

# IMU-Kalibrierung und Lernverfahren mittels Transformer-Netzwerk für die Lokalisierung<sup>1</sup>

## IMU Calibration and Learning Using a Transformer Network for Localization

Hossein Shoushtari, Harald Sternberg

Inertiale Lokalisierungsansätze spielen eine zentrale Rolle bei Navigationssystemen und basieren üblicherweise auf der doppelten Integration von dynamischen Beschleunigungssensorwerten. Diese Verfahren stoßen jedoch auf erhebliche Herausforderungen bei der präzisen Verfolgung komplexer Bewegungen, wie sie etwa bei Smartphones oder Wearables auftreten, und neigen dazu, im Laufe kurzer Zeit zu drifteten. Darüber hinaus wirft das Bestreben, Lokalisierung mit diesen handelsüblichen Geräten zu realisieren, die Frage auf, ob eine individuelle Kalibrierung der integrierten Inertialsensoren (engl.: Inertial Measurement Unit, IMU) erforderlich ist, was sich in der Praxis kaum umsetzen ließe. Der Beitrag verfolgt das Ziel, diese Herausforderungen durch einen State-of-the-Art-überwachten Lernansatz zu beschreiben. Dabei wird ein neuartiges Transformer-Netzwerk eingesetzt, um sowohl die Kalibrierung als auch die Lokalisierung simultan zu lösen. Die Verwendung von Lernansätzen als Sensorkalibrierungsmethode kann die Genauigkeit der Vorhersage verbessern. Die Integration von wissensbasierten Merkmalen in das Transformer-Modell ermöglicht eine Verbesserung der Genauigkeit und Zuverlässigkeit der Positionsbestimmung. Die transformatorbasierte Attention-Implementierung dieses Ansatzes verbindet physikalische und lernbasierte Verfahren zu einer fortschrittlichen Methode, die eine genaue und zuverlässige Lokalisierung in verschiedenen Geräteplatzierungen ermöglicht. Darüber hinaus werden umfassende analytische Bewertungen mit einer realistischen Qualitätsbeurteilung unter Verwendung mehrerer Metriken vorgestellt.

**Schlüsselwörter:** IMU, Kalibrierung, Positionsbestimmung, überwachtes Lernen, Transformer-Netzwerk, inertielle Lokalisierung

*Inertial localization approaches play a central role in navigation systems and are usually based on the double integration of dynamic acceleration sensor values. However, these methods face significant challenges in accurately tracking complex movements, such as those that occur in smartphones or wearables, and tend to drift over short periods of time. Furthermore, the desire to implement localization with these commercially available devices raises the question of whether individual calibration of the integrated inertial sensors (Inertial Measurement Unit, IMU) is necessary, which would be difficult to implement in practice. This work aims to overcome these challenges through a state-of-the-art supervised learning approach. A novel transformer network is used to solve both calibration and localization simultaneously. The use of learning approaches as a sensor calibration method can improve the accuracy of the prediction. The integration of knowledge-based features into the transformer model enables an*

<sup>1</sup> Überarbeiteter und erweiterter Beitrag der 23. Internationalen Geodatischen Woche Obergurgl 2025.

*improvement in the accuracy and reliability of position determination. The transformer-based attention implementation of this approach combines physical and learning-based methods into an advanced method that enables accurate and reliable localization in complex device placements. In addition, comprehensive analytical evaluations with realistic quality assessment using multiple metrics are presented.*

*Keywords: IMU, calibration, position determination, supervised learning, transformer network, inertial localization*

## 1 EINLEITUNG

Die inertielle Lokalisierung spielt eine entscheidende Rolle in verschiedenen Anwendungen, einschließlich mobiler Ortungsdienste, Augmented Reality und Servicerobotern. Diese Technologien ermöglichen eine effiziente Verwaltung von Personenströmen, unterstützen bei der Navigation und verbessern das Management von Veranstaltungen und Flächen /Chen et al. 2018/. Schließlich soll eine Navigation gleichermaßen für ein Einkaufszentrum, ein Krankenhaus oder auch einen bewaldeten Friedhof funktionieren können /Willemsen et al. 2013/. Bis zum Jahr 2024 hat die Zahl der Smartphone-Nutzer weltweit etwa 7,1 Milliarden erreicht, was bedeutet, dass inzwischen jeder Mensch auf der Erde ein Smartphone besitzen kann. In Deutschland liegt die Smartphone-Verbreitung allein bei über 80 % der Bevölkerung /Statista 2024/.

Viele Smartphone-Sensoren liefern eine Vielzahl von Daten, aber bedeutet dies auch, dass diese Sensoren kalibriert werden müssen, wie es bei geodätischen Ingenieuraufgaben üblich ist? Dies ist offensichtlich keine partikuläre Lösung, aber in gewisser Weise ist es vielleicht möglich, sich über die Qualität der geschätzten Ergebnisse sicher zu sein, jedoch möglicherweise ohne direkten Kalibrierungsprozess oder eine Schätzung der Kalibrierungsparameter.

Die gesteigerte Rechenleistung und die Möglichkeit, große Datenmengen zu sammeln, haben in den letzten Jahren den Einsatz von Methoden des maschinellen Lernens ermöglicht. Datengetriebene (lernbasierte) Ansätze und Modelle, die auf Bewegungssensordaten, wie beispielsweise Beschleunigungs- und Gyroskop-Sensoren basieren, können die Gehgeschwindigkeit schätzen. Dadurch wird Drift kompensiert und die Gesamtleistung der Positionsbestimmungssysteme erheblich verbessert /Yan et al. 2017/. Dies ist eine der gängigen Methoden, um Drift zu kompensieren, ohne zusätzliche Sensoren oder Umgebungsinformationen hinzuzufügen /Klingbeil 2023/.

Das Ziel lernbasierter Ansätze besteht darin, eine präzise Fußgängertrajektorie zu schätzen, wobei Smartphones oder andere tragbare Geräte unabhängig von ihrer jeweiligen Platzierung mit nur einem Modell verarbeitet werden können. Dazu werden die Daten der Inertialsensoren (engl.: Inertial Measurement Unit, IMU) zunächst in ein einheitliches lokales Koordinatensystem transformiert /Yan et al. 2017/. Jüngste Fortschritte in diesem Bereich umfassen die Integration mehrerer Netzwerke zur Verbesserung der inertialen Lokalisierung. In diesem Zusammenhang beschreiben wir ein Transformer-Modell mit automatischer Merkmalsinteraktion, auch als TAFI (engl.: Transformer with Automatic Feature Interaction) /Shoushtari et al. 2024/. Dieses Modell integriert nahtlos wissensbasierte (oder physikbasierte) Merkmale in das Deep-Learning-Modell und steigert dessen Effektivität damit erheblich. TAFI ermöglicht die gleichzeitige

Verarbeitung von Rohdaten und wissensbasierten Merkmalen innerhalb des Transformer-Netzwerks /Vaswani et al. 2017/, einer State-of-the-Art-Architektur, wie sie auch im Hintergrund von ChatGPT zum Einsatz kommt (siehe Abschnitt 4). Dies führt zu einer präzisen Schätzung der Gehgeschwindigkeit in Form von Geschwindigkeitsvektoren und damit zur Berechnung der relativen Position für die Indoor-Positionierung mit Bewegungssensoren in Smartphones.

