

Eine integrierte Datenstromverarbeitung zur Erfassung von Umweltlagebildern

Andrej Lejman¹, Nick Rüssmeier^{1,2}, Oliver Ferdinand¹, Frederic Stahl¹

¹Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), DFKI Niedersachsen, Marine Perception, Oldenburg, Deutschland

²Jade Hochschule, Fachbereich Ingenieurwissenschaften, Wilhelmshaven, Deutschland

Zusammenfassung

Sensoren werden vielfach zur Erfassung von Objekten und Objektmerkmalen eingesetzt, um ein möglichst vollständiges Situationsbewusstsein für großflächige Gebiete zu schaffen. Hierzu kann beispielsweise ein verteiltes Sensorbox-Netzwerk verwendet werden, indem die Vorteile heterogener Sensoren, das heißt Sensoren unterschiedlichster Typen, kombiniert und deren Sensordaten fusioniert werden. Ein solches Netzwerk besteht dabei aus mehreren einzelnen, geographisch verteilten Sensorbox-Knoten, die die Sensoren einbinden sowie deren Daten simultan erfassen. Über eine integrierte Datenstromverarbeitung können die heterogenen Sensordaten in einer einheitlich festgelegten Datenstruktur abgelegt und bearbeitet werden. Die Datenbasis aller Sensorbox-Knoten ermöglicht es nun, umfassende Umweltlagebilder zu erzeugen. In Häfen können so zum Beispiel Bewegungen einzelner Wasserfahrzeuge wahrgenommen und dokumentiert werden. Die Verarbeitung, Speicherung, Fusionierung sowie der Austausch von Sensordaten zwischen den Sensorbox-Knoten und möglichen Benutzern erfolgt dabei zentral über die sogenannte Middleware. Diese stellt, als Übersetzungsebene im Hintergrund, sowohl die Kommunikation als auch die Verwaltung zwischen Hardware- und Anwendungsebene zur Verfügung. Die Integration einer Middleware aus der Robotik bietet aufgrund ihrer Leistungsfähigkeit und Popularität eine valide Grundlage für eine integrierte Datenstromverarbeitung verteilter Sensoren, da – ähnlich einem Sensorbox-Knoten – heterogene Sensoren in die Roboter integriert werden, um die Umgebung wahrzunehmen.

Mehrere Middleware Alternativen wurden auf Basis der ermittelten Anforderungen, wie etwa der Bedarf eines möglichst genauen Lokalisierungssystems, und ihrer Architekturen selbst miteinander verglichen. Hierbei wurde *Robot Operating System (ROS)* als geeignetste Middleware ausgewählt. Diese Arbeit stellt die Herausforderungen und Lösungen der Integration von ROS als solches Datenstrommanagement in bestehende Sensorbox-Knoten dar und zeigt erste Anwendungen und Ergebnisse.

Keywords: Verteilte Sensornetzwerke, Middleware Integration, Robot Operating System (ROS), Heterogene Sensoren, Datenfusion.

Einleitung

Ein Sensorbox-Knoten bietet eine einheitliche Architektur für Hardware, Software und Datenstromfusion von verteilten optischen Sensoren. Solche Knoten bieten eine Plattform für mehrere Nutzer, um Sensordaten aufzuzeichnen und auszutauschen sowie Anwendungen in verschiedenen Bereichen von Forschung und Entwicklung zu entwickeln und zu evaluieren. Diese Bereiche umfassen unter anderem die Objekterkennung und -verfolgung unter Verwendung von Methoden der künstlichen Intelligenz [1]. Das Sensorbox-Netzwerk ist für den Einsatz in Großraumexperimenten konzipiert, bei denen die Entfernungen zwischen den Sensorbox-Knoten mehrere Kilometer betragen können.

