a predefined area centered at the sensor platform, encompassing a 100 by 100 meter region, discretized with grid size $s \times s$. In the following, $M^t(x, y)$ denotes the grid cell containing x and y in the discretization M for arbitrary x, y at time $t \in \mathbb{N}_0$. For $t = 0$, each grid cell the memory map is initialized to a value of 0.5 to represent an equi-probable state of object presence:

$$M^0(x, y) := 0.5, \quad \forall x, y. \quad (3.7)$$

For each of the raw detections on image at timestamp $t > 0$, c_i^t denotes its confidence, p_i^t its bottom center pixel, and $l_i^t = (x_i^t, y_i^t)$ its coordinate in the world frame. This is obtained from the pixel to 3D projection as described in Equations 3.2 – 3.6. These contribute to the memory map update in the close vicinity of the respective point, in detail:

$$M^t(x, y) = \text{normMean}(M^{t-1}(l_i^t) + c_i^t), \forall i \quad (3.8)$$

$$l_i^t = \text{proj}_{3D}(p_i^t), \quad (3.9)$$

where $\text{normMean}()$ is an operation that shifts the values within the memory map M^t to ensure an average value of 0.5, thus standardizing the probable state distribution. Above, the update step in equation 3.8 is applied to all indices in a 3×3 grid around l_i^t ,

$$(x, y) \in \{x_i^t, x_i^t \pm s\} \times \{y_i^t, y_i^t \pm s\}. \quad (3.10)$$

In the following, $\beta > 1$ is a boosting factor for updating the confidence value. Then the calibrated confidence value $\tilde{c}_i^t$ of the detected object can be formulated as follows:

$$\tilde{c}_i^t = \begin{cases} c_i^t \cdot \beta, & \text{if } M^{t-1}(l_i^t) \geq 0.5, \\ c_i^t / \beta, & \text{if } M^{t-1}(l_i^t) < 0.5. \end{cases} \quad (3.11)$$

Equation 3.11 increases the confidence of a detection if the memory map M^{t-1} from last time step suggests there should be an object and vice versa.

Afterwards, the detected object with the calibrated confidence will be filtered by the confidence threshold or by the non maximum suppression following the standard processes of the detection model, leading to the final detection result.

Figure 3.6 shows a visual representation of the 3D memory map, which is projected back onto the image plane, offering a color-coded likelihood of person presence within each grid cell.

3.4 Experimental Results

This section provides a detailed overview of our hardware used, experimental setup and the results obtained.

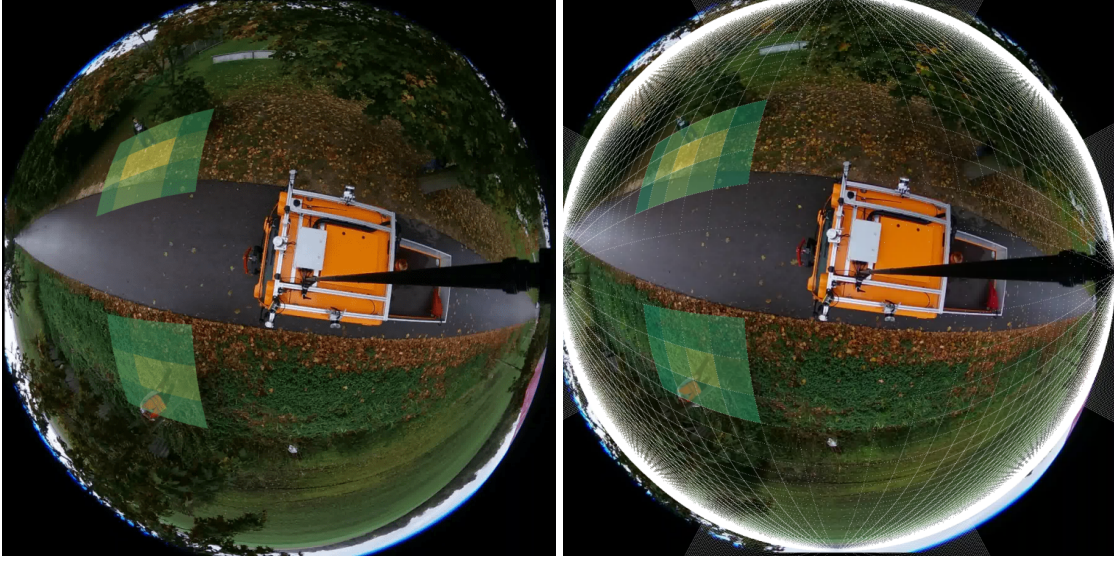


Figure 3.6: Illustration of the reprojected 3D memory map onto the image space, where bright yellow indicates high likelihood areas for person presence. The grid size is set at 1 meter. For optimal clarity, only regions with map values exceeding a selected threshold are used for overlay. Please see how the memory map evolves over time in the supplementary video. The right image visualizes the discretization of the geo-memory map.

3.4.1 Hardware and Tools

- **Ricoh Camera:** the camera Ricoh Theta Z1 was configured to capture image data at a resolution of 1920x1920 pixels at a frequency of 30 Hz. The camera orientation was horizontal, with one lens directed towards the ground and the other skyward. This setup was deliberately chosen to circumvent the complexities and potential inaccuracies associated with image stitching. While this configuration results in the camera pole being visible within the image frame, we contend that the obstruction is minimal and does not significantly impact the overall effectiveness of our method’s evaluation.
- **Ouster Lidar:** The three Ouster Lidar OS1 sensors, each with 64 channels, are operating at 10 Hz. They were mounted on the front, left, and right sides of our mobile platform. The front sensor was angled 20 degrees downward from the horizontal plane, while the side sensors were tilted 60 degrees downward, optimizing the field of view for comprehensive spatial data capture. This configuration ensured a rich point cloud dataset, enhancing the evaluation of our localization model.
- **Custom Annotation Tool for Object Detection:** We developed an annotation

tool² tailored to produce radius-aligned rotated bounding boxes for persons in 360° footage. Our annotation framework characterizes each bounding box by a set of parameters that denote its position, dimensions, and orientation in the image space. Specifically, each bounding box is defined by the coordinates of its center (x_{center}, y_{center}) , its dimensions given by the width and height, and the angle of rotation α relative to the image axes. This precise parameterization allows us to accurately pinpoint the location of a person’s stance in the image space. This annotation approach, as illustrated in the right panel of Figure 3.5, is crucial for the subsequent accurate projection into the person’s 3D location.

3.4.2 Experimental Setup and Results

To evaluate our lens model and geo-localization approach, we synchronized our point cloud and image data. The bottom center pixel of each annotated bounding box in the images is projected into 3D space to obtain a geolocation estimate. For the point cloud data, given the tilted positioning of the lidar sensors, we used RANSAC (Fischler and Bolles, 1981) to define the ground plane’s coefficients. Following this, DBSCAN (Ram *et al.*, 2010) was applied to identify individual clusters, with their centroids projected onto the ground plane to serve as ground truth for our 3D location estimates from the lens model. Figure 3.7 showcases the 3D localization error of our method with respect to the distance from the mobile platform. The error values generally remain below 0.5 meters for distances up to 10 meters, indicating a high degree of accuracy in the localization process. This consistent error margin underscores the model’s reliability, particularly notable in the mid-range distances where the density of data points suggests robust performance despite increasing distance. The plot emphasizes the method’s accuracy in estimating positions in 3D space.

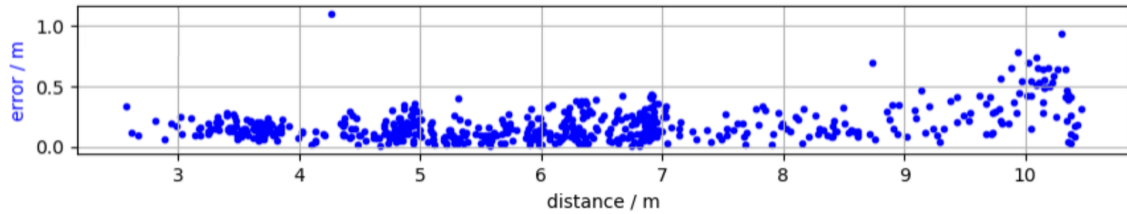


Figure 3.7: The 3D localization error relative to object distance, where the clustering of points below the 0.5-meter error margin up to 10 meters highlights the method’s precision.

To assess the effectiveness of our confidence calibration approach through temporal-spatial memory mapping, we conducted the following experiments: detection performance with vs. without 3D location information, impact of spatial memory map resolution, evaluation of memory map and confident update functions.

² <https://cloud.cs.uni-tuebingen.de/index.php/s/SPcSPDsigYboy7S>

We adopted the DAB-DETR (Liu *et al.*, 2022b) as our primary model for image-based object detection, in alignment with the methodologies proposed in Yang *et al.* (2023a). We trained the model on the LOAF dataset for 50 epochs and then fine-tuned further on our custom annotated dataset for an additional 20 epochs with the default settings.

To assess the efficacy of our temporal-spatial memory mapping approach for confidence calibration, we selected $AP@50$ and $AR10$ as our metrics due to their established reliability in object detection evaluation (Lin *et al.*, 2014; Hosang *et al.*, 2016; Felzenszwalb *et al.*, 2010; Everingham *et al.*, 2014). $AP@50$ provides a balanced measure of precision at a 50% IoU threshold, emphasizing the accuracy of detections, while $AR10$ evaluates the model’s recall, considering the top 10 detection results, to gauge its ability to detect relevant objects without being overwhelmed by false positives. Our experiments were stratified into two distinct scenarios. For the first, absent of occlusions, we explored a variety of memory map grid sizes (0.5, 0.75, 1, and 1.5 meters) and confidence boost factors (1.3, 1.5, and 2). The $AP@50$ observed across these configurations ranged from 0.9700 to 0.9703, closely aligning with the baseline of 0.9702, achieved without incorporating 3D geo-information. This outcome shows the robustness of our detection method even in configurations not optimized for the highest performance, without significantly compromising detection accuracy in scenarios free from occlusions.

In the second scenario, with partial occlusions by tree trunks or leaves, the application of 3D geolocation data significantly elevated the $AP@50$ and $AR10$, highlighting the method’s value in visually complex situations. The $AP@50$ ranges from 0.425 to 0.473, while the baseline is 0.410. Among the 12 settings we explored, ten showed significant improvement over the baseline, underscoring the effectiveness of our approach. The best four performances are quantitatively presented in Table 3.2.

Table 3.2: Experimental results for temporal-spatial memory mapping approach with model DAB-DETR. Here, s is the map grid size, and β is the boosting factor. Columns with - for s and β denote the baseline without memory maps.

	Non-occlusion			Partial occlusion				
	s	β		s	β			
	-	1.5	1	-	0.75	1.5	1	1.5
	β	-	1.3	2	-	2	1.3	2
$AP@50$	0.9702	0.9702	0.9703	0.410	0.425	0.426	0.430	0.473
$AR10$	0.5778	0.5778	0.5778	0.1695	0.176	0.178	0.176	0.198

To test the generalization of our approach, we also conducted the same experiments on other models, e.g. YOLOv7-tiny (Wang *et al.*, 2023) and YOLOv9-tiny (Wang *et al.*, 2024), obtaining similar results. Specifically, on an NVIDIA Orin AGX, our approach with the YOLOv7-tiny model achieved real-time performance of 25 FPS. The results are presented in Table 3.3 and Table 3.4.

Table 3.3: Experimental results with YoloV7-tiny. Here, s is the map grid size, and β is the boosting factor. Columns with - for s and β denote the baseline without memory maps.

	Non-occlusion			Partial occlusion				
s	-	1.5	1	-	0.75	1.5	1	1.5
β	-	1.3	2	-	2	1.3	2	2
AP@50	0.6773	0.7087	0.6989	0.1412	0.1458	0.1543	0.1420	0.1409
AR10	0.4072	0.4093	0.4065	0.0915	0.1127	0.1127	0.1127	0.1127

Table 3.4: Experimental results with YoloV9-tiny. Here, s is the map grid size, and β is the boosting factor. Columns with - for s and β denote the baseline without memory maps.

	Non-occlusion			Partial occlusion				
s	-	1.5	1	-	0.75	1.5	1	1.5
β	-	1.3	2	-	2	1.3	2	2
AP@50	0.5425	0.5698	0.5841	0.0829	0.0869	0.0926	0.0869	0.0925
AR10	0.2972	0.3065	0.3102	0.0559	0.0636	0.0627	0.0636	0.0627

3.5 Conclusion and Outlook

This chapter presented a surround-view object detection and localization system using a single 360° fisheye camera. By projecting detections onto a ground-plane and accumulating them in a continuously updated memory map, the method improved robustness against occlusions and inconsistent predictions, demonstrating that monocular panoramic vision can support reliable situational awareness for ground-based mobile platforms.

The approach, however, is bounded by simplifying assumptions: it models the environment as a flat 2D surface, presumes a stable and downward-facing camera pose, and maintains an egocentric frame of reference. These constraints are appropriate for urban ground vehicles but may not generalize to scenarios with dynamic viewpoints or strong variations in platform motion.

To address such conditions, the next chapter investigates how the memory-map concept can be adapted to aerial platforms, where changes in pose, altitude, and orientation demand globally consistent representations informed by additional sensor metadata.

Chapter 4

Confidence Boosting in 3D for Object Detection on UAVs

While the previous chapter introduced spatial memory maps for enhancing object detection in monocular 360° settings, it operated under the assumption of a ground-based platform with a relatively stable pose and locally accumulated confidence. In contrast, this chapter focuses on object detection from Unmanned Aerial Vehicles (UAVs), where frequent changes in position, altitude, and orientation introduce significant challenges to temporal consistency and spatial reasoning.

To address these challenges, this chapter extends the memory-based framework by integrating onboard UAV metadata—including GPS, IMU, and altitude—to construct a globally referenced confidence map. Rather than accumulating detections in a fixed egocentric frame, this approach projects observations into world coordinates, enabling robust spatiotemporal fusion across dynamic trajectories and complex aerial scenes. This globally aligned representation improves detection reliability and interpretability in both image and real-world space.

In doing so, the system supports metadata-driven confidence boosting, which benefits multiple temporal vision tasks such as video object detection, short- and long-term tracking, and anomaly detection. These results highlight the advantage of using metadata for temporally consistent perception and demonstrate how geometric awareness can further stabilize and enhance UAV-based detection pipelines in unconstrained environments.

4.1 Introduction

When we make predictions about the presence and location of objects, we have an internal understanding of our surrounding world: implicitly, we know where we are in relation to the object and we know about the topology of a given scene. This internal understanding of the surrounding geometry allows us to reason robustly about the existence and location of objects. Furthermore, while we make detection errors when shown ambiguous static scenes, over time, we are able to strengthen our belief about our predictions. This is due to slight changes in appearance caused by different view points, slight movement of objects or just by integrating our predictions over a certain period of time.

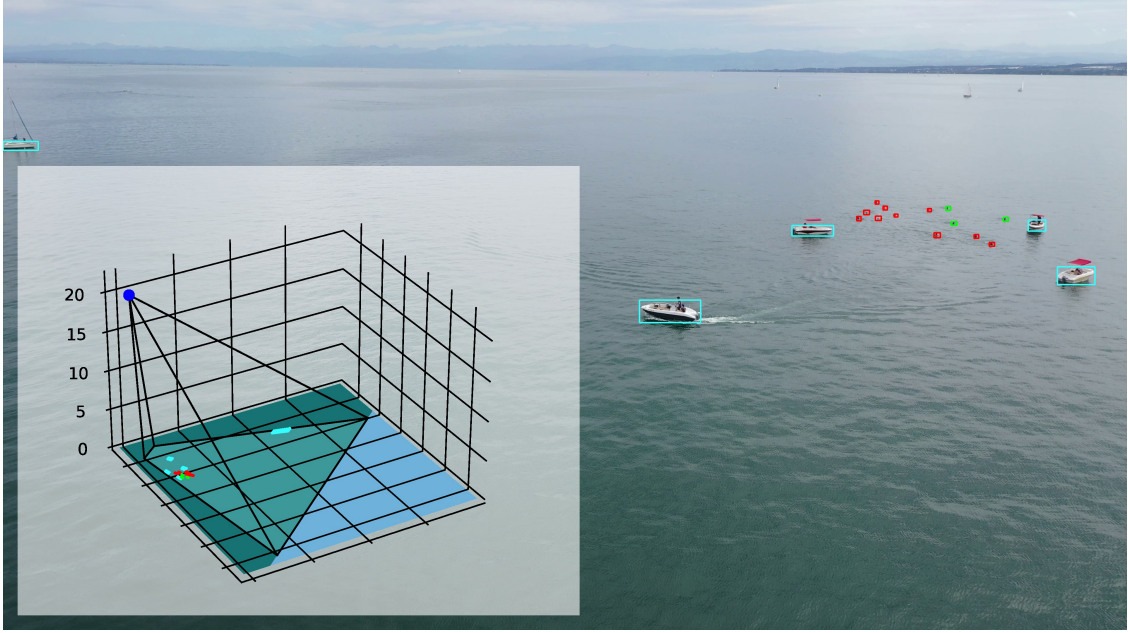


Figure 4.1: Illustration of predictions and the corresponding 3D geometry/topology. We aggregate information over time, taking the correct 3D geometry into account.

We argue that this awareness of our surrounding is particularly important in aerial scenarios, where we need to reason about our environment in the presence of many uncertainties, caused by the smallness of objects. Motivated and inspired by this human-based analogy, we aim to improve several computer vision tasks for object detection and tracking on UAVs.

Conventional techniques for object detection and tracking from UAV perspectives ignore the intrinsic geometry and topology present in UAV-generated imagery, frequently relying on off-the-shelf methods designed for COCO-like (Lin *et al.*, 2014) scenarios or with slight modifications, see e.g. Mittal *et al.* (2020). This makes it difficult for these methods to aggregate uncertain predictions over time, since movements in image space cannot correctly be tracked and, hence, features are hard to be associated temporally.

We argue that this shortcoming can easily be mitigated by leveraging freely available sensors onboard the UAV. GPS alongside compass and IMU measurements allow us to reason about our and the objects' locations in 3D coordinates. In turn, this allows us to create a *temporal memory map* of previous predictions, so that we can aggregate information over time in a geometrically sensible way, resulting in more robust predictions.

In this chapter, we show how correctly considering the 3D geometry allows us to propose a temporal memory map that results in more robust predictions. In particular, our contributions are as follows:

- We derive mathematical formulas detailing the 3D geometry around a UAV.

- We leverage metadata by proposing a memory map to robustify several temporal computer vision tasks, such as video object detection, tracking and anomaly detection.
- We capture the PeopleOnGrass, a benchmark featuring 4K 30 FPS manually annotated video, featuring precise metadata labels, and make it publicly available. Furthermore, we capture and annotate more data for multiple further experiments and release it publicly.
- We show in multiple experiments on diverse benchmarks the utility of our method.

4.2 Related Work

Although video object detection (VOD) on UAVs has become more relevant, with many applications emerging and many network architectures becoming fast enough to deploy on embedded devices, there is still no consensus on which VOD approach seems to be the most promising. Broadly speaking, there are optical flow-based networks, memory networks and tracking-based networks (Wu *et al.*, 2021). While memory networks, such as STDnet-ST (Bosquet *et al.*, 2021), achieve high accuracies, their heavy architectures prohibit their deployment on embedded devices. Optical flow-based networks are faster, but their benefit over single-frame object detectors is limited (Corsel *et al.*, 2023). Most tracking-based networks first do single-frame object detection and use the association for the detection reciprocally (Luo *et al.*, 2019). In fact, the best two VOD models of the last VisDrone-VID challenge were single-frame methods, entirely ignoring the temporal domain (Zhu *et al.*, 2019).

Furthermore, all common video object detectors on UAVs ignore the underlying 3D geometry of the scene and only operate in image space (Wu *et al.*, 2021). While there is much research focusing on geolocation, it is only focused on obtaining world coordinates of objects for downstream tasks, such as following a target (Zhao *et al.*, 2019). Similarly, occupancy networks and other mapping approaches aim at obtaining a map for mapping or scene understanding (Wei *et al.*, 2022).

As opposed to these works, we aim to leverage metadata to build a memory map in GPS space that *improves the video object detection performance*. Since we only rely on freely available metadata onboard the UAV, this approach is viable for small UAVs with a standard RGB camera in low-cost settings (compare to active geolocation via laser (Yang *et al.*, 2019)).

4.3 Deriving Formulas for 3D Geometry

Figure 4.2 illustrates that we would like to know the GPS position of the swimmer as indicated by the orange cross. How do we obtain that? First, we discuss how to obtain

relative coordinates to the UAV, then how to use these to obtain actual world coordinates via passive geolocation.

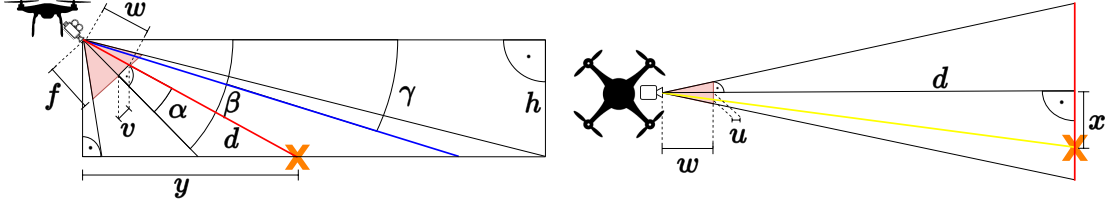


Figure 4.2: Illustration of variables and formulas to estimate the vertical distance y (red lines) and the horizontal distance x (yellow lines, illustrated on bottom left) from the UAV to the swimmer (orange cross).

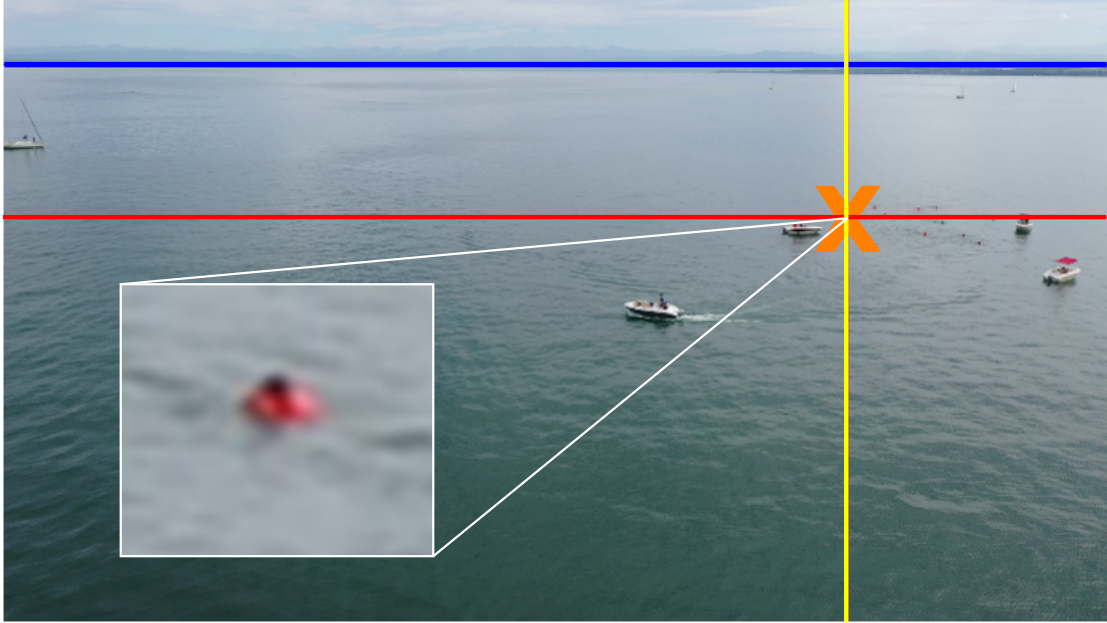


Figure 4.3: Corresponding image example for the computation in Figure 4.2. Here, we have that $h = 20m$ and $\beta = 15^\circ$ resulting in $y = 110m$ and $x = 25m$. This means, that the swimmer is $110m$ in front and $25m$ to the right of the UAV as measured on the ground. Note that we also estimate the horizon line as indicated by the blue line.

4.3.1 Relative Coordinates

We consider a mathematical perspective projection camera model since this resembles the common use-case for cameras on UAVs. We consider our camera to have a focal length f given in pixels. For simplicity, we assume our image to be of 4K (3840x2160)

resolution. Other resolutions follow an analogous derivation. Our camera is looking down at an angle of β , which is the variable gimbal angle. The gimbal also balances a potential UAV roll angle, so that we assume there to be a zero camera roll angle.

First, we estimate the on-ground distance from the UAV to the red line, which we denote as y (see Figure 4.2). We assume the ground to be flat, i.e. we ignore elevation change and the curvature of the earth. The latter is a reasonable assumption for heights that are typical for UAV missions (Bohren and Fraser, 1986), while the first may make a difference in very dynamic terrains.

Using the focal length f and the y-axis pixel position offset v from the horizontal center pixel line of the image, we compute y by computing the angle α via

$$\alpha = \arctan\left(\frac{v}{f}\right). \quad (4.1)$$

This allows us to compute y via

$$y = \tan(90^\circ - (\beta - \alpha))h. \quad (4.2)$$

Having the vertical ground-distance to the object of interest, we compute the horizontal ground-distance x by looking from the top as indicated on the bottom left of Figure 4.2. For that, we need the variables d and w , which both are easily computed via Pythagoras' theorem, which then yield x :

$$d = \sqrt{h^2 + y^2}, \quad w = \sqrt{f^2 + v^2}, \quad x = \frac{u}{w}d. \quad (4.3)$$

For simplicity, we ignored the sign of x , which indicated whether we are on the left or on the right of the vertical center line. Naturally, in the actual implementation, we incorporate that. Likewise for the special cases when an object of interest is behind the UAV, which happens for gimbal angles close to 90° .

Furthermore, we compute the horizon line in image space following a similar derivation as Kiefer and Zell (2023). Ignoring the effect of atmospheric refraction, we estimate the distance to the horizon line l as a function of the height of the UAV as

$$l = 3.57h^{1/2}. \quad (4.4)$$

This approximation is fairly accurate for heights that are typical for our scenarios (far below 1000m) (Bohren and Fraser, 1986). We compute the angle γ to the horizon via

$$\gamma = \arcsin(h/l). \quad (4.5)$$

Using the focal length f and the camera gimbal pitch β , we compute the camera

perspective projection of the horizon on the image plane, which yields the height offset o in pixels to the horizontal center line of the image plane as

$$o = \tan(|\gamma - \beta|) \cdot f \cdot \text{sgn}(\gamma - \beta). \quad (4.6)$$

Finally, we need to truncate o to be within the range of the number of allowable horizontal pixels. We refer to Kiefer and Zell (2023) for a discussion on accuracy. Lastly, note if there is a lens distortion, then we undistort the image before running the above algorithm.