Ein zentraler Aspekt dieses Modells ist die Integration der Sensorkalibrierung direkt in die Positionierungsmethode /Shoushtari et al. 2023a/. Dadurch erfolgt die Kalibrierung implizit auf der Ebene der Positionsschätzung. Die Kalibrierung von Sensoren ist entscheidend für die Gewährleistung der Genauigkeit und Zuverlässigkeit von Sensorsystemen. Dieser Prozess umfasst die Festlegung einer funktionalen Beziehung, welche die Interaktion zwischen den Sensordaten berücksichtigt und Fehlerquellen wie Bias, Skalierungsfehler und Orthogonalitätsfehler einbezieht. Das Ziel ist es, die Messungen mit einer bekannten, fehlerfreien Referenz abzugleichen und die Kalibrierungsparameter für die zukünftige Nutzung zu schätzen. Traditionell wird die Kalibrierung als mathematisches Optimierungsproblem betrachtet. Die Lösung solcher Optimierungsprobleme, insbesondere, wenn sie eine nichtlineare Struktur aufweisen, kann durch den Einsatz von Lernverfahren erfolgreich bewältigt werden. In diesem Beitrag wird ein Modell vorgestellt, das traditionelle Kalibrierungsverfahren mit modernen maschinellen Lernmethoden für die inertielle Lokalisierung integriert. Dieser Ansatz ist besonders geeignet für Mikrosystem-Sensoren, die keine zeitstabilen Kalibrierungsparameter liefern können.

Der Aufbau dieses Beitrags gliedert sich wie folgt: Abschnitt 2 bietet einen umfassenden Überblick über verschiedene Ansätze zur inertialen Lokalisierung und Kalibrierung, wobei die Unterschiede zwischen wissensbasierten und lernbasierten Methoden besonders hervorgehoben werden. Im anschließenden Abschnitt werden diese Ansätze diskutiert, und es wird eine Lösung vorgeschlagen, welche die Vorteile beider Methoden, physikbasierte und lernbasierte Verfahren, kombiniert, um eine optimierte Leistung zu erreichen. Abschnitt 3 beschäftigt sich mit der Architektur des verwendeten Transformer-Netzwerks, das für die Positionsbestimmung und Kalibrierung eingesetzt wird. In Abschnitt 4 werden die experimentelle Methodik, die Datenerhebung und die Vorbereitung der Daten im Detail beschrieben. Die letzten beiden Abschnitte konzentrieren sich auf die Bewertung der erzielten Ergebnisse, fassen die wesentlichen Erkenntnisse zusammen und bieten einen Ausblick auf zukünftige Forschungsmöglichkeiten in diesem Bereich.

## 2 THEORETISCHER HINTERGRUND

### 2.1 Wissensbasierte vs. datengetriebene Ansätze zur Positionsbestimmung

Inertialsensoren bieten einen wissensbasierten Ansatz zur Positionsbestimmung in Innenräumen, indem sie Beschleunigungen und Winkelgeschwindigkeiten erfassen. Algorithmen wie der von /Madgwick 2010/ entwickelte Filter ermöglichen die Berechnung von Rotationsvektoren und Positionsschätzungen durch Integration. Allerdings sind IMUs anfällig für Verzerrungen und Rauschen, insbesondere bei den in Smartphones verbauten Sensoren, was zu Ungenauigkeiten in den Messungen führen kann /Willemsen et al. 2015/. Koppelnavigationsmethoden für Fußgänger (engl.: Pedestrian Dead Reckoning, PDR) tragen dazu bei, diese Probleme zu mildern, indem sie Schrittereignisse erkennen und somit die Positionsgenauigkeit verbessern /Li et al. 2012/. Dennoch bleibt die inertielle Langzeitverfolgung aufgrund kumulierter Fehler ein ungelöstes Problem /Solin et al. 2018/. Daher haben datengetriebene Methoden sich erheblich weiterentwickelt und nutzen maschinelle Lerntechniken, um die Langzeitverfolgung zu verbessern /Chen et al. 2018/. Frühe Ansätze nutzten überwachte Lernverfahren, um Aktivitätstypen (z. B. Laufen oder Rennen) auf Basis von IMU-Daten zu erkennen /Allahbakhshi 2021/ sowie um die Geschwindigkeitsvektoren zu schätzen /Giarmatzis et al. 2012/. Das bedeutet, dass sich auf Basis von IMU-Daten sowohl Klassifikationsaufgaben zur Unterscheidung von Aktivitätstypen als auch Regressionsaufgaben zur Schätzung von Bewegungsparametern in der Innennavigation formulieren lassen. Mit ausreichend Daten und dem Einsatz moderner Machine-Learning-Methoden kann dann eine präzisere Lokalisierung möglich werden.

In einem robusten IMU-Doppelintegrationsdatensatz (engl.: Robust IMU Double Integration, RIDI) konnten die Autoren Geschwindigkeitsvektoren in einem gerätebezogenen Koordinatensystem modellieren und ein rekurrentes neuronales Netz einsetzen, um die Inertialnavigation ohne externe Orientierungsdaten zu optimieren /Yan et al. 2017/, /Chen et al. 2018/. Die robuste neuronale Inertialnavigation (engl.: Robust Neural Inertial Navigation, RoNIN) erweitert diese Ansätze durch den Einsatz fortschrittlicher Architekturen, welche die Lokalisierungsgenauigkeit in schwierigen und variablen Gerätepositionen verbessern /Yan et al. 2019/. Ein weiterer Fortschritt ist das kontextuelle Transformatorennetzwerk für die Inertialnavigation (engl.: Robust Contextual Transformer Network for Inertial Navigation, CTIN). Dieses Verfahren kombiniert einen auf Residual Networks basierenden Encoder mit einem Transformer, um sowohl räumliche als auch zeitliche Informationen zu erfassen und dadurch

die Schätzgenauigkeit deutlich zu verbessern /Rao et al. 2021/. Die neuronale Inertial-Lokalisierung (engl.: Neural Inertial Localization, NILoc) verwendet Transformations-Modelle zur Kartenanpassung unter Verwendung von Trajektorienpunkten, um selbst präzise Karten zu erstellen /Herath et al. 2020/.

### 2.2 Direkte vs. numerische Optimierungsverfahren zur Sensorkalibrierung

Die Kalibrierung von Sensoren stellt ein Optimierungsproblem dar, das sowohl durch direkte als auch durch numerische Ansätze gelöst werden kann (Abb. 1). Direkte Methoden, wie der Gauß-Markov-Algorithmus, lösen Kalibrierungsgleichungen in einer endlichen Anzahl von Schritten und liefern die Ergebnisse, insbesondere wenn die Problemgröße überschaubar ist und die Sensoren zeitstabile Kalibrierparameter bereitstellen können. Direkt bedeutet in diesem Zusammenhang, dass man die Ausgleichungsergebnisse in einem Schritt berechnen kann. Dies steht nicht im Widerspruch zu iterativen Verfahren und nichtlinearen Verbesserungen für die Schätzung von Näherungswerten. Diese Methoden berechnen sowohl die Kalibrierungsparameter der Sensoren als auch deren Qualität und verwenden die Eingabedaten entsprechend. Im Gegensatz dazu verfeinern numerische Optimierungsmethoden, wie das Gradientenabstiegsverfahren, die Lernparameter durch wiederholte Anpassungen. Diese Verfahren sind besonders nützlich für komplexe oder groß angelegte Systeme, bei denen exakte Lösungen schwer zu erzielen sind. Numerische Verfahren bieten Flexibilität und sind effektiv im Umgang mit nichtlinearen Systemen, während direkte Methoden bei sehr großen Datensätzen aufgrund von Rechen- und Speicherbeschränkungen weniger praktikabel sein können.