Ein Netzwerk mit Sensorbox-Knoten wurde bereits in [2] vorgeschlagen und auf Basis der Middleware *Odysseus* [3] entwickelt. Das System wurde bereits erfolgreich eingesetzt, jedoch entstehen Schwierigkeiten bei der Übertragbarkeit der Systemarchitektur auf neue Anwendungsgebiete, zum Beispiel für eine automatisierte Objekterkennung in dynamischen Umgebungen oder die präzisere Zuordnung der geographischen Position von erfassten Objekten. Herausforderungen, die hierbei entstehen, sind (a) Integration neuer Sensoren in die bestehende Middleware, (b) Neukonfiguration bestehender und bereits integrierter Kameras, (c) allgemeine Wartung des Systems wie Softwareupdates, und (d) die Überwachung der Vollständigkeit und Integrität erfasster Datensätze. Diese Limitierungen beeinträchtigen die Flexibilität, Erweiterbarkeit

und Genauigkeit der Datenerfassung und -verarbeitung des bisherigen Sensorbox-Knotens. Die Anpassungsfähigkeit des Systems hängt spezifisch von der genutzten Middleware ab.

Angesichts dieser Herausforderungen besteht Bedarf an der Weiterentwicklung des Systems hin zu einem neuen, universellen Sensorbox-Knoten auf Basis einer geeigneten aktuellen Middleware, der zusätzlich für folgende Zwecke geeignet ist:

- Integration von verteilten Sensoren mit RTK-korrigierter Lokalisierung,
- Verarbeitung und Kommunikation von heterogenen Sensordaten, sowie
- Entwicklung und Erprobung neuer Anwendungen nebst Erfassung von Grundversuchsdatensätzen.

Hieraus lässt sich die folgende Zielstellung ableiten, die mit der vorliegenden Arbeit angestrebt wird:

Integration einer neuen Middleware in die Sensorbox-Knoten unter Berücksichtigung der identifizierten Anforderungen sowie der Erkenntnisse zum aktuellen Stand der Technik und Wissenschaft, um eine flexiblere Plattform für heterogene Sensoren für Forschung und Entwicklung bereitzustellen.

Methodik

Der Entwurf des Sensorbox-Netzwerks basiert auf dem Bedarf, eine robuste und gleichzeitig flexible und erweiterbare Plattform für die Datenerfassung und -verarbeitung bereitzustellen. Das Gesamtkonzept (Abb. 1) sieht vor, dass eine Vielzahl von Sensoren unterstützt wird, deren Daten fusioniert werden und daraus ein umfassendes Umweltlagebild erstellt werden kann. Es ist in drei Ebenen unterteilt: Sensor-, Middleware- und Anwendungsebene.

Die Daten werden in der Sensorebene erhoben. Sensor-Konfigurationen, die aus heterogenen, verteilten Sensoren bestehen, werden über entsprechende lokale Schnittstellen in das Sensorbox-Netzwerk integriert. Darüber hinaus können mit dem Wiedergabemodul die aufgezeichneten Sensordaten wiedergegeben werden, sodass umfangreiche Realszenarien offline verfügbar und wiederholbar sind. Die Sensordaten werden im Anschluss an die Middleware-Ebene übergeben, die den Kern des Sensorbox-Netzwerks bildet und damit den Großteil der Funktionalität des Sensorbox-Knotens bereitstellt. In dieser Ebene erfolgen (a) die intrinsischen Kalibrierungen für Sensoren zur Erfassung von Bilddaten, um mögliche Abbildungsverzeichnungen zu korrigieren, (b) die Georeferenzierung aller Sensordaten anhand des

Lokalisierungssystems (extrinsische Kalibrierung), (c) die Fusion aller heterogenen Sensordaten zu einem gemeinsamen lokalen Lagebild, und (d) die Speicherung in einer homogenen Datenstruktur zur späteren Wiederverwendung.

Die verarbeiteten Sensordaten werden in dieser Stufe des Prozesses über das Kommunikationsmodul des einzelnen Sensorbox-Knoten an die Anwendungsebene und damit in das gesamte Sensorbox-Netzwerk übertragen. Hier erfolgt die Fusion der lokalen Lagebilder von einzelnen Sensorbox-Knoten zu einem Gesamtlagebild, auf dessen Grundlage die Nutzer die jeweiligen Anwendungen entwickeln können.