4.3.2 Absolute Coordinates

Using the relative coordinates of an object (x - and y - ground-distances to UAV), we compute its GPS coordinates based on the UAV's GPS coordinates as follows. Given the camera heading angle θ , we compute the rotation matrix and rotate the relative coordinates of an object to obtain

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (4.7)$$

Finally, we map the relative coordinates to GPS coordinates with the following, where r is the earth radius.

$$la^{object} = la + \frac{y_r}{r} \frac{180}{\pi}, \quad (4.8)$$

$$lo^{object} = lo + \frac{x_r}{r} \frac{180}{\pi} \frac{1}{\cos(lat \pi/180)}. \quad (4.9)$$

We note that we compute the earth radius based on the latitude to account for the slight ellipsoid nature of the earth. Lastly, note that we can reverse the computation to map from 3D to 2D, i.e. the image space (see code that we'll make publicly available).

4.4 Method: Temporal Memory

Classical grid maps in robotics, such as occupancy grid maps, divide the environment into single grid cells and estimate the occupancy probability for each cell (Nuss *et al.*, 2018). We also aim to build temporary memory maps but in the context of modeling dynamic objects of interest. While there are several works extending occupancy grid maps to account for dynamic objects, e.g. Chung and Huang (2010), our focus is on leveraging maps to improve the downstream performance of computer vision detection

and tracking algorithms over time. Hence, we propose a simple class of what we denote *temporal memory maps*.

4.4.1 Map Representation

We consider a UAV-centric map with a fixed size context window, i.e., given the latitude la_t and longitude lo_t of the UAV at discrete time step t , the memory map is defined to be

$$M_t = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \left| \begin{array}{l} x \in \{la_t - \frac{c}{2} + \frac{j}{n} \mid j \in 0, \dots, n(la_t + \frac{c}{2})\} \\ y \in \{lo_t - \frac{c}{2} + \frac{j}{n} \mid j \in 0, \dots, n(lo_t + \frac{c}{2})\} \end{array} \right. \right\}, \quad (4.10)$$

which denotes a quadratic, north-oriented map around la_t and lo_t of edge size c with n^2 equidistantly spaced elements. Note that c and n should be problem-dependent reasonable quantities. This treatment makes sure that the surrounding environment of the UAV can be considered at all times. Note that c should be large enough to account for the relevant field of view of the UAV and n should be large enough to have sufficient resolution. Furthermore, the map will forget information that is outside of the context window, which needs to be considered in downstream applications.

From here, the exact procedure of how to update and leverage the map differs between the computer vision tasks. Broadly speaking, all methods define a memory map on M_t , indicating likelihoods of object locations.

4.4.2 Video Object Detection

After analyzing single-image object detectors, we found that most trained object detectors can detect almost all objects before the non-maximum suppression (NMS) stage, which in our case is doing both, removing duplicate boxes and low-confidence ones. Perhaps unsurprisingly, this is due to object detectors distributing many anchors across the image, from which potential objects are regressed. However, detections tend to cluster around actual objects, thereby achieving theoretical recall values of close to one.

An immediate observation yields that we can filter out predictions based on their sizes. Common object detectors employ multiple stages, responsible for multiple scales. Often, the detectors output predictions, whose sizes are far off from any reasonable quantities. By looking at the distance d and the class indicator of any predicted object, we establish a distance-to-size (size measured by diameter of the box) relation and analyze it on the SeaDronesSee object detection training set (see Figure 4.4). For each class separately, we perform a Gaussian process regression on these data points. This yields a mean function $m(x)$ and a covariance function $cov(x)$. We scale the covariance function by a factor dependent on the distance from the maximal to the minimal bounding box size in a small interval around x . Finally, we check whether a new prediction is inside the resulting accepted area, and if not, we discard that prediction (see Figure 4.5). Table 4.1 shows that applying this procedure to the three best SeaDronesSee Object Detection v2



Figure 4.4: Blue boxes are true positive detections, the red box is a false positive swimmer detection that was successfully filtered out by analyzing the distance-size relation.

Table 4.1: Numbers of removed (Rem.) boxes (B) for best three models trained and tested on SeaDronesSee Object Detection v2.

Model name	#B	Rem.FP	Rem.TP	%
Maritime-VSA (Kiefer <i>et al.</i> , 2023b)	674K	8,087	24	1.2
DetectoRS (Kiefer <i>et al.</i> , 2023b)	838K	6,705	67	0.8
YOLOv7-Sea (Kiefer <i>et al.</i> , 2023b)	378K	4,926	54	1.3

models already removes a small number of false positives (FPs).

Nevertheless, there needs to be a suppression stage because the high recall still comes at the cost of many false positive bounding boxes. We argue that robust detection can be achieved by temporally and geometrically sensible aggregation of many of the low-confidence detections that cluster around actual objects. Since we do this aggregation in actual GPS space, we can cumulate likelihoods from different viewpoints, hence this treatment is theoretically invariant to camera movements, which often cause traditional methods, tracking features across frames, to fail (e.g. see Figure 4.6).

To provide evidence for the initial claim about the high recall before NMS, we train a Faster R-CNN ResNet-18 and a YOLOv7 (configs from Kiefer *et al.* (2023b)) on SeaDronesSee v2 (Kiefer *et al.*, 2023b). For evaluation, we remove the NMS stage and obtain

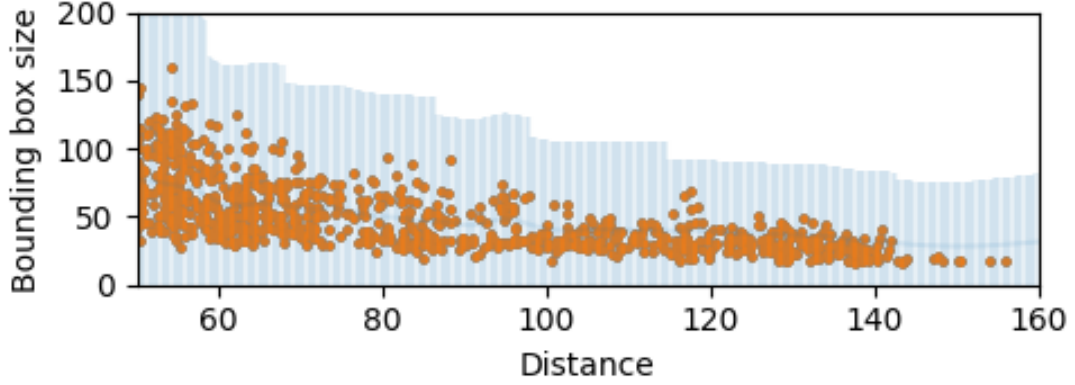


Figure 4.5: Distance (x-axis) to size (y-axis) relation for all swimmers in SeaDronesSee object detection train set (orange dots). The blue area denotes the accepted area of sizes.

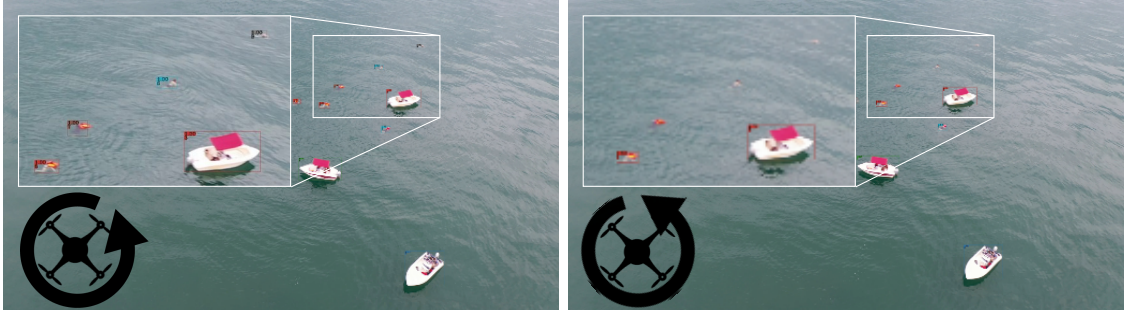


Figure 4.6: A heading change of the camera results in three swimmers (detected in left frame) being lost in subsequent frames (right frame) (DeepSORT (Kiefer *et al.*, 2023b)).

recall values of 97.4 % and 98.8%, resp. Compare that to the recall after the NMS stage: 86.4% resp. 87.6%. See also Figure 4.7 for a sample image showing pre-NMS detections.

This leads us to hypothesize that a large part of missing detections can be obtained (and also false detections can be suppressed) if the confidence scores are modified such that low-confidence detections in *GPS areas*, where there have been detections for several consecutive time steps, are boosted, and, simultaneously, (semi-)confident false positive detections are suppressed if they only appear in certain frames (randomly).

Memory Map Construction:

Therefore, we propose to map all detections before NMS to GPS space as follows. For every box detection $b_i = [x_1^i, x_2^i, y_1^i, y_2^i]$, $i = 1, \dots, m$, we take its bounding box center $[(x_2^i + x_1^i)/2, (y_2^i + y_1^i)/2]$ and compute its GPS coordinates $g^{b_i} = [la^{b_i}, lo^{b_i}]$ via the



Figure 4.7: Before NMS, there are many low-confidence detections, but they cluster around actual objects. Ignoring the confidence, these detections would yield a recall value of one. For visualization, only one-colored boxes.

formulas in Section 4.3. We compute the closest grid window to g^{b_i} via

$$w^{b_i} = \arg \min_{m \in M_t} \|g^{b_i} - m\|. \quad (4.11)$$

Ideally, this is the actual GPS position of that prediction. However, to account for imprecision in the sensor data and deviations from taking the center of the bounding box (which may not be the GPS center of the object), we assign weight to the neighbouring space as well, albeit less, since w^{b_i} is our best single-point prediction¹. Hence, we construct a memory map likelihood over M_t as follows: We add a truncated Gaussian density with radius r at w^{b_i} , i.e.,

$$\tilde{p}_t(x) \leftarrow \tilde{p}_t(x) + se^{-x^2} \text{ for } x \in [w^{b_i} - r, w^{b_i} + r] \subset M_t, \quad (4.12)$$

where we slightly abuse the notation by mixing scalars and vectors (computation to be understood component-wise). Note that we hid the other parameters (constants and variance) in the scaling factor s , which we will *additionally* make dependent on the confidences c_i belonging to b_i .

After every time step, we truncate $\tilde{p}$, such that the memory map values are ≤ 1 at all times:

¹Naively applying this method on all boxes before filtering of incorrectly sized boxes leads to too large or small bounding boxes becoming boosted.

$$p_t(x) = \min(\tilde{p}_t(x), 1). \quad (4.13)$$

Furthermore, we introduce a forgetting factor ϕ , by which the memory map p_t is rescaled after each update step so that the memory map will not be overloaded over time.:

$$p_t(x) \leftarrow \phi p_t(x). \quad (4.14)$$

Note that we explicitly did not model p_t as a probability distribution over M_t as this would introduce a competition of weights among the objects leading to deteriorated results when the number of objects and their prediction certainty vary (similar to a discrete Bayes filter (Fox *et al.*, 2003), although here, we do not have a control update).

Adjusting the Detector Confidences According to p_t :

At time t , each bounding box b_i has a confidence $c_i \in [0, 1]$, which we will update based on p_{t-1} via

$$\hat{c}_i = c_i + p_{t-1}(w^{b_i}). \quad (4.15)$$

Note that we do not update the map based on the updated confidence $\hat{c}_i$ but on the original c_i . Lastly, note that we perform this procedure *class-wise*, so we keep a map for each class separately. After the procedure, we proceed with standard NMS.

4.4.3 Extension to Object Tracking and Reidentification

Similarly as for VOD, a *temporal memory map* helps in tracking scenarios. We apply the memory map to object tracking via a simple extension.

For frame t , we use the GPS location of each tracked object in frame $t - 1$. We take the boosted predictions of frame t closest to the old GPS location with a confidence above a threshold as the new location (in image and GPS space) of the object to be tracked. We will see that this simple formulation of a tracker helps to track objects over time in presence of camera movements, such as in Figure 4.6. Even if a fast camera movement yielding blurry frames results in the object detector missing the object temporally - if an object will be redetected in subsequent frames - the object can be assigned the correct ID again.

While there is benefit in using memory maps for *short-term* tracking tasks, they are especially beneficial in *long-term* tracking, where ground truth objects may leave the frame entirely. The task of reidentification in classical scenarios is solved by feature-based comparison of objects across time. In UAV-based domains, this often fails for small objects as features look similar across different objects. However, the treatment in GPS space allows us to remember where an object was, allowing us to reidentify it in subsequent frames over a pre-defined time horizon.

4.4.4 Extension to Video Anomaly Detection

The goal of video anomaly detection is to output regions of the frames that are considered anomalous, either spatially (from appearance) or temporally (unseen movements). Since anomalies are not well-defined, methods achieve only limited success in completely detecting all anomalies, and the aerial nature leads to highly ambiguous scenes leading to many false positive detections as well (Kiefer and Zell, 2023). Hence, this task is far from solved.

However, we may assume anomalies to be temporally stable, i.e. they exist over a considerable period of time. This makes it ideal to aggregate information at certain GPS positions over a sequence of frames. Since it is hard to explicitly define regions that are anomalous, most methods output an anomaly heatmap, indicating areas of likely anomalies. Therefore, we propose to map the entire image plane to GPS space and aggregate anomalous regions over time in a geometrically reasonable way. We note that this requires an efficient vectorized implementation to still achieve real-time inference. Section 4.5.3 discusses this shortly.

For the correct mapping of pixels to GPS coordinates, we need to make sure that the projection is well-defined. Therefore, we apply the previously introduced horizon cutter, which first computes the horizon and only maps the pixels below it to GPS space. We ignore pixels too far in the horizon as this results in too much noise.

We replace the 2D difference heat map from Kiefer and Zell (2023) with our memory map, which will be added to the previous memory map. After each step, we also apply a forgetting factor ϕ . Afterwards, we apply the same post-processing steps as indicated there.

4.5 Results and Analysis

Since there is only one publicly available dataset for VOD and Multi-Object Tracking that employs precise metadata (SeaDronesSee-MOT (Varga *et al.*, 2022)), we collect and manually annotate another dataset, People-On-Grass-Video (POG-V), depicting people walking on grass and similar terrain (see example in Figure 4.8). It comes with the same type of metadata annotation as SeaDronesSee-MOT. In particular, we used a Zenmuse X5 camera mounted on a DJI Matrice 100 for collection. We collected 10,633 frames in 30 fps videos with 3840x2160 resolution. We annotated 48,802 instances of people using DarkLabel². We made sure to have a great variance w.r.t. to altitude ($h = 10m-100m$) and gimbal pitch angle ($\beta = 17^\circ-90^\circ$).

Note that we focus on the detection of people in both benchmarks only, since this is the hardest class to detect (Kiefer *et al.*, 2023b). For that, we fuse the classes *swimmers*, *swimmers with life jacket* and *life jacket* into a single people class in SeaDronesSee-V (SeaDronesSee-MOT without instance IDs).

²<https://github.com/darkpgmr/DarkLabel>

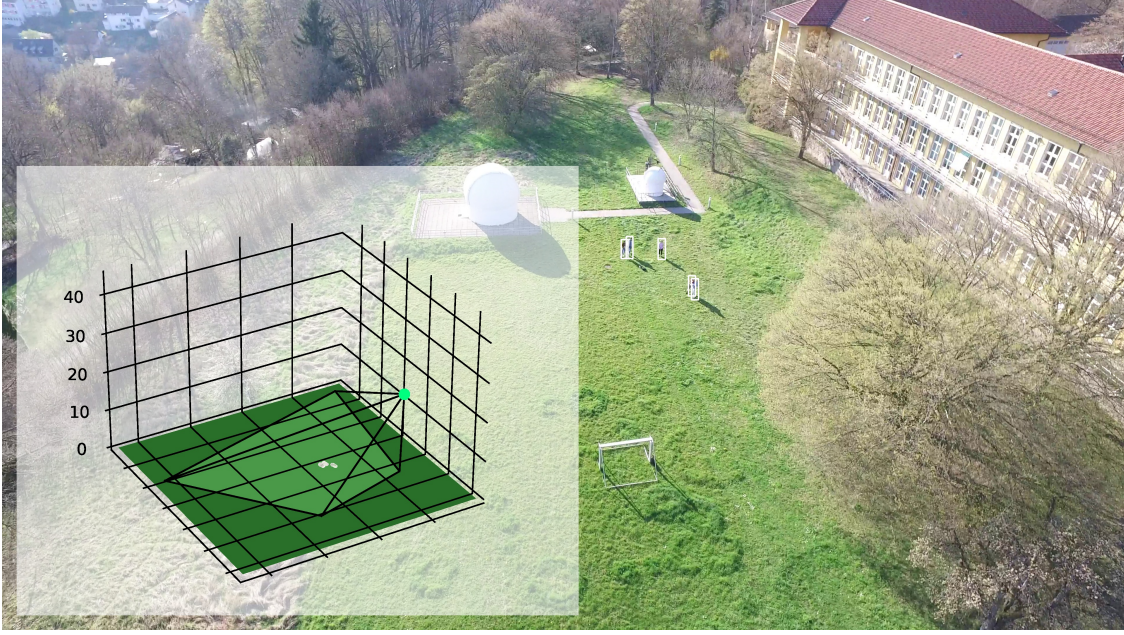


Figure 4.8: Example frame from our captured benchmark POG-V taken at 31m altitude and a gimbal pitch angle of 41° .

For short-term tracking, we also employ SeaDronesSee-MOT. It currently only supports short-term tracking, i.e. objects that left the scene and reappear do not have the same id. Therefore, we relabel a video clip of SeaDronesSee-MOT, such that it also supports long-term tracking, which we need to assess the performance of long-term tracking (reidentification).

For Video Anomaly Detection, we fall back to the OpenWater data set of the Maritime Anomaly Detection Benchmark (Kiefer *et al.*, 2021) since it is the only one featuring metadata.

Finally, we perform additional experiments to evaluate an extension to multiple UAVs in a collaborative scenario.

4.5.1 Video Object Detection

We apply the proposed memory map on the output of a YOLOv7-Tiny (Wang *et al.*, 2023) trained and tested on SeaDronesSee-V and POG-V, respectively. We choose a grid with 0.5m resolution and a size of 300m. We truncate the Gaussian updates with a radius of 6m.

As baselines, we take the well-known Video Object Detectors Deep Feature Flow (DFF) (Zhu *et al.*, 2017b), Flow-guided Feature Aggregation (FGFA) (Zhu *et al.*, 2017d) and the recent Temporal RoI Align (T.R.A.) (Gong *et al.*, 2021) as implemented in Chen *et al.* (2019) with their default configuration.

Table 4.2: Video Object Detection accuracy for Swimmers and People on SeaDronesSee-Video and POG-Video, respectively. The last column denotes running times (wall-clock time in frames per second) benchmarked on an Nvidia Xavier AGX.

Model name	SeaDronesSee-V		POG-V		FPS
	AP ₅₀	AR	AP ₅₀	AR	
DFF (Zhu <i>et al.</i> , 2017b)	53.7	34.2	30.3	18.8	3.4
FGFA (Zhu <i>et al.</i> , 2017d)	53.9	34.1	34.7	24.8	1.6
T.R.A. (Gong <i>et al.</i> , 2021)	60.6	42.9	29.9	19.3	2.4
YOLOv7Tiny	68.2	41.1	81.3	36.2	25.1
+Mem. Map	72.8	44.0	84.4	38.8	25.1

Table 4.2 shows that incorporating our memory maps leads to increased AP₅₀ and AR (average recall with averaged IoU levels 0.5:0.05:0.95 and at most 100 detections) values by successfully boosting the confidence scores. In particular, we achieve a +4.6 and +3.1 AP₅₀ increase over the single-image YOLOv7-Tiny object detector on SeaDronesSee-V and POG-V, respectively. Interestingly, popular video object detectors lack behind in performance because they are not targeted to aerial VOD. Moreover, the increased AR values show that our method detects more objects, which is the main challenge in UAV-based detection.

Figure 4.9 visualizes the temporal memory map, projected to the image space. It shows the heatmap values of likely object locations, which were aggregated from the previous frames. The red areas cluster around actual objects. This results in pre-NMS boxes becoming boosted, as Figure 4.10 shows. While the baseline YOLOv7-Tiny originally did not detect a single swimmer, now we detect eight.

Furthermore, the standard video object detectors are not suitable for deployment on embedded devices as shown in Table 4.2. Our method only adds a negligible time to a standard YOLOv7-Tiny, which can run in real-time on an Nvidia Xavier AGX.

4.5.2 Object Tracking

Applying our method on SeaDronesSee-MOT yields an increase in HOTA and MOTA and a decrease of ID switches and fragmentations over the DeepSORT short-term tracker (see Table 4.3).

To test the reidentification capability, we run our method on top of a DeepSORT with reidentification module on the relabeled video as described earlier. We can successfully reassign the same id to all the objects in the video whereas the baseline fails to do so (4 people are reassigned new IDs.)

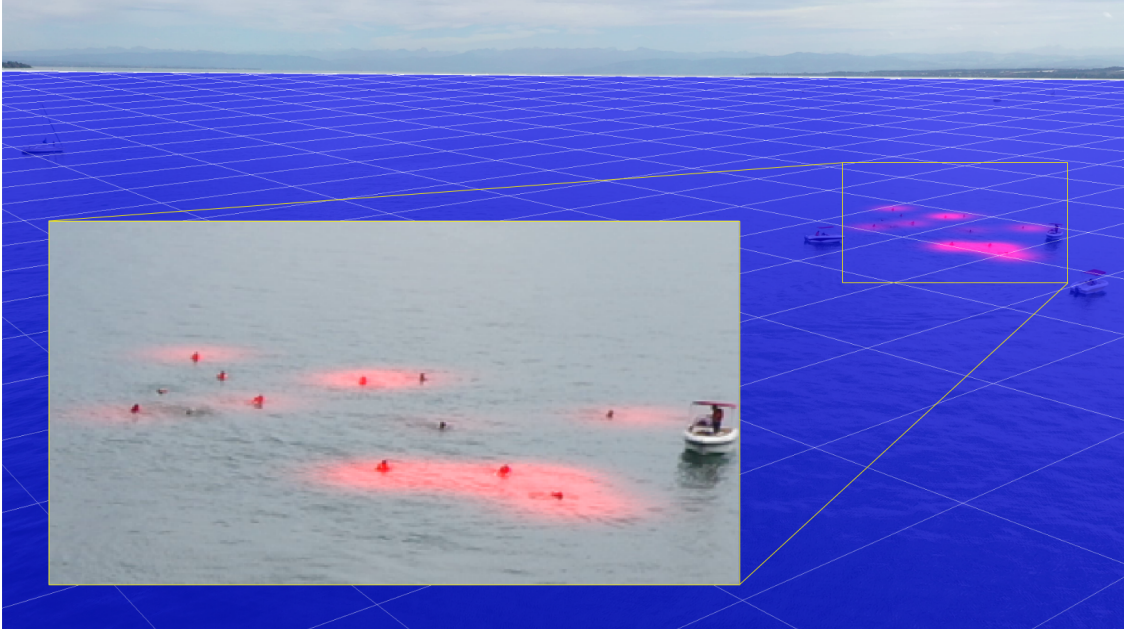


Figure 4.9: Heatmap illustration of memory map on a SeaDronesSee-V test frame. We project actual GPS latitude and longitude lines to image space. We also project the memory map from GPS space to image space and visualize it as a heatmap. The right image part shows that the memory map successfully puts weight around actual swimmers' locations. Compare also to the zoomed-in crop and Figure 4.1 and 4.10.

4.5.3 Video Anomaly Detection

Table 4.4 shows that adding the module to the anomaly detector from Kiefer and Zell (2023) yields an increase of +3 AR to 82.8 (average recall over multiple levels of broadcasting rate; measured differently than average recall from object detection, please refer to Kiefer and Zell (2023)). This performance increase is also reflected in the recall at broadcasting rate of 5%, i.e. there is a +3.8 $R^{p=5\%}$ improvement over the Autoencoder.

Figure 4.11 shows the resulting difference anomaly map aggregated over the last frames and the resulting regions of interest that are returned.

The speed does not increase considerably over the Autoencoder, still achieving real-time inference benchmarked on an Nvidia Xavier AGX as Table 4.4 shows.

4.5.4 Cooperative Detection via Multiple UAVs

The GPS memory map allows for a joint representation of objects' locations. In this section, we demonstrate that it can be leveraged in a collaborative setting with multiple UAVs. This allows for cross-UAV knowledge transfer and leads to more robust predictions. Furthermore, it allows for detections of (partially) occluded objects. For example, Figure 4.12 shows the same scenery captured from different locations at the same time

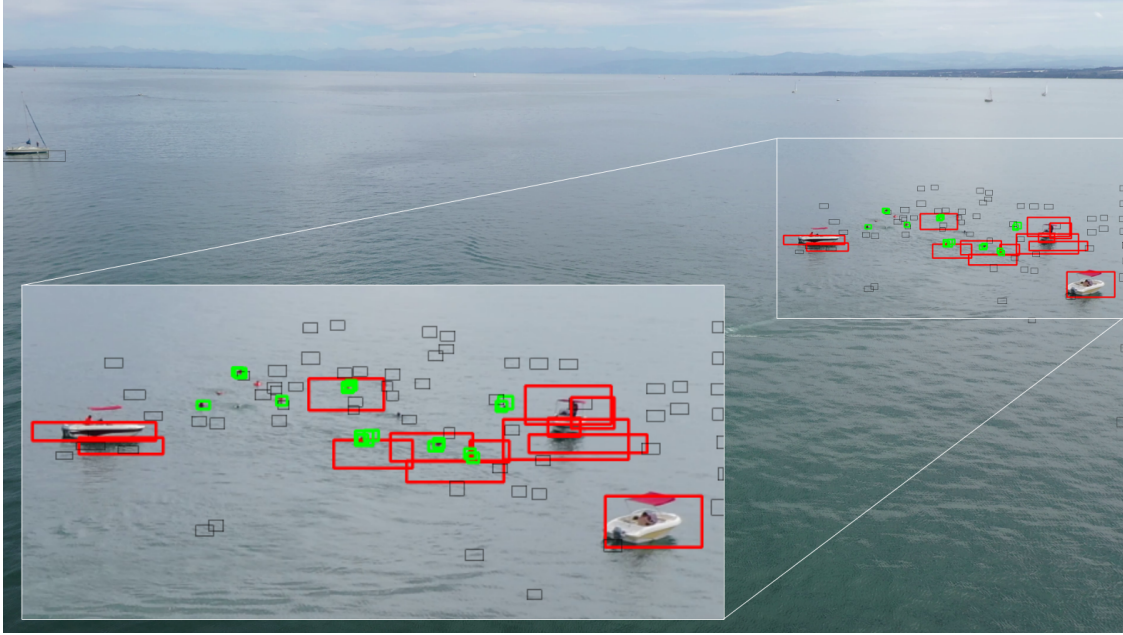


Figure 4.10: Before NMS, many swimmers are predicted (black boxes), several of which are automatically removed due to incorrect sizes (red boxes). Only the confidences of the green boxes are boosted. Applying a confidence threshold of 0.5 yields that 8 swimmers are detected as opposed to none for the baseline (experiment with YOLOv7-Tiny).

with two people are walking on the grass.