Im Folgenden werden die Ausgleichung nach Gauss-Markov und das Gradientenabstiegsverfahren als Beispiele im Rahmen des überwachten Lernens zur Kalibrierung eines Beschleunigungssensors erläutert. Die Kalibrierung eines Beschleunigungssensors kann ohne zusätzliche Referenzmessungen erfolgen, vorausgesetzt, die lokale Erdbeschleunigung ist ausreichend bekannt. Diese Erdbeschleunigung dient als Referenzwert, gegen den die Messdaten des Sensors verglichen werden können. Bei der Kalibrierung werden die Messdaten des Sensors mit Referenzdaten abgeglichen, um mögliche Abweichungen zu erkennen und zu korrigieren. Der funktionale Zusammenhang zur Kalibrierung eines Beschleunigungssensors wird durch die folgenden Gleichungen verdeutlicht:

$$\begin{pmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{pmatrix} = \begin{pmatrix} s_x & r_{xy} & r_{xz} \\ 0 & s_y & r_{yz} \\ 0 & 0 & s_z \end{pmatrix} \times \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}. \quad (1)$$

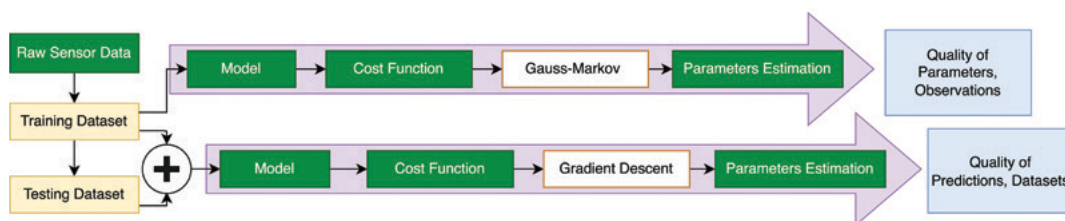


Abb. 1 | Überblick über den Workflow von direkten und numerischen Optimierungsverfahren /Shoushtari et al. 2023b/

Hierbei stellt  $\hat{a}$  die kalibrierten Beschleunigungswerte dar,  $s$  sind die Skalierungswerte,  $a$  sind die rohen Messwerte und  $b$  ist der Offset.

$$g = \sqrt{\hat{a}_x^2 + \hat{a}_y^2 + \hat{a}_z^2}. \quad (2)$$

Diese Gl. (2) berechnet die resultierende Erdbeschleunigung ( $g$ ) auf der Grundlage der kalibrierten Beschleunigungswerte, wenn sich der Sensor in Ruhe befindet und keine dynamischen Bewegungen vorliegen. Gl. (3) beschreibt den funktionalen Zusammenhang aus der Berechnung der Schwerebeschleunigung  $g$  (Gl. (2)) und dem modellierten Fehlermodell (Gl. (1)).

$$g^2 = (s_x \cdot a_x + r_{xy} \cdot a_y + r_{xz} \cdot a_z + b_x)^2 + (s_y \cdot a_y + r_{yz} \cdot a_z + b_y)^2 + (s_z \cdot a_z + b_z)^2. \quad (3)$$

Man kann die neun Kalibrierungsparameter  $s$ ,  $a$  und  $b$  auch als neun unspezifische Parameter  $w_i$  schreiben. Somit ergibt sich durch Umformung von Gl. (3) die Gl. (4) unter der Hinzunahme der Gln. (5) bis (14).

$$g^2 = b + w_1 \cdot a_x + w_2 \cdot a_y + w_3 \cdot a_z + w_4 \cdot a_x^2 + w_5 \cdot a_y^2 + w_6 \cdot a_z^2 + w_7 \cdot a_x a_y + w_8 \cdot a_y a_z + w_9 \cdot a_x a_z. \quad (4)$$

Hierbei ist Gl. (4) identisch mit Gl. (3), wenn:

$$b = b_x^2 + b_y^2 + b_z^2, \quad (5)$$

$$w_1 = 2s_x \cdot b_x, \quad (6)$$

$$w_2 = 2(r_{xy} \cdot b_x + s_y \cdot b_y), \quad (7)$$

$$w_3 = 2(r_{xz} \cdot b_x + r_{yz} \cdot b_y + s_z \cdot b_z), \quad (8)$$

$$w_4 = s_x^2, \quad (9)$$

$$w_5 = r_{xy}^2 + s_y^2, \quad (10)$$

$$w_6 = r_{xz}^2 + r_{yz}^2 + s_z^2, \quad (11)$$

$$w_7 = 2s_x \cdot r_{xy}, \quad (12)$$

$$w_8 = 2(r_{xy} \cdot r_{xz} + s_y \cdot r_{yz}), \quad (13)$$

$$w_9 = 2s_x \cdot r_{xz}. \quad (14)$$

Gl. (3) wird in der Gauss-Markov-Ausgleichung verwendet. Hier werden die Designmatrix  $A$ , basierend auf der linearen Version der Gl. (3), und ein Beobachtungsvektor  $l$  erstellt. Die Designmatrix  $A$  enthält die partiellen Ableitungen der unbekanntenen Kalibrierungsparameter gemäß den Beobachtungsgleichungen. Der Vektor  $l$  ent-

hält die Referenzinformationen, in diesem Fall den Wert der lokalen Erdgravitation  $g$ . Die Schritte sind wie folgt:

$$N = A' \cdot A, \quad (15)$$

$$n = A' \cdot dl, \quad (16)$$

$$Q_x = N^{-1}, \quad (17)$$

$$d_x = Q_x \cdot n. \quad (18)$$

Für den maschinellen Lernansatz kann ein Modell entwickelt werden, das die Referenzdaten basierend auf den Messungen vorher-sagt. Gl. (4) wird hier verwendet – ggf. Verwechslung durch Umschreiben zwischen Gl. (3) und Gl. (4), um den Schwerkraftvektor anhand der gemessenen Beschleunigungsdaten in drei Achsen vorherzusagen. Der Unterschied zwischen dem vorhergesagten Wert ( $g_{w,b}(x^{(i)})$ ) und den Referenzdaten ( $y^{(i)}$ ) ergibt die Kosten, die messen, wie gut das Modell den Ziel-Schwerkraftvektor vorher-sagt. Die Kosten- oder Verlustfunktion kann wie folgt definiert werden:

$$J(w, b) = \frac{1}{2m} \sum_{i=0}^{m-1} (g_{w,b}(x^{(i)}) - y^{(i)})^2. \quad (19)$$

Das Ziel besteht darin, die Kostenfunktion zu minimieren, indem die Parameter  $w_i$  und  $b$  ermittelt werden. Dies erfolgt in Form eines numerischen Ansatzes durch Gradientenberechnung.

