



LUND UNIVERSITY

Efficient Real-Time Localization Systems: Measurements, Algorithms, and Hardware

Yaman, Ilayda

2026

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Yaman, I. (2026). *Efficient Real-Time Localization Systems: Measurements, Algorithms, and Hardware*. [Doctoral Thesis (compilation), Department of Electrical and Information Technology]. Department of Measurement Technology and Industrial Electrical Engineering, Lund University.

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Efficient Real-Time Localization Systems: Measurements, Algorithms, and Hardware

Doctoral Thesis

Ilayda Yaman



LUND UNIVERSITY

Department of Electrical and
Information Technology
Lund, October 2026

Academic thesis for the degree of Doctor of Philosophy, which, by due permission of the Faculty of Engineering at Lund University, will be publicly defended on Tuesday, 13 October, 2026, at 9:15 a.m. in lecture hall E:1406, Department of Electrical and Information Technology, Ole Römers Väg 3, 223 63 Lund, Sweden. The thesis will be defended in English.

The Faculty Opponent will be Professor Sofie Pollin, Katholieke Universiteit, Leuven, Belgium.

<i>Organisation:</i> LUND UNIVERSITY Department of Electrical and Information Technology Ole Römers Väg 3 223 63 Lund Sweden	<i>Document Type:</i> DOCTORAL THESIS	
	<i>Date of Issue:</i> September 2026	
	<i>Sponsoring Organisation(s):</i> Swedish Research Council ELLIIT (Excellence Center at Linköping-Lund in Information Technology)	
<i>Author:</i> Ilayda Yaman		
<i>Title:</i> Efficient Real-Time Localization Systems: Measurements, Algorithms, and Hardware		
<i>Abstract:</i> Localization is fundamental in many applications, including autonomous driving, robotics, emergency services, and wireless communication systems. The achievable localization accuracy depends on the information provided by the measurements, which is shaped by sensor placement, spatial resolution, sampling rate, synchronization accuracy, calibration, noise levels, and environmental effects such as multipath propagation and occlusions. At the same time, practical localization systems must operate under constraints such as limited computational resources, energy consumption, and real-time processing requirements. As a result, localization should be viewed not only as an estimation problem but also as a broader system design problem that involves measurements from sensors, algorithm selection, and hardware implementation. By considering these aspects jointly, this thesis investigates how localization systems can be designed to achieve reliable and accurate performance while remaining suitable for practical real-time applications. This thesis consists of introductory chapters that provide background on localization systems, followed by five papers published in or submitted to scientific conferences and journals. The first paper presents the Lund University Vision, Radio, and Audio (LuViRA) dataset, a public multimodal dataset developed through a collaboration between multiple departments and faculties at Lund University. It provides synchronized measurements from multiple sensor types for indoor robot localization and a common reference for developing, evaluating, and comparing methods across sensing modalities. The second and third papers build on LuViRA by investigating the accuracy, robustness, and practical limitations of different sensors, including a machine-learning-based 5G localization method using massive MIMO channel measurements. The fourth paper shifts the focus to outdoor radio-based localization and develops an adaptive attention-based model that selects between specialized machine learning models according to the propagation environment. Finally, the fifth paper extends this approach towards Transformer-based models and implements an FPGA accelerator to support low-latency and energy-efficient inference. Together, these contributions span measurements, algorithm development, environment-dependent adaptation, and hardware implementation, highlighting the importance of considering the complete localization system for practical real-time deployment.		
<i>Keywords:</i> Localization, multimodal sensing, massive MIMO, machine learning, adaptive models, Transformers, FPGA acceleration, hardware-software co-design.		
<i>Classification System and/or Index Terms</i> –		<i>Language:</i> English
<i>Supplementary Bibliographical Information:</i> –		<i>ISBN (printed):</i> 978-91-6858-032-6
<i>Key title and ISSN:</i> Series of Licentiate and Doctoral Theses; 1654-790X, No. 195		<i>ISBN (digital):</i> 978-91-6858-033-3
<i>Recipient's Notes:</i>	<i>Number of Pages:</i> 154	<i>Price:</i>
	<i>Security Classification:</i> Unclassified	

General Permissions:
I, the undersigned, being the copyright owner and author of the above-mentioned thesis and its abstract, hereby grant to all reference sources permission to publish and disseminate said abstract.

Signature: Ilayda Yaman

Date: 14 September 2026

Efficient Real-Time Localization Systems: Measurements, Algorithms, and Hardware

Doctoral Thesis

Ilayda Yaman



LUND
UNIVERSITY

Department of Electrical and
Information Technology

Lund, October 2026

Ilayda Yaman
Department of Electrical and Information Technology
Lund University
Ole Römers Väg 3, 223 63 Lund, Sweden

Series of Licentiate and Doctoral Theses
ISSN 1654-790X, No. 195
ISBN 978-91-6858-032-6 (printed)
ISBN 978-91-6858-033-3 (digital)

© 2026 Ilayda Yaman
This thesis is typeset using L^AT_EX 2_ε.

Frontispiece: *Visual summary of concepts presented in the thesis, adapted from the corresponding figures.*

Printed by Tryckeriet i E-huset, Lund University, Lund, Sweden.

No part of this thesis may be reproduced or transmitted in any form or by any means without written permission from the author. Distribution of the original thesis in full, however, is permitted without restriction.

To my family and friends.

Abstract



LOCALIZATION is fundamental in many applications, including autonomous driving, robotics, emergency services, and wireless communication systems. The achievable localization accuracy depends on the information provided by the measurements, which is shaped by sensor placement, spatial resolution, sampling rate, synchronization accuracy, calibration, noise levels, and environmental effects such as multipath propagation and occlusions. At the same time, practical localization systems must operate under constraints such as limited computational resources, energy consumption, and real-time processing requirements. As a result, localization should be viewed not only as an estimation problem but also as a broader system design problem that involves measurements from sensors, algorithm selection, and hardware implementation. By considering these aspects jointly, this thesis investigates how localization systems can be designed to achieve reliable and accurate performance while remaining suitable for practical real-time applications.

This thesis consists of introductory chapters that provide background on localization systems, followed by five papers published in or submitted to scientific conferences and journals. The first paper presents the Lund University Vision, Radio, and Audio (LuViRA) dataset, a public multimodal dataset developed through a collaboration between multiple departments and faculties at Lund University. It provides synchronized measurements from multiple sensor types for indoor robot localization and a common reference for developing, evaluating, and comparing methods across sensing modalities. The second and third papers build on LuViRA by investigating the accuracy, robustness, and practical limitations of different sensors, including a machine-learning-based 5G localization method using massive MIMO channel measurements. The fourth paper shifts the focus to outdoor

radio-based localization and develops an adaptive attention-based model that selects between specialized machine learning models according to the propagation environment. Finally, the fifth paper extends this approach towards Transformer-based models and implements an FPGA accelerator to support low-latency and energy-efficient inference. Together, these contributions span measurements, algorithm development, environment-dependent adaptation, and hardware implementation, highlighting the importance of considering the complete localization system for practical real-time deployment.

Popular Science Summary



LOCALIZATION answers the fundamental question: "Where am I?". In everyday life, we localize ourselves using many different senses. We recognize landmarks, follow sounds, and estimate distances based on what we see around us. For example, hearing bells might indicate the direction of a bell tower, helping us understand our relative position. Modern devices do something similar, but with electronic sensors. A phone, robot, or vehicle may use cameras, microphones, or antennas to understand where it is. Each sensor provides a different view of the environment. A camera can give rich visual information, but it may fail in darkness, fog, or when the view is blocked. Radio signals, on the other hand, can often still be measured when there is no direct visual contact.

To develop new localization methods, realistic data is needed. Algorithms cannot be properly designed or evaluated without measurements that represent the conditions they will face in practice. One contribution of this thesis is therefore a real-world dataset with synchronized vision, radio, and audio measurements, together with reference position information. This dataset makes it possible to compare different sensing methods under the same conditions and to study their strengths and limitations.

In modern wireless communication systems, especially 5G, base stations are equipped with many antennas. This technology is known as massive multiple-input multiple-output, or massive MIMO. These antenna arrays are mainly used to improve communication, but they also provide useful information for localization. When a signal travels between a user device and a base station, it interacts with the environment. It may arrive directly, reflect from buildings, or be partially blocked by objects. The measured radio signal therefore contains a kind of fingerprint of the user's position.

Machine learning can be used to learn the relation between these radio fingerprints and physical positions. Instead of explicitly modelling every possible reflection and propagation path, a model can be trained from real measurements. In this thesis, learning-based methods are developed for indoor and outdoor localization using radio signals from a commercial massive MIMO base station. The goal is to estimate the position of a moving user from the measured radio channel.

A challenge in radio localization is that the same model may not work equally well in all situations. For example, measurements with a direct signal path can look very different from measurements where the signal is blocked and mainly reaches the receiver through reflections. To address this, the thesis studies adaptive localization models. Rather than using one fixed model for all conditions, the proposed approach selects the most suitable localization model based on the measured signal. This makes the system more robust to changes in the radio environment.

Accurate algorithms are only useful in practice if they can also run efficiently. Future applications such as autonomous vehicles, mobile robots, extended reality devices, and industrial systems require position estimates with low latency and limited energy consumption. For this reason, the thesis also investigates hardware architectures for real-time radio-based localization. The proposed hardware exploits the structure of the input and the computations inside the machine learning model.

Together, the contributions of this thesis cover several parts of the localization problem: real-world measurement data, learning-based radio localization algorithms, adaptive models for changing signal conditions, and efficient hardware implementation. The work shows how radio signals from communication infrastructure can be used not only for communication, but also for accurate and efficient positioning. By combining real-world measurements, algorithms, and hardware design, this thesis contributes toward localization systems that are real-time and efficient.

Acknowledgments

 My grandfather, Dr. Halil İlbasmış, completed his dissertation entitled “The Hysteresis of Water Retention and Hydraulic Conductivity in the Unsaturated Moisture State of Soil, Using the Example of a Loess Parabraunerde” and was awarded the degree of Doctor of Agricultural Sciences (Dr. sc. agr.) in Göttingen, Germany, on November 11, 1971. Even when I was a child and climbed onto his shoulders, I would still look up to him. He was one of the people who believed in me the most. This thesis is dedicated to his memory.

I have been very lucky to meet so many wonderful colleagues and friends throughout my doctoral studies, from March 2021 to October 2026, and in the following paragraphs I will do my best to acknowledge them. Given how many wonderful people I have had the privilege of crossing paths with along the way, however, this list is inevitably not exhaustive and is presented in no particular order.

My deepest gratitude goes to my main supervisor, **Prof. Liang Liu**, for his kindness, support, and guidance throughout these years, and for always being there to help me work through technical problems. I would also like to thank my co-supervisors, **Prof. Ove Edfors** and **Prof. Kalle Åström**, for their invaluable feedback and many interesting discussions, as well as my former co-supervisor, **Dr. Steffen Malkowsky**, for his support and guidance during the first half of my studies.

A very special thank you goes to all my coauthors: **Prof. Fredrik Tufveson**, **Prof. Zhengya Zhang**, **Dr. Guoda Tian**, **Sijia Cheng**, **Love Bárány**, **Dr. Michiel Sandra**, **Prof. Joachim Rodrigues**, **Allan Anderson**, **Lucas Ferreira**, **Dr. Dino Pjanić**, **Prof. Amir Aminifar**, **Associate Prof. Xuesong Cai**, **Dr. Martin Larsson**, **Patrik Persson**, **Alexander Dürr**, **Dr. Erik Tegler**, **Nikhil Challa**, **Henrik Garde**, and **Jens Gulin**. I have truly enjoyed working with

every one of you, and each collaboration has contributed in its own way to my studies and to what I have learned along the way. Thank you for all the discussions, ideas, challenges, and shared work; it has been a privilege to work with you.

My colleagues in the Integrated Electronic Systems division have also played a big role throughout my doctoral studies: **Assistant Prof. Baktash Behmanesh**, **Prof. Fredrik Edman**, **Prof. Erik Larsson**, **Dr. Per Andersson**, **Dr. Victor Åberg**, **Prof. Henrik Sjöland**, **Dr. Iman Ghotbi**, **Dr. Markus Törmänen**, **Prof. Per Stenström**, **Adjunct Prof. Björn Ekelund**, **Prof. Viktor Öwall**, **Dr. Arturo Prieto Llorens**, **Dr. Jesús Rodríguez Sánchez**, **Dr. Mohammad Attari**, **Dr. Rikard Gannedahl**, **Dr. Siyu Tan**, **Dumitra Iancu**, **Kristoffer Westring**, **Sidra Muneer**, **Kaan Veziroğlu**, **Vilgot Snygg**, **Wenbo Li**, **William Marnfeldt**, **Sergio Castillo Mohedano**, **Sofia Tatidis**, **Josep Ramon Fernández Rull**, **Dimos Tikeris**, **Alex Allfjord**, **Johan Holmstedt**, and **Joel Åhlund**. Thank you for making the division not only a place to work, but also a place filled with conversations, laughter, and good memories.

I am very grateful to **Prof. Buon Kiong Lau** for welcoming me into the Communications Engineering division and, in particular, for including me in its social life. I greatly appreciated being able to join the division meetings, fika, and other activities throughout the years, together with **Prof. Anders J. Johansson**, **Prof. Fredrik Rusek**, **Dr. Johan Thunberg**, **Adjunct Prof. Leif Wilhelmsson**, **Dr. Sara Willhammar**, **Dr. Juan Vidal Alegría**, **Dr. Xuhong Li**, **Dr. Ashkan Sheikhi**, **Dr. Hanieh Aliakbari Abar**, **Dr. Aleksei Fedorov**, **Dr. Galina Sidorenko**, **Paul Berestesky**, **Hedieh Khosravi**, **Linus Ghatnekar Nilsson**, **Jiawang Li**, **Johan Siwerson**, **Neharika Valecha**, **Lina Tinnerberg**, **Junshi Chen**, **Ali Al-Ameri**, **Emil Bergman**, **Yingjie Xu**, and **Yujie Zhang**. Thank you for the discussions, company, and many moments shared along the way. Special thanks go to **Prof. Liesbet Van der Perre** for her invaluable feedback on my presentations and for teaching me about how to communicate research effectively.

Many people across the Department of Electrical and Information Technology (EIT) at Lund University have also contributed, directly or indirectly, to making these years possible. I would like to thank our Head of Department, **Prof. Stefan Höst**, and the administrative staff, **Erik Göthe**, **Linda Bienen**, **Margit Billesö**, **Steffen Moulund**, **Elisabeth Nordström**, and **Tea Larsson**, for all their help and support throughout the years. I am also grateful to **Prof. Maria Kihl** and **Dr. Susanna Lönnqvist** from the management of NextG2Com. During our measurement campaigns, the support of **Sirvan Abdollah Poor** and **Erik Jonsson** with computers, servers, access, and countless other practical matters was invaluable in making our work possible. I would also like to thank **Prof. Daniel Sjöberg** and **Associate Prof. Michael Lentmaier**, who served as Directors of Studies at EIT during my time here,

as well as **Prof. Mattias Borg**, **Prof. Björn Landfeldt**, and **Prof. Thomas Johansson** for the many interesting discussions over the years. Finally, I would like to thank **Assistant Prof. Viktor Larsson** from the Centre for Mathematical Sciences for his help during my midterm review. I am also grateful for my time as a student representative on the LTH research education and research boards (FUN and FN) from 2021 to 2024, and especially to **Prof. Martin Tunér** and **Andrea Nord**. I truly enjoyed being part of this work for three years and learned a great deal from the experience.

Some of my fondest memories from these years come from lunch breaks, coffee breaks, pizza nights, spontaneous conversations, and all the moments in between. I am very thankful to the people who filled these moments with so much laughter and good company: **Dr. Saketh Ram Mamidala**, **Hannes Dahlberg**, **Ben Nel**, **André Andersen**, **Karthik Ram Mamidala**, **Jiawang Li**, **Suleyman Sadikhov**, **Arthur Nijdam**, **Sahar Zamanian**, **Markus Berthilsson**, **Dr. Erik Mårtensson**, **Maggie Tràn**, **Ufuk Özdemir**, **Giovani Chrestani**, **Dr. Fatemeh Akbarian**, **Anahita Baninajjar**, **Dr. Kaan Bür**, **Dr. Farnaz Khodakhah**, **Dr. Paul Stankovski Wagner**, **Dr. Johannes Svensson**, **Baichuan Huang**, **Dr. Denis Nabokov**, and **Yu-Yan Cao**. A very special thank you goes to **Ali Jahangiri** for his help with the figures in this thesis.

During my research stay in the United States, I had the pleasure of joining the Lab for Efficient Application Processors (LEAPs) at the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. I would like to thank **Prof. Zhengya Zhang**, **Dr. Wei Tang**, **Ji Yoon Han**, **Cong Sheng Leow**, **Sangbu Yun**, **Justin Ting**, **Minsik Kim**, **Chen-Chien Kao**, **Tianyu Wei**, **Po-Shao Chen**, **Jack Erhardt**, **Cheng-Hsun Lu**, and **Junkang (Jerry) Zhu** for making my stay both academically rewarding and enjoyable. I am also very grateful to **Prof. Mark Hoover** and **Cassie Cammann** for their hospitality and for making me feel so welcome during my time there. I especially enjoyed our discussions over the occasional late-night dinner, accompanied by good food and wine.

I am also sincerely grateful to my faculty opponent and the members of my examination committee for taking the time to read, discuss, and evaluate this thesis, and for being part of the final step of this journey.

I am fortunate to have had many wonderful friends alongside me during my time in Sweden. Thank you to **Sofya Ignatenko**, **Dr. Haorui Peng**, **Dr. Matthias Mayr**, **Marianne Graw**, **Erik Bolin**, **Dr. Daria Ivina**, **Dr. Ricardo Gonzales**, **Emma Asklund**, **Herman Gidestrand**, **Hamid Yadegar Amin**, **Rimzim Singh**, **Chris Kotlett**, **Filip Hallböök**, **Dr. Joana Campos**, **Robert Dyke**, **Hedieh Akbari**, **Robin Blomquist**, **David Lindqvist**, **Pontus Nordlund**, **Marcus Wiman**, **Marcel Nilsson**, **Linn Rydberg**, and **Dr. Laura Castilla Vallmanya**. A special thank you goes to **Susanne** and **Johan Tegler** for the amazing food and many memorable Easter and Midsummer celebrations,

and, even more importantly, for the wonderful company and conversations over the years.

I am also grateful to those who helped shape my journey long before the start of my doctoral studies. I would like to acknowledge **Mustafa Kemal Atatürk**, whose vision for education, science, and progress has profoundly shaped the opportunities available to generations of young people in Türkiye, including my own. More recently, my first experiences with research, engineering, and collaboration played an important role in shaping the path that eventually led me here. In particular, I would like to thank **Dr. Latif Akçay** and **Prof. Müştak Erhan Yalçın** for the projects, collaborations, guidance, and support along the way.

My friends from Türkiye have been there for me despite the distance and the years, especially **Dilara Koca Yıldız**, my best friend since we were ten years old, and **Dr. Ekin Karasan**. I am also very fortunate to have remained close with many of my friends from my bachelor's studies at Istanbul Technical University: **Sueda Uludağ**, **İnci Pir**, **Dr. Mertol Tüfekci**, **Yonca Bab**, **Emre Bay**, **Burak Mert Gönültaş**, and **Tarik Tamyurek**. I am very grateful that these friendships, many of which began so many years ago, remain such an important part of my life.

To my mum, **Alev İlbasmış Yaman**; my dad, **Ali Mürsel Yaman**; my sister, **Doğa Yaman**; my grandmother, **Sülünay İlbasmış**; my grandfathers, **İshak Yaman** and **Halil İlbasmış**; my aunt, **Figen İlbasmış**; and my great-aunt, **Şule Tokat**: I would not be where I am today without your unconditional love, encouragement, and belief in me. You have shaped who I am, and whatever I have achieved along the way is not mine alone. I am also grateful to my husband's family, **Catarina Wahlgren Sjöstedt**, **Christer Sjöstedt**, **Lovisa Sjöstedt**, and **Elvi Wahlgren**, for welcoming me into their family and for all the love and support they have given me over the years. I feel very fortunate to have the rest of my family and Emil's extended family around me as well.

Finally, to my husband, **Emil Sjöstedt**, who has been by my side through all the sweat and tears, the uncertainty and frustration, and the long and difficult days, weeks, months, and years: thank you for your patience, your support, and for always being there. You have shared both the good and the difficult moments with me, and made the difficult ones a little easier. I could not have made it here without you.

Ilayda Yaman
Lund, April 2026

Contents

Abstract	v
Popular Science Summary	vii
Acknowledgments	ix
Contents	xiii
Preface	xvii
Structure of the Thesis	xvii
Included Papers	xviii
Related Work	xix
Acronyms and Abbreviations	xxi
Glossary	xxi
Symbols	xxv
Glossary	xxv
INTRODUCTION	1
1 Overview	3
1.1 Limitations of GNSS-Based Localization	3
1.2 Practical Use Cases	5
1.3 Sensors and System Requirements	6

1.4 Research Challenges	9
1.5 Research Questions and Thesis Contributions	10
2 Localization Systems and Measurements	13
2.1 Localization State Representation	13
2.2 Sensors	15
2.2.1 Vision Sensors	16
2.2.2 Radio Sensors	17
2.2.3 Audio Sensors	19
2.3 Range Measurements	20
2.4 Bearing Measurements	21
2.5 Calibration	23
2.6 Time Synchronization	25
3 Localization Algorithms	29
3.1 Geometry-Based Algorithms	30
3.2 Data-Driven Algorithms	32
3.2.1 Basics of Neural Networks	33
3.2.2 Fully Connected and Convolutional Neural Networks	35
3.2.3 Transformer Models	36
3.2.4 Recurrent and State-Space Models	37
4 Hardware Implementations	39
4.1 Processing Architectures	39
4.1.1 Systolic Arrays	40
4.1.2 Parallel Vector Engines	40
4.2 Memory Hierarchy and Organization	42
4.3 Dataflow	44
4.4 Quantization	45
4.4.1 Quantization Methods for Deep Learning Models	47
5 Co-Design of Localization Systems	49
5.1 Design Space Exploration and Optimization	50
5.2 Performance Metrics	51
5.3 Case Studies	52
6 Conclusion and Future Work	59

6.1 Future Work	62
Bibliography	65
PAPERS	75
I The LuViRA Dataset: Synchronized Vision, Radio, and Audio Sensors for Indoor Localization	77
II LuViRA Dataset Validation and Discussion: Comparing Vision, Radio, and Audio Sensors for Indoor Localization	87
III High-Precision Machine-Learning Based Indoor Localization with Massive MIMO System	101
IV Adaptive Attention-Based Model for 5G Radio-Based Outdoor Localization	109
V Implementation of an Adaptive Transformer Accelerator for Accurate Outdoor Localization with Massive MIMO	117

Preface

 HIS thesis presents the research contributions developed during my doctoral studies in the Integrated Electronic Systems group at the Department of Electrical and Information Technology, Lund University, Sweden. The main focus of this work addresses localization systems, algorithms, and hardware. The work has been supervised by Professor Liang Liu, Professor Ove Edfors and Professor Kalle Åström.

STRUCTURE OF THE THESIS

The thesis is organized into two parts. The first part provides an overview of the research field and summarizes the main contributions. The second part consists of the appended papers, which detail the individual research results.

- **INTRODUCTION**

Provides a comprehensive overview of the research areas explored during the doctoral studies. It outlines the key problems and challenges addressed in the included papers, establishes the connections between them, and contextualizes the overall contributions. *

- **PAPERS**

The second part includes five papers [1–5] in the main body of the thesis, presented in the following order, along with a summary of my contributions.

*This thesis has in part been produced with the assistance of generative AI model(s), such as ChatGPT, Microsoft Copilot and Gemini. AI models were used to brainstorm ideas, edit content, format and polish the text and improve the language. I have processed the generated text and image and take full responsibility for the content.

INCLUDED PAPERS

The following papers form the main body of this thesis and the respective published or draft versions are appended in the back.

Paper I: I. YAMAN, G. TIAN, M. LARSSON, P. PERSSON, M. SANDRA, A. DÜRR, E. TEGLER, N. CHALLA, H. GARDE, F. TUFVESSON, K. ÅSTRÖM, O. EDFORS, S. MALKOWSKY AND L. LIU, “The LuViRA Dataset: Synchronized Vision, Radio, and Audio Sensors for Indoor Localization”, *IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, May 2024, pp. 11920-11926.

Contribution: The multidisciplinary project and multimodal measurement campaign underlying the LuViRA dataset were led by the author. Contributions include developing the measurement plan, designing the experimental scenarios, synchronizing the data acquisition systems, mitigating interference between sensors, and coordinating the collaboration across the participating research groups. The author was also responsible for conducting the measurements, processing and analyzing the collected data, and integrating the results across the vision, radio, and audio modalities.

Paper II: I. YAMAN, G. TIAN, E. TEGLER, J. GULIN, N. CHALLA, F. TUFVESSON, O. EDFORS, K. ÅSTRÖM, S. MALKOWSKY AND L. LIU, “LuViRA Dataset Validation and Discussion: Comparing Vision, Radio, and Audio Sensors for Indoor Localization”, *IEEE Journal of Indoor and Seamless Positioning and Navigation*, vol. 2, Jul 2024, pp. 240-250.

Contribution: The analysis and validation of the LuViRA dataset were led by the author. Contributions include the design of the evaluation methodology, implementation of localization pipelines for different sensing modalities, and comparative analysis of vision, radio, and audio-based localization. The author was also responsible for interpreting the results and identifying the strengths and limitations of each modality.

Paper III: G. TIAN, I. YAMAN, M. SANDRA, X. CAI, L. LIU AND F. TUFVESSON, “High-Precision Machine-Learning Based Indoor Localization with Massive MIMO System”, *IEEE International Conference on Communications (ICC)*, Rome, Italy, Oct 2023, pp. 3690-3695.

Contribution: The author contributed to the development and evaluation of machine learning-based methods for indoor localization using massive MIMO systems. Contributions include data processing, model implementation, and performance evaluation, with particular focus on analyzing localization accuracy under varying propagation conditions.

Paper IV: I. YAMAN, G. TIAN, D. PJANIC, F. TUFVESSON, O. EDFORS, Z. ZHANG, AND L. LIU, "Adaptive Attention-Based Model for 5G Radio-based Outdoor Localization", *IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Oct 2025, pp. 192-197.

Contribution: The adaptive attention-based localization model for outdoor 5G scenarios was proposed and developed by the author. Contributions include model design, implementation, training, and evaluation across different propagation conditions, as well as analysis of robustness under varying environments.

Paper V: I. YAMAN, S. CHENG, O. EDFORS, AND L. LIU, "Implementation of an Adaptive Transformer Accelerator for Accurate Outdoor Localization with Massive MIMO", *Submitted for review*.