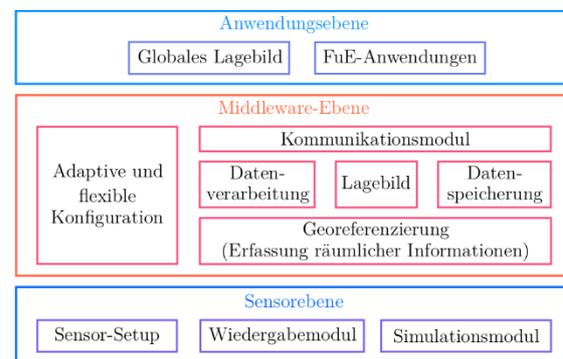


Abb. 1: Schematische Darstellung des Gesamtkonzepts mit Sensor-, Middleware- und Anwendungsebene.

Ein signifikanter Unterschied zum bereits existierenden Sensorbox-Knoten besteht in der Verwendung einer neuen Middleware. Die Entscheidung, ROS [4] als Middleware zu nutzen, ermöglicht eine flexiblere Integration und Verarbeitung der Sensordaten. Dieser Schritt adressiert damit direkt die Herausforderungen, die bei der früheren Verwendung von Odysseus aufgetreten sind. ROS ist eine Open-Source-Software zur Entwicklung wiederverwendbarer Funktionen in autonomen Systemen [5, 6]. Es bietet eine breite Auswahl an nützlichen Erweiterungen und Entwicklungswerkzeugen, einschließlich der Implementierung verschiedener Sensoren über spezifische oder universelle Datenkommunikationsprotokolle. ROS deckt die identifizierten Anforderungen nahezu vollständig ab und verfügt über eine große Gemeinschaft, die zahlreiche ROS-Pakete aktuell weiterentwickelt und bereitstellt. Der Kommunikationsmechanismus basiert auf der *publisher-subscriber*-Architektur, die eine robuste Kommunikation ermöglicht [7]. Neue ROS-Nachrichtentypen können leicht ergänzt werden. ROS verfügt dazu bereits über eine umfangreiche Bibliothek, die eine effektive

Integration heterogener Sensoren ermöglicht. Neue Module und Pakete lassen sich somit schnell entwickeln und testen. Es ist daher für Entwickler sehr flexibel und erlaubt gleichzeitig in verschiedenen Programmiersprachen zu entwickeln. Verschiedene Debugging- und Visualisierungstools ermöglichen sowohl Probleme mit der Software als auch der Hardware zu identifizieren und zu lösen. Darüber hinaus sind Module zur Visualisierung in RViz [8] verfügbar, mit denen Sensordaten in einem fusionierten Lagebild dargestellt werden können.

Um Datenverluste der Sensorbox-Knoten bei Unterbrechung der Netzwerkverbindung zu vermeiden, wurde das Multi-Master-Prinzip [9] anstelle eines einzelnen ROS-Masters für alle im Feld verteilten Sensorbox-Knoten implementiert. Dadurch ist ein Einsatz und Betrieb ohne ständige Netzwerkverbindung zwischen allen Sensorbox-Knoten möglich. Dies trägt besonders zur Ausfallsicherheit bei, da jeder Sensorbox-Knoten nun selbst die Integrität der Datenpakete sicherstellen kann, ohne dabei auf eine permanente Netzwerkverbindung angewiesen zu sein.

Abb. 2 zeigt die Verarbeitungskette der Sensordaten und die Berechnung der Objektposition. Die spezifischen Aufgaben sind in einem ersten Schritt die Erfassung der Sensordaten, separat und individuell für jeden einzelnen Sensorbox-Knoten. Nach der Korrektur der Sensordaten wird die Position von Objekten durch eine Überlagerung von Laserscanner-Punktwolken und Bilddaten in Kombination mit den von Mask R-CNN [10] generierten Segmentierungsmasken bestimmt.