### 3 AUSWAHL DER SCHÄTZMETHODE

Eine zentrale Frage ist, ob ein Ansatz entwickelt werden kann, der sowohl physikalische Kontexte, wie beispielsweise Schrittlänge oder Änderungen der Schrittrichtung, als auch datengestützte Vorteile kombiniert. Insbesondere bei der Verwendung von mikroelektromechanischen Inertialsensoren, bei denen trotz ihrer Einschränkungen eine genaue Positionsbestimmung erforderlich ist, stellt sich die Frage, ob Kalibrierungsparameter berücksichtigt werden können. Diese Fragen können wir durch das Konzept der Kalibrierung des Beschleunigungssensors beleuchten. Bei der oben erwähnten Kalibrierung des Beschleunigungssensors wurden die in Abb. 2 dargestellten Ergebnisse geschätzt. Auf der rechten Seite ist der Unterschied zwischen der geschätzten lokalen Erdgravitation im Quadrat ( $g^2$ ), unter Berücksichtigung der Kalibrierungsparameter in Gl. (3),

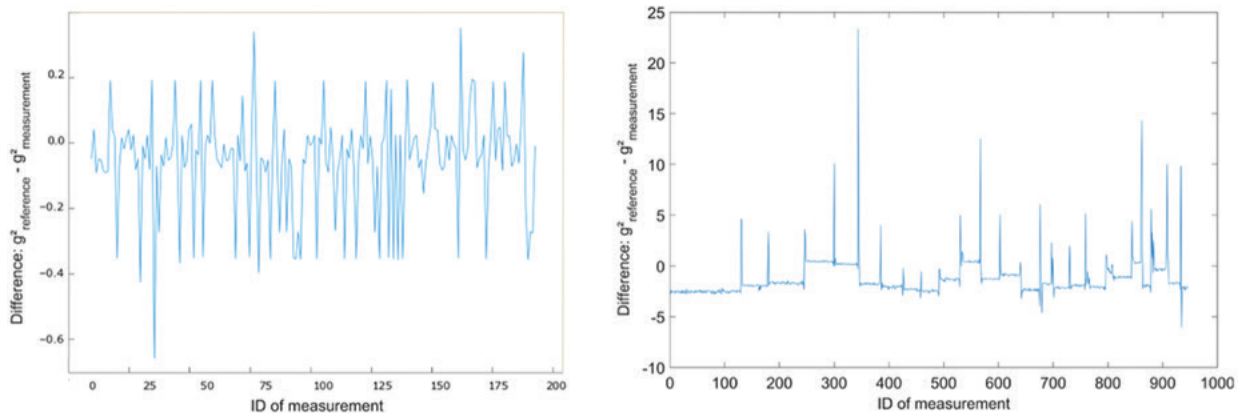


Abb. 2 | Abweichungen der Beobachtung im jeweiligen Verfahren bei der Schätzung von  $g^2$  [(m/s<sup>2</sup>)<sup>2</sup>] mit direktem (rechts) vs. numerischem (links) Optimierungsverfahren, als Konzeptnachweis

zu sehen. Die linke Seite zeigt dasselbe, jedoch nur für die Daten, die im numerischen Verfahren zur Minimierung der Kostenfunktion nicht verwendet wurden (Unseen Data). Das bedeutet: Beim direkten Ansatz werden alle zu Beginn der Ausgleichung vorliegenden Daten gemeinsam genutzt und entsprechend in der Evaluation berücksichtigt. Im Vergleich dazu werden in Lernverfahren etwa 80 % der Daten als Input oder Beobachtungen verwendet und die restlichen ca. 20 % der Daten werden nicht im Lernprozess, sondern in der Evaluation berücksichtigt.

Das überwachte Lernverfahren zeigt bessere Vorhersageergebnisse, selbst für experimentelle Daten, die nicht im Optimierungsprozess verwendet wurden. Der entscheidende Punkt ist, dass, wenn wir die Positionsbestimmung als ein gelerntes Regressionsmodell darstellen könnten, eine separate Kalibrierung möglicherweise nicht mehr erforderlich wäre. Kalibrierung und Positionsbestimmung könnten dann simultan gelöst werden. Ausführliche Informationen sind zu finden in /Shoushtari et al. 2023b/.

Die Aufgabe besteht nun darin, ein überwachtes Lernverfahren zu entwickeln, das sowohl wissensbasierte Merkmale als auch Rohdaten akzeptiert und präzise Werte für die Positionsbestimmung liefert.

## 4 TRANSFORMER-NETZWERK

TAFI nutzt eine fortschrittliche temporale Aufmerksamkeitsfusion (engl.: Attentionsfusion), indem es ein faltendes neuronales Netzwerk und transformatorbasierte Mechanismen kombiniert, um eine robuste raumzeitliche Vorhersage zu ermöglichen. Aufbauend auf der Arbeit von /Vaswani et al. 2017/ zielt dieses Netzwerk darauf ab, die Geschwindigkeitsvektoren von Sensoren zu schätzen, indem es sowohl die räumlichen Beziehungen zwischen den IMU-Sensorenwerten als auch kontextuelle Informationen einbezieht und zeitliche Abhängigkeiten zwischen den Elementen modelliert. Diese Architektur (Abb. 3) kombiniert räumliches, kontextuelles und zeitliches Decoding, um die komplexen Wechselwirkungen zwischen den IMU-Daten über die Zeit hinweg zu erfassen. Durch die Integration von eindimensionalen *Convolutional*-Schichten und *Multi-Head-Self-Attention*-Mechanismen kann das Modell sowohl lokale Muster extrahieren als auch langfristige Abhängigkeiten modellieren. Dies

verbessert die Fähigkeit zur genauen Schätzung von Objektgeschwindigkeiten erheblich.

Der Netzwerkinput besteht aus IMU-Messungen, wobei jede Spalte transformierte Beschleunigungs- und Gyroskop-Daten repräsentiert. Diese Daten werden in einem Buffer gesammelt, der zeitlich geordnete Sequenzen erzeugt. Diese Sequenzen werden dann parallel an den *Spatial Encoder* und die *Context Component* weitergeleitet.

### 4.1 Spatial Encoder

Der *Spatial Encoder* verarbeitet die Zeitsequenzen und extrahiert räumliche Einbettungen, die hierarchische Abhängigkeiten darstellen. Dieser Prozess beginnt mit der Verarbeitung der Eingabedaten durch eindimensionale *Convolutional*-Schichten, die anschließend durch *Batch-Normalisation* und eine lineare Transformation verfeinert werden. Dadurch wird die räumliche Information in einen hochdimensionalen Raum eingebettet. Diese eingebetteten Daten werden anschließend von einem *Transformer Encoder* mit zusätzlichen *Convolutional*-Schichten weiterverarbeitet.