Contribution: The hardware accelerator for the adaptive Transformer-based localization model was designed by the author. Contributions include architectural design, development of dataflow and processing strategies, implementation, and evaluation in terms of latency, power consumption, and computational efficiency.

RELATED WORK

The following papers are not included in the thesis, but summarize related work that I was involved in.

Paper vi: I. YAMAN, A. ANDERSEN, L. FERREIRA AND J. RODRIGUES, "FLoPAD-GRU: A Flexible, Low Power, Accelerated DSP for Gated Recurrent Unit Neural Network". *34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI)*, Campinas, Brazil, 2021, pp. 1-5.

Paper vii: G. TIAN, I. YAMAN, M. SANDRA, X. CAI, L. LIU AND F. TUFVESSON, "Deep-Learning-Based High-Precision Localization With Massive MIMO". *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, 2024, pp. 19-33.

Paper viii: L. BÁRÁNY, I. YAMAN, O. EDFORS, A. AMINIFAR, L. LIU, "Multi-User XR Offloading via Massive MIMO: A System-Level Analysis Using a Real-Life Dataset", *2026 IEEE Workshop on Signal Processing Systems (SiPS)*, accepted for publication.

Acronyms and Abbreviations

Here, important acronyms and abbreviations that recur throughout the thesis are listed. Acronyms and abbreviations that are used only within a narrow context are omitted.

GLOSSARY

AI	Artificial Intelligence
AoA	Angle-of-Arrival
ASIC	Application-Specific Integrated Circuit
BS	Base Station
CDF	Cumulative Distribution Function
CIR	Channel Impulse Response
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSI	Channel State Information
DL	Deep Learning
DoA	Direction-of-Arrival
6DoF	Six Degrees of Freedom
FCNN	Fully Connected Neural Network
FF	Flip-Flop
FFN	Feedforward Neural Network
FFT	Fast Fourier Transform

FPGA	Field-Programmable Gate Array
GCC-PHAT	Generalized Cross-Correlation with Phase Transform
GNSS	Global Navigation Satellite System
GPP	General Purpose Processor
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
IFFT	Inverse Fast Fourier Transform
IMU	Inertial Measurement Unit
IS	Input Stationary
LiDAR	Light Detection and Ranging
LoS	Line of Sight
LSTM	Long Short-Term Memory
LuMaMi	Lund University Massive MIMO testbed
LUT	Lookup Table
LuViRA	Lund University Vision, Radio, and Audio
MAC	Multiply ACcumulate
ME	Mean Euclidean Error
MHA	Multi-Head Attention
MIMO	Multiple-Input Multiple-Output
MISO	Multiple-Input Single-Output
ML	Machine Learning
MoE	Mixture-of-Experts
MPC	Multipath Component
MSE	Mean Squared Error
NLoS	Non Line of Sight
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Division Multiplexing
ORB	Oriented FAST and Rotated BRIEF
OS	Output Stationary
PDP	Power Delay Profile
PE	Processing Element
PTP	Precision Time Protocol
PTQ	Post-Training Quantization
QAT	Quantization-Aware Training

Radar	Radio Detection and Ranging
RANSAC	RANdom SAMple Consensus
ReLU	Rectified Linear Unit
RF	Radio Frequency
RGB-D	Red-Green-Blue-Depth
RNN	Recurrent Neural Network
RS	Row Stationary
RSS	Received Signal Strength
RTT	Round-Trip Time
SfM	Structure-from-Motion
SGD	Stochastic Gradient Descent
SIMO	Single-Input Multiple-Output
SISO	Single-Input Single-Output
SLAM	Simultaneous Localization and Mapping
SRAM	Static Random-Access Memory
SSM	State-Space Model
STE	Straight-Through Estimator
TDoA	Time-Difference-of-Arrival
ToA	Time-of-Arrival
ToF	Time-of-Flight
TPU	Tensor Processing Unit
TTL	Transistor-Transistor Logic
UE	User Equipment
UTC	Coordinated Universal Time
UWB	Ultra-Wideband
WS	Weight Stationary
XR	Extended Reality

Symbols

Some symbols might be used for different parameters in different contexts; but the intended meaning is clarified in the corresponding text.

GLOSSARY

γ	Azimuth angle
$\mathbf{K}_c$	Camera intrinsic matrix
$h(\tau)$	Channel impulse response
$\mathbf{H}[k]$	Channel matrix at subcarrier k
d	Distance
η	Elevation angle
$\hat{\mathbf{p}}$	Estimated position vector
$\mathbf{T}$	Homogeneous transformation matrix
e_i	Localization error for sample i
$\mathcal{L}(\theta)$	Loss function
θ	Model parameters
N_r	Number of receive antennas
N_t	Number of transmit antennas
θ^*	Optimized model parameters

f_θ	Parameterized localization model
ϑ	Pitch angle
Δp	Position difference
$\mathbf{p}$	Position vector
v	Propagation speed
P_r	Received signal power
ϕ	Roll angle
$\mathbf{R}$	Rotation matrix
t_{tof}	Time of flight
Δt	Time offset
$\mathbf{t}$	Translation vector
ψ	Yaw angle

INTRODUCTION

 HIS chapter introduces the problem of localization* and its importance in modern applications ranging from robotics and autonomous systems to wireless communications. It outlines the fundamental challenges associated with estimating position and the surrounding environment of a target object under varying conditions, highlighting limitations of existing solutions such as Global Navigation Satellite System (GNSS) in indoor and dense urban scenarios. The need for robust, low-latency, and energy-efficient localization systems is motivated through practical use cases and system requirements. Furthermore, the chapter defines the scope of the thesis by considering localization measurements, algorithm design, and hardware implementation as parts of the same system-level problem.

1.1 LIMITATIONS OF GNSS-BASED LOCALIZATION

The need for localization has driven innovation for centuries. Long before modern electronic navigation systems, people relied on tools like the magnetic compass, first invented in China over a thousand years ago, to find their way. Sailors used compasses to navigate vast oceans, while explorers carried them to traverse unfamiliar lands.

Today, one of the most common localization technologies is GNSS, which is widely used for outdoor navigation. GNSS receivers estimate their position by measuring the propagation time of signals transmitted from multiple

*Localization typically refers to estimating the position of an object relative to its surrounding environment, whereas positioning denotes the determination of an object's spatial coordinates within a defined coordinate system [6,7]. In this thesis, the terms are used interchangeably for simplicity.

satellites. Since the receiver clock is not perfectly synchronized with the satellite clocks, the receiver must estimate both its position and its clock offset. For three-dimensional positioning, at least four visible satellites are typically required: three unknowns correspond to the receiver coordinates, and one unknown corresponds to the receiver clock offset. While GNSS performs well in open-air environments, its accuracy can degrade in indoor spaces, tunnels, dense urban areas, and other settings where satellite signals are blocked or reflected [8]. In urban canyons, for example, tall buildings restrict sky visibility and introduce multipath propagation, which can significantly reduce positioning performance [9,10]. Therefore, many applications require localization methods that remain reliable under more challenging and realistic conditions.

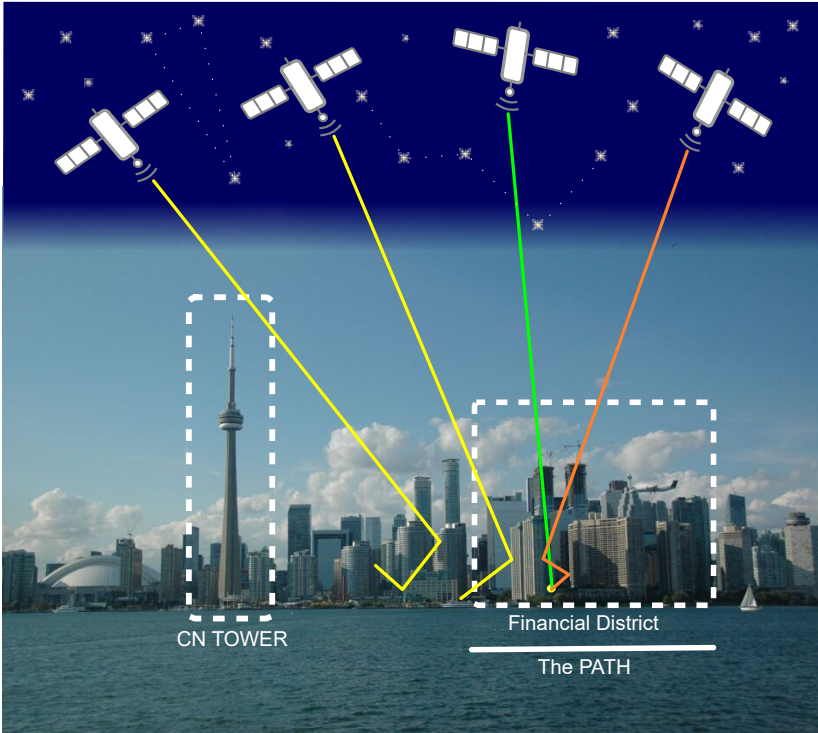


Figure 1.1: Illustration of the urban canyon effect in Toronto’s financial district. The target object is marked with a yellow diamond. A direct LoS satellite path is shown in green, while an indirect multipath component is shown in orange. A satellite without a direct LoS path is shown in yellow.

Toronto’s financial district, illustrated in Fig. 1.1, provides one example of this challenge. In such an environment, GNSS performance can degrade

because satellite signals may be blocked by surrounding buildings or received through reflected Non Line of Sight (NLoS) paths. Beneath the same district lies PATH, a mostly underground pedestrian walkway network spanning more than 30 km. Although PATH provides sheltered pedestrian access during Toronto's harsh winters, GNSS-based positioning is generally unavailable in such environments. This example shows that localization is challenged by both outdoor urban canyons and indoor or underground environments, where reliable positioning remains difficult. The thesis addresses both settings by investigating localization methods under varying outdoor propagation conditions and in indoor environments where satellite-based positioning is unavailable.

1.2 PRACTICAL USE CASES

In practical systems, the estimated position is usually used as an input to a larger task, such as navigation, tracking, control, mapping, or interaction with the surrounding environment. The required localization performance therefore depends on the application. A system that tracks assets in a warehouse may only need coarse position estimates, while a mobile robot, autonomous vehicle, or Extended Reality (XR) device requires more frequent and accurate updates [11, 12].

In robotics, localization is a central part of autonomous operation. A mobile robot needs to know its position in order to plan a path, avoid obstacles, and interact with objects or people around it. Similar requirements exist in smart factories, where robots, tools, products, and vehicles may need to be tracked continuously. In these environments, the localization system must often work indoors, where satellite-based positioning is not reliable, and where the surroundings can change due to moving objects, machines, or people.

Autonomous driving is another application where localization is critical. A vehicle needs to estimate its position with respect to roads, lanes, obstacles, and other vehicles. In this case, localization is not only used to determine the global position of the vehicle, but also to understand its local position within the surrounding traffic environment. Urban areas make this problem more difficult, since buildings can block satellite signals and create reflections. Tunnels, parking garages, and dense city streets are examples where additional sensing methods are needed.

XR applications also place strong requirements on localization. In augmented and virtual reality, the position and orientation of the device must be updated frequently to match the motion of the user. Even small delays or errors can reduce the quality of the experience, and induce motion sickness. For this reason, XR systems commonly rely on Inertial Measurement Units

(IMUs) together with other sensors, such as cameras or radio measurements, to support fast and continuous tracking.

Localization is also useful in applications such as agriculture, emergency response, and healthcare. In agriculture, positioning can support autonomous machines and monitoring of large outdoor areas. In emergency response, localization can help track people or equipment in buildings where GNSS is unavailable. In healthcare and assisted living, indoor localization can be used to monitor movement or support safety-critical services.

1.3 SENSORS AND SYSTEM REQUIREMENTS

Different localization applications impose different requirements on the localization system. These include positioning accuracy and latency, as well as constraints on update rate, computational resources, and energy consumption. For this reason, real-world localization systems are often designed around the requirements of the target application. Example use cases and their requirements are given in Table 1.1.

Table 1.1: Use cases and their requirements [13, 14].

Use case	Positioning accuracy	Latency
Augmented reality (sports)	< 0.5 m	< 20 ms
Augmented reality (professions)	< 0.1 m	< 10 ms
Virtual reality (gaming)	< 0.1 m	< 1 ms (control) < 100 ms (data)
Realtime digital twins (manufacturing)	0.1 m	< 1 ms - 50 ms
Patient monitoring (wearables)	< 1 m	< 200 ms
Human-robot co-working	–	1 ms
Tracking (goods & inventory)	< 0.1 m	< 100 ms - 1 s
Tracking (people in large venues)	< 1 m	< 1 s
Tracking (Robots & vehicles)	< 0.1 m	< 10 ms
Smart home	< 0.1 m	< 100 ms

These requirements show that localization performance cannot be defined by accuracy alone. Many applications also require frequent position updates

and low latency, while practical systems must operate within available computational and energy constraints.

A localization system estimates the position of a device or object based on measurements from the surrounding environment. An illustration of different sensors used to estimate the position and trajectory of a target object is shown in Fig. 1.2. In some applications, the system may also need to provide a more comprehensive understanding of the environment, for example through mapping, tracking, or interaction with other objects. Systems that directly provide position information, such as GNSS, are useful in many outdoor scenarios; however, they do not provide a general solution to the localization problem, as discussed above.

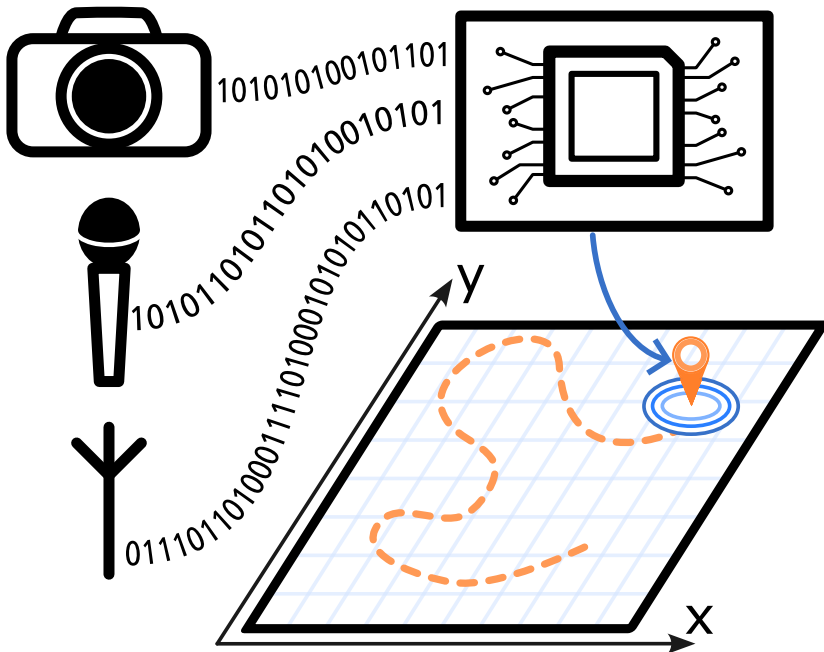


Figure 1.2: Cameras, microphones, and antennas can be used to calculate location information.

Different sensors provide different types of information for localization. Cameras capture images of the environment. One of the most common ways to use cameras for localization is by extracting key features from images and using them as anchor points. This approach is widely studied in the field of computer vision. Visual Simultaneous Localization and Mapping (SLAM) is a widely used technique that allows devices to build a map of their surroundings while simultaneously estimating their position within that

map. Vision-based methods can provide rich spatial information, but they are sensitive to illumination, occlusion, texture, and weather conditions.

Radio-based localization uses wireless signals, such as Wi-Fi, Bluetooth, or cellular signals. One advantage of radio signals is that they can often be measured without direct visual contact between the transmitter and receiver. This makes them useful in indoor and obstructed environments. A common approach is fingerprinting, where signal characteristics are collected at known positions and later compared with new measurements to estimate location.

Modern wireless systems provide new possibilities for radio-based localization. In particular, massive Multiple-Input Multiple-Output (MIMO) systems use large antenna arrays, which provide spatial information about the propagation environment. These measurements can be useful for estimating position, since the received signal depends on the location of the user and the surrounding objects. At the same time, the use of many antennas and wide bandwidths increases the amount of data that has to be processed.

Audio-based localization uses sound measurements to estimate position or direction. Microphone arrays can be used to estimate the direction of a sound source, and time-based methods can estimate position from the difference in arrival time of a signal at different receivers. Audio can be useful when visual information is limited, but it is affected by noise, reverberation, and the geometry of the environment. One fascinating use case is studying the behavior of bats, where researchers analyze echolocation signals to track their movement in the wild. In everyday applications, microphone arrays enable voice assistants like Alexa or Google Home to detect the direction of a speaker's voice and respond accordingly.

In practical systems, several sensors are often combined because each modality has different strengths and limitations. For instance, an autonomous device may use cameras, IMUs, and radio measurements together to improve robustness under changing conditions.

The choice of sensor modality can be considered a system-level design decision because it determines how the environment is observed and influences the available measurements, algorithms, and hardware. Accordingly, the papers in this thesis consider vision-, radio-, and audio-based localization systems. When the localization system is considered at the measurement, algorithm, and hardware levels, the sensor type is included at the measurement level, together with the characteristics of the resulting measurements. These measurement choices then influence the subsequent algorithm and hardware design.

The choice of sensor and localization method therefore depends on the application, the environment, and the system requirements. A method that performs well in one setting may not be suitable in another. This makes the

sensing setup, the available measurements, the algorithm, and the implementation constraints important parts of the overall localization system.

1.4 RESEARCH CHALLENGES

The previous section shows that different applications impose different requirements on localization systems, including accuracy, latency, and implementation constraints. This creates challenges not only in selecting a suitable sensing modality, but also in collecting and processing measurements under realistic conditions.

Vision, radio, and audio systems all provide different types of information about the environment. However, comparing these modalities is not straightforward. To make a fair comparison, the measurements should be collected under the same conditions, synchronized in time, and associated with a reliable ground-truth position. Otherwise, it becomes difficult to separate the effect of the sensor from the effect of the measurement setup. Simulated data can be useful for controlled studies, but it may not capture all practical effects that appear in real measurements, such as hardware impairments, synchronization errors, calibration errors, noise, and environmental changes.

Even after the data has been collected, several practical challenges remain. Measurements from different sensors are often represented in different coordinate systems and have different sampling rates, resolutions, and noise characteristics. The data must therefore be aligned, calibrated, and validated before it can be used for localization. This is especially important for multi-modal datasets, where inconsistencies between sensors can directly affect the evaluation of the localization algorithms.

After selecting a sensing modality, the next challenge is to design algorithms that are accurate and reliable under realistic conditions. In radio-based localization, the measured channel depends strongly on the propagation environment. Indoor and outdoor scenarios can have very different channel characteristics, and LoS, NLoS, and mixed conditions can lead to different measurements. A method that works well in one environment may therefore lose accuracy when the propagation conditions change. This makes robustness and adaptability important requirements for learning-based radio localization.

Accuracy alone is not sufficient for practical deployment. Many localization applications require frequent position updates, low latency, and limited power consumption. This becomes challenging when the localization algorithm uses high-dimensional sensor data or complex machine learning models. For example, massive MIMO radio measurements can provide rich spatial information, but they also increase the amount of data that must be processed.

Similarly, machine learning models can improve localization performance, but they introduce computational and memory-access costs that may be difficult to support on resource-constrained platforms.

1.5 RESEARCH QUESTIONS AND THESIS CONTRIBUTIONS

The thesis therefore treats localization as a co-design problem across measurements, algorithms, and hardware implementation. Design choices across these three domains are considered together to achieve the required localization accuracy and real-time performance under realistic operating conditions, while limiting computational, memory, hardware resource, and energy costs. The research presented in this thesis addresses the following questions:

1. How can measurements, algorithm design, and hardware implementation be considered jointly with system requirements and constraints to achieve high localization accuracy, real-time operation, and reduced energy consumption?
2. How can localization algorithms based on different modalities be developed, evaluated, and compared using real-world measurements? What are the advantages and limitations of vision-, radio-, and audio-based localization?
3. How can 5G massive MIMO measurements be used for localization, and how do measurement structure and system requirements affect algorithm selection and the trade-off between localization accuracy and computational complexity?
4. How can propagation conditions and measurement characteristics be used to design learning-based radio localization algorithms that improve localization performance while reducing computational and hardware implementation costs?
5. How can adaptive learning-based radio localization models be implemented in hardware while considering latency, computational resources, and energy requirements?

The thesis addresses these questions across real-world data collection, algorithm development, and hardware realization. An overview of the measurement, algorithm, and hardware design parameters considered in the thesis is shown in Fig. 1.3.

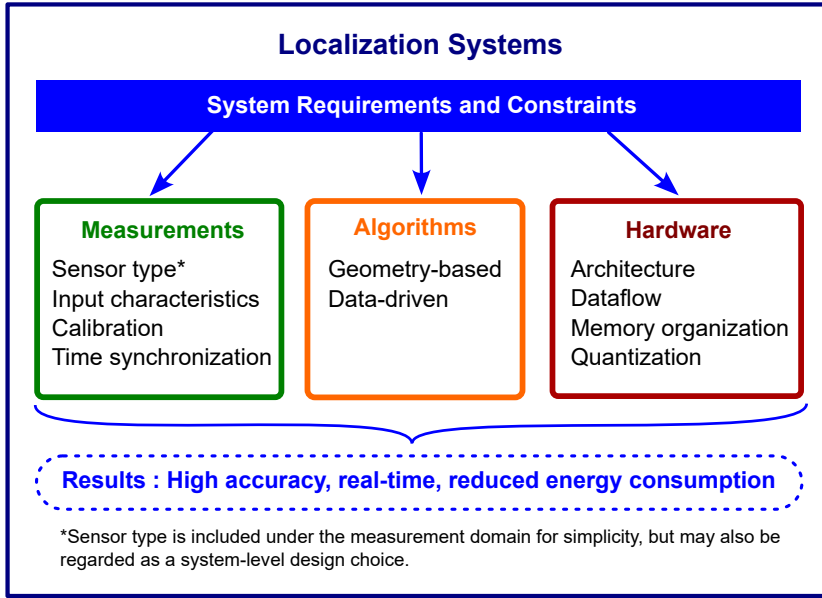


Figure 1.3: Co-design of localization systems across measurement, algorithm, and hardware implementation.

The main contributions of the thesis can be summarized as follows:

- Joint analysis and optimization of selected design choices across the measurement, algorithm, and hardware implementation levels of indoor and outdoor localization systems.
- Development and evaluation of a synchronized multimodal localization dataset containing vision, radio, and audio measurements together with accurate ground truth positions, enabling localization methods based on different sensing modalities to be evaluated under common measurement conditions.
- Investigation of learning-based localization using 5G massive MIMO measurements, including how the structure of the radio measurements influences model selection, localization accuracy, and computational complexity.
- Development of adaptive radio localization using real-world measurements from a commercial 5G massive MIMO base station under varying propagation conditions. Measurement characteristics are used to select specialized localization models, improving localization accuracy while reducing computational complexity.

- Based on the adaptive models, development and implementation of processing architecture for low-latency adaptive radio localization, including quantization, mixed dataflow, and sparsity-aware processing.

Localization Systems and Measurements



LOCALIZATION systems combine sensor measurements with models and coordinate representations to estimate the position or pose of a target. This chapter first introduces the main system concepts used throughout the thesis, including state representation, coordinate frames, and the distinction between point-source and rigid-body models. The notion of pose as a Six Degrees of Freedom (6DoF) representation is established. Different sensing modalities are then considered, with focus on vision, radio, and audio systems. The sensing modality determines the type of information available for localization as well as properties such as measurement resolution and sensitivity to the surrounding environment. Two fundamental types of localization measurements, range and bearing, are introduced and related to the different sensing modalities [15].

Measurements collected using physical sensors are affected by the measurement setup and hardware. In contrast to simulated data, real measurements may contain hardware impairments, noise, calibration errors, synchronization errors, and variations in the measurement conditions. These effects make data collection more challenging, but they are also essential for evaluating localization systems under realistic conditions. Calibration and time synchronization are therefore discussed as important parts of the measurement process, allowing measurements from different sensors to be related to common spatial and temporal references.

2.1 LOCALIZATION STATE REPRESENTATION

Point source and rigid body motion are two models for describing objects in localization problems. A point source treats an object as a single point,

neglecting its size and rotation, and is described by three translational degrees of freedom in 3D space. In contrast, a rigid body has six degrees of freedom: three translational and three rotational. The *pose* of an object describes both its position and orientation.

Accordingly, if a sensor is modeled as a point source, its state is defined solely by its position, whereas a 3D object requires both position and orientation. The position is represented by the translation vector $\mathbf{t} = [x; y; z]^\top$, while the orientation can be represented using Euler angles (yaw ψ , pitch ϑ , roll ϕ) or by a rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$, obtained from the composition of individual rotation matrices ^{*}:

$$\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\vartheta)\mathbf{R}_x(\phi), \quad (2.1)$$

where

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.2)$$

$$\mathbf{R}_y(\vartheta) = \begin{bmatrix} \cos \vartheta & 0 & \sin \vartheta \\ 0 & 1 & 0 \\ -\sin \vartheta & 0 & \cos \vartheta \end{bmatrix} \quad (2.3)$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

The full pose can then be compactly expressed using a homogeneous transformation matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \quad (2.5)$$

which fully characterizes the rigid-body transformation in three-dimensional space. In practical systems, such as the motion capture setups used in the Lund University Vision, Radio, and Audio (LuViRA) dataset described in this thesis, reference data is provided both as three-dimensional positions and full 6DoF poses. Rigid objects need to be tracked with at least three non-collinear markers to enable estimation of both translation and orientation, while the individual markers themselves are modeled as point sources.

^{*}A right-handed coordinate system is used, with positive rotations defined according to the right-hand rule. Left-handed coordinate systems and alternative rotation conventions are also used in the literature [15].

Localization is inherently dependent on the choice of coordinate frame. A distinction is typically made between global and local coordinate systems. The global frame defines an absolute reference, often tied to a fixed environment or external positioning system, whereas the local frame is attached to the sensor and evolves over time.

Transformation matrices are used to express measurements from different sensors in a common reference frame. This is necessary when sensors use different coordinate systems, since their measurements must be represented in the same frame before they can be compared or combined. A similar issue appears when evaluating the absolute trajectory error. If the estimated trajectory and the ground truth trajectory are not already expressed in the same coordinate frame, a rigid transformation can first be estimated between them [16]. The absolute trajectory error can then be computed from the differences between corresponding poses of the estimated and reference trajectories [17]. Relative trajectory metrics, on the other hand, compare changes in pose over a time interval and are therefore less dependent on the choice of global coordinate frame. In a dataset, the relationship between different cameras (i.e. two imagers and the color camera) and the transformation matrix between the cameras and IMU are obtained and used as extrinsic parameters of the camera.