Having a joint location likelihood representation allows the UAVs to share information. We apply the same memory map from before on both UAVs' object detectors' output, but this time, we average their memory maps in the overlapping region. For that, we take an EfficientDet-D0 trained on POG (Kiefer *et al.*, 2021) and test it on the following data. We capture four minutes of footage in a similar environment as POG from the viewpoint of two UAVs, a DJI Matrice 210 and a DJI Mavic 2 Pro, denoted 2AVs. We varied variables, such as altitude, pitch and heading viewing angle and GPS location. We annotated the people visible in both video streams and compare the performance of the single-frame object detector with the memory map from before. We obtain an AP_{50} of 61.3 for the single-frame object detector compared to an AP_{50} of **68.9** for the memory map.

To show the utility of a joint memory map for tracking, we compare a fast single-object tracker, PrDiMP18 (Danelljan *et al.*, 2020), to a simple tracker based on the memory map. For that, we apply an EfficientDet-D0 on a consecutive subset of 2AVs that contains partial occlusions of a person in one video stream (blue camera in Fig. 4.12). While the baseline tracker cannot handle the occlusion and fails to track the person behind the tree entirely (4.3 AP_{50}) for the blue camera, our method leverages the joint memory map that transfers knowledge from the red UAV to the blue (86.3 AP_{50}).



Figure 4.11: Difference anomaly map aggregated in 3D space over several frames (top) and resulting regions of interested (bottom).

Table 4.3: Multi-Object Tracking accuracy on SeaDronesSee-MOT. We used the output of DeepSORT and built on top a memory map to become more robust towards id switches and fragmentations.

Model name	HOTA $\uparrow$	MOTA $\uparrow$	IDs $\downarrow$	Frag $\downarrow$
ByteTracker (Kiefer <i>et al.</i> , 2023b)	65.0	76.9	68	841
DeepSORT (Kiefer <i>et al.</i> , 2023b)	66.6	80.0	44	805
+Memory Map	67.2	80.8	35	721

Table 4.4: Video Anomaly Detection accuracy on SeaDronesSee-MOT. We built our memory maps on top of the Autoencoder (Kiefer and Zell, 2023).

Model name	AR $\uparrow$	$R^{p=5\%}\uparrow$	FPS
Gaussian Mixture Model (Kiefer <i>et al.</i> , 2023b)	45.9	2.6	17
Mean Filter (Kiefer <i>et al.</i> , 2023b)	73.9	54.3	50
Frame Differencing (Kiefer <i>et al.</i> , 2023b)	76.1	54.8	62
Autoencoder (Kiefer <i>et al.</i> , 2023b)	79.8	71.0	27
+Memory Map	82.8	74.8	27

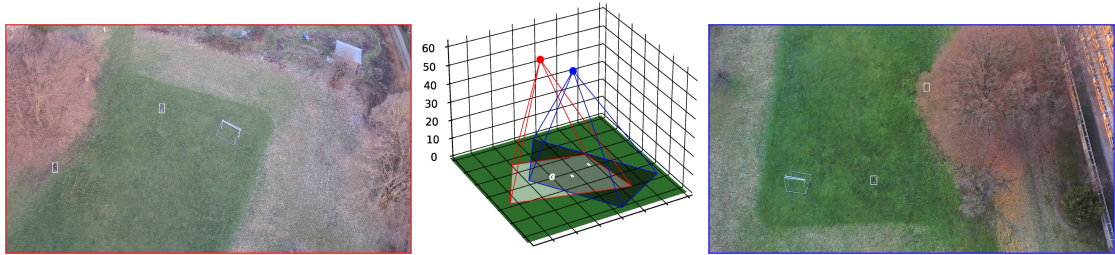


Figure 4.12: Two UAVs inspect the same scenery from different locations at the same time. The red camera (left) has a good view on both objects and the goal (G) while the blue camera (right) is underexposed (brightened for visualization) and one person is partially occluded by a tree. We aggregate both memory maps together to obtain more robust detections. Without considering the underlying geometry, this is not possible with standard video object detectors.

The joint memory map also allows for reidentification in long-term tracking tasks *while the object is moving*. To test this, we take a standard DeepSORT trained on POG with the default Reid model within mmtracking (Chen *et al.*, 2019) as baseline. We take a subset of 2AVs where one camera leaves the scenery entirely, while the other is tracking the object throughout. When reappearing, it immediately uses the track id information from the other camera, while DeepSORT failed to reidentify the object, which we believe to be attributed to the object size.

4.5.5 Accuracy and Calibration

To test the accuracy of the passive geolocation, we put a GPS sensor (C94-M8P from ublox), which is accurate up to two centimeters, on a Robotnik Summit XL mobile robot. We launched a DJI Mavic 2 Pro and assessed the 2D-3D-projection quality by comparing the bounding boxes as follows. We project the GPS location to image space, resulting in the red bounding box of Figure 4.13. The projection already is quite accurate, yielding an offset of 112 pixels in a 4K image. Naturally, this very much depends on many factors, such as altitude, viewing angle and more.

To alleviate constant measurement errors, induced by an imperfect calibration at the start of the UAV, we may alleviate it by recalibrating using a known GPS location of an object. We take the robot from Figure 4.13 and compare its bounding box to the projected bounding box. We optimize for parameters, such as the heading angle, gimbal angle and GPS measurements (latitude, longitude, altitude). Doing this over multiple images yields offsets for these parameters. Testing on a subsequent flight around the robot yielded more accurate geolocations, as illustrated in Figure 4.13. A detailed analysis is beyond the scope of this chapter and we refer, e.g., to Albrektsen and Johansen (2017).

4.6 Conclusion and Outlook

This chapter demonstrated that incorporating UAV metadata into confidence maps provides a principled way to stabilize detection and improve temporal consistency in aerial video. By projecting detections into geo-referenced world coordinates, we achieved more robust performance across object detection, tracking, and anomaly detection tasks, without imposing significant computational overhead. These results highlight the benefits of coupling metadata with vision-based detection, enabling interpretable and geometry-aware reasoning in dynamic flight scenarios.

Nevertheless, reliance on metadata introduces certain limitations: calibration errors, restricted applicability in GPS-denied environments, and inefficiencies when full localization is unnecessary. In applications where the primary requirement is estimating object distances—rather than constructing globally referenced memory maps—the overhead of maintaining spatial maps becomes disproportionate to the task.



Figure 4.13: Example calibration procedure: We know the GPS position of the robot. Projecting the GPS location to image space yields the red bounding box. Taking the distance to the yellow box as loss, we optimize the gimbal pitch angle, heading angle, and GPS longitude and latitude parameter offsets that might have occurred, leading to a better calibrated metadata measurement. After calibration across several frames from different viewpoints, we obtain the orange box.

This motivates the next chapter, which explores a more direct, learning-based approach to object distance estimation. Focusing on unmanned surface vehicles (USVs) in maritime environments, we shift from metadata-driven 3D mapping to lightweight monocular distance regression, aiming to provide a cost-efficient alternative to traditional ranging sensors.

Chapter 5

Approximate Supervised Object Distance Estimation on Unmanned Surface Vehicles

The use of UAV pose metadata in the previous chapter improved detection stability through spatial referencing. However, when the task requires only object distance—rather than full localization—maintaining a global memory map represents an inefficient use of resources. From this perspective, a more direct and lightweight solution is desirable.

This chapter addresses that need by introducing a learning-based formulation that estimates object-level distances directly from images, thereby bypassing the need for external pose information or global mapping. The approach is particularly motivated by maritime applications, where Unmanned Surface Vehicles (USVs) are increasingly deployed for tasks such as navigation, monitoring, and inspection. In these settings, reliance on conventional sensors such as LiDAR, radar, or depth cameras is often impractical: they are costly, prone to noise or sparsity, and demand extensive calibration efforts.

To overcome these limitations, we collected a dedicated dataset of images captured from USVs, annotated with bounding boxes and corresponding distance measurements. Leveraging this data, we extend a standard object detection architecture with a regression branch that predicts the distance to each detected object. This integrated formulation enables the detector not only to recognize targets but also to infer their real-world distances, providing a compact, cost-efficient, and interpretable alternative to traditional distance measurement techniques. By aligning more closely with the way human operators intuitively estimate distance from visual cues, the proposed method offers a lightweight and straightforward solution for monocular 3D perception in maritime environments.

5.1 Introduction

Unmanned surface vehicles (USVs) and autonomous boats promise efficient autonomous navigation and enhanced safety. However, the development and operational efficiency of USVs are hampered by the limitations of current sensing and perception technologies (Liu *et al.*, 2016).

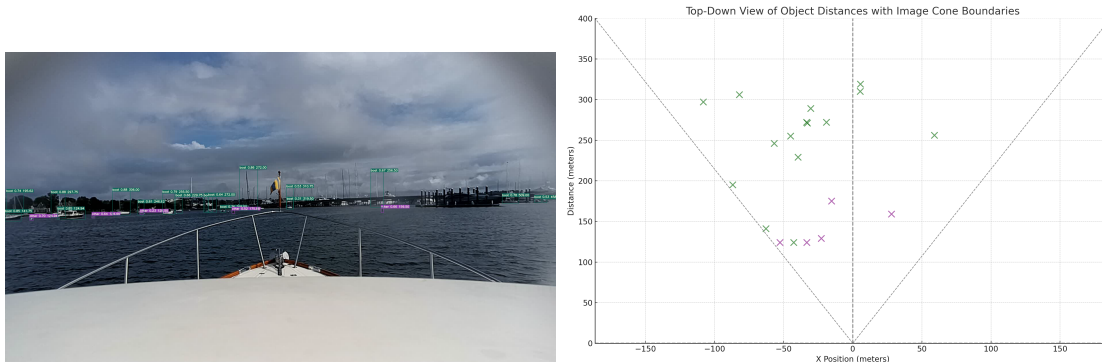


Figure 5.1: Example scene and its bounding box and distance predictions using our method (left). Projected detections onto the 2D plane (right).

Traditional sensors for distance estimation, such as LiDAR, radar, sonar, and depth cameras, play a crucial role in navigation and object detection (Jorge *et al.*, 2019). LiDAR exhibits high costs and sparsity of data. It requires complex processing to interpret sparse point clouds, making it less effective in dynamic, cluttered maritime environments.

Radar systems, while robust in harsh weather conditions, suffer from their own drawbacks. While less expensive than LiDAR, they are still orders of magnitude more expensive than monocular cameras. Their primary technical limitation is the inherent noise in radar signals, which leads to challenges in accurately detecting and classifying small or distant objects (Zhuang *et al.*, 2016). This limitation is particularly critical in busy maritime channels where the ability to detect small objects can be crucial for collision avoidance.

Sonar sensors are less effective for detecting surface-based obstacles due to their primary design for water depth estimation (Bae *et al.*, 2015). Besides their costliness and limited utility in surface object detection, sonar systems also raise environmental concerns. The acoustic emissions from sonar are known to be disruptive to marine life (Abate 1, 2010). These sound waves, often loud, can disturb, disorient, or even harm aquatic animals, especially marine mammals that rely on sound for navigation and communication.

Depth cameras offer a potential alternative, providing rich spatial data about the surroundings. However, they require a very large baseline for distances beyond 10-20m as the disparity error scales quadratically with distance. Moreover, precise calibration is necessary and currently, there is a lack of specialized datasets tailored to marine environments (Jorge *et al.*, 2019).

Monocular depth estimation leverages neural networks to estimate depth from single images (for each pixel). While promising in theory, this technique is hindered by the quality and variety of available datasets. Existing datasets are predominantly land-based and do not represent the unique challenges of maritime environments, such as water

reflections, varying lighting conditions, and diverse objects (Yang *et al.*, 2025).

Lastly, trigonometry-based approaches by estimating the orientation of the camera via an onboard IMU (or horizon finding methods), and subsequent ray-casting yield poor results due to the difficulties of obtaining a precise own-orientation in the presence of many different acceleration forces (Liu *et al.*, 2016) and as this projection is ill-posed with such acute viewing angles (Mou and Wang, 2018).

We propose a novel approach to approximate distance estimation on USVs using supervised object detection. We circumvent the limitations of traditional sensors by leveraging video-metadata pairs comprising images with manually annotated bounding boxes and corresponding distance measurements. We obtain the distance measurements by leveraging the onboard CAN bus to get the "NMEA"(National Marine Electronics Association) data stream to collect GPS, heading, and more, and compare to chart data, which we collected from open sources, such as the National Data Buoy Center. Using this as a test dataset, we propose an object detector-based neural network to predict distances of individual objects based on visual cues, akin to human estimations. For that, we also human-label distance by letting annotators gauge distances based on the image context.

This method has the advantage that any video data can be used to train a system to become better at distance estimation, without the need for any other costly or hard to configure sensors. Moreover, it does not affect the runtime of the underlying object detector, making it a cheap and simple solution.

We demonstrate the practical application of our method by running experiments on embedded hardware and analyzing its performance in combination with multi-object trackers. Our main contributions are as follows:

- We propose an approach to approximate distance estimation in USVs using supervised object detection, a cheap and simple method for gauging distances to other objects without adding any runtime requirements.
- We open-source a comprehensive maritime dataset of manually annotated images with bounding box data and distances alongside a labeling tool to associate objects in image space with their counterparts in chart data space, enabling effective training of a machine learning model for distance estimation.

5.2 Related Work

Research in the intersection of computer vision and USVs has seen progress in areas, such as horizon estimation (Hashmani *et al.*, 2020; Singhal *et al.*, 2023; Kiefer and Zell, 2023), semantic segmentation (Bovcon *et al.*, 2019; Bovcon and Kristan, 2021), panoptic segmentation (Žust *et al.*, 2023; Nirgudkar *et al.*, 2023), anomaly/obstacle detection (Prasad *et al.*, 2018), heading estimation (Kiefer *et al.*, 2023f; CaptainAI, 2022), and more. Please see the survey or workshop papers on the current progress of maritime com-

puter vision (Kiefer *et al.*, 2023c, 2024; Prasad *et al.*, 2017; Lyu *et al.*, 2022; Steiniger *et al.*, 2022; Saleh *et al.*, 2022).

The field of object detection has matured within the last years from CNN-based two-stage detectors (Ren *et al.*, 2015a; Carranza-García *et al.*, 2020; Zou *et al.*, 2023), one-stage detectors (Ouchra and Belangour, 2021; Diwan *et al.*, 2023; Vijayakumar and Vairavasundaram, 2024), up to transformer-based detectors (Arkin *et al.*, 2023; Li *et al.*, 2023; Amjoud and Amrouch, 2023). While there are attempts on making transformer-based detectors embedded-friendly (Chen *et al.*, 2022; Ma *et al.*, 2022; Mehta and Rastegari, 2021), the default choice for embedded detectors are CNN-based one-stage detectors, such as the YOLO series (Zhou *et al.*, 2024). In our experiments, we mostly rely on these as they are the most prevalent in this domain. In the maritime domain, object detection has often been connected to obstacle detection for downstream obstacle avoidance (Jiang *et al.*, 2024; Jin *et al.*, 2024; Kiefer *et al.*, 2023c). We refer to Kiefer *et al.* (2024) for a discussion on challenges in maritime object detection.

Since many vessels don't employ an automatic identification system (AIS) and it also only transponds signals slowly, the need of distance estimation through other means is inevitable (Qu *et al.*, 2023). There are many radar-based approaches, focusing on reducing the noise or classifying radar blobs (Li *et al.*, 2019; del Rey-Maestre *et al.*, 2018; Almeida *et al.*, 2009). However, radar suffers from the usual challenges of reflectivity and lack of interpretability (Robinette *et al.*, 2019). LiDAR-based approaches are investigated as well, but lack the resolution for wider distances (Ferreira *et al.*, 2022; Zha and Huang, 2022; Han *et al.*, 2017). Both sensor types are considerably more expensive than cameras.

Hence, vision-based solutions are in the main focus of current USV-based computer vision research. Traditional approaches are triangulation-based, e.g. stereo-vision approaches were investigated in Sinisterra *et al.* (2014); Kaplowitz (2022); Sinisterra *et al.* (2017), where depth information is obtained by comparing the disparity between two horizontally displaced cameras. This method provides accurate distance estimation for objects at varying ranges, but the accuracy diminishes with increasing distance due to reduced disparity. E.g., experiments on KITTI show a poor performance for distances beyond $\sim 100\text{m}$ (Pinggera *et al.*, 2014). Also, the maritime domain has mostly poor features, often only showing moving water and sky. Furthermore, stereo cameras are prone to misconfiguration and very sensitive to calibration.

Traditional monocular approaches rely on geometric triangulation based on knowing the ego-pose and camera intrinsics (Mousazadeh *et al.*, 2018; Kiefer *et al.*, 2023f). However, these approaches are ill-posed with small variations in self-orientation estimation leading to large variations in pixel space (Ali *et al.*, 2020). Mansour *et al.* (2019); Liu *et al.* (2022a) show that parallax-based approaches can outperform stereo vision for greater distances, but it remains inaccurate for larger distances.

Lately, fully monocular depth estimation approaches by means of end to end neural networks have been investigated as a means of gauging distances (Ming *et al.*, 2021; Bhoi, 2019; Yang *et al.*, 2025). While promising, these approaches rely on large amounts

of accurate depth maps, either from stereo cameras, lidar, or synthetic data. Furthermore, they tend to work in indoor scenes with limited distances only (Roussel *et al.*, 2019).

Contrary to these approaches, we focus on approximate object-based supervised distance estimation (Davydov *et al.*, 2022). The maritime domain has unique challenges and our approach has the benefit to work with larger distances, can be trained fully end to end without any considerable overhead and is simple to integrate into existing object detection architectures. Please see Section 5.5 for an analysis and comparison of different methods.

Lastly, to facilitate this research direction, we collected and annotated a large dataset with bounding box labels accompanied with distances. Current maritime CV datasets focus on object detection, multi-object tracking, semantic segmentation (Su *et al.*, 2023) or focus on other sensor modalities (De Wind *et al.*, 2010). However, CV-based distance estimation is a relatively young domain without any publicly available datasets.

5.3 Methodology

5.3.1 Model Architecture

We employ the YOLOv7 and YOLOv9 series of object detectors for our task. These detectors are known for being on the Pareto front of accuracy and real-time performance (Wang *et al.*, 2024). To adapt these models for distance estimation, we modify the architecture to predict an additional output for distance as part of each anchor as shown in Figure 5.2.

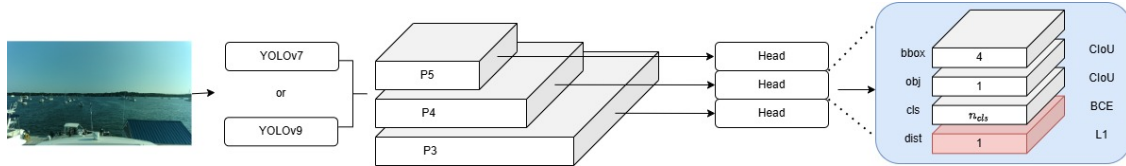


Figure 5.2: Architecture of our proposed approach at the example of YOLOv7 and YOLOv9. We leave the base architecture of the networks the same except for adding a distance loss branch to the heads (in red).

The adjusted model architecture integrates distance prediction by including an extra output neuron for each anchor. This additional output allows the model to predict the distance to the detected object directly.

However, these networks predict numbers that are "well-behaved", in that they are mostly centered around zero and are of low magnitude (Wang *et al.*, 2024). Directly predicting the metric distances would result in unstable training or very poor performance. Hence, we experimented with various normalization strategies for the distance prediction branch, including linear scaling, logarithmic scaling, and hybrid approaches to handle different ranges and distributions of distances.

5.3.2 Distance Normalization Strategies

We explored four main normalization strategies for the distance prediction:

Linear Normalization

The linear normalization scales the actual distance d to a normalized value y in the range $[0, 1]$:

$$y = \frac{d}{d_{\max}} \quad (5.1)$$

During inference, we recover the predicted distance $\hat{d}$ from the network's output y :

$$\hat{d} = y \times d_{\max} \quad (5.2)$$

Logarithmic Normalization

For logarithmic normalization, we apply a logarithmic transformation and scale to $[0, 1]$:

$$y = \frac{\log(d + \varepsilon)}{\log(d_{\max} + \varepsilon)} \quad (5.3)$$

Here, ε is a small positive constant to avoid taking the logarithm of zero (e.g., $\varepsilon = 1$).
During inference:

$$\hat{d} = \exp(y \times \log(d_{\max} + \varepsilon)) - \varepsilon \quad (5.4)$$

Linear Negative Normalization

To map distances to the range $[-1, 1]$:

$$y = 2 \left(\frac{d}{d_{\max}} \right) - 1 \quad (5.5)$$

During inference:

$$\hat{d} = \left(\frac{y + 1}{2} \right) \times d_{\max} \quad (5.6)$$

Logarithmic Negative Normalization

Combining logarithmic scaling with mapping to $[-1, 1]$:

$$y = 2 \left(\frac{\log(d + \varepsilon)}{\log(d_{\max} + \varepsilon)} \right) - 1 \quad (5.7)$$

During inference:

$$\hat{d} = \exp \left(\left(\frac{y+1}{2} \right) \times \log(d_{\max} + \varepsilon) \right) - \varepsilon \quad (5.8)$$

5.3.3 Training and Inference

By applying these normalization strategies, we ensure that the network's distance predictions y are within ranges suitable for neural network outputs. During training, the loss is computed between the normalized predicted distances and the normalized ground truth distances. During inference, the inverse transformations are applied to map the network's outputs back to actual distance values $\hat{d}$.

5.3.4 Model Training

The model is trained using a composite loss function that balances several components:

- **Objectness Loss:** Penalizes incorrect predictions of object presence.
- **Classification Loss:** Ensures accurate classification of detected objects.
- **Localization Loss:** Measures the accuracy of bounding box predictions.
- **Distance Loss:** Evaluates the accuracy of the distance predictions.

We leave the loss functions for the first three losses unchanged, and use L_1 as the distance loss.

5.3.5 Dataset Collection and Annotation

Our evaluation dataset includes images captured from USVs, manually annotated with bounding boxes and distances. We captured the images using rented boats on the east coast of the US. For ground truth evaluation (only for that - for our proposed method, we use manually labelled data), we determined distances to static objects such as buoys and docked boats using the USV's GPS position and heading and verified manually with known chart positions. For the chart data, we manually downloaded and fused buoy data for the US from NOAA.

Our evaluation dataset consists of 1,000 images captured from USVs in various maritime environments, including open waters, harbors, and coastal regions, see example scenes in Fig. 5.3. Each image is annotated with bounding boxes around objects such as boats, buoys, and other obstacles, along with their respective distances from the USV. Distances are measured using GPS positioning for static objects. This comprehensive dataset ensures robust training and evaluation of our model.



Figure 5.3: Random samples from our collected dataset for object detection and distance estimation to illustrate the scenes.

We created our own labeling tool, which combines all these components: bounding box labeling, accompanying chart data integration, and association labelling. See Figure 5.4 for a screenshot from its UI. We’re making it publicly available. Also see Figure 5.5 for a distance histogram. Furthermore, see Fig. 5.6 to illustrate the non-trivial relationship between image size and distance.

5.3.6 Human Labeling for More Data

To avoid the need for metadata alongside the image data, we also explored the approach of having human annotators gauge distances to objects. For this, we asked a professional labeling service company to label bounding boxes and alongside of it distances. With this approach, we obtained another 5,000 images with pseudo-distances. We’ll explore in the experiment section how useful this data is.

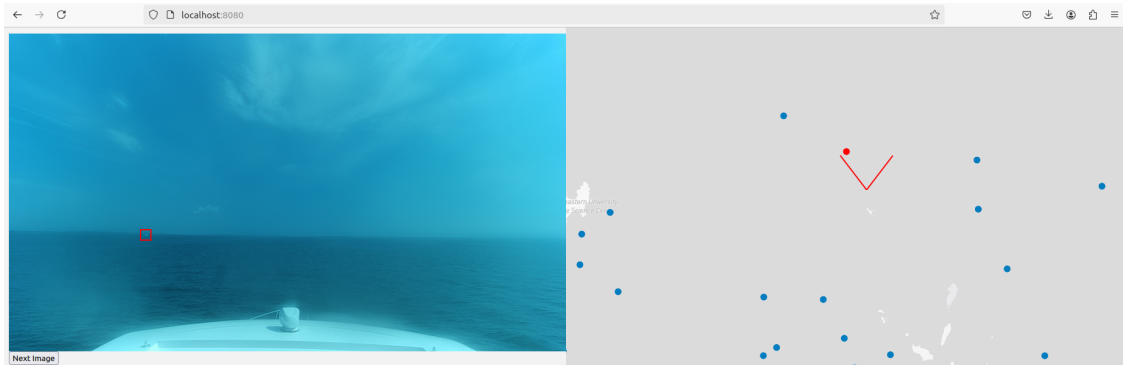


Figure 5.4: Labeling tool used for getting bounding boxes and associating chart data with vision data.