Der *Transformer Encoder* und der *Convolutional Encoder* bestehen aus einer Anzahl von identischen Schichten  $N$ , die eine Eingabesequenz von räumlichen Einbettungen in eine Sequenz kontinuierlicher Repräsentationen transformieren. Jede Schicht kombiniert einen standardmäßigen *Transformer Encoder* mit zusätzlichen *Convolutional*-Schichten, um die Verarbeitung der Eingabesequenzen zu optimieren. Dabei verwendet der *Transformer Encoder* *Self-Attention*-Mechanismen und positionweise *Feed-Forward*-Netzwerke. Detaillierte Informationen zum Basis-Transformer-Modell finden sich bei /Vaswani et al. 2017/.

### 4.2 Kontextuelle Komponente

*Contextual Component* des TAFI-Modells integriert als kontextuelle Informationen die wissensbasierten Merkmale in die Modellvorhersagen (Tab. 1). Zunächst verarbeitet diese Komponente die Zeitsequenzen und extrahiert relevante Merkmale. Anschließend werden kontextuelle Einbettungen durch eine eindimensionale

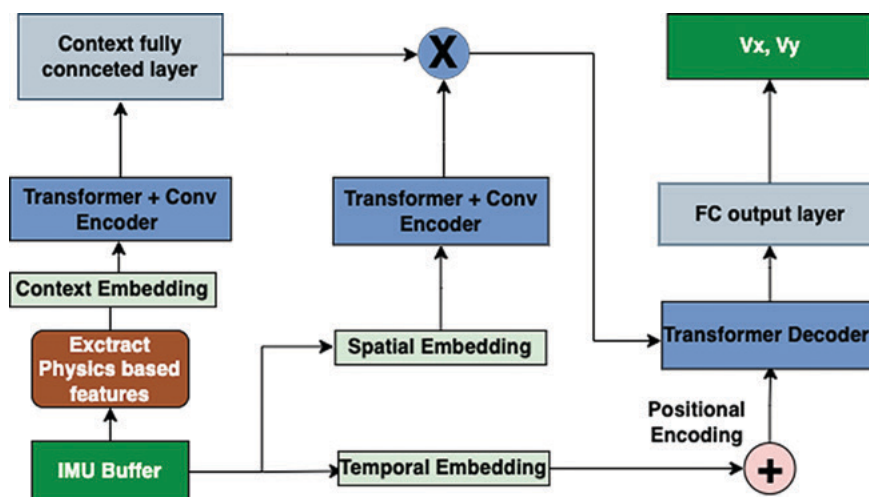


Abb. 3 | TAFI-Modell-Architektur

Merkmale	Beschreibung
Mittelwert, Standardabweichung, Max, Min, Energie, Summe, Amplitude und Wertebereich	Berechnen Sie die genannten Merkmale auf dem berechneten Vektor der Größe, indem Sie die Quadratwurzel aus der Summe der Quadrate der Vektorkomponenten ( $x, y, z$ ) ziehen.
Korrelation	Berechnen Sie den Korrelationskoeffizienten zwischen zwei Vektoren. Dieser misst den Grad des linearen Zusammenhangs zwischen den beiden Signalen und gibt den Grad ihrer Synchronisation oder Korrelation an.
Spitzen- und Tiefstwerte zählen	Zählen Sie die Anzahl der Spitzen und Talsohlen in den Magnitude-Vektoren. Sie stellen signifikante Änderungen oder Extrempunkte in den Bewegungssignalen dar.
Schrittlänge	Die Schätzung der Schrittlänge gibt die bei jedem Schritt zurückgelegte Strecke an.
Simpson-Integration	Führen Sie die Simpson-Integration über die konstanten Vektoren mit der gleichen Länge wie die Größenvektoren durch. Diese Integration quantifiziert den kumulativen Effekt oder die Fläche unter der Kurve der Bewegungssignale.
Madgwick-Algorithmus	Abwurf der Quaternionen für die geschätzte Orientierung des Madgwick-Filters.

Tab. 1 | Beschreibung der wissensbasierten Merkmale

*Convolutional*-Schicht erzeugt, gefolgt von einer *Batch-Normalisation* und einer linearen Transformation. Dieser Ansatz ermöglicht es dem Modell, kontextuelle Abhängigkeiten und Wechselwirkungen innerhalb der Eingangsdaten zu erfassen.

Um die effektive Nutzung der kontextuellen Informationen zu unterstützen, werden auch hier der *Transformer Encoder* und der *Convolutional Encoder* eingesetzt. Diese lernen die kontextuellen Abhängigkeiten, indem zusätzliche kontextuelle Daten integriert werden. Eine abschließende vollständig verbundene Schicht (engl.: Fully Connected Layer) generiert die Ausgabe-Repräsentationen. Diese strukturierte Herangehensweise ermöglicht es dem Modell, die kontextuellen Informationen effizient zu nutzen und die Genauigkeit der geschätzten Geschwindigkeiten zu verbessern.

### 4.3 TEMPORAL DECODER

Der *Temporal Decoder* des Modells generiert Vorhersagen über zeitliche Abläufe, indem er räumliche und kontextuelle Repräsentationen mit zeitlichen Informationen integriert. Der *Decoder* beginnt mit der Verarbeitung der Zeitsequenz und erstellt eine zeitliche Einbettungsrepräsentation. In Anlehnung an die Arbeit von Vaswani et al. 2017/ wird ein lernbarer linearer *Positional Encoder* verwendet, um die Positionsinformationen zu codieren. Diese Daten werden dann zusammen mit den zuvor berechneten Merkmalen in den *Transformer Decoder* eingespeist.

Der *Decoder*, ähnlich wie der *Encoder*, besteht aus einer Anzahl identischer Schichten  $N$ . In jeder Schicht wird zunächst eine *Masked-Self-Attention* auf die Ausgabe der Positionscodierung angewendet, gefolgt von einer Normalisierung. Anschließend wird die *Multi-Head Attention* zwischen der normalisierten *Self-Attention* und der Kombination aus räumlichen und kontextuellen Merkmalen berechnet, gefolgt von einer *Feed-Forward-Operation* und einer weiteren Normalisierung.

Die zeitliche Einbettung wird durch eindimensionale *Convolutional*-Schichten und eine *Batch-Normalisation* erreicht. Dieser Schritt extrahiert relevante zeitliche Abhängigkeiten und bereitet die Eingabedaten für die weiteren Operationen vor. Durch die Fusion von räumlichen, kontextuellen und zeitlichen Informationen ermöglicht der *Temporal Decoder* umfassendes Lernen und präzise Vorhersagen. Diese Integration bietet dem Modell tiefere Einblicke in die

Eingabedaten und verbessert sowohl die Vorhersagegenauigkeit als auch die Interpretierbarkeit.

### 4.4 FCOutputModule

Das *FCOutputModule* ist für die Generierung der finalen Vorhersagen verantwortlich. Es nutzt eine vollständig verbundene Schicht, um die gesammelten Informationen in den Ausgaberaum zu mappen. Das Modul besteht aus einer Sequenz von linearen Funktionen, *ReLU*-Aktivierungen (engl.: Rectified Linear Unit, ReLU) und *Dropout*-Schichten, um komplexe Abbildungen zu ermöglichen und Überanpassung (engl.: Overfitting) zu verhindern. Die *Dropout*-Schichten setzen während des Trainings zufällig einige Ausgabewerte auf null, um die Modellrobustheit zu erhöhen.