2.2 SENSORS

The choice of sensors determines what physical information can be used to estimate position. Different sensors measure different physical quantities: cameras capture visual information, antennas receive radio signals affected by the propagation environment, and microphones capture acoustic signals and their timing differences. Other sensors, such as Radio Detection and Ranging (Radar), Light Detection and Ranging (LiDAR), and IMUs, are also widely used in localization and robotics. Radar and LiDAR actively probe the environment using reflected signals, while IMUs measure the acceleration and angular velocity of the device itself. However, the main focus of this thesis is on systems based on cameras, antennas, and microphones, which are introduced in the following subsections.

Sensing methods can also be broadly divided into passive and active approaches. Passive systems measure signals that are already present in the environment, such as images formed by ambient light or sounds captured by microphones. Active systems, on the other hand, transmit a signal and use the received response for localization, as in radar, sonar, and many radio-based systems.

Ground truth systems are also important when collecting datasets for localization research. They provide reference positions or trajectories that can be used for training, evaluation, and comparison of localization algorithms. As an example, motion capture systems can provide highly accurate marker positions using multiple calibrated cameras placed around the measurement area. This makes them useful as external reference systems in controlled indoor experiments. However, they require LoS visibility to the markers, careful calibration, and a fixed measurement setup, and are therefore mainly used for dataset collection and evaluation.

2.2.1 VISION SENSORS

Cameras provide visual observations of the environment and are widely used in vision-based localization. Depending on the camera setup, the measurements may consist of color images, grayscale images, stereo image pairs, or depth images. These observations can be used to estimate the motion and pose of the camera by detecting visual structure in the scene and relating this structure across multiple frames.

In a monocular setup, a single camera is used. The corresponding localization or mapping approach is referred to as monocular SLAM. Since a single image does not directly provide absolute depth, monocular systems suffer from scale ambiguity. Depth can only be inferred from camera motion and feature correspondences over time, and the true metric scale cannot be recovered without additional information [18].

Stereo cameras and depth sensors address this limitation by providing depth-related information. In a stereo setup, depth is estimated by comparing corresponding points observed by two cameras with a known relative pose. Red-Green-Blue-Depth (RGB-D) cameras provide both color and depth images, where the depth image gives the distance from the camera to visible surfaces. Depending on the sensor, depth can be obtained using stereo matching, active infrared projection, structured light, or Time-of-Flight (ToF) measurements. When color and depth are captured by different sensing elements, the images may have different fields of view and must be registered to a common image frame before they can be used jointly.

The use of multiple images is also central to Structure-from-Motion (SfM), where three-dimensional scene structure and camera motion are estimated jointly from feature correspondences and geometric constraints across views. These principles are used in many vision-based localization systems, including visual odometry and SLAM, where the camera pose is estimated from changes in the observed scene over time.

Compared with radio or audio sensors, cameras provide rich spatial information in each frame, but their temporal resolution is limited by the

frame rate. Vision systems commonly operate at frame rates on the order of 10-60 Hz, although this depends on the sensor and acquisition setup. Since a camera observes the environment from a particular viewpoint, its state is described by both position and orientation. Therefore, vision-based localization typically estimates the full camera pose rather than only a point position.

2.2.2 RADIO SENSORS

Radio systems consist of antennas that transmit and receive electromagnetic signals through the wireless channel. Depending on the antenna configuration, a radio system can be classified as Single-Input Single-Output (SISO) when both the transmitter and receiver have one antenna, Single-Input Multiple-Output (SIMO) or Multiple-Input Single-Output (MISO) when only one side has multiple antennas, and MIMO when both sides have multiple antennas. A SISO system is illustrated in Fig. 2.1 [19].

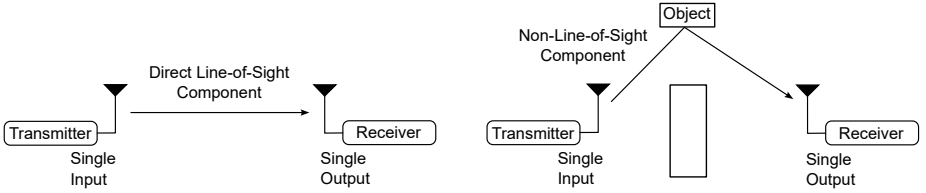


Figure 2.1: Illustration of a wireless communication channel between transmit and receive antennas.

In cellular systems, communication takes place between Base Stations (BSs) and User Equipments (UEs). In the uplink, the UE transmits to the BS, while in the downlink, the BS transmits to the UE. Other radio technologies, such as WiFi and Ultra-Wideband (UWB), are also used for localization, with different tradeoffs in coverage, bandwidth, infrastructure requirements, and accuracy.

For a radio link with N_t transmit antennas and N_r receive antennas, known pilot signals can be transmitted to estimate the channel between the transmitter and receiver. For an Orthogonal Frequency Division Multiplexing (OFDM) system, the received signal on subcarrier k can be expressed as

$$\mathbf{y}[k] = \mathbf{H}[k]\mathbf{x}[k] + \mathbf{n}[k], \quad (2.6)$$

where $\mathbf{x}[k] \in \mathbb{C}^{N_t}$ is the transmitted signal vector, $\mathbf{y}[k] \in \mathbb{C}^{N_r}$ is the received signal vector, $\mathbf{H}[k] \in \mathbb{C}^{N_r \times N_t}$ is the complex-valued channel matrix for subcarrier k , and $\mathbf{n}[k] \in \mathbb{C}^{N_r}$ represents additive noise [13]. The collection of channel estimates across the subcarriers describes the frequency-domain

channel response and is commonly referred to as Channel State Information (CSI).

The measured channel depends on how the transmitted signal propagates through the environment and is affected by path loss, shadowing, reflections, diffraction, scattering, and the antenna configuration. The wireless channel typically contains multiple propagation paths, referred to as Multipath Components (MPCs). These paths arise from interactions with the surrounding environment and arrive at the receiver with different delays, amplitudes, phases, and angles. This structure provides information about the position of the transmitter or receiver, but it also depends strongly on the propagation conditions. In particular, the LoS and NLoS cases shown in Fig. 2.1 differ in their multipath structure and in the degree to which the measured channel is distorted by indirect propagation. Both the magnitude and phase of the channel coefficients can therefore contain useful localization information [20,21]. In practical systems, however, stable and calibrated phase information can be difficult to obtain because of hardware impairments, synchronization errors, and phase variations between measurements.

The channel described above can also be represented in the delay domain using the Channel Impulse Response (CIR). For an OFDM system, a sampled delay-domain representation can be obtained by applying an Inverse Fast Fourier Transform (IFFT) across the frequency-domain channel estimates. For a single transmit-receive antenna pair, the continuous-time baseband CIR can be written as

$$h(\tau) = \sum_{\ell=1}^L \alpha_{\ell} \delta(\tau - \tau_{\ell}), \quad (2.7)$$

where L is the number of multipath components, α_{ℓ} is the complex gain of path ℓ , τ_{ℓ} is the corresponding propagation delay, and $\delta(\cdot)$ is the Dirac delta function. The delays and complex gains therefore describe the delay-domain multipath structure of the channel.

The Power Delay Profile (PDP) is derived from the CIR and describes how the received signal power is distributed over delay:

$$\text{PDP}(\tau) = \mathbb{E} \left[|h(\tau)|^2 \right]. \quad (2.8)$$

The PDP is useful because it provides a compact description of the relative strength and delay of the different propagation paths.

A single antenna at the transmitter or receiver can often be modeled as a point, while an antenna array has a fixed geometry defined by the positions of its elements. The known spacing between the antenna elements makes it possible to compare the received signals across the array and relate their differences to the direction or position of the transmitter or receiver.

Massive MIMO systems

Massive MIMO systems extend the multiple-antenna principle by using large antenna arrays, typically at the BS. The large number of antennas can provide high spatial resolution and make it possible to observe fine differences in the radio channel across space. This is useful for localization, since the measured CSI depends on the position of the UE, the surrounding propagation environment, and the array geometry.

The use of multiple antennas enables spatial diversity and spatial multiplexing. In localization, the same spatial information can be used to distinguish between different user positions. Larger bandwidths improve delay resolution, while shorter wavelengths and larger array apertures improve the ability to resolve spatial differences in the channel. In practical systems, CSI is often estimated for multiple antennas, subcarriers, and time instances, resulting in high-dimensional channel measurements.

One example of such a system is the Lund University Massive MIMO testbed (LuMaMi), which was among the first real-time massive MIMO testbeds developed for wireless communication research [22]. Throughout this thesis, massive MIMO testbeds are used as BSs for radio-based localization, including in **Papers I-III**. These systems typically provide measurements at rates ranging from tens to hundreds of Hz, depending on the pilot signaling configuration and system design.

The measured channel can also be transformed into alternative representations that expose structure useful for localization. For example, applying an IFFT across the frequency-domain subcarriers gives a delay-domain representation, while applying a Fast Fourier Transform (FFT) across the antenna dimension gives a beam-domain representation. Combining these transformations results in a beam-delay representation, which exposes structure across the spatial and delay dimensions and is useful for learning-based localization methods [23].

2.2.3 AUDIO SENSORS

Audio-based localization relies on sound sources and microphones, often arranged in arrays, to capture acoustic signals from the environment [24]. Microphones sample sound pressure at rates in the kHz range, providing high temporal resolution compared with many visual sensors. The recorded signals contain information about the source, the propagation path, and the environment, but this information is affected by the acoustic content of the source, background noise, reverberation, and the speed of sound.

In many systems, the sound source is modeled as a point source, while the microphone array is assumed to have a fixed and calibrated geometry [25]. The known spacing between microphones makes it possible to compare the

signals received at different microphones and relate their differences to the source direction or position. Reflections from walls, floors, and objects create multipath components, while reverberation can make the direct path harder to identify. In addition, the speed of sound depends on environmental conditions, such as temperature, which can introduce errors in localization systems based on acoustic propagation. For more details on acoustic sensing and sound propagation for localization, please see [26].

2.3 RANGE MEASUREMENTS

Range measurements describe the distance between a signal source and a receiver. The most direct approach is based on the signal propagation time. If the ToF is denoted by t_{tof} , the corresponding propagation distance is

$$d = v \cdot t_{\text{tof}}, \quad (2.9)$$

where d is the propagation distance and v is the propagation speed. For radio signals, v is approximately the speed of light, whereas for acoustic signals it is the speed of sound.

The ToF is not always observed directly. A Time-of-Arrival (ToA) measurement records the time at which a signal reaches the receiver. If the transmission time is known and the transmitter and receiver share a common timing reference, the ToF can be obtained from the difference between the arrival and transmission times.

When the transmitter and receivers do not share a common clock, relative arrival-time measurements can be used instead. Time-Difference-of-Arrival (TDoA) measurements describe the difference in arrival time of the same transmitted signal at two receivers. For receivers i and j , the corresponding distance difference, Δd_{ij} , can be written as

$$\Delta d_{ij} = d_i - d_j = v \cdot (t_i - t_j), \quad (2.10)$$

where d_i and d_j are the propagation distances from the source to receivers i and j , respectively, and t_i and t_j are the corresponding arrival times measured at those receivers. The time difference $t_i - t_j$ removes the unknown transmission time, but the receivers must still share a synchronized time reference, or their relative clock offsets must be known or estimated.

Another common approach is round-trip-time-based ranging, where a signal is sent from one device to another and a response is returned. In this case, the propagation time is estimated from the measured Round-Trip Time (RTT), after accounting for any known processing or turnaround delay. This avoids the need for tight clock synchronization between the two devices, but the accuracy depends on how well the response delay is known or controlled.

In vision systems, depth measurements provide distance information for visible surfaces. A depth map assigns a depth value to each image pixel or image region. Depending on the sensor and coordinate convention, this value may represent distance along the camera's optical axis or distance along the corresponding viewing ray. Depth maps can be obtained using stereo cameras, structured-light systems, or ToF-based depth cameras. An RGB-D camera provides both color images and corresponding depth measurements.

In radio systems, distance can also be inferred from measurements that are not based directly on propagation time. One such measurement is Received Signal Strength (RSS), which describes the received signal power. For a received signal $r(t)$, the average received power can be written as

$$P_r = \mathbb{E} \left[|r(t)|^2 \right]. \quad (2.11)$$

Received power generally decreases with propagation distance according to an environment-dependent path-loss model and can therefore provide coarse range-related information [15]. However, RSS does not determine distance directly, and its relationship to distance is affected by shadowing, antenna orientation, multipath propagation, and variations in transmit power. Similar distance-dependent attenuation can occur in acoustic systems, although the corresponding propagation models differ.

In microphone arrays, arrival-time differences between microphones can be estimated from recorded audio using methods such as Generalized Cross-Correlation with Phase Transform (GCC-PHAT) [27]. These time-delay estimates can be converted into propagation-distance differences using the speed of sound and subsequently used to estimate the source direction or position.

2.4 BEARING MEASUREMENTS

Bearing measurements describe the direction from a sensor or receiver array toward a target or signal source. Unlike range measurements, they do not directly determine the distance between the sensor and the target. In two dimensions, a bearing is fully described by a single azimuth angle. In three dimensions, it is described by two angles: an azimuth γ and an elevation η .

Given a bearing (γ, η) , the unit vector pointing toward the target in the sensor's local frame is

$$\mathbf{u}_s(\gamma, \eta) = \begin{bmatrix} \cos \eta \cos \gamma \\ \cos \eta \sin \gamma \\ \sin \eta \end{bmatrix} \in \mathbb{R}^3, \quad \|\mathbf{u}_s(\gamma, \eta)\| = 1. \quad (2.12)$$

The direction can be expressed in the world frame using the sensor-to-world rotation matrix $\mathbf{R}_{ws}$:

$$\mathbf{u}_w = \mathbf{R}_{ws} \mathbf{u}_s(\gamma, \eta). \quad (2.13)$$

Together with the sensor position $\mathbf{p}_s$, the transformed direction defines a ray along which the target must lie:

$$\mathbf{p}_w(d) = \mathbf{p}_s + d \mathbf{u}_w, \quad d \geq 0. \quad (2.14)$$

Here, d denotes the distance from the sensor to the target along the ray [28]. A single bearing measurement determines the direction $\mathbf{u}_w$ but leaves d unknown. The target position therefore cannot be recovered without additional information, such as a range measurement or bearing measurements obtained from multiple sensor positions.

Calibrated cameras provide bearing information through image coordinates. A point observed in an image corresponds to a viewing direction in the camera coordinate frame. A single image observation therefore constrains the target to lie along a viewing ray but does not determine its distance from the camera. Range can instead be recovered by combining observations from multiple viewpoints, using known object geometry, or incorporating depth information.

In radio systems, the Angle-of-Arrival (AoA) describes the direction from which a signal reaches a receiving antenna array. It can be estimated from the relative phase or time differences observed across the array elements. Since these differences depend on the antenna locations and signal wavelength, AoA estimation requires knowledge of the array geometry and calibration of the relative phase and timing responses of the receiver channels.

In audio systems, the Direction-of-Arrival (DoA) of a sound source can similarly be estimated using a microphone array. The direction is inferred from the relative arrival times or phases of the acoustic signal at different microphones. For example, pairwise time delays estimated using GCC-PHAT can be related to the source direction through the known microphone geometry and the speed of sound. In the far-field approximation, the incoming wavefront is treated as approximately planar, and the measured time differences primarily describe direction rather than distance. For nearby sources, the curvature of the wavefront may also contain range information.

Across vision, radio, and audio systems, converting the sensor observations into bearing measurements requires knowledge of the sensor geometry and relevant calibration parameters.

2.5 CALIBRATION

The calibration required for localization depends on the sensor, the type of measurement, and the algorithm used. Range estimation based on propagation time requires compensation for fixed delays in the measurement system and accurate time synchronization, which will be discussed in the next section. Bearing estimation instead depends strongly on the sensor geometry and on parameters such as camera intrinsics or relative phase and delay offsets across microphone and antenna channels.

For cameras, calibration determines the parameters that relate a 3D point in the world to its corresponding 2D image coordinate. Using the pinhole camera model, a world point $\tilde{\mathbf{p}}_w = [x_w, y_w, z_w, 1]^T$ is projected onto a point $\tilde{\mathbf{p}}_c = [u, v, 1]^T$ in the camera frame as

$$s\tilde{\mathbf{p}}_c = \mathbf{K}_c \begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw} \end{bmatrix} \tilde{\mathbf{p}}_w, \quad \mathbf{K}_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.15)$$

where s is a scale factor, $\mathbf{K}_c$ is the intrinsic camera matrix, and $\mathbf{R}_{cw}$ and $\mathbf{t}_{cw}$ describe the rotation and translation from the world coordinate frame to the camera coordinate frame. The parameters f_x and f_y are the focal lengths in pixel units, and (c_x, c_y) is the principal point of the camera in pixel units.

In practical cameras, the ideal pinhole projection is also affected by lens distortion. This means that the observed image points may deviate from their ideal locations, especially near the image boundaries. Lens distortion is commonly described using radial distortion, which bends straight lines outward or inward, and tangential distortion, which can occur when the lens and image sensor are not perfectly aligned. These distortion effects are estimated during calibration together with the other camera parameters.

Camera calibration therefore estimates both intrinsic parameters, such as focal length, principal point, and distortion, and extrinsic parameters, which describe the camera position and orientation relative to a world frame or another sensor. Although approximate intrinsic parameters may be provided by the manufacturer or through factory calibration, they may not be accurate enough for all localization tasks. Therefore, the selected camera is often calibrated directly using a known calibration target, such as a checkerboard, captured from several viewpoints, as illustrated in Fig. 2.2 [29].

Since the geometry of the target is known, the detected corner locations can be compared with their predicted image locations, and the calibration parameters can be estimated by minimizing the reprojection error. The resulting intrinsic and extrinsic parameters are typically stored in calibration files, often using the YAML format, which is widely supported by computer

vision and robotics frameworks such as OpenCV and ROS. Storing the calibration results allows distortion correction, coordinate transformations, and pose estimation algorithms to consistently use the same camera model across different applications.

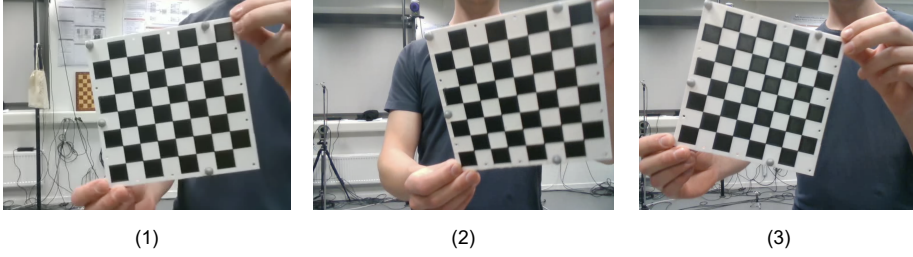


Figure 2.2: Example of a checkerboard, captured from several viewpoints.

For radio-based localization, the calibration requirements depend on the channel features and localization method being used. Methods that explicitly exploit the antenna-array geometry require sufficiently accurate knowledge of the antenna-element positions and, when estimates are expressed in a room or global coordinate frame, the position and orientation of the array. Calibration of the Radio Frequency (RF) hardware may additionally be required to account for differences among the RF chains. In particular, AoA estimation depends on the spatial response of the antenna array and requires accurate knowledge of the array geometry and the relative gain, phase, and delay responses of the RF chains [30, 31]. In contrast, supervised Machine Learning (ML)-based fingerprinting can learn a direct mapping from measured channel features to position without separately calibrating every RF chain, as demonstrated in **Papers II-III**. This relies on the relevant hardware- and array-dependent effects remaining consistent between training and deployment. Changes in the antenna configuration or RF-chain responses due to hardware replacement, temperature variation, or component drift may otherwise degrade localization performance.

Audio calibration mainly concerns the microphone geometry and differences among the microphone acquisition channels. For array-based DoA or TDoA estimation, the microphone positions must be known because the expected arrival-time differences depend on the relative positions of the microphones. These positions may be obtained through direct measurement or estimated from recorded audio. In audio localization, this is often referred to as self-calibration, sensor-network self-calibration, or structure from sound, and may involve jointly estimating unknown microphone and sound-source positions [32, 33]. Calibration may also compensate for differences in microphone sensitivity and relative phase and delay responses. The importance of

these parameters depends on the localization method: gain differences mainly affect amplitude-based methods, while relative phase and delay errors are particularly important for bearing and TDoA-based methods.

The practical effort required for calibration differs across sensor types. Camera calibration is well established and can often be performed using a single calibration target. In systems with multiple cameras, the target must provide sufficient shared observations to determine the position and orientation of each camera in a common coordinate frame. Radio-array calibration can be particularly demanding when localization relies on phase differences across the antenna elements. Differences among the RF chains can be characterized using conducted measurements, for example by applying known signals through cable connections. If the response of the antenna array itself must also be characterized, over-the-air measurements may be required. Such measurements can be performed in a controlled environment, such as an anechoic chamber, to reduce the influence of surrounding reflections. Microphone-array calibration becomes more demanding when the microphones are distributed across the environment or recorded using separate acquisition devices, since the microphone positions and differences in the gain, phase, or delay responses of the acquisition paths may need to be characterized.

The LuViRA campaign presented in **Paper I** illustrates these different calibration requirements. The cameras were calibrated because their intrinsic and extrinsic parameters were required to relate image measurements to the camera geometry and the common coordinate system. For the radio system, no separate calibration of the antenna-array response for localization was performed, since the learning-based methods in **Papers II and III** used the measured channel responses directly. For the audio system, the known microphone positions were used in the localization algorithm, although the selected method can also jointly estimate unknown microphone and source positions. The localization calculations also depend on the assumed speed of sound, which varies with environmental conditions such as temperature.

2.6 TIME SYNCHRONIZATION

Measurements collected by sensors and ground truth systems are usually assigned timestamps using the clocks of their acquisition devices. These clocks must refer to a common time scale so that measurements associated with the same target state can be identified. This is required both when jointly processing measurements from multiple sensors and when comparing sensor measurements or localization estimates with ground-truth positions. Without correct time alignment, the measurement and reference position

may correspond to different positions of a moving target, leading to errors in training and evaluation [29]. Unix time is commonly used to represent timestamps as the elapsed time since 1 January 1970 in Coordinated Universal Time (UTC).

A common physical event can be used to approximately align sensor streams. One example is a movie clapper, as illustrated in Fig. 2.3. When the clapper is closed in front of the camera, it creates a visual event in the image sequence and an acoustic event in the microphone recordings. These two events can then be matched in time to synchronize the camera and audio measurements. In the same setup, reflective markers used by the motion-capture system are also visible, allowing the motion-capture measurements to be aligned with the camera and audio recordings. However, this approach requires manual inspection of each recorded stream to identify and match the clapper event. The resulting time alignment is limited by the sensor sampling rates, since the event may occur between camera frames or extend over several audio samples. Differences in the delay between the physical measurement and timestamp generation may introduce an additional offset between the sensor streams. It also corrects only the initial time difference; if the devices use independent clocks, differences in their clock rates may cause the streams to drift apart during longer recordings.

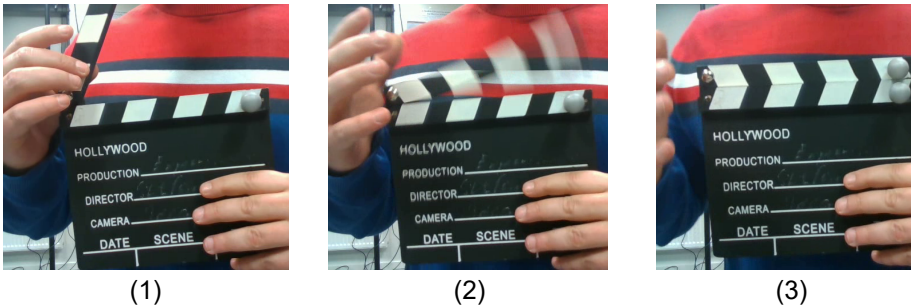


Figure 2.3: Example of a clapper providing a common visual and acoustic event for aligning camera and microphone measurements. Reflective markers can also make the event observable by a motion-capture system.

The sensors may also operate at different sampling rates, as illustrated in Fig. 2.4. Their measurements therefore do not generally occur at exactly the same times, even when their clocks are synchronized. A shared time reference allows measurements from the different streams to be associated using their timestamps, for example by selecting the closest samples or interpolating between measurements.

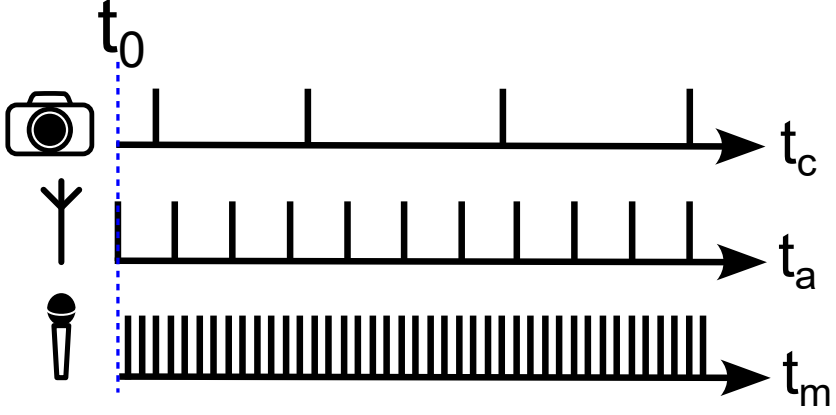


Figure 2.4: Illustration of camera, microphone, and antenna streams sampled at different rates over the same time interval. A common time reference allows measurements to be associated despite their different sampling instants.

Network-based synchronization can establish a shared time reference and limit clock drift throughout a recording. Network Time Protocol (NTP) periodically adjusts computer clocks using timing information exchanged over a network. It is widely used in computer networks and can provide millisecond-level synchronization, although the achieved performance depends on the network and system configuration [34, 35]. For systems requiring higher timing precision, Precision Time Protocol (PTP), standardized in IEEE 1588, supports sub-microsecond synchronization between devices connected over a network [36].

The required synchronization accuracy depends on the target motion, sensor sampling rates, and how the measurements are used. As an example, a target velocity of $v_r = 0.1$ m/s is considered, corresponding approximately to the robot speed in the measurement campaign presented in **Paper I**. A time offset Δt then introduces an approximate position difference of

$$\Delta p \approx v_r \Delta t. \quad (2.16)$$

A time difference of 10ms therefore corresponds to approximately 1mm of motion, while a difference of 100ms corresponds to approximately 1cm.

In the LuViRA measurement system, synchronization was established using a combination of network-based clock synchronization, hardware signals, and shared physical events. A Raspberry Pi operated as a local NTP server and periodically synchronized the acquisition computers, including the computer used for audio recording. This provided a common time reference across the acquisition computers and limited clock drift during the recordings.