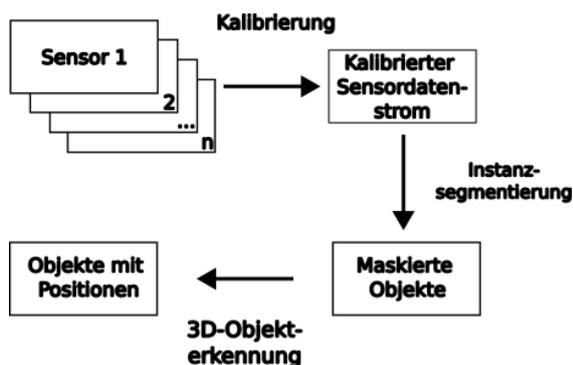


Abb. 2: Schematische Darstellung der Sensordatenverarbeitung vom einzelnen Sensor bis zum erkannten und im Raum lokalisierten Objekt.

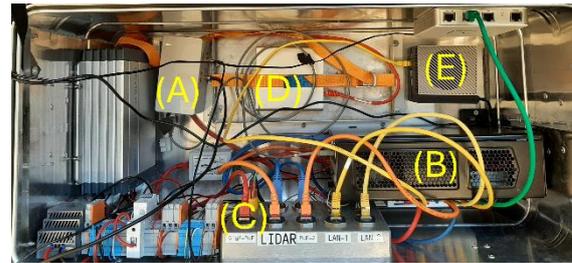


Abb. 3: Detaillierte Darstellung des Hardware-Aufbaus eines Sensorbox-Knotens: (A) GNSS- und RTK-Rover zur Berechnung der genauen Position aus der GNSS-Antenne und Korrekturdaten einer RTK-Basisstation. (B) Hardwareplattform für die Verarbeitung von Sensordaten; (C) Anschlussfeld mit Energie- und Sensoranschlüssen; (D) CloudKey als Netzwerktechnik für die Vernetzung aller im Feld verteilten Sensorbox-Knoten; (E) Edge Device für Evaluierungen von KI-Anwendungen während eines Feldeinsatzes.

Implementierung

Auf Basis der in [2] bereits erzeugten Sensorbox-Knoten wurden neben der Implementierung der neuen Middleware ROS nun auch Modifikationen und Erweiterungen an der Hardware mit zum Beispiel erweiterten Netzwerkkomponenten und Sensoren vorgenommen. Abb. 3 zeigt den inneren Aufbau eines solchen Sensorbox-Knotens.

Um im Sensorbox-Netzwerk eine infrastrukturunabhängige Kommunikation über weite Strecken im Untersuchungsgebiet zu ermöglichen, werden je nach Anforderung unterschiedliche Kommunikationsmodule verwendet: Für kurze Strecken können neben kabelgebundener Ethernet-Verbindung auch herkömmliche outdoorfähige WiFi-Antennen genutzt werden (UAP-AC-M-Pro, Ubiquiti Networks, USA). Bei größeren Distanzen hingegen können Richtfunkantennen zum Einsatz kommen (Nanobeam NBE-M5AC-500, Ubiquiti Networks, USA).

Zur Unterstützung von KI-Anwendungen im Feldeinsatz ist jeder Sensorbox-Knoten zudem mit einem Edge-Device (NVIDIA Jetson AGX Orin) ausgestattet. Durch die Implementierung von ROS auf diesem Gerät ermöglicht es eine direkte Anbindung an den Datenstrom und gewährleistet somit eine unmittelbare, leistungsfähige Verarbeitung der Sensordaten am Einsatzort.

Die eingesetzten optischen Sensoren umfassen dabei:

- einlagige Laserscanner (SICK LMS-153),
- mehrlagige Laserscanner (SICK MRS6124, 24 Lagen),
- visuelle Kameras (Basler acA2040-gc), und
- Infrarotkameras (Optris PI450).