Durch diese Abfolge von Operationen kann das *FCOutputModule* die kombinierten Repräsentationen effektiv in die gewünschten Ausgangswerte (Geschwindigkeiten in den Achsen  $x$  und  $y$ ) umwandeln und somit präzise und relative Geschwindigkeiten ( $V_x, V_y$ ) vorhersagen.

## 5 EXPERIMENTE

### 5.1 Datenerfassung

Für die Entwicklung und Evaluierung des Positionsbestimmungsmodells wurden Daten mithilfe eines Smartphones Huawei Mate 20 5G erfasst. Die gesammelten Daten umfassen 7,3 Stunden synchronisierte Sensorwerte in 62 Trajektorien mit Gehzeiten zwischen 2 und 45 Minuten. Die Daten stammen aus der alltäglichen Nutzung eines Smartphones unter verschiedenen Bewegungsmustern, darunter Stehen, langsames Gehen, normales Gehen, schnelles Gehen, Seitwärtsgehen und Umherwandern (Abb. 4).

Die Datenerfassung wurde in verschiedenen Gerätepositionen durchgeführt, die typische Nutzungsszenarien widerspiegeln. Dazu gehörte das Halten des Smartphones in der Hand mit Blick auf den Bildschirm, in der Tasche, in einer Handtasche, in einem Rucksack und das Schwingen des Arms während des Gehens. Ergänzend wurden auch Daten auf einem Kinderwagen und einem 3D-Roboterkan



Abb. 4 | Platzierungen des Geräts (oder Gerätepositionen) von links nach rechts: am Arm schwingend, telefonierend, in der Hand, in der Tasche, im Rucksack, in der Handtasche, am Kinderwagen, am 3D-Roboterkran

erfasst, um Bewegungen wie das Benutzen einer Rolltreppe oder eines Fahrstuhls zu simulieren. Detaillierte Informationen sind in den Artikeln der Datengenerierung enthalten /Shoushtari et al. 2023a/.

Die erfassten Daten setzen sich aus einer Kombination von Smartphone-Sensordaten, 5G-Uplink-Time-Difference-of-Arrival (UL-TDoA-)Positionsbestimmungen sowie der Fused Location Provider API für Android zusammen, die GPS-, Bluetooth- und Wi-Fi-Daten verwendet, um sowohl in Innenräumen als auch im Freien Positionsdaten zu liefern. Zu den verwendeten Sensoren gehören Inertialsensoren zur Erfassung von Beschleunigung und Winkelgeschwindigkeit (6-Achsen-IMU) sowie ein Magnetometer zur Erfassung des Magnetfelds (3 Achsen). Darüber hinaus stehen Werte zur Bestimmung der linearen Beschleunigung, des Gravitationsfelds, der Position und der Schrittzahl zur Verfügung, die vom Android-Betriebssystem berechnet wurden. Die Sensorwerte wurden mit einer Frequenz von 200 Hz aufgezeichnet und anschließend interpoliert, um eine einheitliche Abtastfrequenz für alle Sensoren zu gewährleisten. Detaillierte Informationen sind in den Artikeln des Datasets enthalten /Shoushtari et al. 2023a/.

## 5.2 Vorbereitung der Eingabedaten

Um die Herausforderungen bei der Nutzung von Smartphone-Sensoren für präzise Positionierung zu meistern, verwenden wir ein gleitendes Intervall, wie in /Yan et al. 2019/ beschrieben. Hierbei werden IMU-Daten in Sequenzen der Länge  $m = 200$  (entsprechend einer 1-Sekunden-Zeitspanne bei 200 Hz Sensorfrequenz) aufgeteilt. Diese  $m$ -Blöcke werden in Schritten von 50 Messwerten gebildet. Jede Sequenz  $[x_{t-m}, \dots, x_t]$  umfasst  $m$  aufeinander folgende Messungen von Beschleunigung und Gyroskop. Innerhalb jedes Zeitfensters werden die Sensormessungen mithilfe eines berechneten Rotationsvektors in das Navigationskoordinatensystem transformiert, um konsistente Daten zu gewährleisten. Zielgeschwindigkeiten werden durch Berechnung der Verschiebung über einen bestimmten Zeitraum abgeleitet, während Algorithmen wie der erweiterte Kalman-Filter implementiert von /Yan et al. 2019/ und der Madgwick-Algorithmus /Madgwick 2010/ verwendet werden, um die Gravitationskomponente zu entfernen.

Für die Vorhersage der endgültigen Geschwindigkeit jeder Sequenz  $[acc_x, acc_y, acc_z, gyr_x, gyr_y, gyr_z]$  extrahieren wir zusätzlich zu den IMU-Daten  $m = 200$  kontextuelle Merkmale  $[c_1, \dots, c_{25}]$ . Diese Merkmale umfassen sowohl statistische Maße als auch physikalische Kontexte, die aus der gesamten Eingabesequenz berechnet

werden. Die Funktion zur Merkmalsextraktion liefert Einblicke in die zentrale Tendenz, Variabilität, Verteilung und Größe der Bewegungssignale, was dem Modell hilft, versteckte Muster zu erkennen. In Anlehnung an /Allahbakhshi et al. 2021/ zielt das in Tab. 1 beschriebene Verfahren zur Merkmalsextraktion darauf ab, aus dem Eingangstensor sinnvolle Informationen zu extrahieren, welche die Beziehungen und Interaktionen zwischen IMU-Messwerten erklären. Diese Kontextmerkmale können entweder durch Verkettung mit den IMU-Messwerten oder als zusätzliche Eingabe verwendet werden.

## 5.3 Training

Die Implementierung des Modells erfolgte unter Verwendung von PyTorch 1.7.1. Die Verlustfunktion (engl.: Loss Function), die zur Optimierung des Modells herangezogen wurde, ist das mittlere Abweichungsquadrat (engl.: Root Mean Squared Error, RMSE) der relativen Positionen innerhalb jedes Zeitfensters. Für das Training wurde der Adam-Optimizer /Kingma & Ba 2014/ über 250 Epochen eingesetzt. Zur Vermeidung von Überanpassung (engl.: Overfitting) wurde ein Early-Stopping-Kriterium mit einer Wartezeit (engl.: Patience) von zehn Epochen implementiert, das auf der Leistung des Validierungsdatensatzes basiert. Das Modell nutzt eine Architektur mit zwei Heads und zwei Schichten, wobei die verborgene Größe auf 32 festgelegt wurde. Die Transformer-Struktur orientiert sich an den ersten beschriebenen Transformern aus /Vaswani et al. 2017/.