For the radio system, the UE and LuMaMi were connected by cable to provide a common frequency reference and eliminate carrier-frequency offset, avoiding the need for over-the-air frequency synchronization.

A Transistor-Transistor Logic (TTL) signal generated by the Qualisys synchronization unit at the start and end of each motion-capture recording was detected by the Raspberry Pi and supplied to the audio system. For the audio recordings, this signal was converted into an audio pulse and recorded on a separate synchronization channel, while all microphones were connected to the same sound card to maintain synchronization between the microphone channels. Additional events generated by the radio synchronization circuit and the clapperboard were used to verify the temporal alignment between the radio, vision, audio, and ground-truth measurements. These references also allowed remaining systematic time offsets to be estimated and corrected during post-processing. According to Eq. (2.16), millisecond-level synchronization was sufficient for the robot speed and the millimetre-level accuracy of the motion-capture system. The complete synchronization setup and its verification for each sensor system are described in more detail in **Paper I**.

Localization Algorithms



LOCALIZATION algorithms use measurements and representations obtained from the sensing system to estimate position. Geometry-based algorithms determine position from geometric relationships, signal propagation, or feature correspondences, whereas data-driven algorithms learn mappings from measurement representations to position. Geometry-based methods depend on how well the assumed geometric and measurement models represent the actual measurement conditions, as well as on synchronization, calibration, and measurement quality. Data-driven methods instead depend on representative training data and their ability to generalize to new conditions.

Localization algorithms can also be developed depending on whether they use a single measurement or a sequence of measurements. Single-snapshot methods estimate the position from one sensor observation at a time. Multi-snapshot methods use temporal information from consecutive observations, which can improve stability and tracking performance.

Temporal information can also be incorporated through filtering techniques, such as Kalman or particle filters, which combine measurements over time with previous position estimates to exploit information about the motion of the target. A filter may be included within the localization algorithm or applied to its position estimates afterward. This can reduce fluctuations caused by measurement noise, compensate for brief missing measurements, and produce trajectories that are more consistent with the motion of the target. The choice of filter depends on the system model and the form of the uncertainty [37,38]. Although filtering is outside the scope of this thesis, it is relevant to distinguish the localization algorithm itself from subsequent filtering of its estimates. The focus is therefore on the localization methods rather than on the filtering techniques used to process their outputs.

3.1 GEOMETRY-BASED ALGORITHMS

Some of the main geometry-based algorithms are triangulation, trilateration, and multilateration. These methods use the range or bearing measurements introduced in the previous chapter. The reference points may be cameras, BSs, microphones, or other objects with known positions. They are briefly defined as follows and illustrated in Fig. 3.1:

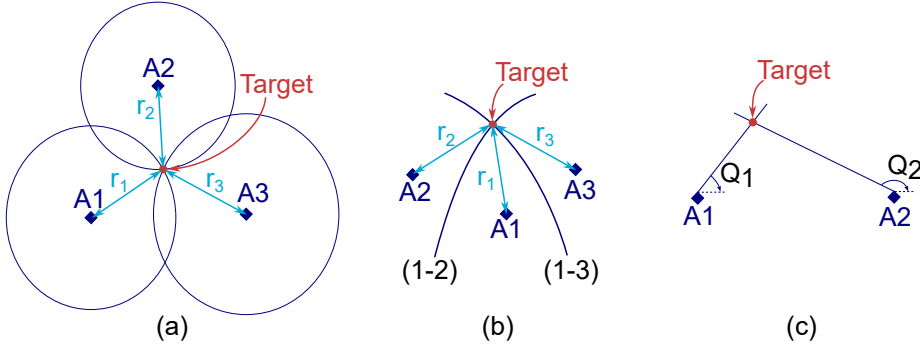


Figure 3.1: (a) ToA-based trilateration, (b) TDoA-based multilateration, and (c) AoA-based triangulation for two-dimensional localization using anchors denoted by A.

1. **Trilateration** estimates position from distances to known reference points [26]. In two-dimensional localization, each distance measurement defines a circle centered at a reference point. The target position is estimated from the intersection of these circles. With three reference points at known coordinates (x_i, y_i) and distance estimates d_i , the position (x, y) satisfies

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2, \quad i = 1, 2, 3. \quad (3.1)$$

In ideal conditions, the three circles intersect at the target position, as illustrated in Fig. 3.1(a). In practical systems, noise and measurement errors usually prevent an exact intersection, and the position is estimated using optimization or least-squares methods.

2. **Multilateration** generalizes this idea to multiple reference points or to relative distance measurements. In many localization systems, the term is used for TDoA-based positioning, where the measured quantity is not the absolute distance to each receiver, but the difference in distance between pairs of receivers. For receivers i and j , the distance-difference constraint can be written as

$$d_i - d_j = v(t_i - t_j), \quad (3.2)$$

where v is the propagation speed, and t_i and t_j are the arrival times at receivers i and j , respectively. In two-dimensional localization, each distance-difference measurement defines a hyperbola, and the target position is estimated from the intersection of multiple such constraints, as seen in Fig. 3.1(b).

3. **Triangulation** estimates position from angles measured with respect to known reference points. In two-dimensional localization, each angular measurement defines a bearing line from a known reference point [39]. The target position is estimated from the intersection of two or more bearing lines. Therefore, triangulation is commonly associated with AoA- or DoA-based localization, an example is shown in Fig. 3.1(c).

Robust estimation methods are used when some measurements or associations are unreliable. One common example is RANdom SAMple Consensus (RANSAC), which repeatedly estimates a model from small subsets of the data and selects the solution supported by the largest number of measurements [40]. This is useful when feature matches, time-delay estimates, or other measurement constraints contain outliers. In audio localization, such outliers may arise from reverberation, where reflections from walls, the floor, the ceiling, and surrounding objects produce delayed and attenuated copies of the emitted sound. These reflections can obscure the direct propagation path and lead to inconsistent delay estimates. In the audio localization system presented in **Papers I and II**, the SFS2 algorithm is therefore combined with RANSAC to estimate positions from microphone measurements while reducing the effect of such inconsistencies [41].

The localization problem can be extended to include simultaneous map construction. When the sensor trajectory and a map of the surrounding environment are estimated jointly, the problem is referred to as SLAM. These estimates are interdependent: the map is constructed using the estimated sensor poses, while the sensor poses are determined relative to the estimated map. In **Papers I and II**, Oriented FAST and Rotated BRIEF (ORB)-SLAM3 is used as a camera-based SLAM method and provides a reference for evaluating localization from the collected image data. It estimates the camera trajectory and map from geometric relationships between matched image features and uses robust estimation to reduce the effect of incorrect correspondences. For example, RANSAC is used during place recognition to estimate the transformation between matched map regions [42]. The method combines feature extraction and matching with pose estimation, map construction, and optimization.

3.2 DATA-DRIVEN ALGORITHMS

Data-driven approaches use statistical learning to map measurements to spatial coordinates without explicitly relying on geometric models. A data-driven algorithm learns a function that maps an input representation directly to a position estimate, without modeling the underlying propagation geometry. Such a method is defined by three components: the input representation, the learning model, and the training task.

The input representation can be derived from images, audio recordings, or radio-channel measurements. When measurement representations are associated with known reference positions, they can be interpreted as fingerprints. In radio-based localization, fingerprints use location-dependent channel characteristics, including RSS, the CIR, multipath structure, or derived representations such as the PDP. Since the measured channel depends on the positions of the transmitter and receiver relative to the surrounding environment, nearby locations often produce similar fingerprints, while different locations may produce distinguishable fingerprints.

Fingerprints collected at known reference positions can be stored in a database or used to train a learning-based model. During localization, a new measurement can be compared with the stored fingerprints using methods such as nearest-neighbor matching or processed by a trained model that directly estimates the position. In the latter case, the localization task is commonly formulated as regression, where the predicted coordinates are compared with known reference coordinates during training [23]. In **Papers II-V**, location-dependent radio-channel representations are used as fingerprints, and learning-based models map them directly to spatial coordinates.

During training, each sample consists of an input measurement representation $\mathbf{x}_i \in \mathbb{R}^N$ and a corresponding ground truth position $\mathbf{p}_i \in \mathbb{R}^D$, where $D = 2$ for two-dimensional localization and $D = 3$ for three-dimensional localization. The learned mapping

$$f_{\theta} : \mathbb{R}^N \rightarrow \mathbb{R}^D \quad (3.3)$$

is parameterized by θ , typically the weights of a neural network, and produces a position estimate

$$\hat{\mathbf{p}}_i = f_{\theta}(\mathbf{x}_i). \quad (3.4)$$

Given a training set $\{(\mathbf{x}_i, \mathbf{p}_i)\}_{i=1}^M$ of M samples, the parameters θ are obtained by minimizing an empirical loss that quantifies the discrepancy between estimated and ground truth positions. A common choice for position regression is the Mean Squared Error (MSE),

$$\mathcal{L}_{\text{MSE}}(\theta) = \frac{1}{M} \sum_{i=1}^M \|\hat{\mathbf{p}}_i - \mathbf{p}_i\|_2^2 = \frac{1}{M} \sum_{i=1}^M \sum_{k=1}^D (\hat{p}_{i,k} - p_{i,k})^2. \quad (3.5)$$

The parameters are optimized as

$$\theta^* = \arg \min_{\theta} \mathcal{L}_{\text{MSE}}(\theta), \quad (3.6)$$

typically via stochastic gradient-based optimization (e.g., Stochastic Gradient Descent (SGD)) with gradients computed through backpropagation. After training, θ^* is fixed, and inference on a new, previously unseen measurement $\mathbf{x}_i$ is performed as $\hat{\mathbf{p}}_i = f_{\theta^*}(\mathbf{x}_i)$; in real-time localization, this step must additionally satisfy the latency, throughput, memory, and energy constraints of the target system.

The formulation above corresponds to supervised learning, where labeled ground truth positions are available during training. Unsupervised methods can instead be used to learn representations without labeled data, while reinforcement learning is mainly relevant when localization is part of a sequential decision-making problem. Among ML methods, Deep Learning (DL) models have been widely used for both localization and SLAM. For example, DL has been applied to vision-based SLAM [43], as well as vision-, radio-, and audio-based localization [21, 44–46]. Different architectures suit different input structures: Fully Connected Neural Networks (FCNNs) process fixed-size feature vectors, Convolutional Neural Networks (CNNs) exploit local structure in images or radio-channel representations, attention-based models learn relationships between input components, and sequence models such as Recurrent Neural Networks (RNNs), and Mamba process ordered measurements over time.

3.2.1 BASICS OF NEURAL NETWORKS

Neural networks are one way to implement the learnable function $f_{\theta}(\cdot)$ introduced above. The parameters θ correspond to quantities such as weight matrices and bias vectors, which are updated during training. Given an input vector $\mathbf{x} \in \mathbb{R}^N$, a fully connected layer can be expressed as

$$\mathbf{y} = g(\mathbf{W}\mathbf{x} + \mathbf{b}), \quad (3.7)$$

where $\mathbf{y} \in \mathbb{R}^M$ is the layer output, $\mathbf{W} \in \mathbb{R}^{M \times N}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^M$ is the bias vector, and $g(\cdot)$ denotes a non-linear activation function. An example neural network with two layers is shown in Fig. 3.2.

Other important components include:

- **Activation functions:** introduce non-linearity into neural networks, allowing them to approximate non-linear mappings between inputs and outputs. Without non-linear activation functions, multiple layers of affine transformations could be reduced to a single affine transformation, limiting the network's ability to model complex non-linear

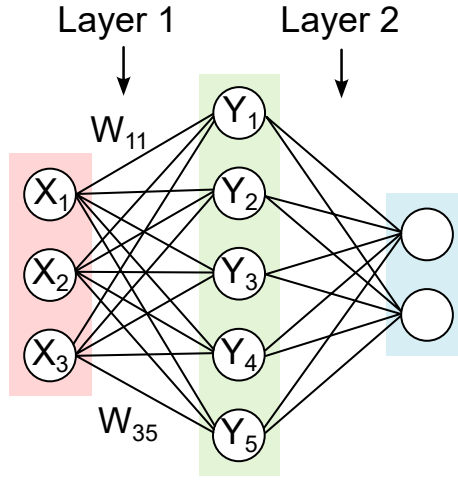


Figure 3.2: Illustration of a neural network with two layers [47].

relationships in the data. Common activation functions include the Rectified Linear Unit (ReLU), leaky ReLU, and sigmoid functions, as illustrated in Fig. 3.3.

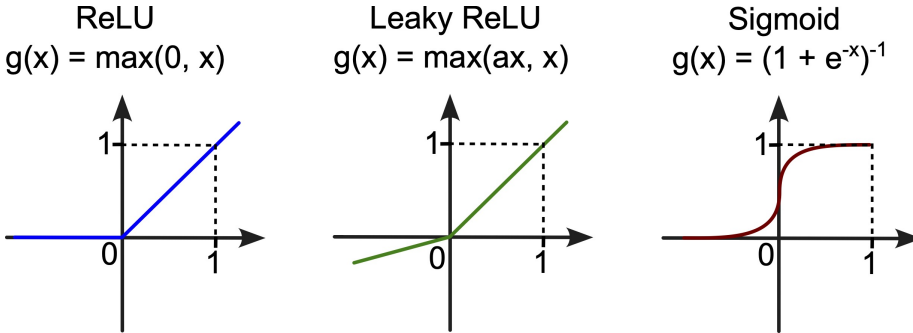


Figure 3.3: Examples of commonly used activation functions.

- **Normalization layers:** normalize intermediate activations to improve training stability and convergence. Different layers compute the normalization statistics over different dimensions. Batch normalization uses statistics estimated across a group of training samples processed together, while layer normalization computes them across the feature dimension for each input independently. Instance and group normalization instead normalize individual samples over spatial dimensions or groups of features. The normalized activations are typically scaled

and shifted using learnable parameters. The choice of layer depends on the network architecture, input representation, and batch size.

- **Pooling layers:** reduce the size of an intermediate feature representation by combining values within local windows. In one-dimensional max pooling, the maximum value is selected within each window along one dimension of the input. This reduces the number of values passed to later layers and can therefore decrease the number of parameters and computations in the following parts of the model.

For an input length L_{in} , kernel size K_s , stride S_t , and padding P_a , the output length of a one-dimensional pooling layer is

$$L_{\text{out}} = \left\lfloor \frac{L_{\text{in}} + 2P_a - K_s}{S_t} \right\rfloor + 1. \quad (3.8)$$

In this thesis, max pooling is used between the encoder layer and the FCNN to reduce the size of the feature representation before the final position-regression layers. For example, using a kernel size of $K_s = 4$, stride $S_t = 4$, and padding $P_a = 1$ downsamples the input while preserving the strongest activation within each pooling window.

- **Dropout:** is a regularization technique used to reduce overfitting by randomly setting a fraction of intermediate values in the network to zero during training. In standard dropout, this is typically applied to layer activations. During inference, dropout is disabled and the complete network is used.

3.2.2 FULLY CONNECTED AND CONVOLUTIONAL NEURAL NETWORKS

FCNNs consist of fully connected layers in which each output neuron receives input from all neurons in the preceding layer. A fully connected layer can be written as

$$\mathbf{y} = g(\mathbf{W}\mathbf{x} + \mathbf{b}), \quad (3.9)$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors, $\mathbf{W}$ and $\mathbf{b}$ are learned parameters, and $g(\cdot)$ is an activation function. In localization, FCNNs can learn a direct mapping from a fixed-size measurement representation to spatial coordinates. However, spatial, temporal, or sensor-specific relationships are not explicitly encoded by the fully connected structure.

CNNs instead apply shared kernels across structured inputs, allowing local patterns to be extracted while preserving spatial or sequential relationships. For a two-dimensional input $\mathbf{X}$ and kernel K_f , the convolution operation is

$$Y(i, j) = \sum_m \sum_n X(i - m, j - n) K_f(m, n), \quad (3.10)$$

where $Y(i, j)$ is the resulting feature-map value. Depending on the measurement representation, convolution can be applied across image coordinates, time samples, frequency bins, antennas, or other structured dimensions.

FCNNs and CNNs have been used in data-driven localization to learn mappings from measurement representations to positions. FCNNs are well suited to fixed-size feature vectors, while CNNs can exploit local structure in representations such as images, channel matrices, and fingerprint maps. In **Papers II-III**, FCNNs are used for radio-based localization by mapping channel-derived input representations directly to spatial coordinates.

3.2.3 TRANSFORMER MODELS

Transformers are attention-based models that learn relationships between different parts of an input representation and assign larger weights to the most relevant information. Since self-attention does not inherently represent input order, Transformers commonly add positional encoding to indicate the position of each input element. Self-attention processes all elements in parallel and can capture dependencies across time steps, sensors, antennas, or spatial features. Given an input $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{model}}}$ containing N elements, for a single attention operation, the query, key, and value matrices are obtained as

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V. \quad (3.11)$$

where $\mathbf{W}_Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, and $\mathbf{W}_V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are the learnable projections for the queries, keys, and values, respectively. Scaled dot-product attention is then computed as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} \right) \mathbf{V}, \quad (3.12)$$

where d_k is the dimension of the query and key vectors. The resulting attention weights determine how strongly each input element contributes to the output.

Multi-Head Attention (MHA) applies several attention heads in parallel using different learned projections:

$$\text{head}_i = \text{Attention}(\mathbf{X}\mathbf{W}_i^Q, \mathbf{X}\mathbf{W}_i^K, \mathbf{X}\mathbf{W}_i^V), \quad (3.13)$$

$$\text{MHA}(\mathbf{X}) = \text{Concat}(\text{head}_1, \dots, \text{head}_l) \mathbf{W}^O, \quad (3.14)$$

where l is the number of attention heads [48]. Different heads can capture complementary relationships within the input representation.

Each attention layer is typically followed by a position-wise Feedforward Neural Network (FFN). For localization, attention-based models can learn

relationships across measurements, sensor channels, or time steps and emphasize the information most relevant to position estimation. **Papers IV and V** investigate attention-based models for radio-based localization.

3.2.4 RECURRENT AND STATE-SPACE MODELS

RNNs process sequential inputs by updating a hidden state that carries information from previous time steps [49]. Given an input $\mathbf{x}_t$ at time t , a basic recurrent update is

$$\mathbf{h}_t = f(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{b}_h), \quad (3.15)$$

$$\mathbf{y}_t = g(\mathbf{W}_y \mathbf{h}_t + \mathbf{b}_y), \quad (3.16)$$

where $\mathbf{h}_t$ is the hidden state and $f(\cdot)$, $g(\cdot)$ are activation functions. Variants such as the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) use gating mechanisms to control how information is retained or discarded over longer sequences [50, 51].

Mamba is a sequence model based on selective State-Space Models (SSMs), with the recurrent form

$$\mathbf{h}_t = \bar{\mathbf{A}}_t \mathbf{h}_{t-1} + \bar{\mathbf{B}}_t \mathbf{x}_t, \quad (3.17)$$

$$\mathbf{y}_t = \mathbf{C}_t \mathbf{h}_t + \mathbf{D} \mathbf{x}_t. \quad (3.18)$$

where $\bar{\mathbf{A}}_t$ is the state-transition matrix, $\bar{\mathbf{B}}_t$ maps the input to the state, $\mathbf{C}_t$ maps the state to the output, and $\mathbf{D}$ represents the direct input-to-output mapping [52]. Unlike conventional SSMs with fixed parameters, Mamba introduces input-dependent state-space parameters, allowing the state dynamics to adapt to the input at each time step while maintaining linear computational complexity with respect to the sequence length.

In localization, recurrent and state-space models can exploit sequences of measurements to capture temporal dependencies and changes in the observed environment. In temporal or multi-snapshot localization, recurrent and state-space models can use information across consecutive measurements to capture temporal dependencies that are not available to single-snapshot methods.

4

Hardware Implementations

 EFFICIENT execution of localization algorithms requires careful consideration of hardware constraints, including power consumption, latency, computational resources, and memory requirements. Processing architecture design plays a critical role in addressing these constraints [53]. Different processing platforms, including General Purpose Processors (GPPs), Field-Programmable Gate Arrays (FPGAs), and Application-Specific Integrated Circuits (ASICs), offer distinct trade-offs in flexibility, performance, energy efficiency, and memory organization [54].

This chapter discusses processing architectures, including systolic arrays and parallel vector engines, dataflow strategies and memory hierarchy and organization. It also introduces quantization as a method for reducing arithmetic, storage, and memory-bandwidth requirements.

4.1 PROCESSING ARCHITECTURES

Processing architectures determine how computational units, memories, interconnections, and control mechanisms are organized to execute an algorithm in hardware. The basic computational unit is commonly referred to as a Processing Element (PE) and may contain arithmetic units, such as multipliers, adders, or Multiply ACcumulates (MACs), together with registers or local memory. The organization and interconnection of these PEs determine how operations are performed and how data are distributed and reused. This section presents systolic arrays and parallel vector engines as two processing architectures for matrix and vector operations.

4.1.1 SYSTOLIC ARRAYS

In many workloads, transferring data between memory and processing units consumes significantly more energy and time than the arithmetic operations themselves [55,56]. Systolic arrays address this challenge by maximizing data reuse and minimizing memory accesses [57]. Instead of repeatedly fetching the same data from external memory, data elements are loaded and reused across multiple computations as they propagate through the array. In a systolic architecture, data flows rhythmically between PEs, similar to the flow of blood through the circulatory system. Data is streamed from memory through multiple interconnected PEs before the results are written back to memory. Since the computation schedule and data movement pattern are predetermined, the need for explicit instruction control between operations can be significantly reduced. Each PE typically performs simple operations, such as MAC computations, and may include local memory for temporary storage and data reuse.

For example, as shown in Fig. 4.1, an output-stationary systolic array can be used to perform the following matrix multiplication:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}. \quad (4.1)$$

However, the regular structure of a systolic array requires the input data to be supplied in the correct order and at the correct time so that the required operands meet at each PE. As seen in the example above, additional buffering may therefore be needed to align the inputs, while filling and draining the array introduces some latency. For large and regular matrix operations, the computational benefits can outweigh these overheads. For example, Google's Tensor Processing Units (TPUs) have used systolic arrays with dimensions of 256×256 and 128×128 [58].

The performance of a systolic architecture therefore depends on balancing computational throughput with memory bandwidth and the cost of data movement between the PEs. This is achieved by organizing many simple PEs into a regular array structure and carefully optimizing the dataflow between neighboring PEs.

4.1.2 PARALLEL VECTOR ENGINES

A parallel vector engine distributes vector elements across several computational lanes that operate concurrently. As illustrated in Fig. 4.2 (b), the lanes receive their operands directly rather than passing data between neighboring

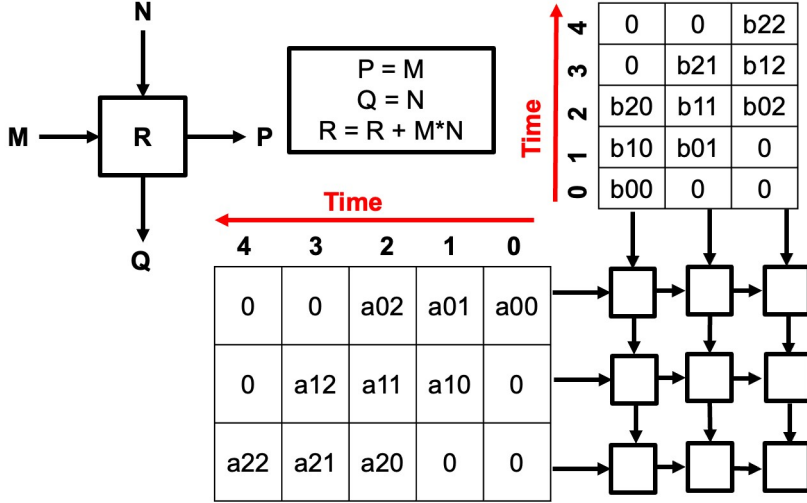


Figure 4.1: Example of matrix multiplication using a systolic array architecture with output stationary dataflow.

PEs, as in a systolic array. The products computed in parallel are then combined using a reduction structure, such as an adder tree. When the same operand is used by several lanes, it may be broadcast to the lanes while the remaining operands are supplied individually.

For the matrix multiplication in Eq. 4.1, elements from a row of the first matrix and a column of the second matrix can be distributed across the available lanes. The element-wise products are computed in parallel and accumulated to obtain an output element. When the inner matrix dimension exceeds the number of lanes, the dot product is divided across several clock cycles and the resulting partial sums are accumulated.

Compared with systolic arrays, parallel vector engines can provide greater flexibility for computations with varying dimensions or irregular execution patterns. Operands are supplied directly to the computational lanes, and the number of active lanes can be adapted to the operation. This is advantageous for irregular workloads, such as zero skipping, where a conventional systolic array may require additional control and buffering to preserve the required data alignment and can suffer from reduced PE utilization when the regular execution pattern is disrupted. Systolic arrays are instead well suited to large and regular matrix operations, where predictable data movement between neighboring PEs enables efficient local data reuse.

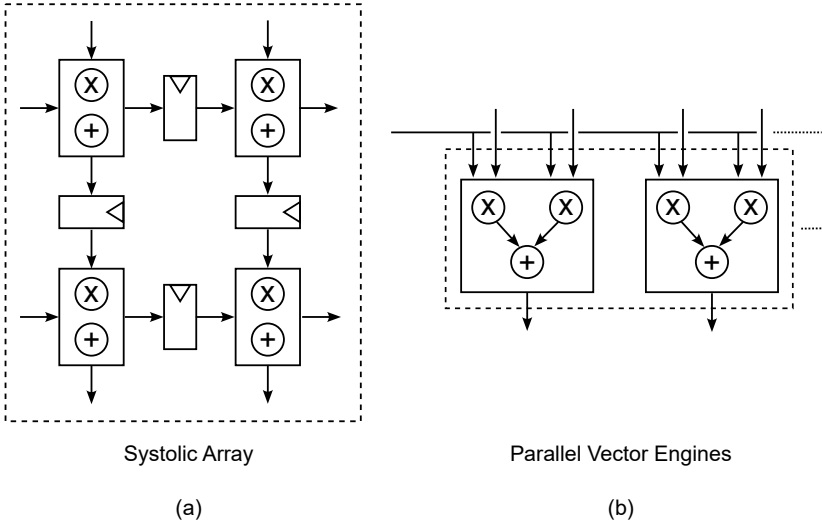


Figure 4.2: Comparison of (a) a systolic array, where data propagate between neighboring PEs, and (b) parallel vector engines, where parallel products are combined using reduction structures. Adapted from [59].