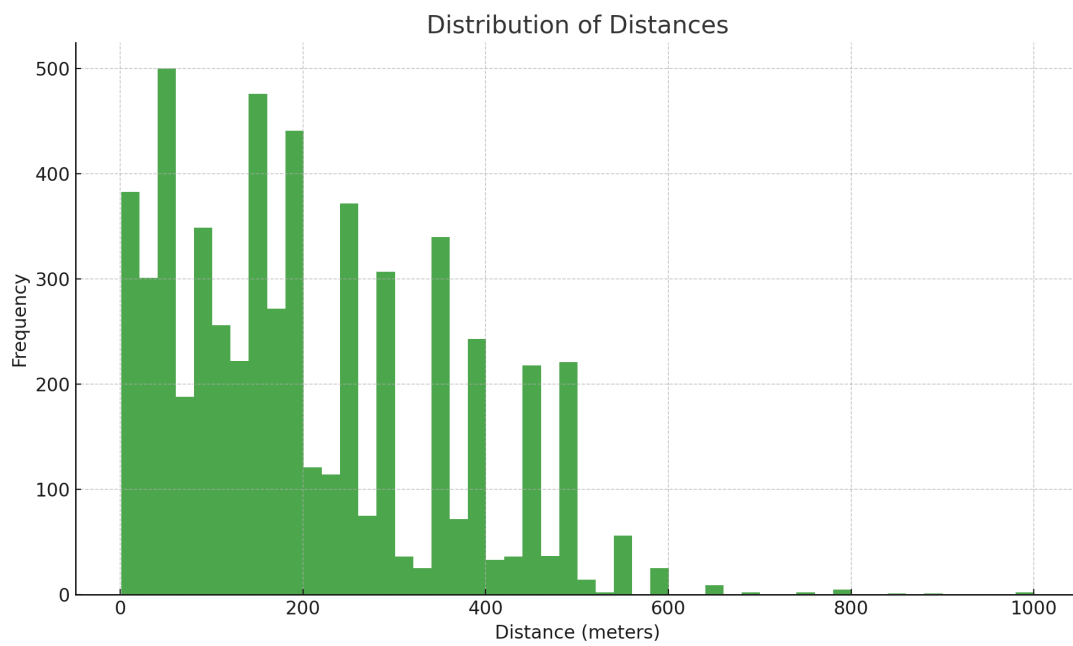


Figure 5.5: Distribution of distances in our dataset.

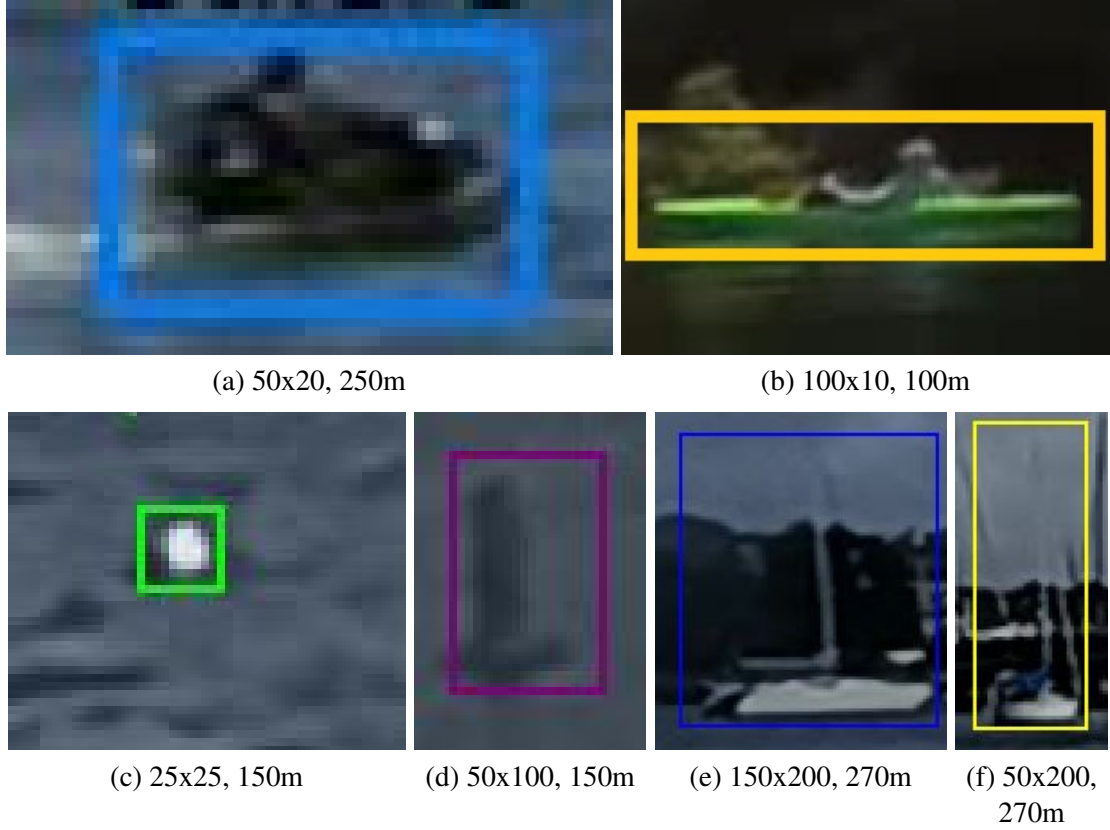


Figure 5.6: Despite the same image space area of (a) and (b), the distances vary significantly due to different types of boats. Vice versa, same-distance buoys (c) and (d) occupy a different image area. This effect can also happen due to viewpoint changes, such as in (e) and (f).

5.4 Evaluation Metrics

To evaluate the performance of our object detection and distance estimation model, we use two primary metrics: mean Average Precision (mAP) and a novel weighted distance error metric. We calculate mean Average Precision (mAP) using an Intersection over Union (IoU) threshold of 0.5 to determine correct detections.

5.4.1 Weighted Distance Error Metric

The weighted distance error metric evaluates the accuracy of the distance predictions while considering the confidence of each detection. For each detected object i , let d_i be the ground truth distance, $\hat{d}_i$ be the predicted distance, and c_i be the confidence score of the detection. The distance error e_i is given by $e_i = |d_i - \hat{d}_i|$.

To weight the distance error by the confidence, we define the weighted distance error

E as:

$$E = \frac{\sum_{i=1}^M c_i \cdot e_i}{\sum_{i=1}^M c_i}$$

where M is the total number of detections. This metric ensures that higher confidence detections have a greater impact on the overall error, reflecting their presumed accuracy.

Similar to the mAP calculation, we only consider distance errors for detections whose IoU with the ground truth exceeds a predefined threshold. This ensures that only sufficiently accurate object detections contribute to the weighted distance error metric.

5.5 Experiments

In this section, we evaluate the performance of our proposed approach for approximate distance estimation on unmanned surface vehicles (USVs). We conduct experiments to assess both object detection accuracy and distance estimation precision. The evaluation includes comparisons between different model architectures, analysis of distance estimation errors across distance intervals and object types, and benchmarking against other distance estimation methods. All experiments are performed using our maritime dataset described in Section 5.3.

5.5.1 Object Detection Performance

We first evaluate the object detection performance of our modified YOLO models to assess the impact of the proposed changes. Table 5.1 presents the precision, recall, mean Average Precision at IoU threshold 0.5 (mAP@0.5), mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95), and inference speed measured in frames per second (FPS). The models are tested on an NVIDIA Orin AGX using the DeepStream toolkit, with speed measured as wall-clock time from camera input to display output.

As shown in Table 5.1, the larger models (YOLOv7 and YOLOv9-M) achieve higher detection accuracy, with YOLOv9-M attaining the highest mAP@0.5 of 81%. The tiny models offer faster inference speeds, with YOLOv9-Tiny reaching up to 72.8 FPS, making them suitable for real-time applications where computational resources are limited.

Please see Figure 5.1 for a qualitative result of YOLOv7 with integrated distance estimation.

5.5.2 Distance Estimation Accuracy

To assess the distance estimation performance, we analyze the weighted distance error across different distance intervals and object types. Before selecting the normalization strategy for the distance prediction, we conducted preliminary tests with the different

Table 5.1: Object detection performance and speed as measured on an NVIDIA Orin AGX, using the DeepStream toolkit. Speed is in wall-clock time from camera to display.

Metric	Pr	Re	mAP@0.5	mAP@0.5:0.95	FPS
YOLOv7-T	0.82	0.68	0.73	0.45	56.5
YOLOv7	0.85	0.72	0.78	0.50	45.2
YOLOv9-T	0.83	0.70	0.76	0.48	72.8
YOLOv9-M	0.87	0.74	0.81	0.53	48.8

normalization methods described in Section 5.3, including linear normalization, logarithmic normalization, linear negative normalization, and logarithmic negative normalization. We found that only the basic linear normalization yielded satisfactory results. The other methods either did not converge during training or resulted in unstable predictions. Therefore, we proceeded with the basic linear normalization in our experiments.

Table 5.2 presents the weighted distance errors (in meters) for boats and buoys over distance ranges from 0 to 500 meters (clipped), using different model architectures.

Table 5.2: Weighted distance error by distance interval (in meters) and object type (**boats** and **buoys**) for different models.

Distance	YOLOv7-T		YOLOv7		YOLOv9-T		YOLOv9-M	
	Bt	By	Bt	By	Bt	By	Bt	By
0-100	19.1	16.0	18.2	15.1	20.4	17.0	18.7	15.8
100-200	39.6	33.3	37.8	32.0	41.1	34.9	38.5	33.1
200-300	47.7	55.5	45.9	53.6	50.2	57.0	48.1	54.8
300-400	63.4	91.1	61.2	89.1	65.7	92.6	62.8	90.4
400-500	76.7	131.2	74.6	129.0	78.4	133.6	76.0	130.9
Avg.	49.3	65.4	47.5	63.8	51.2	67.0	48.8	65.0

The results indicate that the distance estimation error increases with the distance to the object, which is expected due to reduced visual cues at greater distances. Our models consistently perform better on boats than buoys, likely because boats are larger and have more distinct features. Among the models, YOLOv7 and YOLOv9-M achieve lower average distance errors, demonstrating the effectiveness of our approach in accurately estimating distances.

5.5.3 Comparison with Other Methods

We compare our proposed method with traditional triangulation-based distance estimation (based on orientation via pitch and roll) and state-of-the-art monocular depth estimation models (dense distance values are projected into bounding box by taking the median

Table 5.3: Comparison of Distance Estimation Methods Based on Ground Truth Data for USVs.

Method	MDE (m)	Outl. (%)	FPS
YOLOv7 + Triangulation	35.2	40.0	45.2
Depth Anything V2 (Small)	43.5	50.0	10
Depth Anything V2 (Base)	27.1	37.5	2
Depth Anything V2 (Large)	19.2	25.0	0.5
Proposed YOLOv7	14.9	12.0	45.2

distance within a box). Table 5.3 summarizes the mean distance error (MDE) in meters, the percentage of outliers, and the inference speed in FPS for each method.

Our proposed method achieves the lowest mean distance error and the smallest percentage of outliers while maintaining real-time performance. Traditional triangulation methods and monocular depth estimation models either have higher errors or are computationally intensive, making them less suitable for real-time USV applications. Depth Anything v2 struggles with distances beyond approx. 60m as it is not trained to do so and there is no dataset that can be used to train this model.

5.5.4 Robustness Towards Camera Extrinsics

We assess the sensitivity of distance estimation to camera extrinsics by simulating small roll and pitch variations. Roll is simulated by rotating the image ($\pm 5^\circ$) and pitch by shifting it vertically ($\pm 20\text{px}$). Table 5.4 shows the mean distance error (MDE) for a trigonometric approach versus our proposed method. The trigonometric approach suffers notably from these perturbations, while the learning-based approaches are more robust.

Table 5.4: Impact of Roll and Pitch Variations on Distance Estimation (MDE in m).

Transformation	Trigonometric	Depth Anything	Proposed
Baseline	35.2	19.2	14.9
Roll $\pm 5^\circ$	46.6	18.7	15.2
Pitch $\pm 20\text{px}$	63.1	21.0	15.5
Roll + Pitch	67.2	20.4	15.3

5.5.5 Loss Balancing and Ablation Studies

We conducted ablation studies to understand the impact of the distance loss gain on the model’s performance for both object detection (OD) and distance estimation. As shown in Table 5.5, the distance loss gain significantly affects the balance between OD accuracy and distance estimation precision.

Table 5.5: Ablation Studies on Distance Loss Gain Settings (Other loss components are fixed: Box Loss = 0.05, Classification Loss = 0.3, Objectness Loss = 0.7)

Distance Loss Gain (distance)	0.001	0.01	0.1
mAP@0.5 (OD)	89.5%	86.4%	72.1%
MAE (m) (Distance)	19.7	14.2	9.8

Table 5.6: Impact of Tracking and Smoothing on Distance Estimation Performance for YOLOv9-M.

Method	MDE (m)	Outl. (%)	FPS
Without Tracking	18.4	22.5	48
With Tracking (SORT + Avg.)	14.9	12.0	45

The results indicate that increasing the distance loss gain improves distance estimation accuracy (lower MAE) but significantly reduces OD performance (lower mAP). A lower distance loss gain (e.g., 0.001) maintains higher OD accuracy but at the cost of less precise distance predictions. Balancing these loss components is crucial for optimizing both OD and distance estimation tasks.

5.5.6 Smoothing the Distance Estimates

To handle distance estimation in video sequences, where objects move and change positions over time, we use the Simple Online and Realtime Tracking (SORT) algorithm (Bewley *et al.*, 2016). SORT tracks objects across frames by combining object detection results with a Kalman filter and the Hungarian algorithm for data association, ensuring that each detected object maintains a consistent identity across frames.

Alongside tracking, we keep a running average of the distance estimates for each tracked object. This simple averaging technique helps to smooth the distance predictions over time, filtering out outliers caused by sudden detection errors or rapid changes in object appearance. The approach is computationally efficient and provides stable distance estimates, which are crucial for downstream tasks like collision avoidance in dynamic maritime environments.

By combining SORT with a running average for distance smoothing, we achieve robust and real-time performance on Unmanned Surface Vehicles (USVs), ensuring that distance predictions are both accurate and temporally consistent. Table 5.6 shows how tracking and smoothing reduces the Mean Distance Error (MDE) and the percentage of outliers.

Table 5.7: Distance Estimation Errors for Human-Labeled Ground Truth Compared to Chart-Based Ground Truth by Object Type and Distance Range.

Dist Range	Boats		Buoys	
	MAPE (%)	MAE (m)	MAPE (%)	MAE (m)
0-100	45.2	17.8	52.7	20.4
100-300	64.9	48.3	70.3	56.1
300+	84.1	98.2	89.2	115.7

5.5.7 Evaluation of Human-Labeled Distances

To assess the accuracy of human-labeled distances compared to chart-derived ground truth data, we examine human estimations of distances to boats and buoys in maritime settings. The errors in human distance estimation generally increase with the distance to the object.

We categorized the ground-truth data, which has been labeled manually as well, into three ranges: Close (0-100 m), Medium (100-300 m), and Far (300+ m). For each range, we computed the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) to quantify deviations from chart data. The results are summarized in Table 5.7.

Table 5.7 shows human-labeled distance estimates are more accurate at closer ranges, with lower MAE and MAPE. However, accuracy decreases for medium and far ranges. For example, at distances over 300 m, the MAE for boats and buoys exceeds 98 m and 115 m, respectively, with MAPE approaching 90%. Notably, for various use-cases it is not important to get a precise distance estimate if an object is far away.

5.6 Conclusion and Outlook

We introduced a supervised approach for approximate distance estimation on USVs, extending an object detection model with a regression branch to jointly predict object presence and distance. The method achieves real-time performance while providing a cost-effective alternative to conventional range sensors, making it particularly suited for maritime platforms with strict resource constraints.

Nevertheless, several limitations must be acknowledged. A key limitation is the sensitivity of the model to camera configuration, including field of view (FoV), focal length, zoom, and mounting position. Since the network learns to associate apparent object size and visual cues with distance under a specific setup, deployment with different FoVs or zoom levels can lead to inaccurate predictions. For example, a narrower FoV may make distant objects appear larger, resulting in systematically underestimated distances. The approach therefore lacks intrinsic invariance to such variations. Additionally, the evaluation was conducted under a constrained field of view and in a relatively uniform environment dominated by a flat water surface, which reduces the generality of the re-

sults.

These limitations suggest directions for future work. One path is to incorporate explicit camera parameters or calibration data into the learning process to improve robustness across setups. Another is to extend the evaluation to more diverse imaging conditions. The next chapter pursues the latter direction by generalizing distance estimation from rectilinear maritime imagery to omnidirectional 360° vision, addressing new challenges such as projection distortion, spatial discontinuities, and the variability of more complex operational domains.

Chapter 6

Learning-Based Distance Estimation for 360° Single-Sensor Setups

The preceding chapter demonstrated that object distance can be effectively learned from monocular rectilinear imagery. However, the evaluation was limited to a constrained field of view and a relatively unique environment, namely scenes dominated by a flat water surface. This chapter extends the scope to omnidirectional 360° vision, where challenges arise from extreme projection distortion and spatial discontinuities. In addition, increased scene variability arising from diverse application scenarios becomes an important factor to consider. By evaluating the approach on multiple datasets representing diverse domains, this chapter demonstrates its applicability to a broader spectrum of mobile perception scenarios, ranging from structured urban environments to unstructured maritime and open-field settings.

Traditional distance estimation methods in omnidirectional vision often rely on geometric calibration and trigonometric inference. While theoretically precise, these approaches are highly sensitive to lens parameters and prone to failure under real-world variability. To overcome these limitations, this chapter extends the architecture of the previous chapter to spherical imagery, enabling direct distance inference from raw 360° fisheye inputs without explicit calibration.

The approach is systematically evaluated on three 360° datasets: LOAF, ULM360, and BOAT360, covering distinct environments and sensor setups. Experimental results show that the proposed learning-based model outperforms both geometry-based baselines and alternative learning methods in accuracy and robustness. These findings establish that omnidirectional distance estimation can be achieved reliably with a lightweight framework suitable for robotics, autonomous navigation, and surveillance.

6.1 Introduction

Accurate distance estimation is a fundamental requirement for a wide variety of perception tasks, including autonomous vehicle navigation, robotic manipulation, and surveillance. In particular, accurately determining how far away an object lies becomes essential for safe decision-making and interaction in dynamic environments. Although stereo or

multi-camera setups and LiDAR systems have been adopted in many industrial applications, their high cost and complexity can be prohibitive in resource-constrained scenarios. Monocular depth estimation with a single pinhole camera is more affordable but suffers from a limited field of view (FOV). By contrast, using a single 360° fisheye camera offers a more affordable, lightweight, and comprehensive way to capture the surrounding scene (Yang *et al.*, 2023b; Quan *et al.*, 2024).

However, the extreme distortion inherent in fisheye lenses poses unique challenges for distance estimation. Classical methods typically rely on carefully calibrated lens models and trigonometric formulas to project 2D image coordinates into 3D space (Quan *et al.*, 2024). These geometry-based approaches perform well under stable camera intrinsics and extrinsics, lighting, minimal occlusion, and uniform lens characteristics. Yet, any deviation from assumed calibration conditions, such as change in camera extrinsics or intrinsics, unmodeled distortions or dynamic lighting, can yield substantial errors in practice (Genovese, 2024). Moreover, adopting these methods across different camera models, scene types, and environmental conditions often requires extensive recalibration.

Conventional trigonometric and polynomial-based lens models offer a mathematically elegant solution for fisheye image projection. Nevertheless, their accuracy relies on the completeness of the lens distortion parameters and the static nature of the environment. For instance, once the height of the camera or the angle of the lens changes, the precomputed calibration parameters may no longer be valid. Similarly, the presence of reflective surfaces, moving platforms, or occlusions can significantly degrade the performance of purely geometric methods. Furthermore, these approaches often treat each image frame independently, lacking the capacity to learn and refine distance estimations from large, diverse datasets.

Advances in deep learning have catalyzed a shift toward data-driven solutions for visual perception tasks, including monocular depth estimation in regular perspective images. Without explicit geometric formulas, neural networks learn mappings between pixel intensities and depth values from training data alone. Although research on omnidirectional depth estimation is less extensive than on standard pinhole cameras, the potential for a learning-based system to adapt to fisheye distortions and dynamic environments is immense. Using convolutional architectures, these methods can capture high-level context and local texture information to predict distances more robustly than purely trigonometric models.

In summary, our key contributions are as follows:

- **A lightweight yet highly effective learning-based network** for real-time object distance estimation in omnidirectional images.
- **Extensive empirical validation**, demonstrating the superiority of our approach over both geometric and deep learning baselines through testing on multiple datasets, including real-world deployment scenarios.
- **Benchmark dataset contribution**, providing high-quality, annotated datasets for

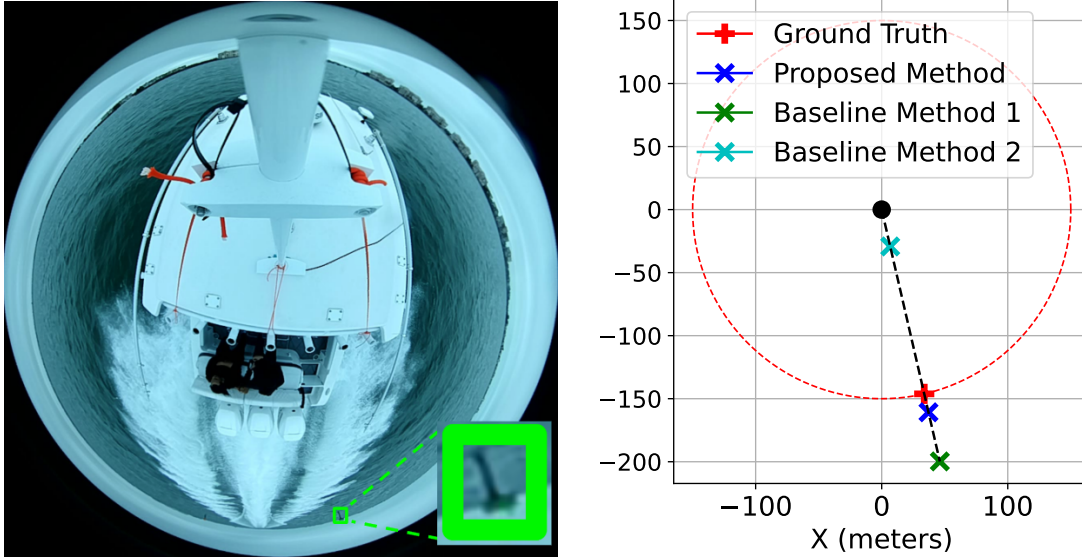


Figure 6.1: As the boat accelerates, our learning-based method estimates 165m, closer to the ground truth (150m) than the geometry method (205m) and DepthAnythingV2 (Yang *et al.*, 2025) (30m), showing robustness to camera pose changes in dynamic environments.

omnidirectional distance estimation to facilitate future research upon paper acceptance.

6.2 Related Work

Distance estimation is a fundamental aspect of numerous perception tasks, including robotic manipulation, autonomous vehicle navigation, and surveillance. Traditional stereo vision techniques, such as Semi-Global Matching (Hirschmuller, 2007) and deep learning-based methods like PSMNet (Chang and Chen, 2018), provide dense depth estimation but require well-calibrated stereo pairs and suffer in textureless or occluded regions. On the other hand, LiDAR-based methods (Chen *et al.*, 2017) offer highly accurate depth measurements and are widely used in autonomous driving, but they are expensive and resource intensive.

Monocular 360° fisheye cameras present an affordable and comprehensive solution for capturing full environmental context, but the distortions introduced by such optics pose unique challenges for depth estimation. Traditional geometry-based techniques rely on calibrated lens models and trigonometric formulas to project 2D image coordinates into 3D space (Yang *et al.*, 2023b; Quan *et al.*, 2024). However, the reliance on precise lens calibration makes them less adaptable to diverse environmental conditions and real-world distortions, as highlighted by Son *et al.* (2023).

Advances in deep learning have enabled data-driven approaches to monocular depth

estimation, which learn mappings between pixel intensities and depth values directly from training data. Eigen et al. introduced a multi-scale deep network for depth map prediction, demonstrating the potential of learning-based methods to extract meaningful depth information from single images (Eigen *et al.*, 2014). These techniques have since evolved to incorporate self-supervised learning, as seen in Godard et al.’s work, which introduces a left-right consistency loss to improve monocular depth estimation (Godard *et al.*, 2017).

Adapting learning-based methods to omnidirectional images presents additional challenges due to fisheye distortions and non-uniform image projections. Rey-Area et al. proposed 360MonoDepth (Rey-Area *et al.*, 2022), which projects 360° images onto tangent planes to create perspective views that can be processed using existing depth estimation models. The individual depth predictions are then merged using deformable multi-scale alignment and gradient-domain blending to construct a dense, high-resolution 360° depth map. Similarly, Yan et al. introduced FisheyeDistill (Yan *et al.*, 2022), which employs an ordinal distillation loss to transfer depth information from a teacher model, enhancing performance on fisheye images. Li et al. developed OmniFusion (Li *et al.*, 2022), a method that transforms omnidirectional images into less-distorted perspective patches and integrates 3D geometric priors with 2D image features using a transformer-based architecture. Kumar et al. proposed UnRectDepthNet (Kumar *et al.*, 2020), a self-supervised framework that eliminates the need for image rectification, preserving the full field of view while reducing computational costs during inference. Zhang et al. proposed Depth-Anything-V2 (Yang *et al.*, 2025), a large-scale general-purpose depth estimation model designed to handle diverse scene variations. Yet, its reliance on extensive supervised training makes it unsuitable for scenarios where pixel-wise depth annotations are unavailable. Furthermore, its high computational cost limits its feasibility for real-time deployment in autonomous navigation.

Applications of these methods in robotics and autonomous systems have demonstrated substantial benefits, particularly in object geolocalization and obstacle detection. Syn-DistNet (Kumar *et al.*, 2021b) integrates monocular depth estimation with semantic segmentation to enhance depth discontinuities and improve object localization accuracy. This synergy is critical for autonomous navigation and robotic perception, where accurate distance estimation directly influences decision-making and real-time obstacle avoidance.

Our approach differs from existing methods by emphasizing object-based depth estimation, enabling real-time performance suitable for deployment in resource-constrained environments. By optimizing inference efficiency, we achieve accurate depth predictions while maintaining computational feasibility, making our system particularly well-suited for robotics and surveillance applications.

While traditional geometry-based methods retain their relevance for controlled environments and well-calibrated setups, learning-based approaches offer superior adaptability and robustness, particularly for real-world scenarios where lens distortions and environmental variations cannot be easily corrected. The discussed neural network-based

methods highlight the potential of deep learning in advancing monocular distance estimation for omnidirectional imagery, making them strong candidates for deployment in robotic perception, surveillance, and autonomous navigation applications.

6.3 Methodology

6.3.1 Datasets

We use three datasets, all featuring large-FOV ($> 180^\circ$) downward-facing fisheye cameras but collected under different conditions to test the generalization and robustness of our approach, as summarized in Table 6.1.