Um Feldeinsätze besser vorzubereiten und die Handhabung der Sensoren im Feld flexibler zu gestalten, werden alle Sensoren eines Sensorbox-Knotens auf einem speziell entwickelten Sensorträger montiert und ausgerichtet (Abb. 4). Der Sensorträger integriert Sensoren, die Primärdaten wie Bilddaten erzeugen, sowie weitere Sensoren, die Zusatzinformationen zur Lokalisierung bereitstellen, beispielsweise die Distanz vom Sensor zum Objekt. Die relative räumliche Ausrichtung und Distanzen von Sensoren auf einem Trägersystem werden zuvor einmalig vorbestimmt. Hierüber wird eine extrinsische Kalibrierung vereinfacht und die automatisierte Erfassung der Objekte und ihrer Position im Raum erleichtert. Obwohl Kameras primär nur zweidimensionale Informationen von Objekten liefern, ist es möglich, diese präzise mit den Distanzinformationen der von den Laserscannern erzeugten Punktwolken zu überlagern. Dadurch wird eine Georeferenzierung der Objekte ermöglicht.

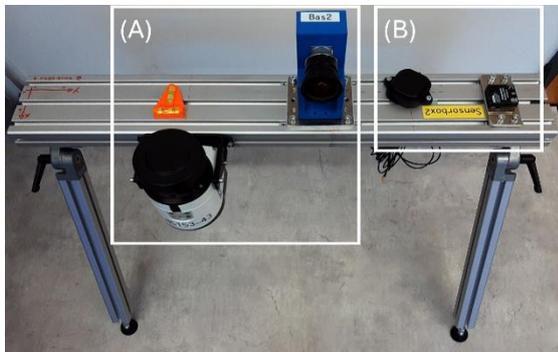


Abb. 4: Sensorträger zur direkten Implementierung von Lokalisierungs- und Orientierungsänderungen während des Einsatzes mit (A) Laserscanner (LMS-153) und visueller bzw. Infrarotkamera (blaues Gehäuse), und (B) Positionierungsmodul mit GNSS-Antenne und IMU-Einheit.

Eine absolute Position der jeweiligen Sensorträger wird durch ein präzises Lokalisierungssystem erfasst. Hierfür werden Multiband-GNSS-Informationen mit Echtzeit-Kinematik-Korrekturen (RTK) von einer nahegelegenen Referenzstation kombiniert.

Die Orientierungsinformationen der im Feld eingesetzten Sensoren werden in Verbindung mit einer Inertial Measurement Unit (IMU) direkt in den Datenstrom integriert. Die Georeferenzierung aller Sensordaten ist entscheidend für die Genauigkeit der Lagedarstellung.

Anwendungsszenario maritime Lagebilder

Um die Wirksamkeit und Anpassungsfähigkeit der Sensorbox-Knoten zu demonstrieren, werden spezifische Anwendungsfälle in realen Umgebungen getestet. Die Tests konzentrieren sich auf die Erfassung und Analyse von Daten in dynamischen Umgebungen, wie sie in einem Jachthafen auftreten. Die Validierung in solchen Szenarien bestätigt die Fähigkeit des Sensorbox-Netzwerks, vollständige und fehlerfreie Datensätze zu liefern, die für Forschungs- und Entwicklungsanwendungen unerlässlich sind.

Ein Beispiel für den Einsatz der neu entwickelten Sensorbox-Knoten zur Erstellung maritimer Lagebilder ist der Nassauhafen in Wilhelmshaven. Die Übersichtskarte in Abb. 5 zeigt die verteilten Standorte der Sensorbox-Knoten, die an Land mit Sensor Ausrichtung aufs Wasser aufgestellt wurden. Sie dienen dazu, Schiffsbewegungen zu erkennen, ihre Position zu verfolgen und ein umfassendes Lagebild des Hafengebiets zu erstellen.