The added flexibility of parallel vector engines can increase the requirements on the memory system and interconnect, including support for varied memory access patterns. Different operands may need to be supplied independently to the lanes, while values that are reused across several lanes may need to be broadcast. Systolic arrays reduce these requirements by propagating data through a regular array, but provide less flexibility for computations with varying dimensions or irregular execution patterns. The choice between the two architectures therefore depends on the regularity and dimensions of the computation, available data reuse, memory bandwidth, interconnect requirements, and the required degree of parallelism.

4.2 MEMORY HIERARCHY AND ORGANIZATION

The energy required to execute an operation depends not only on the arithmetic itself, but also on where its operands are stored and how often they must be moved. Data locality is therefore important in accelerator design: temporal reuse allows frequently used data to remain in local storage across several operations, while spatial reuse allows the same data to be shared among multiple PEs.

The memory hierarchy is therefore organized to exploit these forms of locality. Frequently reused operands and intermediate results can be retained in registers or local memories close to the PEs, while data shared across several PEs can be stored in an on-chip buffer or transferred directly between processing elements. Large inputs and model parameters may initially reside in external memory and be transferred to on-chip Static Random-Access Memory (SRAM) before processing. In FPGA implementations, on-chip storage can be mapped to dedicated block memories, while small local buffers and lookup tables may also be implemented using distributed Lookup Table (LUT)-based memory.

Memory bandwidth determines how quickly operands can be supplied to the PEs and results can be written back. As the number of parallel computational units increases, a single memory port may not provide enough data per clock cycle. On-chip memories can therefore be divided into several independent banks that are accessed simultaneously, increasing the aggregate bandwidth. The distribution of data across these banks can be selected according to the expected access pattern, allowing several matrix or vector elements to be read in parallel. On the other hand, increasing the number of banks increases the area overhead because each memory bank requires additional peripheral circuitry.

The memory layout must also match the access pattern of the computation. For a matrix stored in row-major order, the elements of a row occupy consecutive memory locations, whereas the elements of a column are separated by the row length. Row-wise accesses can therefore be supplied sequentially, while column-wise accesses may require strided reads or a transposed representation. This distinction is important for matrix multiplication and matrix decomposition, where rows, columns, or both may be accessed repeatedly. A matrix can be stored in transposed form when column-wise access dominates, or the memory can be organized to support several access directions.

For example, the parallel vector memory in [60] distributes matrix elements across multiple memories and uses data shufflers to support access to a complete row, column, or diagonal in the matrix. The required index mapping mechanism is calculated according to [61, 62]. This allows the same computational units to process different matrix operations without repeatedly rearranging the data in software, as seen in Fig. 4.3. Memory organization is therefore closely connected to the selected dataflow, which determines which operands remain locally stored and when data are transferred between memory levels.

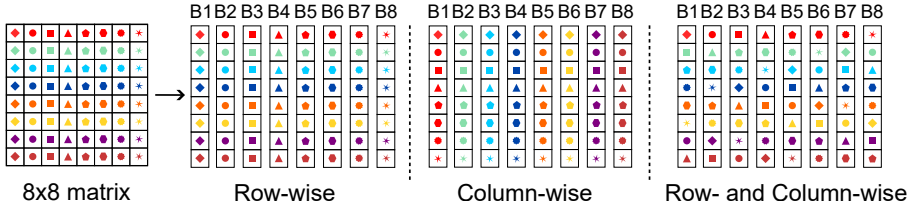


Figure 4.3: Mapping of the elements of an 8×8 matrix in a parallel memory system with 8 memory banks, denoted by $B_1, \dots, B_8$, allowing rows, columns, and diagonals to be accessed without bank conflicts. [60].

4.3 DATAFLOW

Efficient hardware design depends strongly on memory-access patterns and data reuse. In DL accelerators, a large fraction of the implementation cost can arise from moving inputs, weights, and intermediate results through the memory hierarchy rather than from the arithmetic operations themselves. Dataflow selection determines how these data are mapped to the PEs, which values remain locally available, and which values must be transferred between PEs or memory levels. Temporal reuse allows values to remain locally available across several operations, while spatial reuse allows values to be shared or forwarded among multiple PEs. Increasing such reuse can reduce memory traffic, execution latency, and energy consumption. Common dataflows include output-, input-, and weight-stationary execution, while hybrid approaches combine several forms of reuse.

- **Output Stationary (OS):** Each PE accumulates a specific output element or partial sum. The partial sum remains stationary while weights and input activations are supplied to the PE, reducing the movement of intermediate results. An example is shown in Fig. 4.4.
- **Input Stationary (IS):** Input activations (feature maps) are kept stationary in the PE registers. This is often used when the input data is significantly larger than the weights, or in specific streaming scenarios where buffering inputs is more efficient than buffering weights.
- **Weight Stationary (WS):** Filter weights are pre-loaded and kept fixed within the PEs. Input activations and partial sums stream through the array. This can be advantageous when weights are used across many input batches (e.g., in large convolutional layers).
- **Row Stationary (RS):** Row-stationary execution combines several forms of reuse by mapping rows of inputs and weights across the PE array, allowing reuse of inputs, weights, and partial sums [56].

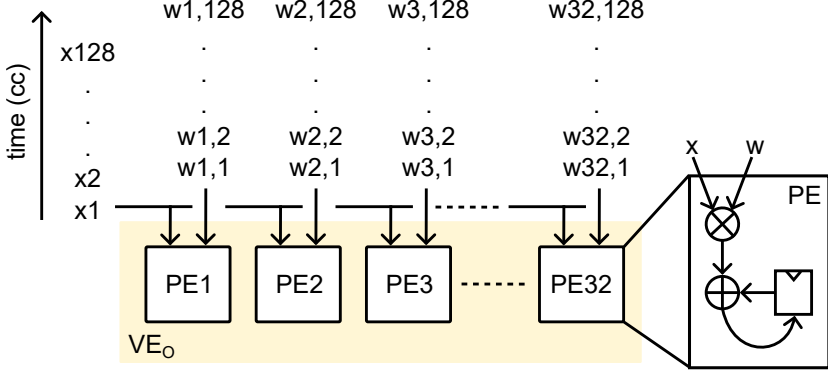


Figure 4.4: Example processing architecture of an OS dataflow taxonomy.

Since different operations provide different reuse opportunities, no single dataflow is optimal for all computations. For example, when both operands of a matrix multiplication are generated during execution rather than stored as pre-computed parameters, an OS dataflow can keep the partial sums locally while the input operands are streamed into the computation. This reduces the storage and memory traffic associated with intermediate results. In an image processing system, the same input data may be reused to compute multiple different features, and an IS dataflow can keep these inputs locally to reduce repeated memory accesses. In contrast, when pre-computed weights are reused across many input samples or operations, a WS dataflow can keep the weights close to the PEs and reduce repeated accesses to the memory hierarchy.

Dataflow selection must therefore consider the computation dimensions, memory organization, available on-chip storage, degree of parallelism, and hardware resources together with the latency, resource, and energy requirements [47].

4.4 QUANTIZATION

This section considers affine integer quantization, which reduces computational cost and memory footprint by mapping floating-point values to low-precision integer representations [63]. It is parameterized by a scaling factor and zero-point. For a set of values $\mathbf{X}$ that shares the same quantization

parameters, these are computed as

$$\begin{aligned} \text{scale} &= \frac{\beta - \alpha}{\beta_q - \alpha_q}, \\ \text{zeroPoint} &= \text{clip}\left(\text{round}\left(\alpha_q - \frac{\alpha}{\text{scale}}\right), \alpha_q, \beta_q\right), \end{aligned} \quad (4.2)$$

where $\alpha_q = -2^{(\text{num_bits}-1)}$, $\beta_q = 2^{(\text{num_bits}-1)} - 1$, and α and β denote the minimum and maximum values of $\mathbf{X}$.

Quantization is applied element-wise. The quantized representation is obtained as

$$\mathbf{X}_q = \text{clip}\left(\text{round}\left(\frac{\mathbf{X}}{\text{scale}} + \text{zeroPoint}\right), \alpha_q, \beta_q\right), \quad (4.3)$$

and de-quantization is given by

$$\mathbf{X} \approx \text{scale} (\mathbf{X}_q - \text{zeroPoint}). \quad (4.4)$$

Table 4.1 summarizes representative hardware cost trends across integer and floating-point formats. These formats are not exhaustive, as number formats designed specifically for ML have also been developed, including bfloat16 [64] and FP8 formats such as E4M3 and E5M2 [65].

Table 4.1: Representative energy cost trends for various operations in 45 nm, 0.9 V [55].

Format	Add	Multiply
INT8	0.03 pJ	0.2 pJ
INT32	0.1 pJ	3.1 pJ
FP16	0.4 pJ	1.1 pJ
FP32	0.9 pJ	3.7 pJ

Low-precision fixed-point arithmetic can reduce the resource requirements of arithmetic units on FPGA platforms. Table 4.2 shows representative LUT and Flip-Flop (FF) utilization for multiplier and adder IP cores synthesized using different number formats, as reported by Guo et al. [66]. In this study, Fixed8 denotes an 8-bit fixed-point implementation, which corresponds most closely to the INT8 arithmetic considered in this thesis. Compared with Fixed8, the FP32 multiplier uses approximately $9.4\times$ more LUTs and $10.7\times$ more FFs, while the FP32 adder uses approximately $53.8\times$ more LUTs and $93.6\times$ more FFs. Although the exact resource utilization depends on the device and implementation of the arithmetic IP cores, these results illustrate the potential hardware savings of low-precision arithmetic and motivate its use for deploying DL models on resource-constrained hardware.

Table 4.2: Representative FPGA resource utilization for individual arithmetic units [66].

Format	Multiplier		Adder	
	LUT	FF	LUT	FF
Fixed8	75	80	8	8
Fixed32	1112	1143	32	32
FP16	221	303	211	337
FP32	708	858	430	749

4.4.1 QUANTIZATION METHODS FOR DEEP LEARNING MODELS

Two main approaches are commonly used to quantize neural networks: Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT). PTQ applies quantization after training and can be effective for models with sufficient redundancy, but may introduce significant accuracy degradation in compact models [67]. In contrast, QAT simulates quantization and dequantization during the forward pass, allowing the model to adapt to reduced precision during training. Since rounding and clipping operations are non-differentiable, the Straight-Through Estimator (STE) can be used to approximate their gradients during the backward pass [68].

The scale and zero-point are commonly estimated using observers that collect information about the ranges of weights and activations. In PTQ, this is performed using a representative calibration dataset after training, whereas in QAT, the observers operate while the model is trained and may be fixed during the later training stages. Min-max calibration uses the observed minimum and maximum values, while alternative methods, such as percentile-based calibration, clip a small fraction of extreme values to reduce the influence of outliers [69]. When input distributions vary across scenarios, the calibration data should represent these variations so that the selected quantization ranges remain suitable during inference.

For weights, symmetric quantization is commonly used, with the zero-point set to zero, because weight distributions are often approximately centered around zero. This eliminates offset terms in multiplication and simplifies the hardware implementation. For strongly asymmetric weight distributions, however, asymmetric quantization may reduce the quantization error at the cost of additional hardware complexity. Per-channel symmetric quantization can also account for differences in the weight ranges across output channels while retaining a zero-point of zero. Activations and intermediate values are more commonly quantized asymmetrically because their distributions are not

necessarily centered around zero.

The hardware design space therefore includes several related choices, including the processing platform, memory organization, dataflow, processing architecture, numerical precision, and degree of parallelism. Techniques such as stationary dataflows, local data reuse, quantization, and sparsity exploitation can reduce computation and data movement, although their benefits depend on the algorithm and the available hardware resources. These choices should therefore not be considered independently. The next chapter takes a broader view of this design space by considering the localization system, algorithm, and hardware implementation together.

5

Co-Design of Localization Systems

 LOCALIZATION systems are often evaluated based on their positioning accuracy, but practical deployment requires a broader view. The sensing system must provide measurements that contain useful information about the target position, the algorithm must extract this information reliably, and the hardware must process the data within latency, energy, and resource constraints. These parts are strongly connected: the input representation and measurement environment affect the selection and design of the DL model, while the resulting model determines the computational and memory requirements of the implementation. Hardware capabilities and constraints can, in turn, influence the choice of algorithm, model structure, numerical precision, and input representation.

Previous works have studied similar system-level approaches in spatial Artificial Intelligence (AI), where real-time perception requires co-design across sensors, algorithms, and processors [70]. Real-time localization and mapping systems also require joint consideration of algorithm choice, hardware architecture, localization accuracy, computational performance, and energy consumption [71]. These studies have largely focused on visual perception and SLAM, whereas the perspective considered here also includes localization based on radio and audio measurements. In radio systems, algorithm-hardware co-design has been studied for deep-learning-based MIMO channel estimation and energy-efficient 5G positioning [72,73]. This chapter considers localization systems as a co-design problem across the measurement, algorithm, and hardware levels. It describes our methodology for design space exploration and optimization, introduces the relevant performance metrics and design parameters, and presents case studies illustrating the resulting design decisions.

5.1 DESIGN SPACE EXPLORATION AND OPTIMIZATION

Design space exploration is guided by the performance metrics and design parameters used throughout this thesis, which are described in the next section. The performance metrics provide a basis for formulating optimization objectives and constraints, such as reducing localization error, latency, energy consumption, and hardware-resource use while accounting for limits on inference time or available memory. Since these objectives can conflict, design space exploration generally aims to identify appropriate trade-offs rather than optimize one metric independently.

The design space is formed by identifying the parameters that can be varied at the measurement, algorithm, and hardware levels, together with the values or alternatives considered for each parameter. A comprehensive design space exploration would ideally use models of the relevant design abstractions and integrate them within a common framework for joint optimization across the three levels. Jointly exploring all parameters, however, results in a large number of possible combinations, making exhaustive evaluation impractical. The case studies in this thesis therefore focus on selected subsets of the design parameters and characteristics summarized in Table 5.1.

Table 5.1: Design parameters and characteristics considered in this thesis.

Measurement	Algorithm	Hardware
Sensor type – Camera, antenna, microphone Measurement characteristics – Range and bearing measurements, propagation conditions, input sparsity Calibration – Intrinsic and extrinsic parameters Time synchronization – NTP, PTP, hardware triggering, common clock	Geometry-based methods – Triangulation, trilateration, multilateration Data-driven methods – Model type (FCNN, CNN, attention-based, RNN, Mamba), model structure and size, activation and normalization, temporal processing, adaptive model selection	Hardware platform – CPU, GPU, FPGA, ASIC Processing architecture – Systolic arrays, parallel vector engines Dataflow – OS, IS, WS, RS Memory organization – Memory hierarchy, memory access patterns Quantization – Number format, bit width, scaling

The selection of these parameters can also be guided by characteristics observed in the measurements and by bottlenecks identified in the algorithm or hardware implementation. For example, radio-channel measurements may contain spatial and delay-domain structure, while image and audio measurements contain different forms of spatial and temporal information. Propagation conditions and beam-domain sparsity can also motivate changes to the algorithm or the amount of computation performed. At the algorithm-

hardware boundary, the selected model must be mapped to the target processing platform [74,75]. Operations such as matrix multiplication, normalization, and activation functions have different implementation costs, while the movement of model parameters and intermediate values can become a major bottleneck.

5.2 PERFORMANCE METRICS

The accuracy of a localization system is commonly evaluated by comparing the estimated position with a reference ground truth position. For each position estimate, the localization error is computed from the distance between the estimated and ground truth positions; a smaller error corresponds to a more accurate estimate. When several position estimates are evaluated, the resulting error values are used to compute metrics and error distributions such as the Mean Euclidean Error (ME), median error, percentile error, variance of the error, and the Cumulative Distribution Function (CDF) of the positioning error.

In this thesis, the 2D localization error is computed as the Euclidean distance between the estimated and ground truth positions, for each sample. For a two-dimensional position estimate, this error is given by

$$e_i = \|\hat{\mathbf{p}}_i - \mathbf{p}_i\|_2 = \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}, \quad (5.1)$$

where $\hat{\mathbf{p}}_i$ and $\mathbf{p}_i$ denote the estimated and ground truth positions for sample i , respectively.

In addition to accuracy, latency and throughput are important metrics for practical localization systems. In hardware implementations, latency refers to the time between providing a localization measurement to the processing hardware and receiving the corresponding position estimate. Throughput is the number of position estimates produced by the processing hardware per unit time.

For pipelined designs, new inputs can be accepted before previous inputs have completed. The time between two consecutive inputs accepted by the hardware is referred to as the initiation interval. Once the pipeline is fully occupied, consecutive outputs are produced at the same interval, so a shorter initiation interval corresponds to a higher throughput.

The required throughput depends on the rate at which localization measurements become available. The sensor sampling rate sets the measurement arrival rate for a single sensor stream, while multiple users or targets can further increase the number of measurements that must be handled within the same time interval. For real-time processing of a single measurement

stream, the initiation interval must not exceed the sensor sampling period. The required accuracy and latency depend on the localization use case, as presented in Chapter 1.

Energy consumption is another important performance metric, especially for battery-powered or embedded devices. A localization algorithm that requires frequent memory accesses, large matrix operations, or high computational precision can increase both power dissipation and energy use. Higher power dissipation also increases thermal-management requirements, while energy per inference and average power determine the operating time of battery-powered devices. These metrics are therefore often considered together with accuracy and latency when evaluating practical implementations.

Hardware constraints also affect the choice of localization algorithm. Available memory limits the size of the model parameters and intermediate data, while the number of processing elements, memory bandwidth, and supported arithmetic precision determine how efficiently the algorithm can be executed. Other practical constraints include chip area, hardware cost, and scalability.

5.3 CASE STUDIES

The case studies below illustrate how observations at the measurement, algorithm, and hardware levels lead to co-design decisions throughout this thesis. Case Study 1 examines how different propagation environments can motivate adaptive model selection. Case Study 2 analyzes how structural and temporal relationships within the input features—along with the hardware costs of model implementation—dictate model selection, demonstrating that the architecture achieving the lowest localization error is not necessarily optimal when balanced against throughput, latency, and memory constraints. Case Study 3 exploits sparsity in radio-channel representations to skip unnecessary computations and reduce execution time. Case Study 4 considers replacing activation functions with alternatives that require less computation in hardware while limiting the resulting degradation in localization accuracy.

Case Study 1 - Effect of Measurement Environment on Localization Algorithms and Hardware

Localization systems may operate under measurement conditions that vary across environments. Radio measurements are affected by, for instance, LoS, NLoS, and mixed propagation conditions, while vision-based localization can be affected by lighting and occlusions, and audio-based localization by reflections and background noise. Changes in the surrounding geometry, sensor placement, and propagation environment can therefore produce different input distributions across buildings, routes, or cities. A single model trained to

cover all such conditions must learn a broad mapping from the measurements to position and may not provide the same localization performance in every environment.

Adaptive localization algorithms address this problem by changing the processing strategy based on the current measurements, operating environment, or computational constraints. At the measurement level, the system may select or weight sensors according to their reliability. For example, visual measurements may be useful when image features are clearly visible, while radio measurements may provide information when the target is not directly visible. At the input-representation level, the system may process selected parts of the measurement, such as antennas, beams, delay bins, or time samples in radio-based localization. This can reduce the computational load by focusing the processing on input components that are more informative for the current position estimate.

At the algorithm level, the localization problem can be divided into smaller sub-problems by using specialized models for different environments, sensor conditions, or propagation scenarios. A single generalized model trained to cover all conditions must learn a broader mapping from measurements to position, even though some features may only be useful in specific environments. Specialized models reduce the complexity of the learning problem by limiting each model to a smaller and more homogeneous subset of the data. A lightweight routing module can then select the model that best matches the current input. This idea was further explored in **Paper IV**, where environment-dependent model selection was used for adaptive attention-based 5G radio localization.

While originally proposed in 1991 to isolate independent subtasks within a single dataset [76], this foundational concept has expanded into a macro-level framework for spatial and temporal adaptation [77]. This idea is related to transfer learning and domain adaptation in localization, where models are adapted across time, space, or devices, as well as to conditional computation and Mixture-of-Experts (MoE) architectures, where a gating mechanism selects the most relevant expert for each input.

Adaptive processing is also relevant for hardware implementation because real-time localization systems must satisfy latency, memory, and energy constraints, explored in **Paper V**. Applying the same computation to every input may result in redundant processing and inefficient utilization of computational and memory resources. This cost can be reduced by skipping computations, activating only the required model parameters, or choosing a dataflow that matches the selected computation. The measurement determines which parts of the input are reliable or informative, the algorithm determines how this information is used for position estimation, and the

hardware determines whether the selected computation can be executed within the available resource budget.

Case Study 2 - DL Model Selection Based on Input Characteristics

The selection of a DL model depends on whether each position is estimated from a single measurement snapshot or a sequence of consecutive observations, as well as on the spatial, temporal, or other structural relationships among the input elements. For a single measurement snapshot represented as a feature vector, such as RSS values from several anchors or estimated distances to known reference points, a feedforward model such as an FCNN may be sufficient. For structured input representations, the position or index of each input element can carry additional information. Examples include neighboring pixels in an image, antennas in an array, and beam indices and delay bins in a radio-channel representation. Although these inputs can be flattened and processed by an FCNN, doing so does not directly exploit their spatial arrangement or ordering. CNNs can capture local patterns, while attention-based models can capture relationships between more distant input elements.

When consecutive measurement snapshots are available, temporal relationships can be used to capture motion or changes in the propagation environment. The measurement rate, required update rate, and selected window size determine how much historical information can be incorporated into each position estimate. RNNs and state-space models such as Mamba [52] process ordered measurements by updating an internal state, while attention-based models can capture dependencies among measurements within a temporal window. Figure 5.1 summarizes how different model types exploit relationships among input elements. CNNs extract features from local input regions, Transformers relate input elements through attention, and RNNs capture sequential dependencies through a recurrent state.

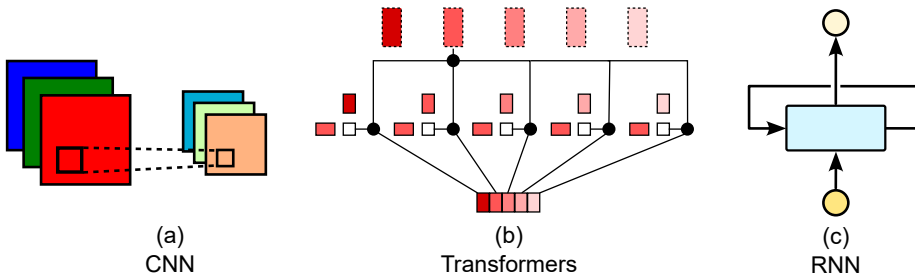


Figure 5.1: Simplified illustration of feature extraction in (a) CNNs, using local neighborhoods, (b) Transformers, using relationships between input elements through attention, and (c) RNNs, using sequential dependencies.

Model selection must also consider localization accuracy together with throughput, latency, energy consumption, and memory requirements. Models differ substantially in computational complexity and memory footprint. An FCNN has a fixed-size input and output, and its parameter count and computational cost scale with these dimensions rather than with sequence length. Convolutional layers are effective for local pattern extraction, but capturing long-range dependencies may require deeper stacks or larger kernels. RNNs process measurements sequentially, updating a fixed-size internal state at each step; this keeps memory and per-step computation low, but the sequential dependency prevents parallelization across time steps and can lead to vanishing or exploding gradients over long sequences [78]. Self-attention can directly relate input elements across the sequence, but its computational and memory complexity scales quadratically with sequence length [48].

The studies presented in this thesis illustrate several of these model-selection choices. In **Papers II and III**, an FCNN learns a direct mapping from the radio input features to the target position. In **Paper IV**, an attention-based model is instead used to exploit relationships between beam components within a single measurement snapshot. The beam indices form the sequence dimension of the attention model and therefore represent structure within one snapshot rather than a temporal sequence of consecutive measurements. **Paper V** further investigates how the computational structure of this model affects the processing architecture and dataflow.

The models in **Papers IV and V** do not exploit temporal relationships between consecutive measurement snapshots. Extending attention-based or state-space models to temporal measurement sequences therefore remains a possible direction for future work. For instance, in [79], an FCNN is compared with a Mamba-based model using the dataset introduced in **Paper I**. The Mamba model achieves comparable localization accuracy with a substantially smaller model size, making it well suited for memory-constrained edge hardware.

Case Study 3 - Exploiting Radio-Channel Sparsity for Reduced Execution Time

Sparsity refers to the presence of zero-valued elements in data or model parameters that can allow the corresponding storage, memory accesses, or computations to be omitted. In DL models, sparsity may arise within the model, for example through zero-valued activations or pruned weights, or from the input data. From a hardware perspective, the organization of the zero-valued elements is important. Fine-grained sparsity occurs at the level of individual elements, whereas coarse-grained or structured sparsity occurs when complete rows, columns, channels, or other groups can be omitted.

The sparsity granularity therefore affects how efficiently the corresponding arithmetic operations and memory accesses can be skipped.

Fine-grained sparsity, such as individual zero-valued activations produced by ReLU, can reduce the number of arithmetic operations. However, exploiting isolated zeros may require additional detection and indexing logic and can result in irregular memory accesses. The location of the zero-valued elements may also change between inputs, requiring the hardware to identify which elements can be skipped during execution. Sparsity-aware execution is therefore beneficial only when the reduction in computation and data movement outweighs the overhead of identifying and managing the sparse pattern [80].

Coarse-grained sparsity is generally easier to exploit because larger parts of the computation can be skipped using simpler control logic. In radio-channel measurements, such sparsity can arise from the propagation environment and the selected channel representation. For a beam-delay representation, the received energy may be concentrated in a limited number of beams and delay components, while other parts of the representation contain considerably less energy. These values are not necessarily zero, but components below a selected energy threshold can be treated as insignificant for the subsequent computation. Grouping the sparsity at the row level allows complete beam rows to be omitted rather than checking individual channel values.

The selected representation determines whether the sparse structure appears in a form that can be exploited efficiently, while the algorithm determines which components can be omitted without significantly degrading the localization result. The hardware must then support skipping the corresponding memory accesses and arithmetic operations with sufficiently low control overhead. The choice of sparsity granularity therefore depends not only on the algorithm, but also on how the data are represented, stored, and processed in hardware.

In **Paper V**, row-wise skipping is implemented by identifying low-energy beam components and omitting the corresponding computations. The selected threshold determines the trade-off between the number of skipped rows, the reduction in execution time, and the resulting localization accuracy. A higher threshold can increase the amount of skipped computation, but may also remove channel information that is useful for localization.

Case Study 4 - Activation Function Selection for Reduced Computational Complexity

Nonlinear functions provide another example of how algorithm selection can be influenced by hardware implementation cost. The ReLU function is relatively simple to implement because it only requires a comparison and

selection operation. Piecewise-linear functions can provide greater flexibility while retaining an implementation based mainly on comparisons, additions, and multiplications or shifts. More complex functions, such as sigmoid and softmax, may require exponentiation, division, or LUTs. These functions may also require higher numerical precision, especially for intermediate values used in exponentiation and normalization, which can make low-bit implementations more difficult.