Table 6.1: Dataset Characteristics

Dataset	Platform	Camera Mounting	Environment
LOAF	Static	Fixed height (varied per scene)	Indoor & Outdoor
ULM360	Moving vehicle	Mounted on a pole	Urban
BOAT360	Moving boat	Mounted on a mast	Open water

Dataset	Distance GT	Image Resolution
LOAF	Manual with reference ruler	512×512
ULM360	LiDAR-based annotation	1920×1920
BOAT360	Chart Data	1452×1452

- **LOAF (public dataset):** Captured from a static camera setup at various heights. Distance ground truth is established using a reference ruler, with one end positioned directly beneath the fisheye camera to serve as a calibration reference. Each annotated person in the dataset includes a bounding box along with the manually measured distance information (Yang *et al.*, 2023b).
- **ULM360 (our dataset):** Collected in Ulm, Germany, using a mobile vehicle equipped with a downward-facing fisheye camera mounted on a pole. To provide precise ground-truth distances, the vehicle was equipped with three Ouster LiDARs, which allowed us to map each manually labeled bounding box to its corresponding point cloud distance (Quan *et al.*, 2024).
- **BOAT360 (our dataset):** Collected from a moving boat, with the fisheye camera mounted on a mast. Unlike the previous datasets, where distances were measured using structured references (rulers or LiDARs), object distances in BOAT360 were estimated by experienced sailors. This dataset presents additional challenges due to boat dynamics, lack of static reference points, and dynamic environmental conditions.

Each dataset has different size and resolution, providing an opportunity to evaluate the generalization ability of our method across diverse real-world conditions. Additionally, we generate equirectangular representations of both the original captured images and their corresponding annotated bounding boxes. This ensures compatibility with alternative image processing techniques and allows us to evaluate the generalization capabilities of our method. Representative samples of the datasets and their equirectangular conversions with annotations are shown in Figure 6.2, highlighting the diversity of imaging conditions considered in our evaluation.

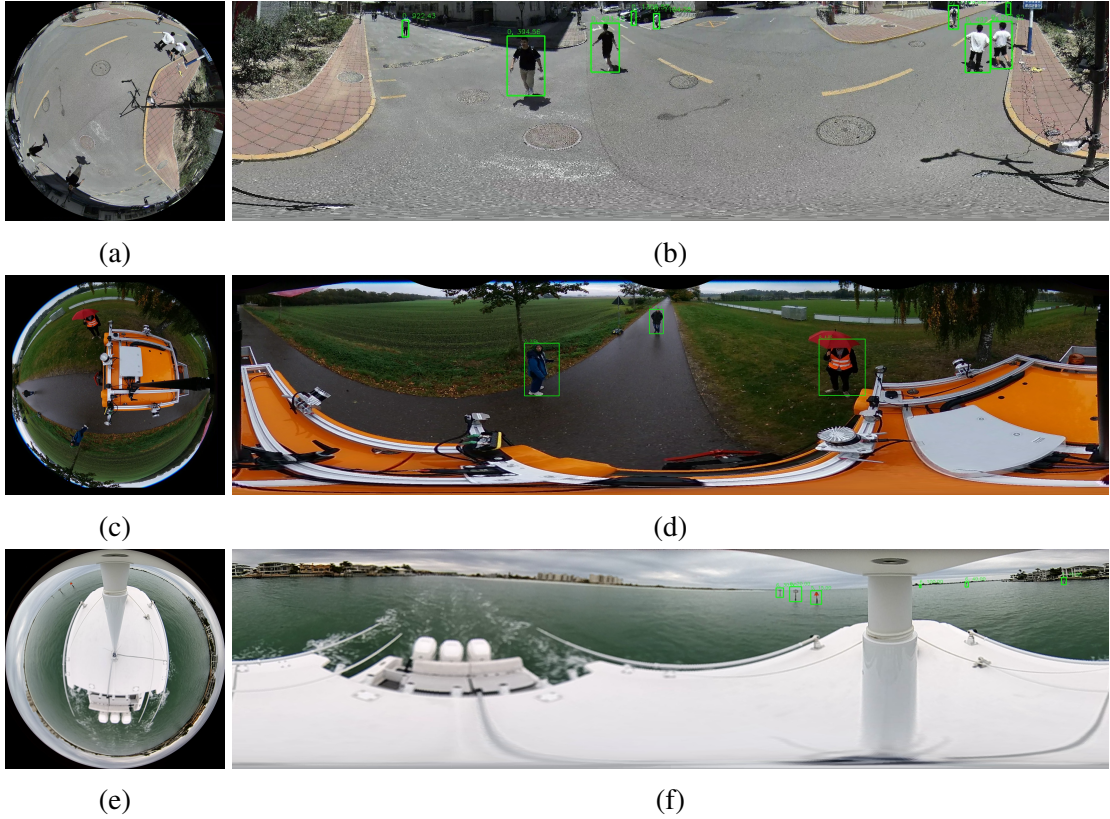


Figure 6.2: (a, c, e) shows samples of raw images from dataset LOAF, ULM360, BOAT360. Their equirectangular-transformed images and the corresponding annotations are shown as (b, d, f).

6.3.2 Model Architecture

For our task, we utilize the YOLOv7 object detection architectures, which are recognized for their very good balance between accuracy and real-time performance, positioning them on the Pareto front of efficiency (Wang *et al.*, 2023). To enable these models to estimate object distances, we extend their architecture by incorporating an additional

output neuron per anchor, responsible for predicting the distance, as illustrated in Figure 6.3.

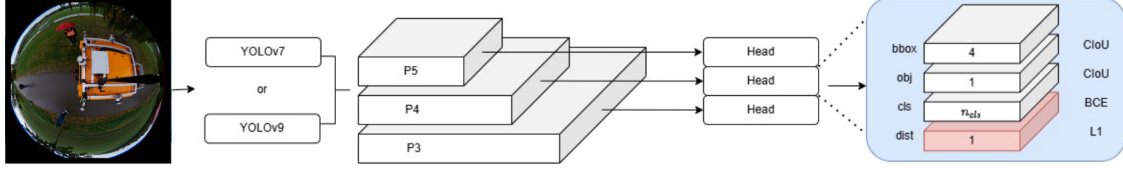


Figure 6.3: Architecture of our proposed approach, illustrated using YOLO models. We integrate a dedicated distance estimation branch (red) with tailored feature transformations and loss formulations, enabling simultaneous object detection and accurate depth estimation while maintaining real-time performance.

6.3.3 Distance Normalization Strategies

Integrating this additional output into the detection framework allows the model to estimate object distances simultaneously with object detection. However, a direct regression of metric distances poses challenges due to the natural tendencies of these networks to predict values that are centered around zero and of relatively low magnitude (Wang *et al.*, 2023). Training instability and performance degradation can occur when predicting raw distances. To address this, we experimented with two normalization techniques for the distance estimation branch: linear scaling and logarithmic transformation tailored to different distance distributions and ranges.

Linear Normalization

In linear normalization, the actual distance d is scaled to a normalized value y within the range $[0, 1]$.

Logarithmic Normalization

A logarithmic transformation is applied for normalization to better handle variations across a wide range of distances. This allows the model to penalize errors in the estimation of closer objects more heavily compared to objects farther away. The transformation is given by:

$$y = \frac{\log(d + \varepsilon)}{\log(d_{\max} + \varepsilon)}, \quad \varepsilon \geq 1 \quad (6.1)$$

where $d_{\max}$ is a predefined maximum distance, and ε is a constant to prevent numerical instability when $d = 0$. The original distance is then recovered during inference as:

$$\hat{d} = \exp(y \cdot \log(d_{\max} + \epsilon)) - \epsilon \quad (6.2)$$

These normalization strategies ensure that distance predictions remain well-scaled for neural network outputs. During training, the model minimizes the loss between the normalized predictions and ground truth distances. In inference, the predicted values are mapped back to real-world distances using the respective inverse transformations. Our experiments revealed no significant performance differences between these strategies. Due to its simplicity, we selected linear normalization for subsequent experiments.

6.3.4 Model Training

The model is trained using a multi-component loss function that balances different objectives:

- Objectness Loss: Encourages correct detection of object presence.
- Classification Loss: Ensures accurate identification of detected objects.
- Localization Loss: Measures how precisely bounding boxes match ground truth.
- Distance Loss: Optimizes the accuracy of predicted distances.

The first three loss functions remain unchanged from the original YOLO framework, while the distance loss is computed using the L_1 norm.

The total loss function is formulated as:

$$\mathcal{L}_{total} = \lambda_{obj}\mathcal{L}_{obj} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{loc}\mathcal{L}_{loc} + \lambda_{dist}\mathcal{L}_{dist} \quad (6.3)$$

where λ_{obj} , λ_{cls} , λ_{loc} , and λ_{dist} are weighting coefficients that control the contribution of each loss term. The distance loss is defined as:

$$\mathcal{L}_{dist} = \frac{1}{N} \sum_{i=1}^N |\hat{d}_i - d_i| \quad (6.4)$$

where N is the number of detected objects, $\hat{d}_i$ is the predicted distance for object i , and d_i is the ground truth distance.

6.4 Evaluation Metrics

To assess our model's performance in both object detection and distance estimation, we use mean Average Precision (mAP) (Lin *et al.*, 2014) at different Intersection over Union

(IoU) thresholds (Hosang *et al.*, 2016) for object detection, and assess distance estimation using both absolute and confidence-weighted error metrics, which defined as following.

6.4.1 Absolute Distance Error

The absolute distance error measures the mean absolute difference between predicted and ground truth distances, quantifying the discrepancy for detected objects. It directly evaluates distance prediction accuracy but does not consider the model’s confidence in each detection.

6.4.2 Confidence-Weighted Distance Error

To incorporate the model’s confidence in its predictions, we introduce a confidence-weighted distance error metric. This metric ensures that errors from high-confidence detections contribute more significantly to the final evaluation, aligning the error measurement with the model’s certainty.

Given a set of M detected objects, for each object i , let c_i denote the confidence score assigned by the model. The confidence-weighted distance error E_w is computed as:

$$E_w = \frac{\sum_{i=1}^M c_i \cdot |d_i - \hat{d}_i|}{\sum_{i=1}^M c_i}. \quad (6.5)$$

Unlike standard absolute error metrics, this formulation down-weights the impact of low-confidence detections while emphasizing the accuracy of high-confidence predictions. As a result, it provides a more reliable assessment of distance estimation performance, particularly in real-world scenarios where model confidence is crucial for decision-making.

Following the standard practice in mAP calculations to ensure meaningful evaluation, distance errors are only considered for detections that meet a predefined IoU threshold with ground truth objects, filtering out false positives. This approach ensures both the absolute distance error and confidence-weighted distance error reflects errors for valid detections rather than spurious predictions.

6.5 Experiments and Results

This section presents an analysis of the experimental results, focusing on evaluating the proposed learning-based distance estimation approach. The key research questions addressed include:

1. Whether adding a distance estimation head affects object detection accuracy.

2. Whether our proposed learning-based model outperforms a well-calibrated geometric model and other learning-based approaches.
3. Whether our proposed models are more robust to changes in camera parameters compared to geometric methods.
4. Whether training directly on fisheye images is better than using equirectangular projections. The question arises because fisheye images preserve raw spatial distortions that might aid depth estimation, while equirectangular projections provide axis-aligned bounding boxes that could better benefit pretrained object detection models.
5. Whether incorporating another camera height estimation head improves distance prediction. This question arises because camera height is a crucial parameter in geometric distance estimation, and learning it jointly with object detection may provide additional supervisory signals that improve depth prediction.

To achieve this, we consider multiple model configurations, each designed to investigate different aspects of object detection and distance estimation. The following models are employed in our study:

- *Y7-w/oDist*: A baseline object detection model YOLOv7 without the distance estimation head. In this configuration, the distance loss weight is set to zero, meaning the network is trained solely for object detection.
- *Y7-Geo*: A baseline model that utilizes a geometric distance calculation approach based on a well-calibrated fisheye camera model. Given the pixel location of the detected bounding box predicted by YOLOv7, the model estimates distance using precomputed geometric transformations, as illustrated in Figure 6.4.
- *Y7-DAv2*: A baseline model using Depth-Anything-V2 (small) (Yang *et al.*, 2025) estimates pixel-wise depth and averages the n lowest pixel depths within each detected bounding box to determine the object's distance.
- *Y7-D*: A model that extends the baseline *Y7-w/oDist* by incorporating a dedicated distance estimation head. This model jointly optimizes both object detection and distance estimation in an end-to-end manner.
- *Y7-D-Eq*: Similar to *Y7-D*, but trained on equirectangular projections of fisheye images rather than the original fisheye format. This approach aims to assess the impact of input representation on detection and distance estimation accuracy.
- *Y7-D+C*: A model that extends *Y7-D* by adding an auxiliary head for estimating the camera height. The inclusion of camera height estimation aims to provide additional geometric constraints, potentially improving the model's depth estimation performance.

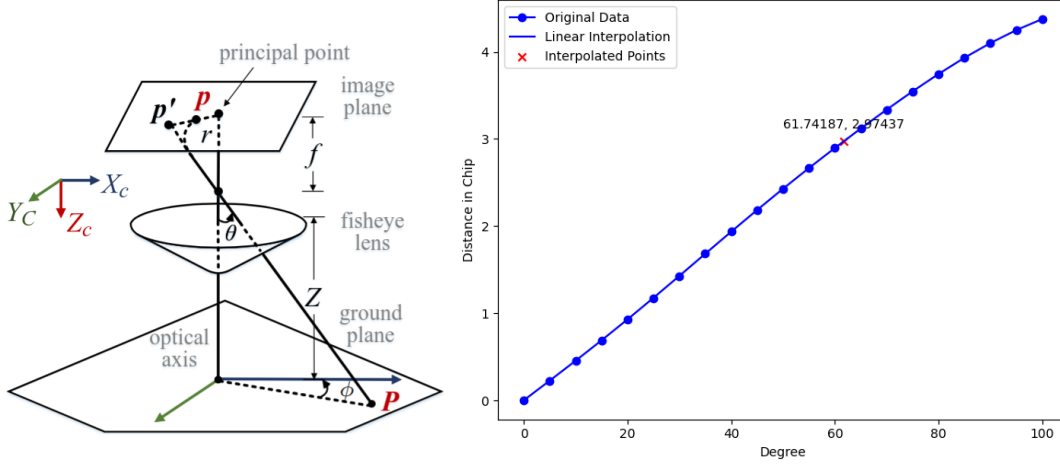


Figure 6.4: (Left) The fisheye lens model for geometric distance calculation. (Right) The interpolation between θ with respect to the optical axis and r in millimeters on the chip. (Quan *et al.*, 2024). As explained in subsection 3.3.2.

All models were trained on their respective datasets for 200 epochs. When training with fisheye images, augmentations such as rotation and flipping were applied, while transformations that could distort the fisheye structure, such as scaling, shearing, and perspective changes, were disabled. For training with equirectangular images, only flipping and color adjustments were used. In the case of *Y7-D* and *Y7-D-Eq*, the weight coefficient λ_{dist} in Equation (6.3) is empirically set to $1/3$, while the remaining $2/3$ is distributed among the other loss components preserving their default ratios from Wang *et al.* (2023). For *Y7-D+C*, λ_{dist} is empirically set to $1/6$, with an additional $\lambda_{\text{ad}} = 1/6$ assigned to the loss of height estimation of the camera. The remaining hyperparameters for model training follow the default settings in Wang *et al.* (2023).

We first evaluate our models on the LOAF dataset and subsequently analyze its generalizability across the ULM360 and BOAT360 datasets. Additionally, we evaluate the real-time inference capability of our model on an NVIDIA Orin AGX using 1024×1024 resolution images, achieving a processing speed of 45 FPS, as shown in Table 6.2, demonstrating its suitability for real-time applications.

Table 6.2: Inference speed (FPS) on 1024×1024 resolution images.

Model	RTX3090	NVIDIA Orin AGX
Y7-D / Y7-Geo / Y7-D-Eq / Y7-D+C	166	45
Y7-DAv2	39	10

6.5.1 Evaluation on LOAF Dataset

Impact of Distance Estimation Head on Object Detection

A crucial aspect of our approach is integrating a distance estimation head into an object detector. A key concern is whether this additional head negatively impacts detection performance. To analyze this, we compare *Y7-w/oDist* against *Y7-D*.

As seen in Table 6.3, adding the distance estimation head results in a slight decrease in precision but an increase in recall and an improvement in mAP@50. These results indicate that integrating a distance estimation head does not significantly degrade object detection accuracy, making it feasible to use a unified network for object detection and distance estimation.

Table 6.3: Object detection and distance estimation performance on LOAF dataset (baselines italicized). Weighted Error and Absolute Error are measured in meters.

Model	Weighted Error	Absolute Error	Precision	Recall	mAP@50	mAP@50:95
<i>Y7-w/oDist</i>	-	-	0.617	0.496	0.409	0.149
<i>Y7-Geo</i>	1.203	10.822	0.653	0.502	0.438	0.169
<i>Y7-DAv2</i>	0.858	10.950	0.653	0.502	0.438	0.169
<i>Y7-D</i>	0.070	0.874	0.653	0.502	0.438	0.169
<i>Y7-D-Eq</i>	0.210	1.609	0.799	0.601	0.631	0.274
<i>Y7-D+C</i>	0.066	0.874	0.690	0.531	0.495	0.229

Comparison to Geometric Methods and other Learning-Based Models

To evaluate the effectiveness of our learning-based approach, we compare the performance of *Y7-D* against both *Y7-Geo* and *Y7-DAv2*. From Table 6.3, the geometric model *Y7-Geo* performs significantly worse, with an absolute error of 10.747 m compared to 0.874 m for our proposed model. Additionally, the weighted error for *Y7-Geo* is 1.153, while *Y7-D* achieves a significantly lower error of 0.070. These results show that our proposed learning-based models generalize better than geometric models, which rely on precise camera calibration and extrinsic parameters.

Compared to the other learning-based baseline *Y7-DAv2*, *Y7-D* also achieves higher accuracy and efficiency. Since our dataset lacks pixel-wise depth annotations, *Y7-DAv2* could not be fine-tuned, likely leading to its suboptimal performance. However, the high computational complexity of *Y7-DAv2* makes it impractical for real-time applications, as in Table 6.2.

Robustness to Changes of Camera Parameters

According to Handel (2007), the image could drift by up to several pixels during the camera warm-up process, which correlates with temperature-induced shifts in the principal

point. To assess the robustness of our approach under such perturbations, we simulate this scenario by shifting the image along the x-axis by one and five pixels and analyzing the resulting distance estimation errors. Table 6.4 summarizes the impact of these shifts on both the geometric baseline and our proposed learning-based method.

Table 6.4: Robustness against principal point shifts. The table compares Absolute Error for Y7-Geo (baseline) and Y7-D (proposed method) on the LOAF dataset under principal point offset.

Offset	Y7-Geo Abs.Err (m)		Y7-D Abs.Err (m)	
	range 28.8 - 36.0 m	Overall	range 28.8 - 36.0 m	Overall
No offset	14.590	10.822	3.490	0.874
1 pixel offset	14.977	10.891	3.649	0.896
Δ	0.387	0.069	0.159	0.022
5 pixel offset	15.606	10.929	3.827	0.920
Δ	1.016	0.107	0.337	0.046

With a one-pixel offset, the geometric method, *Y7-Geo*, already exhibits a small but noticeable increase in overall absolute error (+0.069 m). In contrast, our learning-based approach, *Y7-D*, shows only a 0.022 m rise in absolute error, indicating strong resilience to small perturbations. As the shift increases to five pixels, the performance gap between the two methods widens further, our model maintains lower error growth, confirming its robustness against minor optical misalignments. The increase in absolute error is more severe when estimating far-distance objects in the range of 28 m to 36 m, where geometric methods deteriorate significantly.

Figure 6.1 illustrates the impact of extrinsic camera parameter changes on distance estimation for both models. As the boat accelerates, significant pitch angle variations occur. Our proposed learning-based method estimates a distance of 165m, closer to the ground truth of 150m, whereas the geometry method estimates 205m. This demonstrates the robustness of our approach to camera pose changes in dynamic environments. For a more comprehensive comparison, we select 20 images: 10 from scenarios where the boat is traveling in a straight line at a constant speed, and 10 from scenarios where the boat is accelerating, decelerating, and turning. We then analyze the distance prediction errors from both methods, as shown in Table 6.5.

Table 6.5: Distance estimation errors on BOAT360 for Y7-D (proposed) and Y7-Geo (baseline) under different boat motion conditions: steady (constant speed, straight path) and high dynamic (acceleration, deceleration, and turning).

Movement Condition	Y7-Geo		Y7-D	
	W.Err (m)	Abs.Err (m)	W.Err (m)	Abs.Err (m)
Steady	0.609	58.63	0.299	18.59
High Dynamic	0.629	68.40	0.386	32.16

Learning on Fisheye Images vs. on Equirectangular Projections

We compare *Y7-D* (fisheye) and *Y7-D-Eq* (equirectangular) to assess the impact of the input format. As seen in Table 6.3, while *Y7-D-Eq* achieves a higher mAP@50 (0.631 vs. 0.438), its distance estimation error is significantly worse, with an absolute error of 1.609m compared to 0.874m for *Y7-D*. This suggests that training on raw fisheye images provides better depth estimation performance, likely because equirectangular projections distort object shapes and spatial relationships. However, the superior detection accuracy observed in the equirectangular model suggests that object detection benefits from axis-aligned bounding boxes, aligning with the characteristics of datasets such as COCO (Lin *et al.*, 2014), on which YOLO models are pretrained. Importantly, our method, even when trained on equirectangular images, outperforms the geometric and learning-based baselines, demonstrating its adaptability across projection formats.

Impact of Camera Height Estimation Head on Distance Prediction

In traditional geometric methods, such as shown in Figure 6.4, camera height is a critical factor for accurate distance estimation. To evaluate whether estimating camera height improves learning-based models, we compare *Y7-D+C* against *Y7-D*. Table 6.3 shows that estimating camera height (*Y7-D+C*) slightly improves weighted error (0.066 vs. 0.070) but does not impact absolute distance error. Interestingly, it improves mAP@50 from 0.438 to 0.495, suggesting that incorporating camera height estimation provides additional supervisory signals that enhance object detection. These results indicate that camera height estimation might have acted as a useful auxiliary task, improving both object detection and depth estimation without requiring additional calibration.

To further demonstrate the effectiveness of our approach, we analyze distance estimation performance across different distance ranges. Table 6.6 presents the results, illustrating how weighted and absolute errors vary with object distance, further highlighting the superiority of our learning-based method over the baselines.

Table 6.6: Distance-wise Weighted and Absolute Errors on the LOAF dataset (baselines italicized). Weighted Error and Absolute Error are measured in meters.

Range in meters	<i>Y7-Geo</i>		<i>Y7-DAv2</i>		Y7-D		Y7-D-Eq		Y7-D+C	
	W.Err	Abs.Err	W.Err	Abs.Err	W.Err	Abs.Err	W.Err	Abs.Err	W.Err	Abs.Err
(0.0 - 7.2)	1.588	8.825	0.783	4.924	0.074	0.478	0.328	1.741	0.058	0.369
(7.2 - 14.4)	1.109	12.203	0.894	10.132	0.064	0.777	0.146	1.545	0.072	0.833
(14.4 - 21.6)	0.671	11.081	0.928	15.829	0.066	1.097	0.070	1.208	0.071	1.207
(21.6 - 28.8)	0.510	13.350	0.952	23.372	0.095	2.179	0.090	1.895	0.080	1.974
(28.8 - 36.0)	0.455	14.590	0.958	27.544	0.104	3.490	0.123	3.588	0.075	2.017

6.5.2 Evaluation on ULM360 and BOAT360 Datasets

To assess the generalizability of our approach, we evaluate it on the ULM360 and BOAT360 datasets, which feature different sensor setups and environmental conditions, as mentioned in Subsection 6.3.1.

Performance on the ULM360 Dataset

Table 6.7 and Table 6.8 present the results on this dataset. Overall, several observations are consistent with the findings on LOAF, whereas the effect of adding the camera-height estimation head on detection and distance-estimation performance appears to be dataset-specific.

- Learning-based models outperform geometric methods: *Y7-D* achieves an absolute error of 0.57 m, while *Y7-Geo* records a significantly higher error of 2.44 m.
- The equirectangular model underperforms in distance estimation: *Y7-D-Eq* shows relatively high detection accuracy (mAP@50 of 0.915) but lags behind *Y7-D* in absolute error (0.788 m vs. 0.567 m).
- Distance head integration does not degrade object detection: Comparing *Y7-w/oDist* and *Y7-D*, we observe a slight trade-off in detection recall to obtain distance information.
- In contrast to the observations on LOAF, the additional camera-height estimation head does not improve detection or distance-estimation performance on this dataset. A similar pattern is observed on BOAT360. This difference may be related to the more dynamic camera poses in these mobile-platform scenarios.

Table 6.7: Object detection and distance estimation performance on ULM360 and BOAT360 (baselines italicized).

Dataset	Model	W.Err (m)	Abs.Err (m)	Precision	Recall	mAP@50	mAP@50:95
ULM360	<i>Y7-w/oDist</i>	-	-	0.946	0.914	0.893	0.451
	<i>Y7-Geo</i>	0.396	2.443	0.963	0.897	0.908	0.525
	<i>Y7-D_{Av2}</i>	0.795	5.107	0.963	0.897	0.908	0.525
	<i>Y7-D</i>	0.063	0.567	0.963	0.896	0.905	0.525
	<i>Y7-D-Eq</i>	0.105	0.788	0.955	0.914	0.915	0.539
	<i>Y7-D+C</i>	0.078	0.645	0.955	0.819	0.855	0.438
BOAT360	<i>Y7-w/oDist</i>	-	-	0.666	0.468	0.456	0.164
	<i>Y7-Geo</i>	0.617	65.359	0.565	0.553	0.437	0.173
	<i>Y7-D_{Av2}</i>	0.980	94.255	0.565	0.553	0.437	0.173
	<i>Y7-D</i>	0.358	29.544	0.565	0.553	0.437	0.173
	<i>Y7-D-Eq</i>	0.362	27.935	0.686	0.511	0.469	0.175
	<i>Y7-D+C</i>	0.447	29.968	0.553	0.540	0.423	0.154

Table 6.8: Distance-wise Weighted and Absolute Errors on ULM360 and BOAT360 datasets (baselines italicized). Weighted Error and Absolute Error are measured in meters.