Für die Modellarchitektur wurden ein Padding von 1 und eine Kernelgröße von 3 gewählt, was zu einer anfänglichen Lernrate von 0,0001 führte. Die Batch-Größe von 128 erwies sich als optimal für die Leistung des Modells. Sowohl das Training als auch die Tests wurden auf einer NVIDIA RTX 2080Ti GPU durchgeführt, um die Konsistenz mit den Baseline-Experimenten sicherzustellen. Diese beschreibenden Implementierungsparameter ermöglichen eine präzise Anpassung des Modells an die spezifischen Anforderungen der Kalibrierung und Positionierung mit Inertialsensoren und garantieren eine robuste Leistung sowohl bei der Training- als auch der Testphase.

## 6 AUSWERTUNG

Bei der Bewertung wurden fünf Kriterien verwendet, darunter der absolute Trajektorienfehler (engl.: Absolute Trajectory Error, ATE), der als RMSE über die gesamte Trajektorie berechnet wird, sowie zwei neue Messgrößen: Dauer und Entfernung, in der die Abweichung der

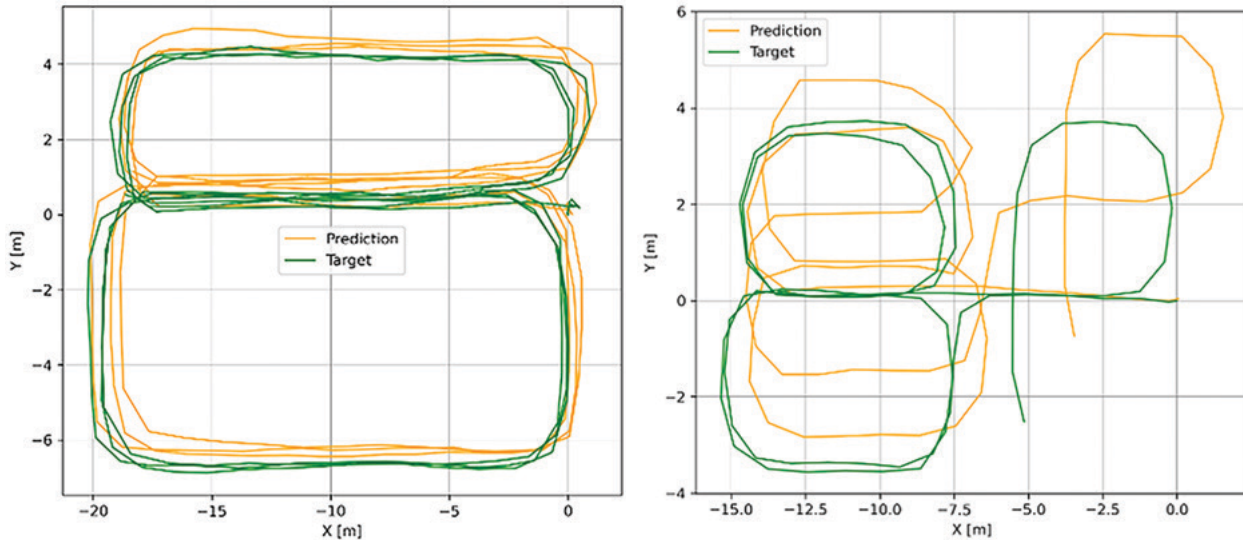


Abb. 5 | Die ausgewählten Diagramme veranschaulichen die Leistung des TAFI-Modells bei verschiedenen Seen- (links) und Unseen-Datensätzen (rechts). Hinweis: Die  $x$ - und  $y$ -Achsen haben unterschiedliche Skalierungen.

Positionsschätzung unter  $\pm 1$  Meter liegt (engl.: Time Below 1 m (TB1) und Distance Below 1 m (DB1)), zur Schätzung von Fehlern über feste Zeit- und Entfernungsintervalle. Zusätzlich wurden der RMSE und der Streukreisradius (engl.: Circular Error Probable, CEP) als wichtige Leistungsindikatoren verwendet, die anhand von kumulativen Verteilungsfunktionen mit einem Konfidenzniveau von 50 % und 90 % bewertet wurden. Dieser Bewertungsrahmen zielt darauf ab, sich stärker an realen Szenarien und Nutzerbedürfnissen zu orientieren und die Relevanz und Anwendbarkeit des Modells zu verbessern.

Bei der Bewertung des vorgeschlagenen Modells wird die Leistung sowohl bei gesehenen (engl.: Seen) als auch bei ungesehenen (engl. Unseen/Test-)Datensätzen unserer Definition berücksichtigt (Abb. 5). Wir teilten den gelesenen Datensatz auf, indem wir 80 % für das Training und die restlichen 20 % für die Validierung verwendeten. Wir haben eine zusätzliche Unterscheidung zwischen gesehenen und ungesehenen Datensätzen vorgenommen.

Wie man sehen kann, ist der absolute Fehler sowohl im gesehenen als auch im ungesehenen Datensatz ausreichend für die Fußgängernavigation mit Smartphone oder Wearables, da eine

angemessene Raumgenauigkeit gewährleistet ist. Die Raumgenauigkeit beschreibt die Genauigkeit, mit der Personen in der Lage sind, sich zu einem bestimmten Zielraum zu navigieren. Der Fehler im ungesehenen Datensatz ist deutlich höher als im gesehenen, was darauf hindeutet, dass die Lokalisierungsgenauigkeit des TAFI-Modells mit einem großen Datensatz verbessert werden kann. Die Trajektorien, die für die Fußgängernavigation verwendet werden, sind oft 10 bis 15 Minuten lang. Dabei genügt es, ca. jede Minute eine Positionskorrektur mit einer Genauigkeit von 1 Meter vorzunehmen, um die Gesamtgenauigkeit zu erhalten, da die relativen Abweichungen mit dem oben beschriebenen Modell für ca. 1 Minute unter 1 Meter bleiben, wie in Tab. 2 zu sehen ist. Das bedeutet, dass es genügt, diese Optimierung und Korrektur nur einmal pro Minute durchzuführen. Dies wäre energieeffizient und passt zu realen Anwendungen und kann sogar den Einsatz zusätzlicher Infrastrukturen vermeiden.

## 7 ZUSAMMENFASSUNG UND AUSBLICK

Um den Lernprozess zu beschleunigen und die Komplexität zu reduzieren, haben wir automatische wissensbasierte Merkmale in die Architektur eines Transformermodells integriert. Dieser strategische Ansatz hat zu vielversprechenden Ergebnissen im Bereich der Inertiallokalisierung geführt, einer zentralen Herausforderung in der Robotik und Navigation, bei der die Position von mobilen Objekten mithilfe von Inertialsensoren bestimmt wird.