The computational complexity depends not only on the nonlinear function itself, but also on whether it is evaluated independently for each element or jointly across several elements. Sigmoid operates independently on each input element and can be approximated using a LUT. Each value can therefore be processed as it becomes available, without waiting for the remaining elements in the row. Softmax, in contrast, requires exponentiation of all elements, accumulation of the exponential values, and normalization by their sum. The normalized output for an element cannot be produced until the sum over the complete row is known.

This dependency has direct consequences for the hardware implementation. The intermediate values for a softmax row must typically be buffered while the sum is accumulated and then accessed again during normalization. This requires additional storage and introduces latency before the normalized outputs can be produced. Softmax also requires reduction and reciprocal operations, whereas sigmoid does not require communication or normalization across the elements of a row. Replacing softmax with sigmoid in DL models can therefore reduce the required intermediate storage and execution time, provided that the resulting accuracy remains acceptable, as explored in **Paper V**.

Activation functions can also be modified or approximated to reduce computational complexity. Piecewise-linear approximations and alternative functions such as RSigELU illustrate how the shape of a nonlinear function can be adjusted while retaining selected properties of existing activation functions [81]. The function is therefore selected by considering both localization accuracy and the arithmetic, memory, and control requirements of its hardware implementation.

The case studies presented in this section show how selected design choices across sensing and measurement, algorithms, and hardware implementation can be considered together with the corresponding performance metrics. Depending on the application, these metrics may include localization accuracy, latency, throughput, energy consumption, and hardware cost. A complete co-design methodology would require a framework with appropriate models at the different abstraction levels and an optimization procedure for exploring

the resulting design space with respect to these metrics. Developing and validating such models across the measurement, algorithm, and hardware domains is a substantial task that extends beyond the scope of a single PhD thesis and would benefit from joint efforts across the corresponding research areas. The approach taken in this thesis is therefore based on prior design experience together with local analysis and optimization, including models of resource and energy consumption and analyses of latency and throughput, to guide the design choices investigated in the individual case studies. Extending these methods toward systematic design space exploration and joint optimization across all three domains represents an important direction for future work.

Conclusion and Future Work

HE presented work demonstrates that accurate and efficient localization can be achieved by jointly considering sensing modalities, algorithm design, and hardware implementation. By exploiting modality-specific characteristics and structured signal representations, the proposed approaches enable robust localization under practical constraints on latency, power, and computational resources.

The contributions of this thesis are presented in two parts. The first part (Chapters 1-6) provides a system-level overview of localization systems, measurements, algorithms, and hardware implementation. Chapter 1 introduced the motivation and scope of the work, outlining the challenges of localization and the need to consider measurements, algorithms, and hardware jointly. Chapters 2 to 4 then presented the main concepts related to localization systems and measurements, localization algorithms, and hardware implementation, respectively. Chapter 5 brought these domains together through a co-design perspective and showed, through selected case studies, how measurement properties, algorithmic choices, and hardware constraints can be considered jointly to support accurate and low-latency localization. Finally, this current chapter summarizes the main findings of the thesis, discusses the limitations of the presented work, and identifies directions for future research.

The second part consists of the collection of papers, which present the detailed research contributions. A key foundation of this work is the development of representative measurement data collected in realistic environments.

In **Paper I**, a multimodal public dataset, referred to as the LuViRA dataset, is developed. The dataset includes synchronized measurements from cameras, microphones, a massive MIMO testbed, a UE, and an IMU, together with ground truth labels, enabling consistent evaluation across sensing modalities.

In **Paper II**, the LuViRA dataset is analyzed and validated through a comparison of vision-, radio-, and audio-based localization methods. The study highlights the strengths and limitations of the different sensors and demonstrates their complementary nature. Localization using radio signals and machine learning is further investigated in subsequent works.

In **Paper III**, high-precision indoor localization is studied using channel measurements from a 5G massive MIMO testbed. The paper investigates how radio-channel measurements can be used with machine-learning models to estimate the user position.

In **Paper IV**, the radio-based localization problem is extended to a more challenging outdoor setting with multiple spatially separated trajectories. In this case, the propagation conditions vary between the different measurement areas, including LoS and NLoS conditions. To address this, an adaptive attention-based model is proposed for outdoor 5G localization. The model selects an environment-specific localization model based on the measured signal, improving robustness under different propagation conditions.

The efficient implementation of the adaptive Transformer-based localization model is addressed in **Paper V**. A dedicated hardware accelerator is designed to reduce inference latency, power consumption, and computational cost. The architecture exploits the inherent structure of the radio signals and the localization model, using fixed-point arithmetic, mixed dataflow, and sparsity-aware computation.

Together, these contributions address different parts of efficient localization system design, from multimodal data acquisition and sensor comparison to radio-based machine learning, adaptive modeling, and hardware implementation. The resulting design space includes parameters and characteristics that can be varied at the system, algorithm, and hardware levels, together with performance metrics that guide their selection. The co-design methodology, design parameters, and performance metrics considered in this thesis are illustrated in Fig. 6.1. Table 6.1 summarizes the levels considered in each paper.

The comparison highlights the importance of treating localization system design as a co-design problem, since decisions across the measurement, algorithm, and hardware domains affect one another and must be balanced against the target performance and implementation constraints. The results show that jointly considering these domains can support accurate and low-latency localization.

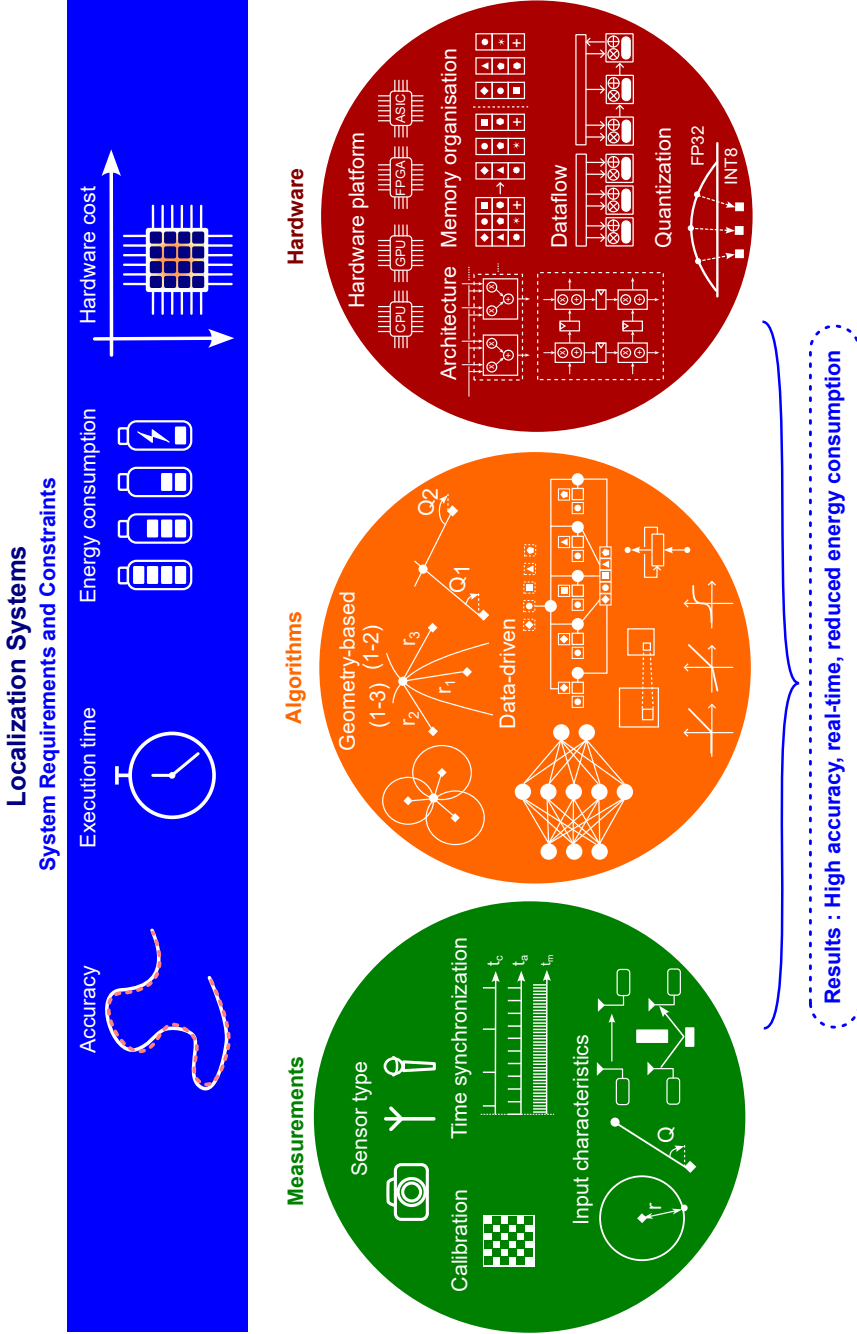


Figure 6.1: Design parameters and performance metrics considered in the co-design of localization systems across the system, algorithm, and hardware implementation.

Table 6.1: Scope of each paper in terms of the measurement, algorithm, and hardware levels.

Paper	Measurement	Algorithm	Hardware
I	✓	×	×
II	✓	✓	×
III	✓	✓	×
IV	✓	✓	✓
V	✓	✓	✓

6.1 FUTURE WORK

Several directions remain open for future research, building upon the system-level approach adopted in this thesis. These directions span sensing, algorithm design, and hardware implementation, with the aim of further improving robustness, accuracy, and efficiency of localization systems. One possible future scenario, a smart factory with fully autonomous robots and drones, is shown in Fig. 6.2. One of the most critical challenges in such a scenario is accurate and real-time localization. Drones can be localized using microphones [82].

A natural extension of this work is the joint use of multiple sensing modalities. Some of the common sensors in the environment are cameras, radio receivers and transmitters and microphones and speakers. While individual sensors such as radio, vision, and audio provide complementary information, their integration remains a challenging problem due to differences in sampling rates, coordinate systems, noise characteristics, and failure modes.

Future work may explore sensor fusion strategies that combine these modalities at different levels, including early fusion at the feature level, intermediate fusion within learned representations, and late fusion at the decision level. In particular, learning-based fusion approaches that adaptively weight sensor contributions based on environmental conditions could significantly improve robustness in scenarios where one or more modalities degrade, such as low-light conditions for vision. Some works that explore fusion of these sensors are: [83–85].

Beyond the localization scenarios considered in this thesis, the proposed methods can also be extended to distributed localization systems. Positioning with distributed massive MIMO has been studied using neural networks with probabilistic output layers, where spatially separated surfaces provide complementary channel observations for estimating the user position [86]. This direction is particularly relevant for future wireless systems, where

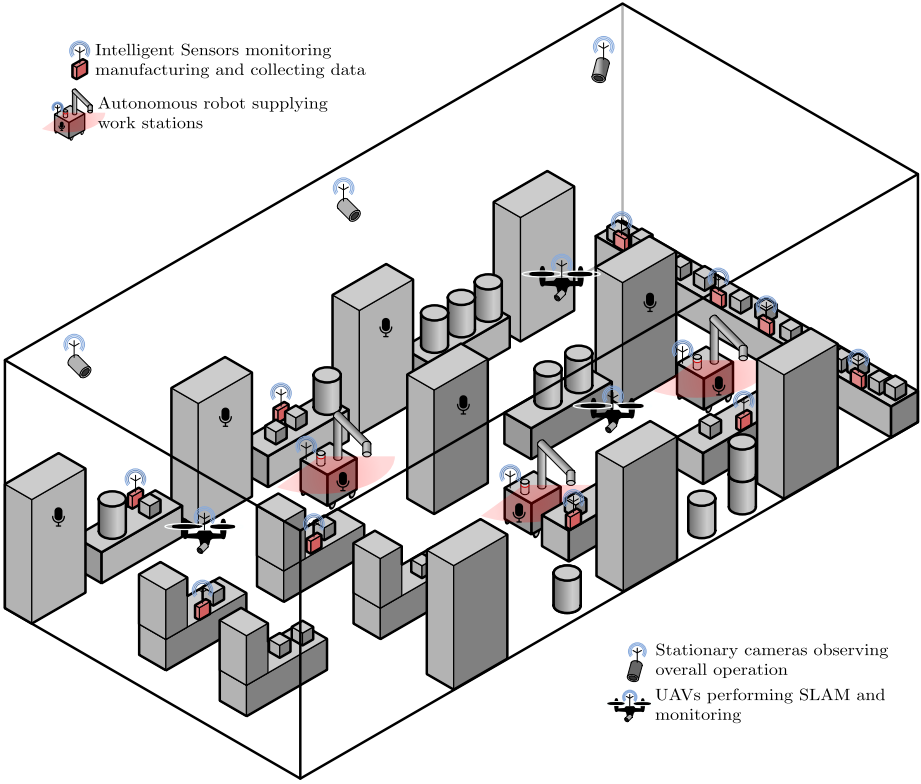


Figure 6.2: Illustration of a future fully autonomous smart factory with cameras, antennas, and microphones.

sensing and communication infrastructure may be distributed across several access points, surfaces, or panels. In such systems, learning-based localization algorithms must not only process high-dimensional channel representations, but also combine information from multiple spatially separated nodes while accounting for uncertainty in the position estimate.

The same high-dimensional channel representations used for localization also contain rich spatial information that can be exploited in related wireless applications, such as beamforming and channel estimation [87,88]. Learning-based models could, for instance, be used to predict suitable beam directions or reconstruct channel state information in scenarios with limited measurements. Exploring the interplay between localization and communication tasks may therefore enable joint optimization frameworks, where positioning information is directly used to improve system-level communication performance.

Another important direction is the integration of additional sensing modalities. IMUs, which provide high-rate motion information, can complement other systems by improving temporal resolution and enabling dead-reckoning in the absence of external references. Similarly, distributed MIMO systems, where multiple spatially separated base stations cooperate, offer increased spatial diversity and improved coverage. Combining such systems with learning-based localization models may enable more accurate and scalable solutions, particularly in large or complex environments.

From a hardware perspective, further exploration of quantization techniques remains highly relevant. While this thesis demonstrates the feasibility of low-precision implementations, additional gains may be achieved through QAT, mixed-precision architectures, and dynamic precision scaling [89]. In particular, adapting quantization levels based on input characteristics could improve the trade-off between accuracy and efficiency. Moreover, incorporating quantization into the training process more tightly, for example through hardware-in-the-loop approaches, may lead to models that are optimized for deployment.

Finally, the concept of adaptive models can be extended further. While this work focuses on environment-dependent adaptation, future research may investigate more fine-grained adaptive systems that dynamically adjust model complexity, attention mechanisms, or hardware dataflow strategies based on the input characteristics, propagation conditions, or hardware constraints. This also includes exploring architectures such as MoE models and retraining strategies that incorporate newly collected measurement data after deployment. Such approaches are particularly relevant for real-world localization systems operating in dynamic and unpredictable environments.

Overall, advancing localization systems will require continued integration across sensing, algorithms, and hardware. The directions outlined above provide a pathway toward more robust, scalable, and energy-efficient solutions for future applications.

Bibliography

- [1] I. Yaman, G. Tian, M. Larsson, P. Persson, M. Sandra, A. Dürr, E. Tegler, N. Challa, H. Garde, F. Tufvesson *et al.*, “The LuViRA Dataset: Synchronized Vision, Radio, and Audio Sensors for Indoor Localization,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. doi: 10.1109/ICRA57147.2024.10610237 pp. 11 920–11 926.
- [2] I. Yaman, G. Tian, E. Tegler, J. Gulin, N. Challa, F. Tufvesson, O. Edfors, K. Åström, S. Malkowsky, and L. Liu, “LuViRA Dataset Validation and Discussion: Comparing Vision, Radio, and Audio Sensors for Indoor Localization,” *IEEE Journal of Indoor and Seamless Positioning and Navigation*, vol. 2, pp. 240–250, 2024. doi: 10.1109/JISPIN.2024.3429110
- [3] G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu, and F. Tufvesson, “High-Precision Machine-Learning Based Indoor Localization with Massive MIMO System,” in *ICC 2023 - IEEE International Conference on Communications*, 2023. doi: 10.1109/ICC45041.2023.10278664 pp. 3690–3695.
- [4] I. Yaman, G. Tian, D. Pjanić, F. Tufvesson, O. Edfors, Z. Zhang, and L. Liu, “Adaptive Attention-Based Model for 5G Radio-Based Outdoor Localization,” in *2025 59th Asilomar Conference on Signals, Systems, and Computers*, 2025. doi: 10.1109/IEEECONF67917.2025.11443733 pp. 192–197.
- [5] I. Yaman, S. Cheng, O. Edfors, and L. Liu, “Implementation of an Adaptive Transformer Accelerator for Accurate Outdoor Localization with Massive MIMO,” 2026. [Online]. Available: <https://arxiv.org/abs/2605.13507>
- [6] X. Li, “Parametric Radio Channel Estimation and Robust Localization,” Ph.D. dissertation, Lund University. ISBN 978-91-8039-279-2 May 2022.

- [7] B. Hofmann-Wellenhof, K. Legat, and M. Wieser, *Navigation: Principles of Positioning and Guidance*. Springer Vienna, 2011. ISBN 9783709160787
- [8] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, “GNSS Position Integrity in Urban Environments: A Review of Literature,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2762–2778, 2018. doi: 10.1109/TITS.2017.2766768
- [9] J. Gutierrez, R. Gilabert, E. T. Dill, G. Hernandez, D. Kaeli, and P. Closas, “Multipath Mitigation via Clustering for Position Estimation Refinement in Urban Environments,” *ION Pacific PNT*, pp. 556–568, 2024. doi: 10.33012/2024.19605
- [10] L. Wang, P. D. Groves, and M. K. Ziebart, “Multi-Constellation GNSS Performance Evaluation for Urban Canyons Using Large Virtual Reality City Models,” *Journal of Navigation*, vol. 65, no. 3, p. 459–476, 2012. doi: 10.1017/S0373463312000082
- [11] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A Survey of Autonomous Driving: Common Practices and Emerging Technologies,” *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020. doi: 10.1109/ACCESS.2020.2983149
- [12] A. Arun, S. Saruwatari, S. Shah, and D. Bharadia, “XRLoc: Accurate UWB Localization to Realize XR Deployments,” in *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’23. New York, NY, USA: Association for Computing Machinery, 2023. doi: 10.1145/3625687.3625810. ISBN 9798400704147 p. 459–473. [Online]. Available: <https://doi.org/10.1145/3625687.3625810>
- [13] M. Attari, “Application Specific Instruction-set Processors for Massive MIMO Systems,” Ph.D. dissertation, Lund University. ISBN 978-91-8104-247-4 Oct. 2024.
- [14] REINDEER. (2021) REINDEER deliverable D1.1: Use case-driven specifications and technical re-quirements and initial channel model. [Online]. Available: <https://doi.org/10.5281/zenodo.5561844>
- [15] F. Gustafsson, *Statistical sensor fusion*, 1st ed. Studentlitteratur, 2010. ISBN 978-91-44-05489-6. [Online]. Available: <https://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-74627>
- [16] B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *J. Opt. Soc. Am. A*, vol. 4, no. 4, pp. 629–642, Apr 1987. doi: 10.1364/JOSAA.4.000629. [Online]. Available: <https://opg.optica.org/josaa/abstract.cfm?URI=josaa-4-4-629>

- [17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012. doi: 10.1109/IROS.2012.6385773 pp. 573–580.
- [18] X. Gao and T. Zhang, *Introduction to Visual SLAM: From Theory to Practice*. Springer Singapore, Jan 2021. ISBN 978-981-16-4938-7
- [19] A. F. Molisch, *Wireless Communications*, 2nd ed. Chichester, UK: Wiley-IEEE Press, 2011. ISBN 978-0-470-74187-0
- [20] M. Zhu, J. Vieira, Y. Kuang, K. Åström, A. F. Molisch, and F. Tufvesson, "Tracking and positioning using phase information from estimated multi-path components," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015. doi: 10.1109/ICCW.2015.7247265 pp. 712–717.
- [21] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017. doi: 10.1109/PIMRC.2017.8292280 pp. 1–6.
- [22] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. Wong, F. Tufvesson, V. Öwall, and O. Edfors, "The World's First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation," *IEEE Access*, vol. 5, pp. 9073–9088, 2017. doi: 10.1109/ACCESS.2017.2705561
- [23] G. Tian, "Navigating the Future — Machine Learning for Wireless Sensing and Localization," Ph.D. dissertation, Lund University. ISBN 978-91-8104-119-4 Oct. 2024.
- [24] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, ser. Springer Topics in Signal Processing. Berlin, Heidelberg: Springer, 2008, vol. 1. ISBN 978-3-540-78612-2
- [25] C. Evers, H. W. Löllmann, H. Mellmann, A. Schmidt, H. Barfuss, P. A. Naylor, and W. Kellermann, "The locata challenge: Acoustic source localization and tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1620–1643, 2020. doi: 10.1109/TASLP.2020.2990485
- [26] E. Tegler, "From Sound to Structure: Robust Localization of Sources, Sensors, and Surroundings," Ph.D. dissertation, Lund University. ISBN 978-91-8104-763-9 Dec. 2025.

- [27] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976. doi: 10.1109/TASSP.1976.1162830
- [28] H. Sallouha, S. Saleh, S. De Bast, Z. Cui, S. Pollin, and H. Wymeersch, "On the Ground and in the Sky: A Tutorial on Radio Localization in Ground-Air-Space Networks," *IEEE Communications Surveys and Tutorials*, vol. 27, no. 1, pp. 218–258, 2025. doi: 10.1109/COMST.2024.3417336
- [29] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013. doi: 10.1109/IROS.2013.6696514 pp. 1280–1286.
- [30] Q. Zhang, A. Sakhnini, R. Beerten, H. Xiong, Z. Cui, Y. Miao, and S. Pollin, "Robust Localization in OFDM-Based Massive MIMO through Phase Offset Calibration," in *2026 IEEE 6th International Symposium on Joint Communications and Sensing (JC&S)*, 2026. doi: 10.1109/JCS69321.2026.11366003 pp. 1–6.
- [31] R. Pöhlmann, S. Zhang, E. Staudinger, A. Dammann, and P. A. Hoeher, "Simultaneous Localization and Antenna Calibration," in *2022 16th European Conference on Antennas and Propagation (EuCAP)*, 2022. doi: 10.23919/EuCAP53622.2022.9769393 pp. 1–5.
- [32] M. Larsson, "Localization using Distance Geometry: Minimal Solvers and Robust Methods for Sensor Network Self-Calibration," Ph.D. dissertation, Lund University. ISBN 978-91-8039-377-5 Oct. 2022.
- [33] M. Larsson, G. Flood, M. Oskarsson, and K. Åström, "Fast and robust stratified self-calibration using time-difference-of-arrival measurements," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021. doi: 10.1109/ICASSP39728.2021.9414309 pp. 4640–4644.
- [34] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, Jun. 2010. [Online]. Available: <https://www.rfc-editor.org/info/rfc5905>
- [35] Y. Li-Baboud, "Using NTP: Introduction and Recommended Practices," ISMI Technology Transfer, Feb. 2006.
- [36] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008)*, pp. 1–499, 2020. doi: 10.1109/IEEESTD.2020.9120376

-
- [37] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. doi: 10.1115/1.3662552. [Online]. Available: <https://doi.org/10.1115/1.3662552>
 - [38] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002. doi: 10.1109/78.978374
 - [39] R. Whiton, "Dude, Where's My Car? Cellular Navigation for Autonomous Driving," Ph.D. dissertation, Lund University. ISBN 978-91-8039-950-0 Mar. 2024.
 - [40] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, Jun. 1981. doi: 10.1145/358669.358692. [Online]. Available: <https://doi.org/10.1145/358669.358692>
 - [41] K. Åström, M. Larsson, G. Flood, and M. Oskarsson, "Extension of Time-Difference-of-Arrival Self Calibration Solutions Using Robust Multilateration," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021. doi: 10.23919/EUSIPCO54536.2021.9616051 pp. 870–874.
 - [42] C. Campos, R. Elvira, J. J. Gomez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021. doi: 10.1109/TRO.2021.3075644
 - [43] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 16 558–16 569. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/89fcd07f20b6785b92134bd6c1d0fa42-Paper.pdf
 - [44] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. doi: 10.1109/ICCV.2015.336 pp. 2938–2946.
 - [45] S. De Bast, A. P. Guevara, and S. Pollin, "CSI-based Positioning in Massive MIMO systems using Convolutional Neural Networks," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020. doi: 10.1109/VTC2020-Spring48590.2020.9129126 pp. 1–5.