Dataset	Distance Range in meters	<i>Y7-Geo</i>		<i>Y7-D_{Av2}</i>		<i>Y7-D</i>		<i>Y7-D-Eq</i>		<i>Y7-D+C</i>	
		W.Err	Abs.Err	W.Err	Abs.Err	W.Err	Abs.Err	W.Err	Abs.Err	W.Err	Abs.Err
ULM360	(0.0 - 4.0)	0.120	0.458	0.691	2.623	0.249	0.938	0.040	0.154	0.010	0.103
	(4.0 - 8.0)	0.431	2.476	0.789	4.598	0.109	0.838	0.073	0.657	0.093	0.748
	(8.0 - 12.0)	0.302	2.889	0.872	8.456	0.031	0.477	0.020	0.192	0.035	0.306
BOAT360	(0.0 - 100.0)	0.515	28.697	0.971	54.448	0.512	23.506	0.482	18.487	0.667	21.424
	(100.0 - 200.0)	0.723	87.459	0.989	117.872	0.188	25.244	0.237	32.167	0.223	28.450
	(200.0 - 300.0)	0.816	181.859	0.994	223.773	0.612	114.938	0.366	91.500	0.507	99.656

Performance on the BOAT360 Dataset

Tables 6.7 and 6.8 report the results on this dataset, which introduces additional challenges such as dynamic motion and greater distance ranges. Key findings include:

- Learning-based models outperform geometric methods but show higher errors in extreme conditions: *Y7-D* achieves an absolute error of 29.54 m compared to 65.36 m for *Y7-Geo*. However, compared to LOAF and ULM360, the errors are significantly larger due to the increased dynamics and long-range distance estimation.
- Distance estimation error increases at longer distances: A notable trend across BOAT360 is that the absolute error grows with increasing distance bins. This suggests that while the learning model remains effective at short to mid-range distances, its performance degrades in long-range estimations. However, compared to our proposed method, the large absolute distance errors of *Y7-Geo* for ranges exceeding 100 meters indicate that the geometric method is unsuitable for mobile platforms with significant dynamics, as variations in camera pose adversely affect its accuracy.

6.5.3 Summary of Key Findings

Overall, the experimental results across LOAF, ULM360, and BOAT360 reveal several consistent trends:

- Our model maintains object detection accuracy while simultaneously enabling precise distance estimation.
- Our proposed learning-based method consistently outperform geometric model and the other learning-based method in distance estimation.
- Learning with equirectangular projections improves detection performance but degrades distance estimation.

These results underscore the robustness of our learning-based approach, which generalizes well to different environments and sensor setups. While geometric models require precise calibration and are highly sensitive to environmental variations, our deep learning model adapts effectively to diverse conditions.

6.6 Conclusion and Outlook

This chapter introduced a neural learning-based approach for distance estimation from omnidirectional fisheye imagery, extending the monocular rectilinear formulation of the

preceding chapter. By eliminating the need for explicit calibration, the proposed framework proved robust to distortions and environmental variability, and consistently outperformed both geometric baselines and alternative learning methods across three distinct datasets. These results demonstrate that reliable distance estimation can be achieved with a lightweight single-sensor setup, making it suitable for a broad range of mobile perception applications, from structured urban scenes to unstructured maritime and open-field environments.

Nevertheless, the formulation remains rooted in spatial reasoning on a per-frame basis. While this chapter demonstrated the feasibility of generalizing monocular distance regression to omnidirectional vision, it does not yet address challenges that arise from temporal dynamics in video streams, such as motion blur, occlusions, and abrupt appearance changes. The next chapter tackles this aspect by exploring multi-frame integration, showing how temporal information can be leveraged to improve detection robustness without sacrificing the efficiency required for real-time deployment.

Chapter 7

Lightweight Multi-Frame Integration for Robust YOLO Object Detection in Videos

Whereas the preceding chapters focused on improving spatial and geometric reasoning through memory maps and distance estimation, they did not directly address temporal dynamics in video-based object detection. This chapter takes an orthogonal approach by exploring temporal fusion through multi-frame input stacking. Modern detectors such as YOLOv7 typically process frames independently, ignoring the temporal context naturally available in videos. Existing video-based approaches often attempt to capture this context through sophisticated temporal modules, but these solutions significantly increase model size and computational burden. In practical applications such as surveillance, autonomous driving, and maritime monitoring, transient challenges—including motion blur, occlusions, and abrupt appearance changes—can severely degrade the reliability of single-frame detection.

To address these issues, this chapter proposes a simple yet effective strategy: stacking multiple consecutive frames as input to a YOLO-based detector while supervising only the final frame. This formulation leverages temporal information without relying on motion estimation or memory representations, thereby preserving simplicity, computational efficiency, and real-time capability. Extensive experiments on both the challenging MOT20Det dataset and the newly introduced BOAT360 dataset demonstrate that this multi-frame integration substantially improves detection robustness, particularly for lightweight detectors. The results show that our method narrows the gap between compact and heavy detection networks, offering a practical balance between accuracy and efficiency. Moreover, BOAT360 provides a new benchmark of annotated fisheye video sequences captured from a boat, enabling further research in multi-frame object detection under challenging real-world conditions.

7.1 Introduction

Modern object detection methods, notably YOLO-based architectures such as YOLOv7 (Wang *et al.*, 2023), have achieved remarkable performance on single-frame image benchmarks. However, when deployed in video-centric applications such as robotics, surveillance systems, and autonomous driving, these methods often fall short due to their limited ability to utilize temporal information. Single-frame detectors typically treat each video frame independently, failing to exploit valuable cues from preceding frames, thereby reducing robustness to real-world challenges including occlusions, motion blur, and rapid changes in appearance.

To address this limitation, we propose a simple yet effective strategy: feeding a stack of multiple consecutive frames into the detection network and supervising the detection output for only the most recent frame. This approach implicitly enables the network to learn temporal and motion cues directly from raw pixel data at the earliest stage of processing, potentially enhancing the robustness of object detection without significantly increasing the model’s complexity or the training overhead.

Unlike previous approaches that explicitly model temporal dynamics using additional computationally intensive modules such as recurrent networks (Ahmad and Arnd, 2020), optical flow estimation (Zhu *et al.*, 2017d), or temporal attention mechanisms (Wu *et al.*, 2019b; Feichtenhofer *et al.*, 2019), our proposed method integrates seamlessly into existing single-frame architectures with minimal adaptation. Importantly, it does not need dense labels for each frame as opposed to classical video object detection methods. It merely takes in preceding video frames without labels as additional context, effectively categorizing our approach into a sparsely or *weakly supervised* learning method (Zhou, 2018).

Specifically, we extend YOLOv7 and YOLOv7-tiny by adapting the first convolutional layer to accommodate multiple input frames. We explore two convolutional adaptation strategies: direct early fusion by stacking frames at the pixel level, and grouped convolutions where each frame is processed independently before concatenation at the feature level, as illustrated in Figure 7.1

We evaluate our method on the MOT20Det dataset (Dendorfer *et al.*, 2020), a challenging pedestrian detection benchmark known for its crowded environments and frequent occlusions. Our results demonstrate that integrating temporal information boosts the detection performance of lightweight models such as YOLOv7-tiny. Furthermore, we validate the generalizability of our method on our BOAT360 dataset, which involves dynamic scenarios captured by a moving fisheye camera, showcasing our method’s robustness and wide applicability.

In summary, our key contributions are as follows:

- **A lightweight yet highly effective temporal integration strategy for YOLO-based object detectors**, which improves detection robustness while maintaining computational efficiency and minimal architectural modifications.

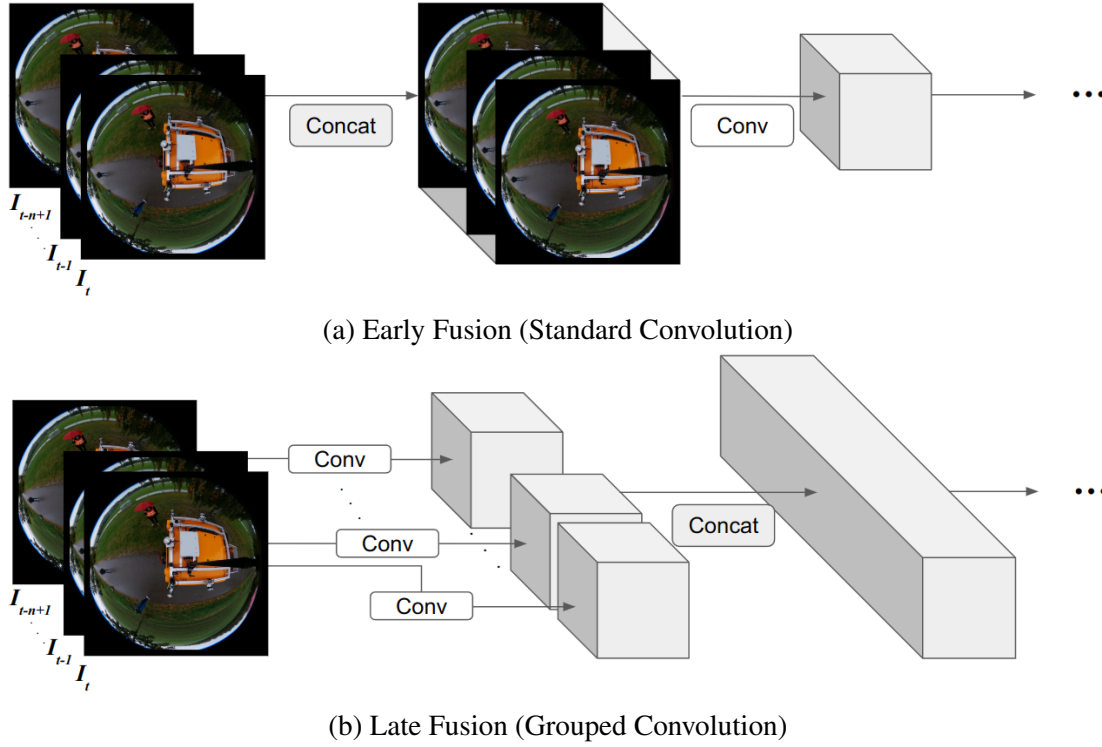


Figure 7.1: Network adaptation strategies for multi-frame object detection. We investigate two methods to extend single-frame detectors to support multiple video frames as input. (Top) Early Fusion (Standard Convolution): Input frames are stacked along the channel dimension and passed through shared convolutions from the first layer. This allows the network to jointly learn spatiotemporal features directly from raw pixels. (Bottom) Late Fusion (Grouped Convolution): Each frame is independently processed by separate convolutional kernels in a grouped convolution setup. Early layers extract low-level features per frame before merging via concatenation. This defers temporal fusion until later in the network. Both methods require modifying only the first few layers, making them lightweight and compatible with existing pretrained models.

- **Extensive empirical validation**, demonstrating performance gains over single-frame YOLO baselines across challenging datasets, including MOT20Det and our own BOAT360 dataset, under conditions of occlusion, motion blur, and rapid appearance changes.
- **Benchmark dataset contribution**, providing the BOAT360 dataset featuring annotated fisheye video sequences captured from a dynamic, waterborne platform, to support further research in multi-frame video object detection dynamic real-world environments.

7.2 Related Work

7.2.1 Single-Frame Object Detection

Object detection has witnessed tremendous progress with deep convolutional networks. Architectures like Faster R-CNN (Ren *et al.*, 2015b), YOLO (Redmon *et al.*, 2016b; Wang *et al.*, 2023), and RetinaNet (Lin *et al.*, 2017) have demonstrated strong performance on still images. Recent models such as YOLOv7 (Wang *et al.*, 2023) further optimize the trade-off between accuracy and speed, enabling real-time performance on edge devices.

However, these models primarily process each frame independently and do not exploit the temporal consistency naturally available in videos. In contrast, our method integrates multiple consecutive frames as input, providing temporal context while maintaining the architectural simplicity and speed advantages of YOLO-based detectors.

7.2.2 Video Object Detection

Video object detection aims to improve detection robustness by utilizing temporal information across frames. Early works proposed simple techniques such as feature aggregation across adjacent frames (Zhu *et al.*, 2017d) or tracking-by-detection pipelines (Stadler, 2023). More recent approaches introduced flow-guided feature warping (Zhu *et al.*, 2017d), temporal attention (Wu *et al.*, 2020), or recurrent memory networks (Pu *et al.*, 2020).

While these methods achieve notable gains, they often require complex architectures or multi-stage training procedures, making them difficult to integrate into real-time systems. Our approach offers simpler alternatives without adding large temporal modeling modules, making it more suitable for lightweight and deployable applications.

7.2.3 Multi-Frame Input with Sparse Supervision

Several studies have explored feeding multiple frames into the detector directly. Simple early fusion by stacking frames was explored in Qu *et al.* (2022), but often limited to two frames or relied on heavy video models such as 3D CNNs (Hour *et al.*, 2019).

Moreover, while most methods supervise every (or almost every) frame in a video sequence (Jiao *et al.*, 2021; Wu *et al.*, 2023; Zhu *et al.*, 2020; Shi *et al.*, 2023), we propose to supervise only a single frame while still benefiting from multiple inputs. This reduces labeling dependency and fits well into settings where annotation density is limited. In fact, many proposed object detection datasets sprung from videos where most frames were discarded in favor of diverse scenes (Varga *et al.*, 2022; Zhu *et al.*, 2020; Kiefer *et al.*, 2023d). Our approach allows to leverage these datasets to better performance almost *for free*.

7.2.4 Model Interpretability in Object Detection

Recent interpretability tools such as Grad-CAM (Selvaraju *et al.*, 2017), Grad-CAM++ (Chattopadhyay *et al.*, 2018), and Eigen-CAM (Muhammad and Yeasin, 2020) help visualize which regions influence predictions. In our study, we apply Grad-CAM++ to qualitatively analyze where the YOLO detector focuses when using single vs. multi-frame inputs.

7.3 Methodology

7.3.1 Problem Formulation

Given a video sequence, our goal is to improve object detection by leveraging information from multiple consecutive frames, while still supervising the model only on a single frame per input stack. This setting reduces annotation costs and introduces temporal context that aids detection under occlusion, motion blur, and appearance changes.

Formally, let $\{I_t\}_{t=1}^T$ be a sequence of RGB frames. Instead of detecting objects based only on frame I_t , we provide the model with a stack of n frames $\{I_{t-n+1}, \dots, I_{t-1}, I_t\}$ and predict object detections for frame I_t only.

7.3.2 Multi-Frame Input Representation

Instead of introducing new temporal modules or recurrent structures, we simply concatenate n consecutive RGB frames along the channel dimension. For instance, if $n = 5$, the input tensor has $3 \times n = 15$ channels, shaped as $(15, H, W)$.

The frames are stacked either as:

- **Adjacent frames:** consecutive frames at native frame rate (for example 25 fps on MOT20Det, 20 fps on BOAT360).
- **Stepped frames:** frames sampled with a fixed gap to span a longer temporal window.

This simple early fusion exposes the model to temporal variations at the pixel level from the first convolutional layer onward.

7.3.3 Network Architecture Adaptation

We adopt YOLOv7-tiny and YOLOv7 (Wang *et al.*, 2023) as our base architectures. To accommodate the increased number of input channels introduced by stacking multiple frames, we modify the first convolutional layer and explore two strategies:

- **Early Fusion (Standard Convolution):** Frames are stacked along the channel dimension, and a standard convolution is applied jointly across all channels. This enables the model to integrate spatial and temporal information from the very first layer.

To leverage pretraining, the weights of the first convolutional layer are initialized by repeating the pretrained single-frame weights n times along the input channel dimension and scaling them by a factor of $1/n$. This initialization ensures that, at the start of training, the response of the first convolution remains consistent with the original single-frame model, facilitating stable optimization and faster convergence.

- **Grouped Convolution (Feature-level Fusion):** The first convolutional layer is restructured as a grouped convolution (Krizhevsky *et al.*, 2017) with n groups, where each group independently processes the channels of one input frame. After this initial independent feature extraction, the feature maps from all frames are concatenated along the channel dimension.

To transfer knowledge from the pretrained single-frame model, each group is initialized with a copy of the original first-layer weights. Subsequently, the second convolutional layer is adjusted in the same way as in early fusion, ensuring compatibility with the expanded number of feature channels.

To provide a concise comparison of these strategies, Table 7.1 illustrates an example modification made to the first convolutional layer when adapting the network for multi-frame input.

Table 7.1: Adaptation of the first convolutional layer for multi-frame input. The table compares the original single-frame design with two multi-frame approaches: Early Fusion Multi-frame and Grouped Convolution Multi-frame. We assume n RGB frames of size 640×640 are input. $N=32$ kernels and stride of 2 are applied.

Model	Kernel Size H, W, C	Output Size H, W, C	Weight Shape N, H, W, C
Single frame	3, 3, 3	320, 320, 32	32, 3, 3, 3
Early Fusion Multi	3, 3, $3n$	320, 320, 32	32, 3, 3, $3n$
Group Conv. Multi	3, 3, 3 (n groups)	320, 320, $32n$	32, 3, 3, $3n$

7.3.4 Training Strategy

A key advantage of our framework is its weak supervision setup: although each input consists of a stack of multiple frames, only the most recent frame I_t in the sequence

is associated with ground-truth annotations. As illustrated in Figure 7.2, the preceding unlabeled frames are treated as temporal context, allowing the model to extract motion patterns and temporal dependencies across time. This design significantly reduces labeling requirements, as it eliminates the need to annotate every frame in a sequence. Our approach is well-suited for scenarios where dense frame-wise annotation is impractical or costly, such as long-duration surveillance footage or onboard recordings from mobile platforms.

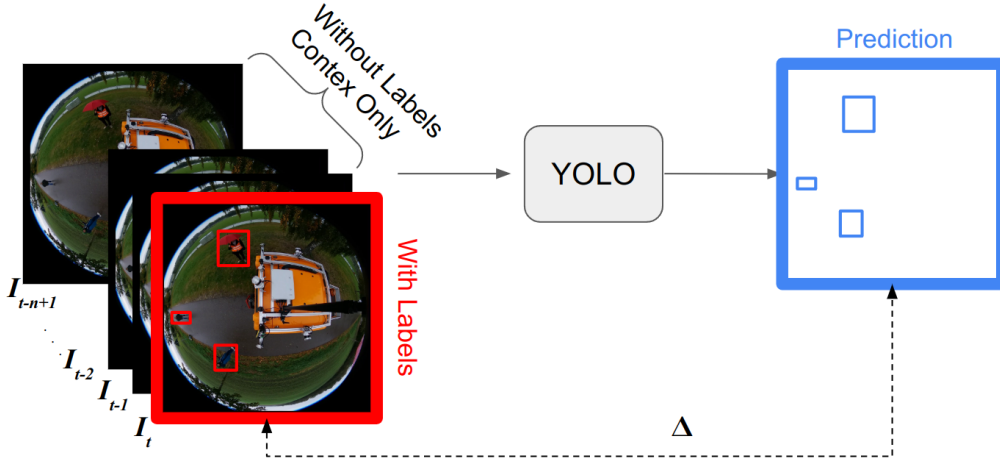


Figure 7.2: Overview of the weakly supervised training setup. Multiple frames are stacked and passed into the YOLO model, but only the latest frame I_t (marked as red) carries supervision. The earlier frames serve as unlabeled context, helping the model extract temporal cues without requiring additional labels. The sparsity of labeled frames can be adapted to the application.

During training, the input is a stack of n frames. Augmentations such as scaling, translation, rotation, and cropping, which alter the spatial characteristics of the image, are applied consistently across all frames within each stack to maintain spatial alignment.

Standard YOLO loss functions for objectness, classification, and bounding box regression are applied. All configurations were trained for 500 epochs, with the best checkpoint (all reached before the final epoch) selected for testing.

7.3.5 Advantages of the Proposed Approach

Our method offers:

- **Simplicity:** Only minor changes to existing models.
- **Efficiency:** Minimal overhead in parameters and computation.
- **Sparse supervision:** Only one frame per stack needs to be labeled.

We demonstrate in experiments that this approach improves detection robustness, particularly in challenging scenes with occlusions and motion artifacts.

7.4 Experiments

7.4.1 Datasets

We evaluate our proposed multi-frame object detection approach using two distinct datasets. These datasets differ in terms of camera setup, environmental conditions, and the complexity of the scenes, providing a robust basis for testing the generalization and performance of our method. The datasets are summarized in Table 7.2.

Table 7.2: Dataset characteristics of multi-frame training and evaluation benchmarks.

Dataset	Platform	Camera Mounting	Camera Type
MOT20Det	Static	Fixed height	Perspective
BOAT360	Mobile (boat)	Mounted on a mast	Fisheye

Dataset	Environment	Image Resolution	# Boxes
MOT20Det	Indoor & Outdoor	downscaled 640×640	2M
BOAT360	Open water	downscaled 1452×1452	509

- **MOT20Det (public dataset):** A large-scale pedestrian detection dataset captured from multiple static surveillance cameras under crowded conditions. Each annotated object includes tight bounding boxes around visible pedestrians, providing a challenging benchmark for object detection, especially under heavy occlusions and dense environments (Dendorfer *et al.*, 2020).
- **BOAT360 (our dataset):** Captured from a moving boat with a mast-mounted fish-eye camera. Annotated objects are floating buoys on water from varying distances. Due to the constant motion of the boat and the unstable environment, BOAT360 offers a unique perspective for evaluating detection models for small objects under dynamic and unstructured background and camera movement scenarios.

Each dataset features varying camera motion characteristics, object densities, and environmental conditions, making them ideal for evaluating the robustness and generalization capabilities of our proposed multi-frame detection method.

As the MOT20Det dataset only provides designated training and testing sets, we further split the official training sequences into 4/5 for training and 1/5 for validation. The official testing sequences are reserved exclusively for final evaluation. For the BOAT360

dataset, we divide the collected video sequences into 3/5 for training, 1/5 for validation, and 1/5 for testing.

7.4.2 Evaluation Metrics

We evaluate detection performance using standard COCO-style metrics (Lin *et al.*, 2014), including Precision (P), Recall (R), mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), and mean Average Precision averaged over multiple IoU thresholds from 0.5 to 0.95 in steps of 0.05 (mAP@0.5:0.95).

We also report model parameters, floating point operations (FLOPS) on NVIDIA RTX3090 and inference speed on Orin AGX embedded GPU.

7.4.3 Results

We evaluate our multi-frame detection framework on the MOT20Det dataset and validate its generalization on the BOAT360 dataset.

Effect of Temporal Context

As shown in Table 7.3, introducing multiple frames substantially improves detection performance. Stacking three adjacent frames increases mAP@0.5 by 4.4 points over the single-frame baseline. Performance further improves with 5 and 7 frames, peaking at 85.5% mAP@0.5 and 47.8% mAP@0.5:0.95.

However, using 9 adjacent frames degrades precision and overall mAP, suggesting that excessively large temporal windows introduce noise rather than useful motion cues. This points to an optimal range of temporal context for balancing new information and stability.

Table 7.3: Impact of number of adjacent frames on detection performance on MOT20Det test set.

YOLOv7-tiny	P	R	mAP@0.5	mAP@.5:.95
1 frame (baseline)	0.877	0.720	0.797	0.413
3 frames adjacent	0.943	0.731	0.841	0.462
5 frames adjacent	0.920	0.725	0.819	0.450
7 frames adjacent	0.946	0.755	0.855	0.478
9 frames adjacent	0.906	0.743	0.829	0.447

Sparse Temporal Sampling

Since 7 adjacent frames yielded the best accuracy in dense settings (Table 7.3), we explore whether similar temporal coverage with fewer, more widely spaced frames could achieve comparable results. As shown in Table 7.4, 3 frames with a step of 3 (spanning 7 frames) and 5 frames with a step of 2 (spanning 9-frame duration) both match or outperform the 7-frame adjacent baseline in mAP and recall. Additionally, a dynamic sampling configuration that emphasizes recent frames while retaining long context performs competitively. Overall, sparse temporal sampling offers an effective and efficient alternative to dense stacking, balancing temporal coverage and input efficiency.

Table 7.4: Impact of temporal step sampling on detection performance on MOT20Det.

YOLOv7-tiny	P	R	mAP@0.5	mAP@.5:.95
7 frames adjacent	0.946	0.755	0.855	0.478
3 frames, step 3	0.926	0.802	0.873	0.490
4 frames, step 2	0.948	0.736	0.844	0.466
5 frames, step 2	0.939	0.782	0.870	0.490
4 f., index [-6,-3,-1,0]	0.947	0.753	0.852	0.470

Early Fusion vs. Grouped Convolution

We investigated a variant where the first convolution layer applies grouped convolution across frames, separating early feature extraction. The intuition was to restrict low-level feature extraction to each frame independently, only mixing temporal information in later layers. However, results in Table 7.5 show consistent performance drops compared to standard early fusion.

This suggests that fusing spatial-temporal information early at the pixel-level allows the network to learn richer joint features, whereas restricting low-level channels by frame (grouping) hinders effective feature integration.

Scaling to Larger Models

We evaluate the applicability of our approach on the larger YOLOv7 model, which offers higher baseline accuracy compared to its tiny variant. As summarized in Table 7.6, adding temporal context yields only marginal improvements over the already strong single-frame baseline. We hypothesize that the high-capacity YOLOv7 model already achieves strong single-frame performance, leaving less opportunity for additional temporal information to further improve detection.

Notably, the 3-frame adjacent setup provides a slight boost in mAP@0.5:0.95 (+0.9), and sparse configurations such as 3-frame step-3 with grouped convolutions yield competitive results with minimal complexity increase.

Table 7.5: Impact of grouped convolution at first layer on detection performance on MOT20Det.

YOLOv7-tiny	P	R	mAP@0.5	mAP@.5:.95
3 frames adj.	0.943	0.731	0.841	0.462
3 frames adj., group	0.920	0.746	0.829	0.444
7 frames adj.	0.946	0.755	0.855	0.478
7 frames adj., group	0.898	0.711	0.804	0.406
3 frames, step 3	0.926	0.802	0.873	0.490
3 frames, step 3, group	0.925	0.736	0.829	0.448

Importantly, the multi-frame version of YOLOv7-tiny closes much of the performance gap to YOLOv7, highlighting the value of temporal modeling especially for compact models intended for deployment on resource-constrained devices.

Table 7.6: Impact of different input settings on YOLOv7 performance on MOT20Det.

YOLOv7	P	R	mAP@0.5	mAP@.5:.95
<i>1 frame (baseline)</i>	0.964	0.779	0.882	0.511
3 frames adjacent	0.954	0.796	0.879	0.520
5 frames adjacent	0.953	0.747	0.852	0.473
7 frames adjacent	0.955	0.774	0.867	0.497
9 frames adjacent	0.850	0.697	0.778	0.411
3 frames, step 3	0.953	0.757	0.855	0.493
7 frames, step 3	0.957	0.729	0.849	0.487
3 frames, step 3, group	0.954	0.798	0.878	0.518
7 frames, step 3, group	0.958	0.767	0.864	0.502

Computational Overhead

An important consideration for deploying multi-frame detectors is the computational cost. Table 7.7 summarizes the number of parameters and GFLOPS on RTX3090 for various configurations.

Notably, multi-frame models incur minimal increase in parameters: e.g., going from a 1-frame to 7-frame input on YOLOv7-tiny only grows the model from 6.006M to 6.011M parameters, by less than 0.1%.