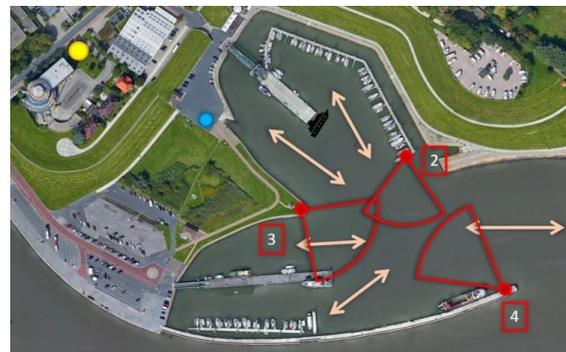


Abb. 5: Einsatz der Sensorbox-Knoten am Nassauhafen in Wilhelmshaven, Deutschland. Rote Punkte markieren Aufstellorte der Sensorträger mit angedeuteten Erfassungsbereichen (rote Kreisabschnitte). Die vorrangig relevanten Bewegungsrichtungen der Objekte sind durch Pfeile angedeutet. Gelber Punkt: Aufstellort der GNSS-Referenz.

Dieser Hafen wurde aufgrund seiner Merkmale und Gegebenheiten ausgewählt, um die Fähigkeit des Sensorbox-Netzwerks umfassend zu demonstrieren. Er zeichnet sich durch eine vielfältige Umgebung mit

unterschiedlichem Verkehrsaufkommen aus. Schiffstypen unterschiedlicher Größe, einschließlich Motoryachten und kleinerer Boote, führen verschiedene Manöver in unterschiedlichen Bereichen des Hafens durch. Darüber hinaus kommen regelmäßig dynamische Szenen vor, zum Beispiel Segelboote, die an Regatten teilnehmen. Die Vielfalt der Szenarien und die dynamischen Veränderungen im Hafengebiet illustrieren, warum der Nassauhafen ein gutes Testfeld für das Sensorbox-Netzwerk darstellt.

In diesem Zusammenhang wurden im September 2022 insgesamt drei Sensorbox-Knoten an drei verschiedenen Positionen eingesetzt. Dabei wurde ein realistisches dynamisches Szenario im Hafengebiet aufgezeichnet, das einen umfassenden Datensatz darstellt und für die Forschung und Entwicklung weiterer Anwendungen genutzt werden kann.

Ergebnisse

Der im Nassauhafen aufgezeichnete Datensatz, enthält Aufnahmen von Booten unter verschiedenen Bedingungen und Konfigurationen. Die gewonnenen Daten dienen als Grundlage für die Berechnung der Objektposition (Abb. 6). Das Mask R-CNN-Modell ermöglicht hier bereits die Instanzsegmentierung einzelner Fahrzeuge, basierend auf vortrainierten Gewichten (COCO-Gewichte [11]), die jedoch auch irrelevante Klassen für die 3D-Erkennung segmentieren und Segelboote manchmal als mehrere Instanzen erkennen. Eigene annotierte Daten wurden hierfür noch nicht verwendet.

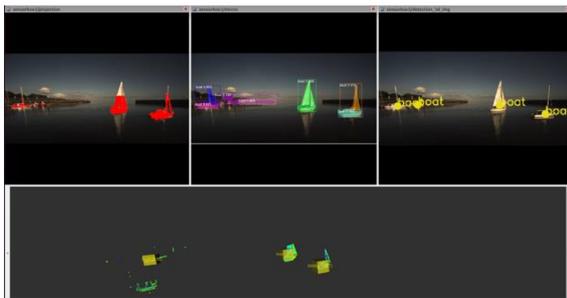


Abb. 6: Darstellung der Daten mit dem ROS-Visualisierungstool RViz. Oben links: Überlagerung der Punktwolkendaten der erfassten Objekte; Mitte: Überlagerung von Mask R-CNN mit 2D-Segmentierungsmaske der Instanzen; Rechts: 3D-Objekterkennung der Boote; Unten: Zeigt die 3D-Welt, einschließlich der Erkennungen als gelbe Rechtecke.