- [46] A. Berg, J. Gulin, M. O'Connor, C. Zhou, K. Åström, and M. Oskarsson, "wav2pos: Sound Source Localization using Masked Autoencoders," in *2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2024. doi: 10.1109/IPIN62893.2024.10786105 pp. 1–8.
- [47] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017. doi: 10.1109/JPROC.2017.2761740
- [48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017. ISBN 9781510860964 pp. 6000–6010.
- [49] J. L. Elman, "Finding Structure in Time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990. doi: https://doi.org/10.1207/s15516709cog1402_1. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1
- [50] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014. doi: 10.3115/v1/D14-1179 pp. 1724–1734. [Online]. Available: <https://aclanthology.org/D14-1179/>
- [51] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [52] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *First Conference on Language Modeling*, 2024. [Online]. Available: <https://openreview.net/forum?id=tEYskw1VY2>
- [53] J. L. Hennessy and D. A. Patterson, *Computer Architecture, Sixth Edition: A Quantitative Approach*, 6th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2017. ISBN 0128119055
- [54] J. L. Hennessy and D. A. Patterson, "A new golden age for computer architecture," *Commun. ACM*, vol. 62, no. 2, pp. 48–60, jan 2019. doi: 10.1145/3282307. [Online]. Available: <https://doi.org/10.1145/3282307>

- [55] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014. doi: 10.1109/ISSCC.2014.6757323 pp. 10–14.
- [56] Y.-H. Chen, J. Emer, and V. Sze, "Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016. doi: 10.1109/ISCA.2016.40 pp. 367–379.
- [57] H. T. Kung, "Why Systolic Architectures?" *Computer*, vol. 15, no. 1, p. 37–46, Jan. 1982. doi: 10.1109/MC.1982.1653825. [Online]. Available: <https://doi.org/10.1109/MC.1982.1653825>
- [58] N. P. Jouppi, D. Hyun Yoon, M. Ashcraft, M. Gottscho, T. B. Jablin, G. Kurian, J. Laudon, S. Li, P. Ma, X. Ma *et al.*, "Ten Lessons From Three Generations Shaped Google's TPUnv4i : Industrial Product," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, 2021. doi: 10.1109/ISCA52012.2021.00010 pp. 1–14.
- [59] H. Genc, S. Kim, A. Amid, A. Haj-Ali, V. Iyer, P. Prakash, J. Zhao, D. Grubb, H. Liew, H. Mao *et al.*, "Gemmini: Enabling Systematic Deep-Learning Architecture Evaluation via Full-Stack Integration," in *Proceedings of the 58th Annual ACM/IEEE Design Automation Conference*, ser. DAC '21. IEEE Press, 2021. doi: 10.1109/DAC18074.2021.9586216. ISBN 9781665432740 p. 769–774. [Online]. Available: <https://doi.org/10.1109/DAC18074.2021.9586216>
- [60] M. Attari, L. Ferreira, L. Liu, and S. Malkowsky, "An Application Specific Vector Processor for Efficient Massive MIMO Processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 9, pp. 3804–3815, 2022. doi: 10.1109/TCSI.2022.3182483
- [61] Y. Liu, L. Liu, and V. Öwall, "Architecture Design of a Memory Subsystem for Massive MIMO Baseband Processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2976–2980, 2017. doi: 10.1109/TVLSI.2017.2732062
- [62] Y. Liu, L. Liu, O. Edfors, and V. Öwall, "An Area-Efficient On-Chip Memory System for Massive MIMO Using Channel Data Compression," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 417–427, 2019. doi: 10.1109/TCSI.2018.2859361
- [63] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A White Paper on Neural Network Quantization," 2021. [Online]. Available: <https://arxiv.org/abs/2106.08295>

- [64] "The bfloat16 numerical format." [Online]. Available: <https://cloud.google.com/tpu/docs/bfloat16>
- [65] P. Micikevicius, D. Stolic, N. Burgess, M. Cornea, P. Dubey, R. Grisenthwaite, S. Ha, A. Heinecke, P. Judd, J. Kamalu *et al.*, "FP8 Formats for Deep Learning," 2022. [Online]. Available: <https://arxiv.org/abs/2209.05433>
- [66] K. Guo, S. Zeng, J. Yu, Y. Wang, and H. Yang, "[DL] A Survey of FPGA-based Neural Network Inference Accelerators," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 12, no. 1, Mar. 2019. doi: 10.1145/3289185
- [67] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," 2017. [Online]. Available: <https://arxiv.org/abs/1712.05877>
- [68] Y. Bengio, N. Léonard, and A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," 2013. [Online]. Available: <https://arxiv.org/abs/1308.3432>
- [69] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, "Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation," 2020. [Online]. Available: <https://arxiv.org/abs/2004.09602>
- [70] A. J. Davison, "FutureMapping: The Computational Structure of Spatial AI Systems," 2018. [Online]. Available: <https://arxiv.org/abs/1803.11288>
- [71] S. Saeedi, B. Bodin, H. Wagstaff, A. Nisbet, L. Nardi, J. Mawer, N. Melot, O. Palomar, E. Vespa, T. Spink *et al.*, "Navigating the Landscape for Real-Time Localization and Mapping for Robotics and Virtual and Augmented Reality," *Proceedings of the IEEE*, vol. 106, no. 11, pp. 2020–2039, 2018. doi: 10.1109/JPROC.2018.2856739
- [72] S. Lyu, Y. She, D. Duan, T. Ni, Y. H. Chan, C. Luo, R. C. C. Cheung, and W. Xu, "SwiftChannel: Algorithm-Hardware Co-Design for Deep Learning-Based 5G Channel Estimation," *IEEE Transactions on Mobile Computing*, vol. 25, no. 6, pp. 8268–8284, 2026. doi: 10.1109/TMC.2025.3647644
- [73] J. Gante, L. Sousa, and G. Falcao, "Dethroning GPS: Low-Power Accurate 5G Positioning Systems Using Machine Learning," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 2, pp. 240–252, 2020. doi: 10.1109/JETCAS.2020.2991024

- [74] H. Peng, S. Huang, S. Chen, B. Li, T. Geng, A. Li, W. Jiang, W. Wen, J. Bi, H. Liu *et al.*, "A length adaptive algorithm-hardware co-design of transformer on FPGA through sparse attention and dynamic pipelining," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, ser. DAC '22. New York, NY, USA: Association for Computing Machinery, 2022. doi: 10.1145/3489517.3530585. ISBN 9781450391429 p. 1135–1140. [Online]. Available: <https://doi.org/10.1145/3489517.3530585>
- [75] L. Lu, Y. Jin, H. Bi, Z. Luo, P. Li, T. Wang, and Y. Liang, "Sanger: A Co-Design Framework for Enabling Sparse Attention using Reconfigurable Architecture," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '21. New York, NY, USA: Association for Computing Machinery, 2021. doi: 10.1145/3466752.3480125. ISBN 9781450385572 p. 977–991. [Online]. Available: <https://doi.org/10.1145/3466752.3480125>
- [76] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 03 1991. doi: 10.1162/neco.1991.3.1.79. [Online]. Available: <https://doi.org/10.1162/neco.1991.3.1.79>
- [77] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," 2017. [Online]. Available: <https://arxiv.org/abs/1701.06538>
- [78] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17-19 Jun 2013, pp. 1310–1318. [Online]. Available: <https://proceedings.mlr.press/v28/pascanu13.html>
- [79] Shengjie Chen and Chenxi Zhang, "Efficient Mamba-Based Accelerator Architecture for Massive-MIMO Indoor Localization Using the LuViRA Dataset," 2026, Master Thesis, Lund University.
- [80] M. Shi, A. Kneip, N. Chauvaux, J. Sun, C. Frenkel, and M. Verhelst, "Sparsity-Aware Hardware: From Overheads to Performance Benefits," *IEEE Solid-State Circuits Magazine*, vol. 17, no. 2, pp. 61–71, 2025. doi: 10.1109/MSSC.2025.3549709
- [81] S. Kiliçarslan and M. Celik, "RSigELU: A nonlinear activation function for deep neural networks," *Expert Systems with Applications*, vol. 174, p. 114805, 2021. doi: <https://doi.org/10.1016/j.eswa.2021.114805>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417421002463>

- [82] E. Tegler, M. Modig, P. Skarin, K. Åström, M. Oskarsson, and G. Flood, "Detection and localization of drones and uavs using sound and vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2025, pp. 6716–6724.
- [83] T. Ishihara, K. M. Kitani, C. Asakawa, and M. Hirose, "Deep Radio-Visual Localization," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018. doi: 10.1109/WACV.2018.00071 pp. 596–605.
- [84] A. Chau, K. Sekiguchi, A. A. Nugraha, K. Yoshii, and K. Funakoshi, "Audio-Visual SLAM towards Human Tracking and Human-Robot Interaction in Indoor Environments," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019. doi: 10.1109/RO-MAN46459.2019.8956321 pp. 1–8.
- [85] T. Zhang, H. Zhang, X. Li, J. Chen, T. L. Lam, and S. Vijayakumar, "AcousticFusion: Fusing Sound Source Localization to Visual SLAM in Dynamic Environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, 2021. doi: 10.1109/IROS51168.2021.9636585 p. 6868–6875. [Online]. Available: <https://doi.org/10.1109/IROS51168.2021.9636585>
- [86] J. R. Sánchez, O. Edfors, and L. Liu, "Positioning for Distributed Large Intelligent Surfaces using Neural Network with Probabilistic Layer," in *2021 IEEE Globecom Workshops (GC Wkshps)*, 2021. doi: 10.1109/GCWkshps52748.2021.9682016 pp. 1–6.
- [87] D. Pjanić, G. Tian, A. Reial, X. Cai, B. Bernhardsson, and F. Tufvesson, "Illuminating the Path: Attention-Assisted Beamforming and Predictive Insights in 5G NR Systems," 2025. [Online]. Available: <https://arxiv.org/abs/2505.18160>
- [88] G. Tian, X. Cai, T. Zhou, W. Wang, and F. Tufvesson, "Deep-Learning Based Channel Estimation for OFDM Wireless Communications," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022. doi: 10.1109/SPAWC51304.2022.9834008 pp. 1–5.
- [89] W. Tang, S.-G. Cho, J.-F. Zhang, and Z. Zhang, "Design and Optimization of Efficient Digital Machine Learning Accelerators: An overview of architecture choices, efficient quantization, sparsity exploration, and system integration techniques," *IEEE Solid-State Circuits Magazine*, vol. 17, no. 2, pp. 30–38, 2025. doi: 10.1109/MSSC.2025.3549361

PAPERS

Paper I

Paper I

Reproduced, with permission, from IEEE

I. YAMAN, G. TIAN, M. LARSSON, P. PERSSON, M. SANDRA, A. DÜRR, E. TEGLER, N. CHALLA, H. GARDE, F. TUFVESSON, K. ÅSTRÖM, O. EDFORS, S. MALKOWSKY, AND L. LIU, “The LuViRA Dataset: Synchronized Vision, Radio, and Audio Sensors for Indoor Localization,” *IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, doi: 10.1109/ICRA57147.2024.10610237.

The LuViRA Dataset: Synchronized Vision, Radio, and Audio Sensors for Indoor Localization

Ilayda Yaman¹, Guoda Tian¹, Martin Larsson², Patrik Persson², Michiel Sandra¹, Alexander Dürr³, Erik Tegler², Nikhil Challa¹, Henrik Garde⁴, Fredrik Tufvesson¹, Kalle Åström², Ove Edfors¹, Steffen Malkowsky¹, and Liang Liu¹

Abstract—We present a synchronized multisensory dataset for accurate and robust indoor localization: the Lund University Vision, Radio, and Audio (LuViRA) Dataset. The dataset includes color images, corresponding depth maps, inertial measurement unit (IMU) readings, channel response between a 5G massive multiple-input and multiple-output (MIMO) testbed and user equipment, audio recorded by 12 microphones, and accurate six degrees of freedom (6DOF) pose ground truth of 0.5 mm. We synchronize these sensors to ensure that all data is recorded simultaneously. A camera, speaker, and transmit antenna are placed on top of a slowly moving service robot, and 89 trajectories are recorded. Each trajectory includes 20 to 50 seconds of recorded sensor data and ground truth labels. Data from different sensors can be used separately or jointly to perform localization tasks, and data from the motion capture (mocap) system is used to verify the results obtained by the localization algorithms. The main aim of this dataset is to enable research on sensor fusion with the most commonly used sensors for localization tasks. Moreover, the full dataset or some parts of it can also be used for other research areas such as channel estimation, image classification, etc. Our dataset is available at: https://github.com/ilaydayaman/LuViRA_Dataset

I. INTRODUCTION

Fully autonomous driving and human-free smart factories are expected to significantly improve life quality [1], [2]. To build an autonomous smart factory, one of the most critical challenges is performing accurate localization and monitoring of autonomous service robots in real-time for such an indoor environment, e.g., a service robot in a factory needs to localize itself within centimeter-level accuracy to perform tasks such as lifting and placing objects. To solve this problem, localization algorithms that use multiple sensors to improve accuracy and reliability have developed immensely [3]–[6]. There are many works that use cameras, radio receivers and transmitters, microphones, and speakers separately to localize objects in indoor environments [7]–[9]. Recent works [10], [11] indicate that these sensors can be used jointly to complement each other and enhance the overall performance. However, to develop, evaluate, and compare algorithms that fuse these sensors, a public dataset that includes simultaneous and synchronized sensor readings from each sensor in the same environment is required.

This work is funded by the Swedish Research Council, Ericsson AB, and ELLIIT (Excellence Center at Linköping-Lund in Information Technology).

The authors are with Lund University, ¹Department of Electrical and Information Technology, ²Centre for Mathematical Sciences, ³Department of Computer Science, ⁴Humanities Lab (Corresponding author email: ilayda.yaman@eit.lth.se)

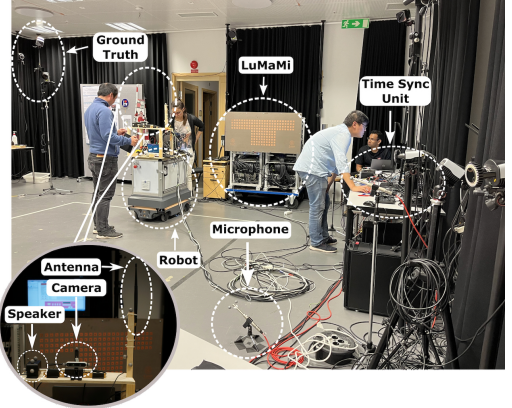


Fig. 1: A photo of the sensors and the ground truth system depicting labels for the different components and the bottom-left corner focuses on the devices on the robot.

Public datasets and benchmarks are key to the evaluation and development of algorithms designed by the scientific community. 12-Scenes [12], KU Leuven [13] and Structure from Sound [14] datasets are examples of public datasets that can be used for vision, radio, and audio based localization algorithms. 12-Scenes is one of many indoor localization datasets based on vision sensors [15], but very few public datasets include radio and audio sensors. Massive multiple-input and multiple-output (MIMO) is an enabling technology for the fifth-generation technology standard for cellular networks, or 5G, which has a scarcity of comprehensive datasets, especially for localization tasks where ground truth labels are required. The KU Leuven dataset is one of the very few datasets that contains measured channel state information (CSI) between a user and a massive MIMO testbed. The StructureFromSound dataset includes audio data collected with 12 microphones. These existing datasets contain the target sensors separately. Fusing separate datasets together for multisensory applications is not possible due to the lack of synchronization of different sensors as well as the different environments where the data has been captured.

In this paper, we describe, to the best of our knowledge, the first publicly available dataset that includes synchronized

data from vision, 5G radio, and audio sensors that are captured in the same indoor environment. We name the dataset that consists of 89 trajectories **the Lund University Vision, Radio, and Audio Dataset** or **LuViRA**. A *Mobile Industrialized Robot* (MIR200) is used to carry the required equipment for all the aforementioned sensors in the Lund University Humanities Lab's Motion Capture (Mocap) Studio. The radio data is recorded using the world's first real-time 5G massive MIMO testbed, *LuMaMi* [16]. A photo of the Mocap Studio, including the sensors, and the ground truth system, is given in Fig. 1 where each system and other required equipment are labeled. The detailed measurement setup, properties of each sensor, the synchronization schemes, key results, and known limitations of the dataset are described in the paper. The dataset is verified using state-of-the-art localization algorithms for the different sensors.

II. MEASUREMENT SETUP

Ground truth systems, that achieve higher localization accuracy than the state-of-the-art algorithms and use the target sensors, are crucial for the precise validation and benchmarking of these algorithms, making them an important component in our measurement campaign. The Lund University Humanities Lab's Mocap Studio consists of 18 networked high-speed infrared *Qualisys* cameras that record at 100Hz and are connected to the *Qualisys Sync Unit*, a synchronization box, for accurate synchronization. The system is calibrated and the localization error is measured to be less than 0.5 mm in the effective area. Due to the high accuracy that is achieved, it is used as a ground truth system throughout all the measurements.

A $4.2 \times 2.5 \text{ m}^2$ area shown in Fig. 2 is chosen as the effective area where the trajectories are recorded. The size of the effective area is decided on the basis of many factors, but the main reason is the coverage of the ground truth system. The ground truth system tracks the 3D positions of the robot and the sensors by placing reflective spherical markers on target objects. Moreover, with a rigid setup of at least 3 markers on an object, the system provides six degrees of freedom (6DoF) data, i.e., 3D orientation plus the 3D position. The ground truth system is the most accurate when several cameras can see the markers and on the sides of the studio, some markers are usually blocked by other objects.

Many precautions are taken to create a controlled environment for all the systems in our measurement campaign. A mobile industrial robot is used to securely transport a camera, a single antenna user equipment (UE), a sound source (speaker), and the equipment connected to them. The camera, antenna, and speaker are placed on a higher-level platform built for this measurement campaign. The platform ensures that the sensors have dedicated and stable locations on the robot and have an unobstructed view of the environment. A top-down view of the sensor placement in the studio is shown in Fig. 2, with sensors marked in yellow and actuators marked in blue. The rounded red rectangle depicts the robot, squares marked with "A", "C" and "S" stand for antenna, camera, and speaker, respectively. The *LuMaMi*

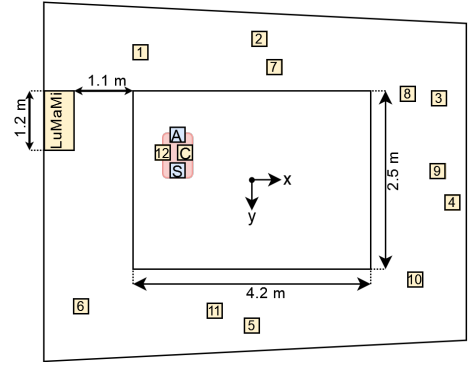


Fig. 2: Top-down view of the studio, containing twelve microphones (1-12), antenna (A), RGB-D camera (C), speaker (S), and LuMaMi testbed.

testbed is placed in the top-left corner, and the placement of the microphones is illustrated as numbered squares.

III. SENSOR SETUP

This section gives detailed descriptions of the vision, radio, and audio systems. A summary of all the sensors recorded and their characteristics are shown in Table I. For completeness, the ground truth system is also added to the table. As can be seen in Table I, different sensors have different snapshot rates per second (ps). Thus, when these sensors are used together, the data should be associated with each other based on their respective timestamps. An example code that uses the sensor with the lowest snapshot rate as a reference and maps the others using the smallest time difference between timestamps is provided with the dataset.

TABLE I: Overview of the four main systems.

System	Snapshot Rate (ps)	Sensor	File Format
Vision	15-30 frames	Intel® RealSense™ depth camera	.png and .bag
Radio	100 channel estimates	LuMaMi testbed with 100 antennas	.mat
Audio	up to 96 k samples	12 microphones	.wav
Ground Truth	100 frames	18 high speed cameras	.txt

A. Vision system

An *Intel® RealSense™ D435i* camera connected to a laptop on the robot is used as the input for the vision system. Each snapshot of visual data includes an RGB or color image, left and right infrared images (obtained by imagers), a depth map, a point cloud, and inertial measurement unit (IMU) data. The IMU data is collected by two sensors: an accelerometer and a gyroscope. The features are captured by using Robot Operating System (ROS) and rosbag (.bag) file

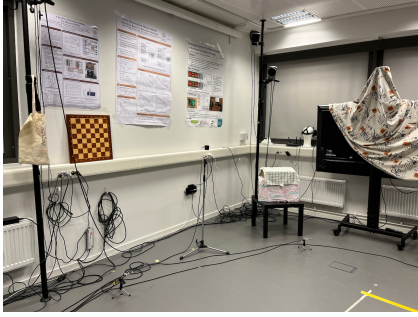


Fig. 3: Examples of the decorations in the environment that can be used in vision-based localization algorithms.

format. A summary of all the features of the camera is given in Table II. The frequency of different features is chosen on the basis of the maximum bandwidth of the USB-2 bus on the laptop. For future dataset creation, we instead suggest choosing a common fps for the RGB, infra cameras and depth map, as well as a common frequency for accel, and gyro. This would simplify usage and a reduced fps would potentially allow higher resolution.

TABLE II: Features and Recording settings of the Camera

Sensor	Snapshot Rate	Resolution	File Format
RGB Camera	30 fps	640x480	.png and .bag
Left Imager	15 fps	640x480	.bag
Right Imager	15 fps	640x480	.bag
Depth Module	15 fps	640x480	.png and .bag
IMU (gyroscope)	400 Hz	N/A	.bag
IMU (accelerometer)	100 Hz	N/A	.bag

While using these data, the effect of the rolling shutter camera can be ignored, since the camera is placed on top of a relatively stable robot moving at a low speed. The left and right imagers embed depth information to aid the generation of depth maps (with the help of a filter), and using these images in stereo-based localization algorithms might give lower accuracy compared to non-filtered images. The Mocap Studio is decorated with posters and objects with different textures to enable feature extraction in vision-based localization algorithms. Some example decorations can be seen in Fig. 3.

B. 5G Radio system

The radio system consists of an user equipment (UE) which is connected to the single antenna mounted on the robot and the *LuMaMi* testbed with 100 antennas (each connected to an individual RF chain). *LuMaMi* operates at a center frequency of 3.7GHz and has a bandwidth of 20 MHz. The UE and *LuMaMi* are connected via a cable to establish frequency synchronization and to eliminate carrier frequency offset. A collection of known signals, called pilot

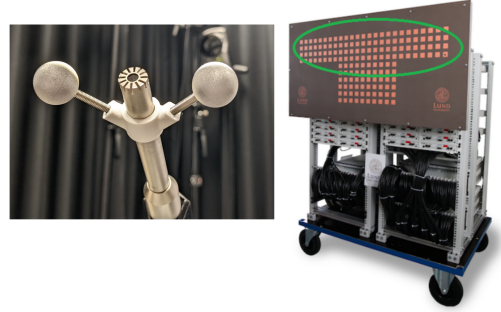


Fig. 4: Markers and the microphone are shown on the left side of the image and the *LuMaMi* testbed where the wide antenna configuration is marked with the green circle is on the right.

signals, are sent by the UE every 10 ms. The *LuMaMi* testbed then captures 100 snapshots of the channel per second (ps). These parameters are recorded and can be used to estimate the position of the mobile antenna in the environment. To exploit more information from the *azimuth* compared to the *elevation* domain, a wide antenna configuration, which is 25×4 , is selected rather than a quadratic one. The *LuMaMi* testbed and the wide antenna configuration are depicted on the right side of Fig. 4. During the measurement campaign, the UE is placed on top of the moving robot, while *LuMaMi* remains static. A universal software radio peripheral (NI X310) connected to a single dipole antenna is used as the UE.

C. Audio system

Audio-based localization uses an array of microphones to calculate the direction or location of a sound source by measuring different metrics, such as the time differences of arrival (TDOA) of sound signals at different microphones and triangulating the source's position based on these TDOA measurements. In our measurement campaign, 12 microphones (*T-bone MM-1*) are set up according to Fig. 2 and connected to a single sound card that is recording to a laptop. A stereo speaker is placed on top of the robot, as well as the 12th microphone. The microphone on the robot is placed as close as possible to the speaker (sound source) and works as a reference to synchronize the speaker with the microphones. In addition to the 12 audio tracks, a synchronization pulse from the ground truth system on start and stop is recorded as a 13th track ("Sync"). To make calculations easier by viewing the sound source as a point source, only one side of the speaker is enabled (playing sound), and the head of the reference microphone is placed directly in front of the sound source. All microphones, except the one on the robot, have two markers placed, as seen on the left side of Fig. 4.

The sampling frequency of the microphones is 96 kHz and, if required, the audio system can localize the speaker at the same frequency. Microphones are placed asymmetrically on the floor and on different heights to avoid microphones being co-linear or co-planar, as this may cause degeneracies when solving for positions. The sound level of every microphone is checked individually and the speaker is tested.

IV. CALIBRATION AND SYNCHRONIZATION

In this section, we outline the calibration and synchronization procedures of different systems, which are key requirements for localization using each sensor and using the sensors together. Calibrating a sensor is important to ensure accurate and reliable measurements, while time synchronization is crucial for multisensory applications to ensure that data from different sensors are aligned in a common temporal reference timeframe, enabling accurate correlation and fusion of information. Moreover, the parameters obtained by the sensor calibration are usually required as input to the localization algorithms.

A. Intrinsic and Extrinsic Calibration of the Sensors

As an overview, the camera (and the built-in IMU) used for the vision system is calibrated intrinsically and extrinsically while the intrinsic calibrations of the sensors used for radio and audio systems are not done by us. Thus, for the radio system, traditional radio-based localization methods cannot be used, since the corresponding amplitude and phase of the antennas are not calibrated.

For the camera, two intrinsic calibrations are performed. For the first calibration, we follow the calibration procedure proposed by the manufacturer for the best performance. The second calibration is done using the *Kalibr* calibration tool [17]. With this calibration, the intrinsic and extrinsic parameters of the camera are extracted. The intrinsic parameters obtained with this method include camera centers, focal length, distortion parameters, and noise information for the IMU. The relationship between different cameras (i.e. two imagers and the RGB camera) and the transformation matrix between the cameras and IMU are obtained and used as extrinsic parameters of the camera.

Temperature changes can significantly affect the calibration of the sensors. The internal temperature change of the devices was not recorded, but the room temperature has been measured for some of the trajectories. The change in temperature also affects the propagation factors of the air and the speed of sound, which can have an impact on the audio localization algorithms.

B. Time Synchronization

We take several steps to make sure that the timestamps can be matched for each sensor (and the ground truth), so that the collected data can be verified with the ground truth data and used for sensor fusion. The block diagram of different systems and the connections that enable the time synchronization are given in Fig. 5. The time synchronization relies on the *Raspberry Pi* and *Qualisys Sync Unit*. An

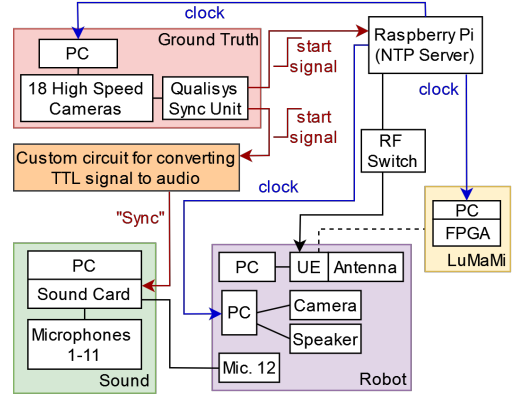


Fig. 5: The overview block diagram of the synchronization system and the connections to different sensors.

Network Time Protocol (NTP) server is used to give the correct timestamps to each system, even though there are other circuits to sync the different systems in some cases.

The NTP server runs on the *Raspberry Pi* and used to synchronize all computers regularly, except the computer used for sound recordings. One of the *Raspberry Pi* Input/Output pins, configured as input with interrupt function, is connected to the *Qualisys Sync Unit*, in order to listen to the short pulse (TTL signal) sent by the ground truth system when the recording starts. This TTL signal is called the “start signal” in Fig. 5 and is also sent to the sound system, connecting all the systems together. A more detailed description of the synchronization methods and the corresponding verification process are as follows.

1) Vision: The NTP server on the *Raspberry Pi* is used to update the clock in the laptop, which is used to record all the vision data. The accuracy of the system is validated up to 33 ms (the frame rate of the camera) with the help of an LED connected to the circuit used for the radio system.

2) Radio: The UE and *LuMaMi* are synchronized using a cable to avoid losing the signal between the receiver and the transmitter. The radio data is recorded on the testbed, thus its computer is connected to the NTP server to get the correct timestamp. To validate the timestamps, the *Raspberry Pi* is connected to an RF switch on the robot, which can turn the transmission of the UE on and off. Afterwards, we plot when the UE starts and stops transmissions and compare the results to the timestamps obtained by the NTP server. The timestamp mismatch results in a maximum ground truth error 0.5 mm, which is negligible.