Inference speed remains real-time, as reported in Table 7.8. Our YOLOv7-tiny multi-frame detector achieves 55 FPS on an NVIDIA Orin AGX, making it suitable for robotics deployments with strict latency constraints.

Table 7.7: Comparison of model size and computational cost for YOLOv7-tiny and YOLOv7 baselines with different multi-frame input configurations.

	YOLOv7-tiny		YOLOv7	
	Params	GFLOPS	Params	GFLOPS
<i>1 frame (baseline)</i>	6,006 k	13.0	36,479 k	103.2
3 frames	6,008 k	13.4	36,481 k	104.6
5 frames	6,010 k	13.7	36,483 k	106.0
7 frames	6,011 k	14.1	36,485 k	107.4
9 frames	6,013 k	14.3	36,486 k	108.8
3 frames, group	6,046 k	15.3	36,520 k	112.1
7 frames, group	6,049 k	16.0	36,523 k	115.0

Table 7.8: Inference speed (FPS) for 1024×1024 resolution images on NVIDIA Orin AGX.

Model	<i>1 frame (baseline)</i>	Multi-frame
YOLOv7-tiny	55	53-55
YOLOv7	43	41-43

Generalization to BOAT360

We further evaluate the model on the BOAT360 dataset, characterized by moving cameras, fisheye distortion, and dynamic water scenes. As shown in Table 7.9, the best results are achieved with 3-frame input, achieving a notable improvement over the single-frame baseline for both YOLOv7-tiny (+12.8 mAP@0.5) and YOLOv7 (+8.1 mAP@0.5). However, with more than 5 frames the performance slightly degrades, likely due to rapid scene changes between distant frames.

This indicates that our multi-frame framework generalizes well to new domains, and that moderate temporal context suffices when scene dynamics are fast.

Finally, to qualitatively validate our findings, we present visual comparisons in Fig. 7.3. The examples showcase challenging scenarios where the multi-frame detector demonstrates clear advantages over the single-frame baseline, including partial object truncation at the image boundary, motion blur, occlusion, and sunlight glare. In these cases, the multi-frame model (blue boxes) exhibits superior robustness, detecting more objects

Table 7.9: Impact of number of adjacent frames on detection performance on BOAT360.

YOLOv7-tiny	P	R	mAP@0.5	mAP@.5:.95
1 frame (baseline)	0.653	0.362	0.411	0.134
3 frames adjacent	0.689	0.617	0.539	0.215
5 frames adjacent	0.555	0.532	0.444	0.146
7 frames adjacent	0.616	0.617	0.537	0.186
9 frames adjacent	0.674	0.617	0.526	0.184
YOLOv7				
1 frame (baseline)	0.729	0.574	0.583	0.234
3 frames adjacent	0.750	0.702	0.664	0.277
5 frames adjacent	0.674	0.66	0.642	0.25
7 frames adjacent	0.577	0.638	0.523	0.208
9 frames adjacent	0.794	0.574	0.606	0.201

correctly compared to the single-frame model (green boxes), highlighting its effectiveness in handling difficult real-world conditions. In addition to detection outputs, we visualize gradient-based class activation maps with Grad-CAM++ (Chattopadhyay *et al.*, 2018) for both the single-frame baseline and our multi-frame model. The activation maps are derived from the penultimate layer of the networks, as the final layer is specialized for producing detection heads and lacks meaningful spatial features. These maps reveal where the models focus their attention during inference. The baseline often produces diffuse activations in the presence of visual challenges. In contrast, the multi-frame model exhibits more localized attention aligned with object regions, suggesting stronger temporal awareness. These qualitative insights further highlight the benefit of incorporating multiple frames during training and inference.

7.5 Conclusion

The chapter presented a simple yet effective strategy to enhance object detection by leveraging multi-frame temporal context within standard single-frame detector architectures. By stacking consecutive frames at the input level and supervising only the detection output corresponding to the latest frame, our approach enriches the early feature extraction process without introducing substantial complexity or computational overhead.

Through comprehensive experiments on the MOT20Det and BOAT360 datasets, we demonstrated that lightweight models such as YOLOv7-tiny benefit significantly from temporal information, achieving up to 8% relative improvement in detection accuracy. Our ablation studies revealed that moderate temporal context (e.g., 3 to 7 frames) and temporal step sampling are crucial for maximizing the performance gains, while early

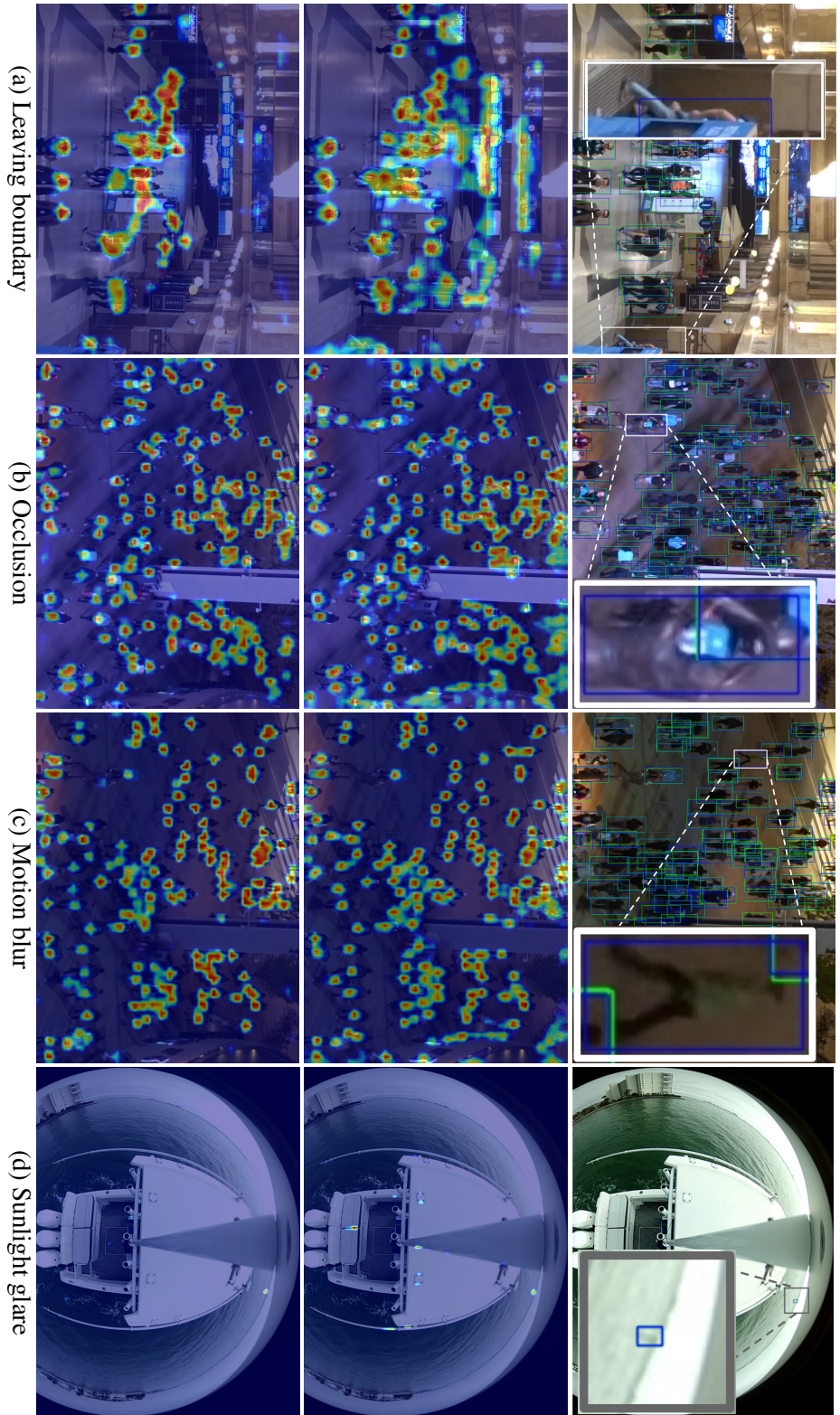


Figure 7.3: Qualitative comparison across challenging video scenarios: (a) leaving the image boundary, (b) occlusion, (c) motion blur, and (d) sunlight glare. **Top row:** detection results comparing the single-frame baseline (green boxes) with our multi-frame YOLOv7 model (blue boxes). **Middle row:** Grad-CAM++ heatmaps from the single-frame baseline. **Bottom row:** heatmaps from our multi-frame model, showing improved focus and spatial understanding contributed by temporal context.

fusion of frames outperforms grouped convolutions. Furthermore, our approach generalizes well to challenging dynamic environments, as evidenced by its strong performance on the BOAT360 dataset.

Importantly, the proposed method scales efficiently, maintaining real-time inference speeds and introducing minimal parameter growth, making it suitable for deployment in robotics and mobile applications. Additionally, we contribute the BOAT360 dataset to foster further research into robust video-based object detection under dynamic and non-ideal conditions.

In future work, one could extend this early fusion paradigm to other detection backbones, explore adaptive frame selection strategies, and investigate integration with lightweight temporal attention mechanisms to further boost video object detection performance while maintaining real-time constraints.

Chapter 8

Conclusions

8.1 Summary

This dissertation investigated how *geometric reasoning*, *temporal fusion*, and *learning-based approaches* can be composed to advance single-sensor perception on mobile platforms operating under realistic constraints. Throughout the chapters, two complementary principles served as the guiding foundation of this work: (i) **spatial memory**, which accumulates and stabilizes evidence over space to localize objects and smooth detection confidence over time, and (ii) **temporal fusion**, which couples adjacent frames to improve robustness, while largely preserving the efficiency required for embedded deployment. These principles were instantiated in five problem settings: surround-view object detection and localization with memory maps for unmanned ground vehicles (UGVs), object detection and localization with memory maps for unmanned aerial vehicles (UAVs), distance estimation with approximate supervision for unmanned surface vehicles (USVs), learning-based distance estimation for single-sensor 360° setups, and multi-frame integration for robust object detection in videos. Together, these results demonstrate that carefully designed inductive biases from projective geometry and scene structure, combined with lightweight temporal fusion, can be integrated into modern detectors to yield measurable and practical gains while keeping models and data requirements modest.

8.2 Contributions by Chapter

Robust 360° Camera Object Detection and Localization with Memory Maps. This chapter introduced a surround-view perception scheme that maintains an explicit, map-like memory to integrate object information over time. The memory acts as a persistent spatial prior that regularizes detections, reduces temporal inconsistencies, and improves detection and localization stability under occlusions and ego-motion. The formulation makes minimal assumptions about platform kinematics and sensor setup, and it remains compatible with efficient, off-the-shelf detectors.

Confidence Boosting in 3D for Object Detection on UAVs. Building on the memory-map mechanism introduced earlier, this chapter extends the concept to UAV-based de-

tection with metadata-aware geo-referencing. It fuses onboard GPS, IMU, and altitude data to align detections across frames within a shared 3D world coordinate frame. A confidence boosting mechanism then exploits geometric and temporal consistency in this geo-referenced space, effectively reducing false positives and improving detection stability under large viewpoint changes and rapid UAV motion.

Approximate Supervised Object Distance Estimation on Unmanned Surface Vehicles. This chapter showed that domain knowledge in maritime scenes can be leveraged to construct *approximate* supervision for distance estimation from a single camera. Despite imperfect labels, the approach achieves competitive distance predictions for USV in scenarios where conventional ranging sensors are impractical. The study also analyzed the failure modes induced by pitch/roll dynamics, reflective surfaces, and low-texture targets, yielding concrete guidance for sensor placement and calibration.

Learning-Based Distance Estimation for 360° Single-Sensor Setups. This chapter generalized distance estimation to broader single-sensor settings, including panoramic and 360° imagery. It proposed a learning framework that remains lightweight yet benefits from geometric priors, improving generalization across cameras and environments. The method emphasizes calibrated uncertainty and prioritizes evaluation metrics directly relevant to downstream navigation and collision avoidance, not only pixel-space error, thus aligning prediction targets with robotic decision-making.

Lightweight Multi-Frame Integration for Robust YOLO Object Detection in Videos. Finally, the dissertation investigated fusion strategies at both the input and intermediate feature levels, where a short stack of adjacent frames was incorporated but supervision was applied only to the latest frame. The design leaves the detector head unchanged, keeps compute growth minimal, and yields robustness to edge cases such as blur, momentary occlusions, and dynamic departures of individuals from the scene. Experiments on public crowded-scene data and our collected fisheye-image dataset (e.g., MOT20Det and BOAT360) demonstrated consistent gains over single-frame baselines at nearly the same throughput, which is essential for embedded platforms.

8.3 Synthesis of Findings

Across these studies, several unifying lessons emerge that extend beyond the scope of individual chapters:

1. **Geometry as a stabilizing prior.** Incorporating geometric structure, even when approximate, provides consistent spatial references that improve detection stability, localization accuracy, and distance estimation, especially when data are scarce or noisy.
2. **Temporal context as a lightweight enhancer.** Short-horizon temporal fusion, either through spatial memory or stacked inputs, captures most of the attainable

temporal benefits while remaining compatible with real-time embedded deployment.

3. **Task-aligned supervision as a driver of utility.** Supervision that reflects downstream robotic needs, such as metric distance estimation, uncertainty calibration, or temporal consistency, leads to perception outputs that are more directly actionable than those optimized for conventional image-based losses.

These insights demonstrate that robust single-sensor perception is achievable without resorting to dense reconstruction or large recurrent architectures. A principled combination of geometric reasoning, temporal information, and task-specific supervision enables such systems to operate reliably under realistic compute budgets. Scientifically, this work contributes a clearer understanding of how geometry, temporal context, and learning complement one another: geometry grounds the representation in physical space, temporal cues provide continuity and stability, and learning adapts to appearance variability and uncertainty.

8.4 Limitations

The work also reveals several limitations. First, *memory-based* accumulation can drift if the egomotion estimate or calibration degrades; periodic re-anchoring or uncertainty-aware updates are needed for long deployments. Second, *approximate supervision* inherits biases from the priors used to generate it, which can limit its applicability to extreme cases such as rough seas, strong heave, or severe camera tilt. Third, in *learning-based distance estimation*, generalization across lenses and extreme FOVs remains challenging without explicitly or implicitly modeling of projection differences. Fourth, although *multi-frame integration* is compute-efficient, its gains depend on carefully chosen frame spacing and latency budgets; overly long context can hurt responsiveness on fast platforms. Finally, all studies are bounded by dataset scale and diversity; the hardest open-world corner cases (nighttime glare and adverse weather) remain underrepresented.

8.5 Outlook and Future Work

Building on these results, several research directions are well motivated and feasible:

1. **Unified spatio-temporal memory.** Replace per-module heuristics with a structured, learnable spatio-temporal memory that jointly encodes past detections, motion, and uncertainty. Such a memory would maintain a consistent world-aligned representation over time, integrate new evidence through updates, and serve as a shared substrate for detection, localization, and distance estimation.

2. **Uncertainty calibration and decision coupling.** Leverage the calibrated uncertainties of detection and distance estimates in downstream planning and tracking modules to achieve confidence-aware perception–action coupling.
3. **Self-supervision from geometry.** Explore geometry-based self-supervision strategies that operate at the object level, for example by leveraging view consistency or motion parallax cues to infer distance without dense labels. Combining such signals with sparse approximate annotations could reduce labeling costs and improve cross-domain generalization.
4. **Projection-aware architectures.** Extend current object-centric perception pipelines by embedding camera projection models (e.g., fisheye or equirectangular) directly into feature sampling and receptive fields. This would enable a single backbone to natively process wide-FOV inputs, potentially improving performance and generalization across different sensor geometries.
5. **Resource-aware perception.** Develop anytime and budgeted variants of the proposed methods that can adapt their computational effort, for example by adjusting frame stacking, memory horizon, or input resolution, to meet power and latency constraints on embedded hardware while still producing usable outputs at different performance levels.
6. **Richer priors for marine and mobile robots.** Extend approximate supervision beyond flat-surface priors to learned sea-state models, IMU-camera co-training, and radar-vision cross-checking, broadening applicability in adverse conditions.
7. **Open-world robustness.** Stress-test under distribution shifts (weather, lighting, scene geometry) and pursue data augmentation or domain adaptation strategies targeted at the most critical failure modes identified in this dissertation.

8.6 Concluding Remarks

The overarching conclusion of this dissertation is that integrating modest but principled amounts of geometric reasoning and temporal context, while maintaining computational efficiency, can meaningfully enhance single-sensor perception. By shaping supervision targets to align with robotic objectives and embedding structural cues to stabilize learning, we obtain detectors and distance estimators that are more robust, interpretable, and practical for deployment. The approaches developed here are intentionally straightforward to adopt, extending existing detectors, and they open pathways for integration with advanced sequence models and uncertainty-aware inference. The empirical evaluations, ablation studies, and analysis of failure modes aim to guide future work toward reliable single-sensor perception under real-world sensing and computational constraints.

Abbreviations

AGX	NVIDIA Jetson AGX (embedded computing platform)
AIS	Automatic Identification System
AP	Average Precision
AP@50	Average Precision at IoU = 0.50
AR	Average Recall
AR10	Average Recall with 10 detections per image
BOAT360	360° vision dataset collected on boats for object detection and localization in maritime environments
BDD100K	Berkeley DeepDrive 100K Dataset
ByteTracker	High-performance multi-object tracker
CAM	Class Activation Map
CAN	Controller Area Network
CNN	Convolutional Neural Network
COCO	Common Objects in Context (dataset for object detection, segmentation, and captioning)
CV	Computer Vision
DAB-DETR	Dynamic Anchor Boxes DEtection TRansformer
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DETR	DEtection TRansformer
DeepSORT	Simple Online and Realtime Tracking with Deep Association Metric
DJI	Da-Jiang Innovations (UAV manufacturer)
EfficientDet	Efficient Object Detector
FGFA	Flow-Guided Feature Aggregation
FLOPs	Floating Point Operations
FOV	Field of View
FPS	Frames Per Second
GFLOPs	Giga Floating Point Operations per Second
GPS	Global Positioning System
GPU	Graphics Processing Unit
Grad-CAM	Gradient-weighted Class Activation Mapping, a method for visual explanation of CNN-based models
GT	Ground Truth
HOTA	Higher Order Tracking Accuracy

Abbreviations

ID	Identifier
IDs	Identifiers (multiple targets)
IMU	Inertial Measurement Unit
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
LOAF	Large-scale OverheAd Fisheye (dataset for person detection and localization)
mAP	mean Average Precision
mAP@0.5	mAP at IoU threshold 0.5 (PASCAL VOC metric)
mAP@.5:.95	mAP averaged over IoU thresholds from 0.5 to 0.95 (COCO metric)
MOT20	Multi-Object Tracking 2020 Challenge (dataset for pedestrian tracking in crowded scenes)
P	Precision, the fraction of predicted detections that are correct
POG-V	People-On-Grass-Video, (dataset of people walking on grass and similar terrain)
R	Recall, the fraction of ground truth instances correctly detected
OS-1	Ouster OS-1 LiDAR sensor
RANSAC	Random Sample Consensus
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicles
ULM360	360° vision dataset collected in the city of Ulm for object detection and localization
USV	Unmanned Surface Vehicle
YOLO	You Only Look Once (object detector family)
YOLOv7	Seventh version of YOLO detector
YOLOv9	Ninth version of YOLO detector

Bibliography

- Abate 1, R. S. (2010). Nepa, national security, and ocean noise: the past, present, and future of regulating the impact of navy sonar on marine mammals. *Journal of International Wildlife Law & Policy*, **13**(4), 326–356.
- Ahmad, B. Q. and Arnd, P. (2020). Recurrent neural networks for video object detection. In *Computer Vision and Pattern Recognition (CVPR)*.
- Albrektsen, S. M. and Johansen, T. A. (2017). Syncboard-a high accuracy sensor timing board for uav payloads. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1706–1715. IEEE.
- Ali, A., Hassan, A., Ali, A. R., Khan, H. U., Kazmi, W., and Zaheer, A. (2020). Real-time vehicle distance estimation using single view geometry. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1111–1120.
- Almeida, C., Franco, T., Ferreira, H., Martins, A., Santos, R., Almeida, J. M., Carvalho, J., and Silva, E. (2009). Radar based collision detection developments on usv roaz ii. In *Oceans 2009-Europe*, pages 1–6. Ieee.
- Amjoud, A. B. and Amrouch, M. (2023). Object detection using deep learning, cnns and vision transformers: A review. *IEEE Access*, **11**, 35479–35516.
- Arkin, E., Yadikar, N., Xu, X., Aysa, A., and Ubul, K. (2023). A survey: object detection methods from cnn to transformer. *Multimedia Tools and Applications*, **82**(14), 21353–21383.
- Bae, H. S., Kim, W.-J., Kim, W.-S., Choi, S.-M., and Ahn, J.-H. (2015). Development of the sonar system for an unmanned surface vehicle. *Journal of the Korea Institute of Military Science and Technology*, **18**(4), 358–368.
- Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and real-time tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468.
- Bhoi, A. (2019). Monocular depth estimation: A survey. *arXiv preprint arXiv:1901.09402*.

- Bohren, C. F. and Fraser, A. B. (1986). At what altitude does the horizon cease to be visible? *American Journal of Physics*, **54**(3), 222–227.
- Bosquet, B., Mucientes, M., and Brea, V. M. (2021). Stdnet-st: Spatio-temporal convnet for small object detection. *Pattern Recognition*, **116**, 107929.
- Bovcon, B. and Kristan, M. (2021). Wasr—a water segmentation and refinement maritime obstacle detection network. *IEEE Transactions on Cybernetics*, **52**(12), 12661–12674.
- Bovcon, B. and Kristan, M. (2022). Wasr—a water segmentation and refinement maritime obstacle detection network. *IEEE Transactions on Cybernetics*, **52**(12), 12661–12674.
- Bovcon, B., Muhovič, J., Perš, J., and Kristan, M. (2019). The mastr1325 dataset for training deep usv obstacle detection models. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3431–3438. IEEE.
- Bovcon, B., Muhovič, J., Vranac, D., Mozetič, D., Perš, J., and Kristan, M. (2021). MODS—A USV-Oriented Object Detection and Obstacle Segmentation Benchmark. *IEEE Transactions on Intelligent Transportation Systems*, **23**(8), 13403–13418.
- Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J., and Tardós, J. D. (2021). Orbslam3: An accurate open-source library for visual, visual–inertial and multi-map slam. *IEEE Transactions on Robotics*, **37**(6), 1874–1890.
- CaptainAI (2022). Captain ai viewpoint estimation model. <https://www.captainai.com/technology/captain-ai-viewpoint-estimation-model/>. Accessed: 2024-02-25.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229.
- Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., and García-Gutiérrez, J. (2020). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*, **13**(1), 89.
- Chang, J.-R. and Chen, Y.-S. (2018). Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5418.
- Chattopadhyay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE.

- Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., and Lin, D. (2019). MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.
- Chen, X., Ma, H., Wan, J., Li, B., and Xia, T. (2017). Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915.
- Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., and Liu, Z. (2022). Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5270–5279.
- Chen, Z., Jain, A., and Gupta, A. (2020). Memory enhanced global-local aggregation for video object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 10557–10564.
- Chung, S.-Y. and Huang, H.-P. (2010). Slammot-sp: simultaneous slammot and scene prediction. *Advanced Robotics*, **24**(7), 979–1002.
- Coors, B., Paul Condurache, A., and Geiger, A. (2018). Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533.
- Corsel, C. W., van Lier, M., Kampmeijer, L., Boehrer, N., and Bakker, E. M. (2023). Exploiting temporal context for tiny object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 79–89.
- Danelljan, M., Gool, L. V., and Timofte, R. (2020). Probabilistic regression for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7192.
- Davydov, Y., Chen, W.-H., and Lin, Y.-C. (2022). Supervised object-specific distance estimation from monocular images for autonomous driving. *Sensors*, **22**(22), 8846.
- De Wind, H., Cilliers, J., and Herselman, P. (2010). Dataware: Sea clutter and small boat radar reflectivity databases [best of the web]. *IEEE Signal processing magazine*, **27**(2), 145–148.
- del Rey-Maestre, N., Mata-Moya, D., Jarabo-Amores, M.-P., Gomez-del Hoyo, P.-J., and Barcena-Humanes, J.-L. (2018). Artificial intelligence techniques for small boats detection in radar clutter. real data validation. *Engineering Applications of Artificial Intelligence*, **67**, 296–308.

- Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., and Leal-Taixé, L. (2020). Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*.
- Diwan, T., Anirudh, G., and Tembhurne, J. V. (2023). Object detection using yolo: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications*, **82**(6), 9243–9275.
- Doll, S., Schulz, R., Schneider, L., Benzin, V., Enzweiler, M., and Lensch, H. P. (2022). Spatialdetr: Robust scalable transformer-based 3d object detection from multi-view camera images with global cross-sensor attention. In *European Conference on Computer Vision*, pages 230–245. Springer.
- Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., and Tian, Q. (2018). Uavdt: A large-scale benchmark for uav object detection and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.
- Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, **27**.
- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, **22**(6), 46–57.
- Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. (2018). Learning so(3)-equivariant representations with spherical cnns. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J. M., and Zisserman, A. (2014). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, **111**, 98 – 136.
- Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(9), 1627–1645.
- Ferreira, F., Kraševac, N., Obradović, J., Milijaš, R., Lončar, I., Bogdan, S., and Mišković, N. (2022). Lidar-based usv close approach to vessels for manipulation purposes. In *OCEANS 2022, Hampton Roads*, pages 1–6. IEEE.

- Fischler, M. and Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, **24**(6), 381–395.
- Fox, V., Hightower, J., Liao, L., Schulz, D., and Borriello, G. (2003). Bayesian filtering for location estimation. *IEEE pervasive computing*, **2**(3), 24–33.
- Genovese, K. (2024). Single-image camera calibration with model-free distortion correction. *Optics and Lasers in Engineering*, **181**, 108348.
- Geyer, C. and Daniilidis, K. (2001). A unifying theory for central panoramic systems and practical implications. *International Journal of Computer Vision*, **45**(3), 175–196.
- Gillner, S., Weiß, A. M., and Mallot, H. A. (2008). Visual homing in the absence of feature-based landmark information. *Cognition*, **109**(1), 105–122.
- Godard, C., Mac Aodha, O., and Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279.
- Godard, C., Mac Aodha, O., Firman, M., and Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838.
- Gong, T., Chen, K., Wang, X., Chu, Q., Zhu, F., Lin, D., Yu, N., and Feng, H. (2021). Temporal roi align for video object recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1442–1450.
- Han, J., Kim, J., and Son, N.-s. (2017). Persistent automatic tracking of multiple surface vessels by fusing radar and lidar. In *OCEANS 2017-Aberdeen*, pages 1–5. IEEE.
- Handel, H. (2007). Analyzing the influences of camera warm-up effects on image acquisition. In *Asian Conference on Computer Vision*, pages 258–268. Springer.
- Hashmani, M. A., Umair, M., Rizvi, S. S. H., and Gilal, A. R. (2020). A survey on edge detection based recent marine horizon line detection methods and their applications. In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–5. IEEE.
- Heinemann, P., Rückstieß, T., and Zell, A. (2004). Fast and accurate environment modelling using omnidirectional vision. *Dynamic Perception*, pages 9–14.
- Hirschmuller, H. (2007). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, **30**(2), 328–341.