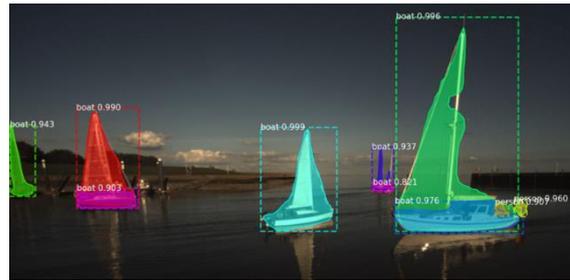


Abb. 7: Beispielhaftes Resultat der Objekterkennung mit Mask R-CNN auf RGB-Bildern der Sensorbox am Standort 3, basierend auf ResNet-101 mit vortrainierten COCO-Gewichten.

Anschließend wird die Segmentierungsmaske genutzt, um ein detailliertes Verständnis der räumlichen Struktur eines Segelbootes zu erhalten. Die Ergebnisse zeigen deutlich, dass Mask R-CNN in der Lage ist, Segelboote zu erkennen und zu segmentieren (Abb. 7). Das Modell ist zudem robust gegenüber kurzen Verdeckungen eines Objektes. Bei der Anwendung von Laserdaten in Kombination der Segmentierungsmaske ist es besonders wichtig, die Position des Segelbootes zu berechnen. Um die dem Segelboot entsprechenden Datenpunkte aus der Punktwolke zu separieren, wird die Distanzinformationen der Laserdaten über die entsprechende Segmentierungsmaske gelegt. Anschließend kann die Position der Koordinaten des Segelbootes bestimmt und als ROS-Nachricht ausgegeben werden.

Diskussion

Das Sensorbox-Netzwerk zeigt ein robustes Verhalten gegenüber Sensorausfällen. Es ist möglich, ausgefallene oder neue Sensoren mit geringem Aufwand in die aktive Aufzeichnung aufzunehmen. Die ROS-Knoten werden dabei im ROS-Berechnungsgraphen durch die Publisher-Subscriber-Architektur isoliert ausgeführt, sodass diese keinen Einfluss auf andere ROS-Knoten nehmen, wenn beispielsweise ein Sensor des ROS-Knotens ausgefallen ist.

Die neuen Sensorbox-Knoten sammeln Daten von verschiedenen Sensoren, die miteinander verbunden und in ein ROS-basiertes System integriert sind.

Die Funktionen dieser Sensorbox-Knoten umfassen in dieser entwickelten Version nun zusätzlich die:

- Intrinsische und extrinsische Kalibrierung aller Sensoren pro Sensorbox-Knoten,
- Georeferenzierung aller verwendeten Sensorbox-Knoten mittels GNSS und RTK-Korrekturen,

- Direkte Steuerung der wichtigsten Sensorparameter über ROS-Befehle ohne proprietäre Software,
- Erfassung vollständiger 3D-Informationen einzelner Objekte aus den im Testfeld gesammelten Sensordaten,
- Auswahl, Start oder Stopp von ROS-Themen, die während des Feldeinsatzes benötigt werden,
- Umfängliche Aufnahmekontrolle der Dateneingabe und -ausgabe.

In der Ausbaustufe verfügt der ROS-basierte Sensorbox-Knoten noch nicht über eine grafische Benutzeroberfläche (GUI) zur Erstellung von Lagebildern. Die vollständige Steuerung des Sensorbox-Knotens erfolgt über das Linux-Terminal. Das ROS-eigene Paket *rqt* bietet Plugins, die auch eine GUI-basierte Steuerung des Sensorbox-Knotens ermöglichen. Das *rqt*-Paket enthält jedoch keine expliziten Plugins zur Erstellung von Lagebildern, da es primär für die Robotik-Anwendung entwickelt wurde. Es ist weiterhin denkbar, eine GUI für die ROS-basierten Sensorbox-Knoten zu entwickeln, die eine komfortable und vollständige Bedienung im Feldeinsatz ermöglicht.