Das TAFI-Modell zeigt die Fähigkeit, die Fehlerakkumulation über die Zeit zu mindern und ist nicht durch spezifische Platzierungsüberlegungen eingeschränkt; es funktioniert als Sensorkalibrierung. Dies kann durch den Einsatz von lernenden Regressionsmodellen im Allgemeinen, einschließlich unseres State-of-the-Art-Modells TAFI, erreicht werden. Unsere Forschung zeigt, dass das transformerbasierte TAFI-Modell nicht nur die Kalibrierungsaufgabe erfolgreich bewältigen kann, sondern auch eine hohe Genauigkeit bei der Positionsbestimmung erreicht. Die vorliegende Untersuchung

Modell	Seen/Unseen	Kriterien	L5IN <sup>+</sup> Datensatz
TAFI	Seen	ATE	0,66 m
		TB1	323,87 s
		DB1	189,43 m
		CEP50	0,81 m
		CEP90	2,47 m
	Unseen	ATE	2,24 m
		TB1	50,80 s
		DB1	31,80 m
		CEP50	1,29 m
		CEP90	4,60 m

Tab. 2 | Leistung des Modells: ATE und CEPs („Niedriger ist besser“), TB1 und DB1 („Höher ist besser“)

kommt zu dem Schluss, dass lernbasierte Ansätze nicht nur dazu genutzt werden können, eine längere Zeit ohne externe Unterstützung durch zusätzliche Sensoren oder Umgebungsdaten zu vermeiden, sondern dass mehrere Smartphones diese lernbasierten Ansätze als inklusive Kalibrierung nutzen können. Die Berechnung der Kalibrationsparameter erübrigt sich in diesem Fall. Für die allgemeine Nutzung, beispielsweise durch Smartphone-Nutzer, ist diese Berechnung nicht erforderlich.

Die gute Qualität der Ergebnisse, die mit dieser datengesteuerten Methode erzielt werden, ist vielversprechend. Dies gelingt durch die Kombination von IMU-Daten und wissensbasierten Ansätzen. Wissensbasierte Ansätze beziehen sich auf das Wissen, das Forscher bisher im Rahmen der physikalischen Gesetze über inertielle Navigation erlangt haben. In der weiteren Forschung werden wir die Leistung des entwickelten Lernmodells mit typischen Inertialmethoden wie Strapdown oder PDR vergleichen.

## DATENVERFÜGBARKEIT

Wir stellen den Code sowie die von uns gesammelten Datensätze als Open Source zur Verfügung, um Forscher zur Weiterentwicklung zu ermutigen.

## DANKSAGUNG

Die Arbeit wurde in Zusammenarbeit mit dem Fraunhofer IIS und parallel zum Forschungsprojekt L5IN durchgeführt. Wir danken den Kollegen beider Teams.

## LITERATUR

- Allahbakhshi, H.; Röcke, C.; Weibel, R. (2021): Assessing the transferability of physical activity type detection models: Influence of age group is underappreciated. In: *Frontiers in Physiology* 12(2021), 738939.
- Chen, C.; Lu, X.; Markham, A.; Trigoni, N. (2018): IONet: Learning to Cure the Curse of Drift in Inertial Odometry. arXiv:1802.02209.
- Giarmatzis, G.; Zacharaki, E. I.; Moustakas, K. (2020): Realtime prediction of joint forces by motion capture and machine learning. In: *Sensors* 20(2020)23, 6933.
- Herath, S.; Caruso, D.; Liu, C.; Chen, Y.; Furukawa, Y. (2022): Neural inertial localization. [https://openaccess.thecvf.com/content/CVPR2022/papers/Herath\\_Neural\\_Inertial\\_Localization\\_CVPR\\_2022\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2022/papers/Herath_Neural_Inertial_Localization_CVPR_2022_paper.pdf) (07.01.2025).
- Kingma, D. P.; Ba, J. (2014): Adam: A method for stochastic optimization. arXiv:1412.6980.
- Klingbeil, L. (2023): Georeferencing of mobile mapping data. Habilitation, Rheinische-Friedrich-Wilhelms-Universität Bonn. <https://hdl.handle.net/20.500.11811/10926> (07.01.2025).
- Li, F.; Zhao, C. et al. (2012): A reliable and accurate indoor localization method using phone inertial sensors. In: *UbiComp 12 – Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 421–430.
- Madgwick, S. (2010): An efficient orientation filter for inertial and inertial/magnetic sensor arrays. [https://courses.cs.washington.edu/courses/cse466/14au/labs/l4/madgwick\\_internal\\_report.pdf](https://courses.cs.washington.edu/courses/cse466/14au/labs/l4/madgwick_internal_report.pdf) (07.01.2025).

Rao, B.; Kazemi, E.; Ding, Y.; Shila, D. M.; Tucker, F. M.; Wang, L. (2021): CTIN: Robust contextual transformer network for inertial navigation. arXiv:2112.02143.

Shoushtari, H.; Harder, D.; Kasperek, M.; Schäfer, M.; Müller-Lietzkow, J.; Sternberg, H. (2023a): Data-Driven Inertial Navigation assisted by 5G UL-TDoA Positioning. In: *Proceedings of the 2023 International Technical Meeting (ITM) of the Institute of Navigation (ION)*, Long Beach, Kalifornien, Jan. 2023, 1169–1183. doi.org/10.33012/2023.18645.

Shoushtari, H.; Willemsen, T.; Sternberg, H. (2023b): Supervised Learning Regression for Sensor Calibration. In: *DGON Inertial Sensors and Systems (ISS)*, 1–20. IEEE.

Shoushtari, H.; Kassawat, F.; Sternberg, H. (2024): Context aware transformer network and in-situ IMU calibration for accurate positioning. In: *14th International Conference on Indoor Positioning and Indoor Navigation*, Okt. 2024, Hongkong. DOI: 10.1109/IPIN62893.2024.10786130.

Solin, A.; Cortes, S.; Rahtu, E.; Kannala, J. (2018): Inertial odometry on handheld smartphones. arXiv:1703.00154.

Statista (2024): Statista – Das Statistik-Portal. <https://de.statista.com> (09.01.2025).

Vaswani, A.; Shazeer, N. M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. (2017): Attention is All you Need. *Neural Information Processing Systems*. arXiv:1706.03762.

Willemsen, T.; Keller, F.; Sternberg, H. (2013): Untersuchungen zur Indoor-navigation mittels aktueller Smartphonetechnologie. In: *allgemeine vermessungs-nachrichten* 120(2013)11-12, 367–374.

Willemsen, T.; Keller, F.; Sternberg, H. (2015): A topological approach with MEMS in smartphones based on routing-graph. In: *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Okt. 2015, Banff (Kanada), 1–6.

Yan, H.; Qi, S.; Furukawa, Y. (2017): RIDI: Robust IMU Double Integration. arXiv:1712.09004.

Yan, H.; Herath, S.; Furukawa, Y. (2019): Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. arXiv:1905.12853.

## AUTOREN



**Dr.-Ing. Hossein Shoushtari**

HAFENCITY UNIVERSITÄT HAMBURG  
PROFESSUR FÜR HYDROGRAPHIE  
UND GEODÄSIE

Henning-Voscherau-Platz 1 | 20457 Hamburg | Deutschland  
hossein.shoushtari@hcu-hamburg.de | <https://www.researchgate.net/profile/Hossein-Shoushtari>



**Prof. Dr.-Ing. Harald Sternberg**

HAFENCITY UNIVERSITÄT HAMBURG  
PROFESSUR FÜR HYDROGRAPHIE  
UND GEODÄSIE

Henning-Voscherau-Platz 1 | 20457 Hamburg | Deutschland  
harald.sternberg@hcu-hamburg.de | ORCID: 0000-0002-1905-2287

Manuskript eingereicht: 03.08.2025 | Im Peer-Review-Verfahren begutachtet