- Hosang, J., Benenson, R., Dollár, P., and Schiele, B. (2016). What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(4), 814–830.
- Hour, R., Chen, C., Sukthankar, R., and Shah, M. (2019). An efficient 3d cnn for action/object segmentation in video. In *British Machine Vision Conference 2019*.
- Hübner, W. and Mallot, H. A. (2007). Metric embedding of view-graphs: A vision and odometry-based approach to cognitive mapping. *Autonomous Robots*, **23**(3), 183–196.
- Jiang, D., Yuan, M., Xiong, J., Xiao, J., and Duan, Y. (2024). Obstacle avoidance usv in multi-static obstacle environments based on a deep reinforcement learning approach. *Measurement and Control*, **57**(4), 415–427.
- Jiao, L., Zhang, R., Liu, F., Yang, S., Hou, B., Li, L., and Tang, X. (2021). New generation deep learning for video object detection: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, **33**(8), 3195–3215.
- Jin, J., Liu, D., Li, F., Dai, Y., Li, L., and Ma, Y. (2024). Wide baseline stereovision based obstacle detection for unmanned surface vehicles. *Signal, Image and Video Processing*, **18**(5), 4605–4614.
- Jin, L., Choi, Y., Kim, G., Kim, J., and Kim, J.-H. (2020). Unifuse: Unidirectional fusion for 360° panorama depth estimation. In *British Machine Vision Conference (BMVC)*.
- Jorge, V. A., Granada, R., Maidana, R. G., Jurak, D. A., Heck, G., Negreiros, A. P., Dos Santos, D. H., Gonçalves, L. M., and Amory, A. M. (2019). A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions. *Sensors*, **19**(3), 702.
- Kannala, J. and Brandt, S. (2006a). A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, **28**, 1335–40.
- Kannala, J. and Brandt, S. S. (2006b). A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(8), 1335–1340.
- Kaplowitz, C. (2022). *Characterization of a Stereo Vision System for Object Classification for USV Navigation*. Master’s thesis, Florida Atlantic University.
- Kiefer, B. and Zell, A. (2023). Fast region of interest proposals on maritime uavs. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3317–3324. IEEE.

- Kiefer, B., Meßmer, M., and Zell, A. (2021). Leveraging domain labels for object detection from uavs. *arXiv preprint arXiv:2101.12677*.
- Kiefer, B., Kristan, M., Perš, J., Žust, L., Poiesi, F., De Alcantara Andrade, F. A., Bernardino, A., Dawkins, M., Raitoharju, J., **Quan, Y.**, Atmaca, A., Höfer, T., Zhang, Q., Xu, Y., Zhang, J., Tao, D., Sommer, L., Spraul, R., Zhao, H., Zhang, H., Zhao, Y., Augustin, J. L., Jeon, E.-I., Lee, I., Zedda, L., Loddo, A., Di Ruberto, C., Verma, S., Gupta, S., Muralidhara, S., Hegde, N., Xing, D., Evangeliou, N., Tzes, A., Bartl, V., Špaňhel, J., Herout, A., Bhowmik, N., Breckon, T. P., Kundargi, S., Anvekar, T., Tabib, R. A., Mudengudi, U., Vats, A., Song, Y., Liu, D., Li, Y., Li, S., Tan, C., Lan, L., Somers, V., De Vleeschouwer, C., Alahi, A., Huang, H.-W., Yang, C.-Y., Hwang, J.-N., Kim, P.-K., Kim, K., Lee, K., Jiang, S., Li, H., Ziqiang, Z., Vu, T.-A., Nguyen-Truong, H., Yeung, S.-K., Jia, Z., Yang, S., Hsu, C.-C., Hou, X.-Y., Jhang, Y.-A., Yang, S., and Yang, M.-T. (2023a). 1st workshop on maritime computer vision (macvi) 2023: Challenge results. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 265–302.
- Kiefer, B., Kristan, M., Perš, J., Žust, L., Poiesi, F., Andrade, F., Bernardino, A., Dawkins, M., Raitoharju, J., **Quan, Y.**, et al. (2023b). 1st workshop on maritime computer vision (macvi) 2023: Challenge results. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 265–302.
- Kiefer, B., Kristan, M., Perš, J., Žust, L., Poiesi, F., Andrade, F., Bernardino, A., Dawkins, M., Raitoharju, J., **Quan, Y.**, Atmaca, A., Höfer, T., Zhang, Q., Xu, Y., Zhang, J., Tao, D., Sommer, L., Spraul, R., Zhao, H., Zhang, H., Zhao, Y., Augustin, J. L., Jeon, E.-i., Lee, I., Zedda, L., Loddo, A., Di Ruberto, C., Verma, S., Gupta, S., Muralidhara, S., Hegde, N., Xing, D., Evangeliou, N., Tzes, A., Bartl, V., Špaňhel, J., Herout, A., Bhowmik, N., Breckon, T. P., Kundargi, S., Anvekar, T., Tabib, R. A., Mudenagudi, U., Vats, A., Song, Y., Liu, D., Li, Y., Li, S., Tan, C., Lan, L., Somers, V., De Vleeschouwer, C., Alahi, A., Huang, H.-W., Yang, C.-Y., Hwang, J.-N., Kim, P.-K., Kim, K., Lee, K., Jiang, S., Li, H., Ziqiang, Z., Vu, T.-A., Nguyen-Truong, H., Yeung, S.-K., Jia, Z., Yang, S., Hsu, C.-C., Hou, X.-Y., Jhang, Y.-A., Yang, S., and Yang, M.-T. (2023c). 1st workshop on maritime computer vision (macvi) 2023: Challenge results. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 265–302.
- Kiefer, B., Žust, L., Kristan, M., Perš, J., Teršek, M., Wiliem, A., Messmer, M., Yang, C.-Y., Huang, H.-W., Jiang, Z., et al. (2023d). The 2nd workshop on maritime computer vision (macvi) 2024. *arXiv preprint arXiv:2311.14762*.
- Kiefer, B., **Quan, Y.**, and Zell, A. (2023e). Memory maps for video object detection and tracking on uavs. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3040–3047.

- Kiefer, B., Höfer, T., and Zell, A. (2023f). Stable yaw estimation of boats from the viewpoint of uavs and usvs. In *2023 European Conference on Mobile Robots (ECMR)*, pages 1–6.
- Kiefer, B., Žust, L., Kristan, M., Perš, J., Teršek, M., Wiliem, A., Messmer, M., Yang, C.-Y., Huang, H.-W., Jiang, Z., Kuo, H.-C., Mei, J., Hwang, J.-N., Stadler, D., Sommer, L., Huang, K., Zheng, A., Chong, W., Lertniphonphan, K., Xie, J., Chen, F., Li, J., Wang, Z., Zedda, L., Loddo, A., Di Ruberto, C., Vu, T.-A., Nguyen-Truong, H., Ha, T.-S., Pham, Q.-D., Yeung, S.-K., Feng, Y., Thien, N. T., Tian, L., Michel, A., Gross, W., Weinmann, M., Carrillo-Perez, B., Klein, A., Alex, A., Solano-Carrillo, E., Steiniger, Y., Rodriguez, A. B., Kuan, S.-Y., Ho, Y.-H., Sattler, F., Fabijanić, M., Šimunec, M., and Kapetanović, N. (2024). 2nd workshop on maritime computer vision (macvi) 2024: Challenge results. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 869–891.
- Kiefer, B., Žust, L., Muhovič, J., Kristan, M., Perš, J., Teršek, M., Mudenagudi, U., Desai, C., Wiliem, A., Kreis, M., Akalwadi, N., **Quan, Y.**, Zhong, Z., Zhang, Z., Liu, S., Chen, X., Yang, Y., Fabijanić, M., Ferreira, F., Lee, S., Lee, J., Lee, K., Yao, S., Guan, R., Huang, X., Ni, Y., Kumar, H., Feng, Y., Cheng, Y.-C., Lin, T.-Y., Lee, C.-M., Hsu, C.-C., Sheikh, J., Michel, A., Gross, W., Weinmann, M., Šarić, J., Lin, Y., Yang, X., Jiang, N., Lu, Y., Feng, F., Awad, A., Lucas, E., Saleem, A., Cheng, C.-H., Lin, Y.-F., Lin, T.-Y., and Hsu, C.-C. (2025a). 3rd workshop on maritime computer vision (macvi) 2025: Challenge results. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 1457–1484.
- Kiefer, B., **Quan, Y.**, and Zell, A. (2025b). Approximate supervised object distance estimation on unmanned surface vehicles. In *2025 IEEE International Conference on Advanced Robotics (ICAR)*. IEEE.
- Kristan, M., Kenk, V. S., Kovačič, S., and Perš, J. (2016). Fast image-based obstacle detection from unmanned surface vehicles. *IEEE transactions on cybernetics*, **46**(3), 641–654.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, **60**(6), 84–90.
- Kumar, V. R., Yogamani, S., Bach, M., Witt, C., Milz, S., and Mäder, P. (2020). Unrect-depthnet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8177–8183. IEEE.
- Kumar, V. R., Yogamani, S., Rashed, H., Sitsu, G., Witt, C., Leang, I., Milz, S., and Mäder, P. (2021a). Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving. *IEEE Robotics and Automation Letters*, **6**(2), 2830–2837.

- Kumar, V. R., Klingner, M., Yogamani, S., Milz, S., Fingscheidt, T., and Mader, P. (2021b). Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–71.
- Li, G., Song, Z., and Fu, Q. (2019). Small boat detection for radar image datasets with yolo v3 network. In *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, pages 1–5. IEEE.
- Li, Y. and Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, **37**(4), 50–61.
- Li, Y., Guo, Y., Yan, Z., Huang, X., Duan, Y., and Ren, L. (2022). Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2801–2810.
- Li, Y., Miao, N., Ma, L., Shuang, F., and Huang, X. (2023). Transformer for object detection: Review and benchmark. *Engineering Applications of Artificial Intelligence*, **126**, 107021.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Liu, F., Lu, Z., and Lin, X. (2022a). Vision-based environmental perception for autonomous driving. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, page 09544070231203059.
- Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., and Zhang, L. (2022b). DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*.
- Liu, Z., Zhang, Y., Yu, X., and Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, **41**, 71–93.
- Losada, C., Mazo, M., Palazuelos, S., Pizarro, D., and Marrón, M. (2010). Multi-camera sensor system for 3d segmentation and localization of multiple mobile robots. *Sensors*, **10**(4), 3261–3279.

- Luo, H., Xie, W., Wang, X., and Zeng, W. (2019). Detect or track: Towards cost-effective video object detection/tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8803–8810.
- Lyu, H., Shao, Z., Cheng, T., Yin, Y., and Gao, X. (2022). Sea-surface object detection based on electro-optical sensors: A review. *IEEE Intelligent Transportation Systems Magazine*, **15**(2), 190–216.
- Ma, H., Xia, X., Wang, X., Xiao, X., Li, J., and Zheng, M. (2022). Mocovit: Mobile convolutional vision transformer. *arXiv preprint arXiv:2205.12635*.
- Mallick, A., Stücker, J., and Lensch, H. (2020). Learning to adapt multi-view stereo by self-supervision. *arXiv preprint arXiv:2009.13278*.
- Mallot, H. A. and Basten, K. (2009). Embodied spatial cognition: Biological and artificial systems. *Image and Vision Computing*, **27**(11), 1658–1670.
- Mansour, M., Davidson, P., Stepanov, O., and Piché, R. (2019). Relative importance of binocular disparity and motion parallax for depth estimation: A computer vision approach. *Remote Sensing*, **11**(17), 1990.
- McCormac, J., Handa, A., Leutenegger, S., and Davison, A. J. (2017). Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4635.
- Mehta, S. and Rastegari, M. (2021). Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*.
- Mei, C. and Rives, P. (2007). Single view point omnidirectional camera calibration from planar grids. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3945–3951.
- Miller, D., Goode, G., Bennie, C., Moghadam, P., and Jurdak, R. (2022). Why object detectors fail: Investigating the influence of the dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4823–4830.
- Ming, Y., Meng, X., Fan, C., and Yu, H. (2021). Deep learning for monocular depth estimation: A review. *Neurocomputing*, **438**, 14–33.
- Mittal, P., Singh, R., and Sharma, A. (2020). Deep learning-based object detection in low-altitude uav datasets: A survey. *Image and Vision computing*, **104**, 104046.
- Mou, X. and Wang, H. (2018). Wide-baseline stereo-based obstacle mapping for unmanned surface vehicles. *Sensors*, **18**(4), 1085.

- Mousazadeh, H., Jafarbiglu, H., Abdolmaleki, H., Omrani, E., Monhaseri, F., Abdollahzadeh, M.-r., Mohammadi-Aghdam, A., Kiapei, A., Salmani-Zakaria, Y., and Makhsoos, A. (2018). Developing a navigation, guidance and obstacle avoidance algorithm for an unmanned surface vehicle (usv) by algorithms fusion. *Ocean Engineering*, **159**, 56–65.
- Muhammad, M. B. and Yeasin, M. (2020). Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE.
- Nirgudkar, S., DeFilippo, M., Sacarny, M., Benjamin, M., and Robinette, P. (2023). Massmind: Massachusetts maritime infrared dataset. *The International Journal of Robotics Research*, **42**(1-2), 21–32.
- Nuss, D., Reuter, S., Thom, M., Yuan, T., Krehl, G., Maile, M., Gern, A., and Dietmayer, K. (2018). A random finite set approach for dynamic occupancy grid maps with real-time application. *The International Journal of Robotics Research*, **37**(8), 841–866.
- Ouchra, H. and Belangour, A. (2021). Object detection approaches in images: a survey. In *Thirteenth International Conference on Digital Image Processing (ICDIP 2021)*, volume 11878, pages 132–141. SPIE.
- Parisotto, E. and Salakhutdinov, R. (2018). Neural map: Structured memory for deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*.
- Philion, J. and Fidler, S. (2020). Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 194–210.
- Pinggera, P., Pfeiffer, D., Franke, U., and Mester, R. (2014). Know your limits: Accuracy of long range stereoscopic object measurements in practice. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pages 96–111. Springer.
- Prasad, D. K., Rajan, D., Rachmawati, L., Rajabally, E., and Quek, C. (2017). Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Transactions on Intelligent Transportation Systems*, **18**(8), 1993–2016.
- Prasad, D. K., Prasath, C. K., Rajan, D., Rachmawati, L., Rajabally, E., and Quek, C. (2018). Object detection in a maritime environment: Performance evaluation of background subtraction methods. *IEEE Transactions on Intelligent Transportation Systems*, **20**(5), 1787–1802.

- Pu, S., Song, Y., Ma, C., Zhang, H., and Yang, M.-H. (2020). Learning recurrent memory activation networks for visual tracking. *IEEE Transactions on Image Processing*, **30**, 725–738.
- Qin, T., Li, P., and Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, **34**(4), 1004–1020.
- Qu, J., Liu, R. W., Guo, Y., Lu, Y., Su, J., and Li, P. (2023). Improving maritime traffic surveillance in inland waterways using the robust fusion of ais and visual data. *Ocean Engineering*, **275**, 114198.
- Qu, M., Cui, J., Su, T., Deng, G., and Shao, W. (2022). Video visual relation detection via 3d convolutional neural network. *IEEE Access*, **10**, 23748–23756.
- Quan, Y.**, Kiefer, B., Messmer, M., Ram Akupati, C., Graser, R., and Zell, A. (2024). Robust single-cam surround view object detection and localization using memory maps. In *International Conference on Pattern Recognition (ICPR)*, pages 105–118. Springer.
- Quan, Y.**, Kiefer, B., Messmer, M., and Zell, A. (2025a). Learning-based distance estimation for 360° single-sensor setups. In *2025 European Conference on Mobile Robots (ECMR)*, pages 1–8. IEEE.
- Quan, Y.**, Kiefer, B., Messmer, M., and Zell, A. (2025b). Lightweight multi-frame integration for robust yolo object detection in videos. In *2025 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE.
- Ram, A., Sunita, J., Jalal, A., and Manoj, K. (2010). A density based algorithm for discovering density varied clusters in large spatial databases. *International Journal of Computer Applications*, **3**.
- Rampinelli, M., Covre, V. B., De Queiroz, F. M., Vassallo, R. F., Bastos-Filho, T. F., and Mazo, M. (2014). An intelligent space for mobile robot localization using a multi-camera system. *Sensors*, **14**(8), 15039–15064.
- Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016a). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016b). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Ren, S., He, K., Girshick, R., and Sun, J. (2015a). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, **28**.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015b). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Resch, B., Lang, J., and Lensch, H. P. (2014). Local image feature matching improvements for omnidirectional camera systems. In *2014 22nd International Conference on Pattern Recognition*, pages 918–923. IEEE.
- Rey-Area, M., Yuan, M., and Richardt, C. (2022). 360monodepth: High-resolution 360° monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3762–3772.
- Robinette, P., Sacarny, M., DeFilippo, M., Novitzky, M., and Benjamin, M. R. (2019). Sensor evaluation for autonomous surface vehicles in inland waterways. In *Oceans 2019-Marseille*, pages 1–8. IEEE.
- Roussel, T., Van Eycken, L., and Tuytelaars, T. (2019). Monocular depth estimation in new environments with absolute scale. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1735–1741. IEEE.
- Saleh, A., Sheaves, M., and Rahimi Azghadi, M. (2022). Computer vision and deep learning for fish classification in underwater habitats: A survey. *Fish and Fisheries*, **23**(4), 977–999.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.
- Shi, Y., Wang, N., and Guo, X. (2023). YOLOv: Making still image object detectors great at video object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2254–2262.
- Singhal, S., Ao, Y., Maas, D., Arsenali, B., and Maranò, S. (2023). Marine vessel attitude estimation from coastline and horizon. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6149–6154. IEEE.
- Sinisterra, A., Dhanak, M., and Kouvaras, N. (2017). A usv platform for surface autonomy. In *OCEANS 2017-Anchorage*, pages 1–8. IEEE.
- Sinisterra, A. J., Dhanak, M. R., and von Ellenrieder, K. (2014). Stereo vision-based target tracking system for an usv. In *2014 Oceans-St. John's*, pages 1–7. IEEE.

- Son, E., Choi, J., Song, J., Jin, Y., and Lee, S. J. (2023). Monocular depth estimation from a fisheye camera based on knowledge distillation. *Sensors*, **23**(24), 9866.
- Stadler, D. (2023). A detailed study of the association task in tracking-by-detection-based multi-person tracking. In *Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*, volume 62, pages 59–85. KIT Scientific Publishing Karlsruhe, Germany.
- Steiniger, Y., Kraus, D., and Meisen, T. (2022). Survey on deep learning based computer vision for sonar imagery. *Engineering Applications of Artificial Intelligence*, **114**, 105157.
- Stürzl, W. and Mallot, H. A. (2006). Efficient visual homing based on fourier transformed panoramic images. *Robotics and Autonomous Systems*, **54**(4), 300–313.
- Su, L., Chen, Y., Song, H., and Li, W. (2023). A survey of maritime vision datasets. *Multimedia Tools and Applications*, **82**(19), 28873–28893.
- Tamimi, H., Andreasson, H., Treptow, A., Duckett, T., and Zell, A. (2006). Localization of mobile robots with omnidirectional vision using particle filter and iterative sift. *Robotics and Autonomous Systems*, **54**(9), 758–765.
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.
- Varga, L. A., Kiefer, B., Messmer, M., and Zell, A. (2022). Seadronessee: A maritime benchmark for detecting humans in open water. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2260–2270.
- Varlashin, V., Semakova, A., and Shmakov, O. (2020). Real-time surround view system for mobile robots. In *Proceedings of 14th International Conference on Electromechanics and Robotics “Zavalishin’s Readings” ER (ZR) 2019, Kursk, Russia, 17-20 April 2019*, pages 465–476. Springer.
- Venugopala, S. H. (2022). Comparative study of 3d object detection frameworks based on lidar data and sensor fusion techniques. In *Journal of Physics: Conference Series*, volume 2232, page 012015. IOP Publishing.
- Vijayakumar, A. and Vairavasundaram, S. (2024). Yolo-based object detection models: A review and its applications. *Multimedia Tools and Applications*, pages 1–40.
- Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7464–7475.

- Wang, C.-Y., Yeh, I.-H., and Mark Liao, H.-Y. (2024). Yolov9: Learning what you want to learn using programmable gradient information. In *European conference on computer vision*, pages 1–21. Springer.
- Wang, F., Yu, S., Dong, J., Wang, X., Gao, P., Guo, Y., and Ma, L. (2020). Bifuse: Monocular 360° depth estimation via bi-projection fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 462–471.
- Wang, J., Li, Y., Zhou, Z., Wang, C., Hou, Y., Zhang, L., Xue, X., Kamp, M., Zhang, X., and Chen, S. (2022). When, where and how does it fail? a spatial-temporal visual analytics approach for interpretable object detection in autonomous driving. *IEEE Transactions on Visualization and Computer Graphics*.
- Wang, W., Sakurada, K., and Kawaguchi, N. (2016). Incremental and enhanced scanline-based segmentation method for surface reconstruction of sparse lidar data. *Remote Sensing*, **8**(11), 967.
- Wei, J., Resch, B., and Lensch, H. P. (2014). Multi-view depth map estimation with cross-view consistency. In *BMVC*.
- Wei, Z., Yao, R., Kang, J., Chen, X., and Wu, H. (2022). Three-dimensional spectrum occupancy measurement using uav: Performance analysis and algorithm design. *IEEE Sensors Journal*, **22**(9), 9146–9157.
- Wu, J. Z., Zhang, D. J., Hsu, W., Zhang, M., and Shou, M. Z. (2023). Label-efficient online continual object detection in streaming video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19246–19255.
- Wu, X., Li, W., Hong, D., Tao, R., and Du, Q. (2021). Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. *IEEE Geoscience and Remote Sensing Magazine*, **10**(1), 91–124.
- Wu, Y., Chen, Y., Yuan, L., and Liu, Z. (2019a). Sequence level semantics aggregation for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7024–7033.
- Wu, Y., Lyu, S., and Yang, M.-H. (2019b). Sequence level semantics aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Wu, Y., Zhang, F., Wang, W., Zhu, Y., and Tang, C.-K. (2020). Video object detection with an aligned spatial-temporal memory. In *European Conference on Computer Vision*, pages 369–385. Springer.

- Yan, Q., Ji, P., Bansal, N., Ma, Y., Tian, Y., and Xu, Y. (2022). Fisheyedistill: Self-supervised monocular depth estimation with ordinal distillation for fisheye cameras. *arXiv preprint arXiv:2205.02930*.
- Yang, L., Li, L., Xin, X., Sun, Y., Song, Q., and Wang, W. (2023a). Large-scale person detection and localization using overhead fisheye cameras. In *ICCV*.
- Yang, L., Li, L., Xin, X., Sun, Y., Song, Q., and Wang, W. (2023b). Large-scale person detection and localization using overhead fisheye cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19961–19971.
- Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., and Zhao, H. (2025). Depth anything v2. *Advances in Neural Information Processing Systems*, **37**, 21875–21911.
- Yang, S., Scherer, S. A., and Zell, A. (2013). An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent & Robotic Systems*, **69**(1), 499–515.
- Yang, S., Scherer, S. A., Schauwecker, K., and Zell, A. (2014). Autonomous landing of mavs on an arbitrarily textured landing site using onboard monocular vision. *Journal of Intelligent & Robotic Systems*, **74**(1), 27–43.
- Yang, X., Lin, D., Zhang, F., Song, T., and Jiang, T. (2019). High accuracy active stand-off target geolocation using uav platform. In *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, pages 1–4. IEEE.
- Zadaianchuk, A., Seitzer, M., and Martius, G. (2023). Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in neural information processing systems*, **36**, 61514–61545.
- Zha, Q. and Huang, Y. (2022). Research on positioning and navigation of usv based on lidar. In *Proceedings of 2021 Chinese Intelligent Systems Conference: Volume I*, pages 700–709. Springer.
- Zhao, X., Pu, F., Wang, Z., Chen, H., and Xu, Z. (2019). Detection, tracking, and geolocation of moving vehicle from uav using monocular camera. *IEEE Access*, **7**, 101160–101170.
- Zhou, X., Koltun, V., and Krähenbühl, P. (2020). Centertrack: Tracking objects as points. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 278–294.
- Zhou, Y., Xia, L., Zhao, J., Yao, R., and Liu, B. (2024). Efficient convolutional neural networks and network compression methods for object detection: A survey. *Multimedia Tools and Applications*, **83**(4), 10167–10209.

- Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National science review*, **5**(1), 44–53.
- Zhu, H., Wei, H., Li, B., Yuan, X., and Kehtarnavaz, N. (2020). A review of video object detection: Datasets, metrics and methods. *Applied Sciences*, **10**(21), 7834.
- Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., Hu, Q., Nie, Q., Cheng, H., Liu, C., Liu, X., *et al.* (2018). Visdrone-det2018: The vision meets drone object detection in image challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0.
- Zhu, P., Du, D., Wen, L., Bian, X., Ling, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., *et al.* (2019). Visdrone-vid2019: The vision meets drone object detection in video challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.
- Zhu, X., Xiong, Y., Dai, J., Yuan, L., and Wei, Y. (2017a). Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2349–2358.
- Zhu, X., Xiong, Y., Dai, J., Yuan, L., and Wei, Y. (2017b). Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358.
- Zhu, X., Wang, Y., Dai, J., Yuan, L., and Wei, Y. (2017c). Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 408–417.
- Zhu, X., Dai, J., Yuan, L., and Wei, Y. (2017d). Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*.
- Zhuang, J.-y., Zhang, L., Zhao, S.-q., Cao, J., Wang, B., and Sun, H.-b. (2016). Radar-based collision avoidance for unmanned surface vehicles. *China Ocean Engineering*, **30**(6), 867–883.
- Zioulis, N., Karakottas, A., Zarpalas, D., and Daras, P. (2018). Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.
- Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, **111**(3), 257–276.

Žust, L., Perš, J., and Kristan, M. (2023). Lars: A diverse panoptic maritime obstacle detection dataset and benchmark. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20247–20257.