Zusammenfassung und Ausblick

In großflächigen und dynamischen Umgebungen ermöglichen verteilte Sensorbox-Knoten die Fusionierung eines umfassenden Lagebildes. Sensordaten heterogener, verteilter Sensoren und Sensorsysteme werden dabei in eine homogene Datenstruktur überführt. Dabei können aus realen dynamischen Szenarien der Testfeldumgebung umfassende Datensätze gewonnen und für wissenschaftliche sowie technische Anwendungen in multidisziplinären Bereichen in weiteren Anwendungen genutzt werden. Der ROS-basierte Sensorbox-Knoten im Netzwerk liefert hierfür vollständig georeferenzierte Informationen annotierter Daten. Die aktuelle Entwicklung beinhaltet daher auch Aspekte für plattformunabhängige Anwendungen.

In Beispielanwendungen zur segmentierten Objekterkennung auf Basis verschiedenster Sensordaten zeigen diese umfassenden Funktionalitäten, auch ohne vortrainierte Gewichtung. Im Ausblick sind allerdings weitere Entwicklungsschritte erforderlich, um die Benutzerfreundlichkeit zu steigern. Die Erweiterung des Systems um eine zentrale GUI steht hierbei im Mittelpunkt der zukünftigen Entwicklungsaktivitäten. Damit wird eine expertenunabhängige und gleichzeitig umfassende Kontrolle des Systems gewährleistet. Perspektivisch könnten auch die

aktuellen Daten und erfassten Objekte direkt im Feldeinsatz hierüber visualisiert werden.

In der aktuellen Implementierung wurden noch keine eigenen annotierten Daten verwendet. Die Integration derartiger Daten wird als nächster wichtiger Schritt zur Verbesserung der Modellgenauigkeit und zur Anpassung an spezifische Anforderungen angesehen. Dadurch würde nicht nur die Genauigkeit der Objekterkennung erhöht, sondern auch die Segmentierung weiter verbessert und an die spezifischen Bedingungen des Einsatzgebietes angepasst.

Danksagung

Die Autoren danken der Hafenaufsicht Wilhelmshaven und Niedersachsen Ports für die Unterstützung bei den Feldversuchen, insbesondere Herrn Helmut Wessels der Schiffsmeldestelle. Der Hauptteil der Arbeit wurde im Rahmen des Projektes CoPDA durchgeführt. Das Projekt wurde vom Bundesministerium für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 01IW19003 gefördert.

Literaturnachweis

- [1] A. Dittmer et al., *Robotik in der Wirtschaftsinformatik. Edition HMD. Springer Vieweg, Wiesbaden* (2023); doi: 10.1007/978-3-658-39621-3_16
- [2] N. Rüssmeier et al., *Journal of Sensors and Sensor Systems* 6, 37-52 (2017); doi: 10.5194/jsss-6-37-2017
- [3] D. Geesen et al., *Ambient Assisted Living. Advanced Technologies and Societal Change. Springer, Berlin, Heidelberg* (2012); doi:10.1007/978-3-642-27491-6_16
- [4] M. Quigley et al., *ICRA workshop on open source software* Bd.3, 3.2 Kobe, Japan, S. 5 (2009).
- [5] Stanford Artificial Intelligence Laboratory et al. (2018). Verfügbar unter: <https://www.ros.org>
- [6] S. Macenski et al., *Science Robotics* 7, (2022); doi: 10.1126/scirobotics.abm6074
- [7] P. Eugster et al., *ACM Comput. Surv.* 35, 114-131 (2003); doi: 10.1145/857076.857078
- [8] A. Mahtani et al., *Packt Publishing, Birmingham, UK.* (2018); ISBN: 9781788623650.
- [9] M. Garzón et al., In: *Robot operating system (ROS)*, 449-483 (2017). doi: 10.1007/978-3-319-54927-9_14
- [10] K. He et al., *Computer Science in arXiv, Cornell University* (2017); doi: 10.48550/arXiv.1703.06870
- [11] T.-Y. Lin et al., *CoRR* (2014); doi: 10.48550/arXiv.1405.0312