3) Audio: Synchronization between microphones is done by connecting all microphones to a single sound card. In order to synchronize the common timestamp, a separate circuit is built to convert the TTL signal from the synchronization box to an audio signal which is passed to the sound card as a separate channel, as described above. For the audio

system, the 13th “Sync” channel is the recording of the pulse, which gets triggered when the mocap system starts and stops recording. The synchronization is verified using a clapperboard. The markers on the clapperboard are tracked by the ground truth system and seen by the vision system. When its two sections are clapped shut, it creates a distinctive sound and simultaneously marks a visual cue.

V. CREATED TRAJECTORIES

There are 89 trajectories collected in this measurement campaign that include all sensors. To meet all the requirements of different systems, we divide the dataset into two parts: the “grid” and “random” data. The “grid” data consist of 75 trajectories while “random” data consist of 14 trajectories. All the trajectories are divided into small sub-trajectories due to the limited amount of continuous data that *LuMaMi* can store at a time due to the limited memory size. Based on previous experiments, we calculate the maximum amount of time that *LuMaMi* can capture data continuously for up to 50 s (which includes the time it takes to establish the synchronization between all the systems). The system setup is the same in all trajectories, and if longer trajectories are required, the data can be used jointly instead.

There are additional trajectories provided with the dataset that include data from one sensor at a time which can aid in calibration, extracting more information, or verifying the sensors separately. These include recordings with the microphones where the speaker is hand-held and the rest of the equipment is turned off, and recordings of the background noise in the environment when all the equipment is running. There are also recordings with the camera, where the camera is hand-held to cover the whole 3D environment, and a chessboard is recorded with different rotations and distances.

A. Grid Data

One of the main considerations in designing trajectories is supporting dense sampling for radio-based localization algorithms. Thus, the spatial sampling of the data should be less than $\lambda/2$ (half a wavelength) according to the Nyquist theorem. For more details on this requirement, we refer the reader to [18]. The wavelength of *LuMaMi* is calculated as approximately 8 cm, and as a result, training data is created such that it consists of measurements of the channel response approximately every 4 cm. In order to generate data that satisfies this condition, we scan the target area in the studio in 75 trajectories and call this part the “grid” data.

To achieve the given resolution, the robot starts in the top-left corner of the effective area. With the help of ROS, the robot is moved 4.2 m in a straight line with a speed of 0.1 m s^{-1} . Fig. 6 shows the overall summary of the environment and the planned trajectory for this part. However, in the actual measurement campaign, a slight deviation from the straight line is seen for each “grid” trajectory. As a result of changes in the trajectories and the limited time available in the Mocap Studio, the $4.2 \times 2.5 \text{ m}^2$ area is covered with 75 parallel trajectories for the “grid” data. Throughout the “grid” data, the speaker plays a chirp-based audio.

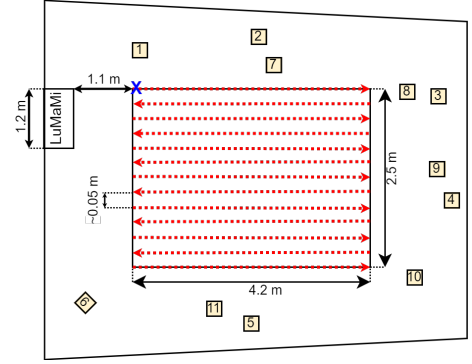


Fig. 6: Red lines depicting the movement of the robot for the “grid” trajectories where the blue cross is the initial location. Yellow numbered squares are the microphones.

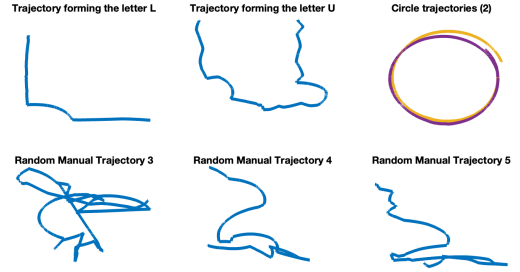


Fig. 7: Example “random” trajectories plotted by using the data from the ground truth system tracking the camera object. Blue lines show one trajectory at a time where yellow and purple lines are used to indicate more than one trajectory.

B. Random Data

We conducted a total of 14 measurements, which are called “random” data, and some of the selected ones are shown in Fig. 7. In this scenario, the robot orientation is not always perpendicular to the antenna array (unlike the “grid” data), while humans act as static or dynamic scatterers in several trajectories. Dynamic environments pose significant localization challenges, including uncertainty and increased computational complexity, with numerous research works addressing these issues [19], [20].

The two circular trajectories are specifically designed for localization algorithms where loop closure is desirable. The two diagonal trajectories are corner cases for audio-based localization algorithms since they can cause some self-calibration methods for the audio system to fail [21]. In the manual trajectories, the robot is controlled by the built-in user interface with the virtual joystick. Thus, the speed of the robot is not constant, and the movements are more unstable and faster compared to the speed and movement of

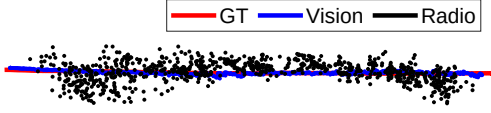


Fig. 8: The localization result of the ground truth (GT), vision, and radio based localization algorithms for *Grid2* trajectory (single antenna is aligned to camera location).

the robot in “grid” data. The “random manual” trajectories number 1, 2, and 6 include people moving around, whereas numbers 3, 4, and 5 do not. For the trajectories with people moving around in them, ground truth drops more frames due to markers not being seen by enough cameras. All the trajectories are in line-of-sight (LoS) from the point of view of *LuMaMi* except for the “random manual” trajectories with people walking around. In some cases in these trajectories, the LoS between *LuMaMi* and the UE is obstructed by people and/or cables. We expect that the dynamic environment in these trajectories affects all localization methods, but these trajectories are particularly hard for radio-based localization algorithms due to the very dynamic environment.

VI. VALIDATION EXPERIMENTS

This section offers a brief overview of the validation experiments done on the dataset to set a baseline for each component. For a more comprehensive understanding, we refer the interested reader to [22]. In order to evaluate the dataset, we use RGB-D Oriented FAST and Rotated BRIEF simultaneous localization and mapping (ORB-SLAM3) [23] for the vision data, machine learning [18] for the massive MIMO radio data, and Structure from Sound2 (SFS2) [24] for the audio data, respectively. Here, we use the *Grid2* trajectory as an example. As shown in Fig. 8, the ORB-SLAM3 algorithm demonstrates a mean localization accuracy of 4.3 cm with a standard deviation (SD) of 2.2 cm for the *Grid2* trajectory. The ground truth trajectory and the estimated trajectory are aligned by using the automated tools provided by [25]. The machine learning algorithm for the radio data is trained on the oddly numbered “grid” trajectories. Evenly-numbered “grid” trajectories are used to test the accuracy of the machine learning algorithm, which results in a mean error of 13 cm and SD of 12 cm. The SFS2 algorithm results in more than 1 m localization error in the “grid” data where a chirp-based audio is used. In trajectories with music-based audio such as the *Random Circle 2* trajectory, the current implementation of the algorithm can achieve up to a mean error of 9.6 cm and SD of 3.1 cm. Detailed evaluation and discussion of these results are not in the scope of this paper but can be found in [22].

VII. KNOWN LIMITATIONS AND REMARKS

The following limitations and remarks should be considered when using the dataset:

- 1) Due to the need for power supply and synchronization, a bundle of cables is connected to the robot. For some of the trajectories, the cables are hanging from the roof to avoid complications in the scene due to the robot’s movement. For the radio system, the cables become a part of the channel, which might affect the accuracy of the localization algorithm. For the vision system, the cables are seen by the camera in some frames and create a dynamic environment. Moreover, the movement of the robot is restricted by the cables that are attached to it.

- 2) In the “grid” data, the robot is facing the same direction of the studio due to the previous remark/limitation (cables). This results in the same viewing direction for the camera and the same direction between the antenna that is connected to the UE and *LuMaMi*. For vision data, “random” data can be used instead.

- 3) The room temperature increases significantly throughout the measurements, due to the equipment in the studio. After the change in temperature is noticed, the studio is measured almost every hour for the second half of the “grid” data. Unfortunately, no temperature data exist for the first half of the “grid” data, where the studio got significantly warmer in the evening. We believe that the temperature was around 28 °C. This is only important for the “grid” data, and no significant temperature change was observed for the “random” data.

VIII. CONCLUSION

Localization is an essential process for many different tasks and devices such as UAVs, autonomous cars, and service robots. Localization algorithms require high computational power, where low power and high efficiency are the key to enabling these new technologies in battery-operated devices. In this paper, we have described a novel, unique, and comprehensive dataset that includes synchronized multi-sensory data (vision, radio, and audio sensors), collected in a controlled environment, together with high precision ground truth labels. This dataset, fused or separately, should allow accurate localization of devices (within cm) in real time. We make the dataset publicly available and give detailed information regarding its creation, drawbacks, and limitations. Overall, we hope that this dataset will contribute to the creation of localization algorithms that maximize the usage of the available data for low-power devices and also provide an in-depth understanding and further development of AI, machine learning, and 5G (and beyond) applications.

ACKNOWLEDGMENT

The authors would like to thank the Humanities Lab, Volker Krueger, and Matthias Mayr from the Department of Computer Science, Anders Robertsson from the Department of Automatic Control, and Sirvan Abdollah Poor, Jesús Rodríguez Sánchez, Jens Gulín and Sara Gunnarsson from the Department of Electrical and Information Technology at Lund University for providing resources, technical support, assistance, and feedback in our measurement campaign. In Memoriam of Anders Robertsson.

REFERENCES

- [1] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination," *Computer Networks*, vol. 101, pp. 158–168, 2016.
- [2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.
- [3] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4670–4677, 2018.
- [4] A. Poulou, J. Kim, and D. S. Han, "A sensor fusion framework for indoor localization using smartphone sensors and wi-fi rssi measurements," *Applied Sciences*, vol. 9, no. 20, 2019.
- [5] B. Xie, L. Yang, Z. Yang, A. Wei, X. Weng, and B. Li, "Fourstr: When multi-sensor fusion meets semi-supervised learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 676–682, 2023.
- [6] J. P. Grisales Campeón, S. López, S. R. de Jesús Meleán, H. Moldovan, D. R. Parisi, and P. I. Fierens, "Fusion of magnetic and wifi fingerprints for indoor positioning," in *2018 Congreso Argentino de Ciencias de la Informática y Desarrollo de Investigación (CACIDI)*, pp. 1–5, 2018.
- [7] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," *Advances in neural information processing systems*, 2021.
- [8] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-Based Multi-Antenna and Multi-Point Indoor Positioning Using Probability Fusion," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2162–2176, 2022.
- [9] S. Zhayida, F. Andersson, Y. Kuang, and K. Åström, "An automatic system for microphone self-localization using ambient sound," in *2014 22nd European Signal Processing Conference (EUSIPCO)*, pp. 954–958, IEEE, 2014.
- [10] T. Ishihara, K. M. Kitani, C. Asakawa, and M. Hirose, "Deep Radio-Visual Localization," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.
- [11] A. Chau, K. Sekiguchi, A. A. Nugraha, K. Yoshii, and K. Funakoshi, "Audio-Visual SLAM towards Human Tracking and Human-Robot Interaction in Indoor Environments," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019.
- [12] J. Valentin, A. Dai, M. Nießner, P. Kohli, P. Torr, S. Izadi, and C. Keskin, "Learning to navigate the energy landscape," *arXiv preprint arXiv:1603.05772*, 2016.
- [13] S. De Bast, A. P. Guevara, and S. Pollin, "CSI-based Positioning in Massive MIMO systems using Convolutional Neural Networks," *arXiv preprint arXiv:1911.11523*, 2019.
- [14] K. Åström, M. Larsson, G. Flood, and M. Oskarsson, "Extension of Time-Difference-of-Arrival Self Calibration Solutions Using Robust Multilateration," in *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 870–874, 2021.
- [15] Y. Liu, Y. Fu, F. Chen, B. Goossens, W. Tao, and H. Zhao, "Simultaneous localization and mapping related datasets: A comprehensive survey," *arXiv preprint arXiv:2102.04036*, 2021.
- [16] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. Wong, F. Tufvesson, V. Öwall, and O. Edfors, "The World's First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation," *IEEE Access*, pp. 9073 – 9088, 2017.
- [17] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1280–1286, 2013.
- [18] G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu, and F. Tufvesson, "High-precision machine-learning based indoor localization with massive mimo system," in *ICC 2023 - IEEE International Conference on Communications*, pp. 3690–3695, 2023.
- [19] S. Garcia-Villalonga and A. Perez-Navarro, "Influence of human absorption of wi-fi signal in indoor positioning with wi-fi fingerprinting," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–10, 2015.
- [20] M. Dubenova, A. Zderadickova, O. Kafka, T. Pajdla, and M. Polic, "D-inloc++: Indoor localization in dynamic environments," in *Pattern Recognition (B. Andres, F. Bernard, D. Cremers, S. Frintrop, B. Goldlücke, and I. Ihrke, eds.), (Cham), pp. 246–261, Springer International Publishing, 2022.*
- [21] Y. Kuang, E. Ask, S. Burgess, and K. Åström, "Understanding toa and tdoa network calibration using far field approximation as initial estimate," in *ICPRAM 2012 - Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods, Volume 2*, pp. 590–596, 2012.
- [22] I. Yaman, G. Tian, E. Tegler, P. Persson, N. Challa, F. Tufvesson, O. Edfors, K. Åström, S. Malkowsky, and L. Liu, "Indoor localization using radio, vision and audio sensors: Real-life data validation and discussion," *arXiv preprint arXiv:2309.02961*, 2023.
- [23] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [24] K. Åström, M. Larsson, G. Flood, and M. Oskarsson, "Extension of Time-Difference-of-Arrival Self Calibration Solutions Using Robust Multilateration," in *29th European Signal Processing Conference (EUSIPCO)*, 2021.
- [25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

Paper II

Paper II

Reproduced, with permission, from IEEE

I. YAMAN, G. TIAN, E. TEGLER, J. GULIN, N. CHALLA, F. TUFVESSON, O. EDFORS, K. ÅSTRÖM, S. MALKOWSKY, AND L. LIU, "LuViRA Dataset Validation and Discussion: Comparing Vision, Radio, and Audio Sensors for Indoor Localization," *IEEE Journal of Indoor and Seamless Positioning and Navigation*, vol. 2, Jul 2024, pp. 240-250, doi: 10.1109/JISPIN.2024.3429110.

LuViRA Dataset Validation and Discussion: Comparing Vision, Radio, and Audio Sensors for Indoor Localization

Ilayda Yaman, *Student Member, IEEE*, Guoda Tian, *Student Member, IEEE*, Erik Tegler, *Student Member, IEEE*, Jens Gulin, *Student Member, IEEE*, Nikhil Challa, *Student Member, IEEE*, Fredrik Tufvesson, *Fellow, IEEE*, Ove Edfors, *Senior Member, IEEE*, Kalle Åström, *Senior Member, IEEE*, Steffen Malkowsky, and Liang Liu, *Member, IEEE*

Abstract—We present a unique comparative analysis, and evaluation of vision, radio, and audio based localization algorithms. We create the first baseline for the aforementioned sensors using the recently published Lund University Vision, Radio, and Audio (LuViRA) dataset, where all the sensors are synchronized and measured in the same environment. Some of the challenges of using each specific sensor for indoor localization tasks are highlighted. Each sensor is paired with a current state-of-the-art localization algorithm and evaluated for different aspects: localization accuracy, reliability and sensitivity to environment changes, calibration requirements, and potential system complexity. Specifically, the evaluation covers the ORB-SLAM3 algorithm for vision-based localization with an RGB-D camera, a machine-learning algorithm for radio-based localization with massive MIMO technology, and the SFS2 algorithm for audio-based localization with distributed microphones. The results can serve as a guideline and basis for further development of robust and high-precision multi-sensory localization systems, e.g., through sensor fusion, and context- and environment-aware adaptation.

Index Terms—Computer vision, indoor localization, massive multiple-input multiple-output (MIMO), multisensor dataset, simultaneous localization and mapping (SLAM).

I. INTRODUCTION

THE need for accurate and reliable location information is rising with the increasing number of applications that require seamless localization, such as smart factories, autonomous driving, and unmanned aerial vehicles (UAVs) [1]. Achieving seamless localization is a critical prerequisite for ensuring the safety, efficiency, and reliability of operations in many scenarios, especially in dynamic conditions such as urban mobility, indoor navigation, and rapidly changing environments. For outdoor scenarios, the real-time kinematic enabled global navigation satellite system (GNSS-RTK) can achieve cm-level accuracy under good conditions [2]. For

indoor scenarios, there exist many different systems using various sensors and technologies for performing localization tasks, including WiFi, Bluetooth, digital cameras, inertial measurement units (IMU), and microphones. Different sensors have their advantages and limitations and there is a need for an in-depth investigation in this area. This investigation requires vision, radio, and audio sensor data to be captured in the same environment at the same time. However, there is very limited work that evaluates these sensors jointly with a realistic dataset, while using 5G technologies such as massive multiple-input multiple-output (MIMO).

Our paper conducts a detailed evaluation of selected existing vision, radio, and audio based localization algorithms and compares their localization performances in a real indoor scenario. To this end, our main contributions are as follows:

- Providing baseline results for the first publicly available dataset that contains synchronized data from vision, radio, and audio sensors, and a ground truth system with localization error less than 0.5 mm: the Lund University Vision, Radio, and Audio (LuViRA) dataset [3]¹.
- Evaluating each sensor type for localizing the target object based on the following aspects using state-of-the-art localization algorithms: localization accuracy, reliability and sensitivity to environment changes, calibration requirements, and potential system complexity.
- Analyzing the advantages and limitations of localization algorithms based on aforementioned sensors coexisting in the same environment for trajectories with static and dynamic (people or objects moving around, sharp movements) environments.

The LuViRA dataset contains massive MIMO channel data captured using the 100-antenna *Lund University massive MIMO* (LuMaMi) testbed [4], RGB-D image streams from an Intel® RealSense depth camera D435i, and audio clips captured by 12 microphones distributed around the measurement area. An overview of the measurement environment can be seen in Fig. 1. We use the 89 trajectories provided in the LuViRA dataset, which consists of 75 “grid” and 14 “random” trajectories.

This work is funded by the Swedish Research Council and Ericsson AB and partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. (Corresponding author: Ilayda Yaman.)

The authors are all with LTH, Lund University, SE-22100 Lund, Sweden (Corresponding author email: ilayda.yaman@eit.lth.se).

¹https://github.com/ilaydayaman/LuViRA_Dataset

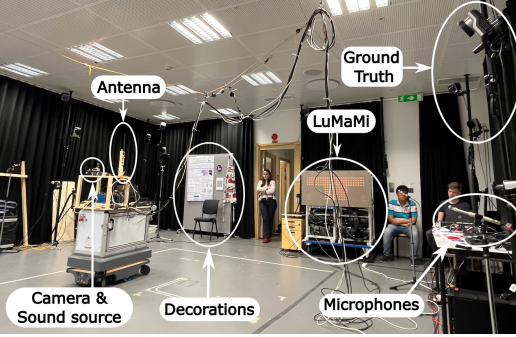


Fig. 1. An overview of the measurement setup during the recording of the *Random circle1* trajectory.

The state-of-the-art algorithms chosen to evaluate the localization accuracy of each sensor are *Oriented FAST and Rotated BRIEF simultaneous localization and mapping* (ORB-SLAM3) [5], a high precision machine-learning (ML) algorithm [6], and *StructureFromSound2* (SFS2) [7]². Based on the difference between the estimated location from these algorithms and the ground truth labels included in the dataset, the localization accuracy is calculated in terms of the mean and standard deviation (SD) of the localization errors. To assess the robustness of the algorithms and sensors, the localization accuracy is evaluated under different conditions, e.g., static and dynamic environments. The sensitivity to calibration errors of different sensors and algorithms is also discussed, from the general aspects of required calibration to precision and frequency requirements. **To the best of our knowledge, this is the first paper that evaluates and compares localization algorithms based on vision, 5G radio, and audio sensors in a realistic and synchronized measurement scenario.**

II. RELATED WORKS

Many academic works on indoor localization study vision, radio, and audio sensors individually. For example, there are many localization systems that use variations of visual sensors such as monocular and stereo, or RGB-D cameras [5] [8]. Moreover, the camera information is frequently fused with other sensors such as IMU, and Light Detection and Ranging (LiDAR), to combat the limitations of the vision-based localization systems in dark, low-texture, and dynamic environments [9]. Some works also use fully distributed multi-robot SLAM systems to achieve robustness and accuracy [10].

On the other hand, the localization accuracy of traditional radio-based localization algorithms, such as Bluetooth and Wi-Fi based fingerprinting, are highly dependent on the setup and measurement environment. To combat this issue, some works use sensors such as cameras and LiDAR jointly with radio-based localization systems [11]. On the other hand, the accuracy requirements are increasing with each generation, and extremely high accuracy is expected to be fundamental

for the sixth generation (6G) systems [12]. Thus, there are many works that explore new enabling technologies, such as distributed large antenna arrays and millimeter-wave transmissions, to improve localization accuracy and robustness [13]. One enabler is massive MIMO antenna arrays, that provide high spatial resolution to fulfill the requirements of accurate localization [14] [15].

Even though audio-based localization is more of a niche area compared to vision and radio based approaches, various methods are available for localizing sound sources [16] [17]. Under-water localization, such as passive sonar, relies on audio signals since it has good propagation properties where both vision and radio are weak in performance [18]. In many cases, the object to localize naturally has an audio signature, whereas vision relies on a separate light source and radio requires active transmission.

There are many indoor localization datasets based on vision sensors [19], but very few public datasets include radio and audio sensors. KU Leuven [14] and *StructureFromSound* [20] are examples of public datasets for radio and audio based localization methods, respectively. Due to the unavailability of public datasets that include vision, radio, and audio sensors, there are very few works that use these sensors jointly to improve the accuracy, reliability, and robustness of the localization tasks [21] [22]. When there are discussions on these sensors together, the sensors are discussed in different scenarios [23]. We believe the number of works that fuse these sensors will increase with the aid of a public dataset with known baselines, and an extensive study on the advantages and limitations of each sensor.

III. BACKGROUND

In this section, the localization algorithms selected for different sensor systems are presented at a high level of abstraction, to provide necessary background information.

A. Vision System

Most vision-based localization algorithms assume no prior knowledge of the environment. Thus the problem of localization extends to simultaneous localization and mapping (SLAM). Briefly, a classical Visual SLAM framework consists of the following steps [24]:

- 1) Feature extraction detects and matches key points in each frame.
- 2) Visual odometry (VO) creates an initial local map by estimating the movement, location, and orientation (pose) of the camera from consecutive frames.
- 3) Loop closing detects if the camera goes back to a previous location.
- 4) Filtering and optimization (i.e. bundle adjustment) create a more accurate camera trajectory and an environment map using estimates from VO and loop closing.

Popular algorithms that solve the SLAM problem include ORB-SLAM3 [5], ROVIO [9], and RTAB-MAP [25]. ORB-SLAM3 is a very well-established, open-source algorithm. Oriented FAST and Rotated BRIEF (ORB) features are used in the visual odometry and loop closing steps, resulting

²<https://github.com/kalleastrom/StructureFromSound2>

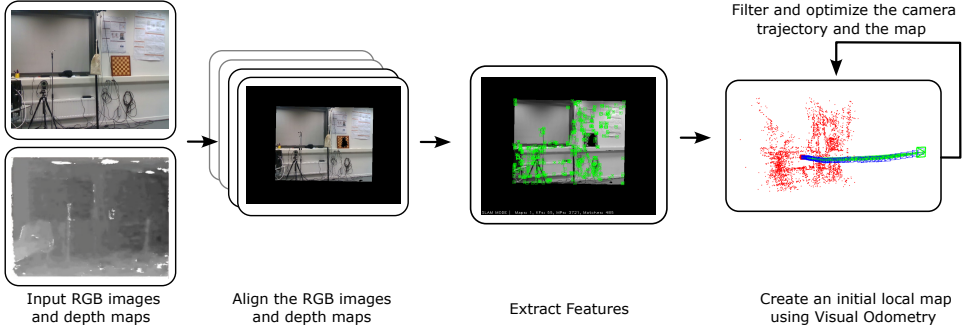


Fig. 2. The ORB-SLAM3 algorithm [5] pipeline for the vision system.

in accurate, robust, and real-time localization and mapping, even in dynamic environments. Moreover, ORB-SLAM3 is designed to support different types of cameras depending on the available hardware in the system. For example, if there is only a single camera available, the monocular ORB-SLAM3 algorithm is used for localization. The accuracy of the monocular ORB-SLAM3 is however limited due to the lack of depth information on the extracted features. Instead, if there is a stereo or depth camera in the environment, a more robust and accurate solution can be obtained by using stereo or RGB-D ORB-SLAM3 algorithms, which are used in this paper for vision-based localization. A high-level diagram of the pipeline using the LuViRA dataset is shown in Fig. 2. The IMU data from the camera unit is not used.

B. Radio System

The accuracy and applicability of different radio-based localization systems differ greatly from each other depending on the system configuration, such as system bandwidth, number of antennas, or input features used. The following three features are widely used:

- **Received signal strength (RSS):** The distance between the transmitter and receiver can be estimated by using a propagation path loss model. This model establishes a link between the received signal strength and the separation distance between the transmitter and receiver.
- **Delay or time based:** The most common time-based methods use the time of arrival (TOA) and time difference of arrival (TDOA), which generally require multiple synchronized receivers. TOA calculates the distance between the transmitter and the receiver by multiplying the signal propagation time with the speed of light. TDOA computes the arrival time differences at the receivers. The location of the transmitter can be estimated from the intersections of hyperboloids created by those time differences.
- **Angle of Arrival (AOA):** Arrival angles of the received signals are calculated and used to estimate the location of the transmitter by triangulation or multilateration techniques using either multiple AOAs or a single AOA together with the measured TOA.

Time and angle based methods can achieve very high accuracy, but they require the antennas to be calibrated, which can be a challenging task in real-life deployments. ML-based methods [6] [26] can relax the calibration requirement significantly but may require a significant amount of training data. E.g., [6] achieves centimeter-level accuracy with a 100-antenna massive MIMO system, by fusing 2 fully connected neural networks (FCNNs). Each FCNN has 8 hidden layers and takes instantaneous spatial covariance matrix and channel impulse responses as inputs, which provide information in both angular and delay domains. Given the advantage of not having to calibrate the antenna array and its relatively low computational complexity, [6] is used for evaluating the radio-based localization system. The ML algorithm pipeline is shown in Fig. 3.

C. Audio System

Audio-based localization extracts distance information from audio recordings. An example of an audio system setup involves multiple stationary microphones recording sound from a single moving source. Simultaneously locating both the sound source and microphones is referred to as the structure from sound (SFS) problem [27]. As in the radio case, the location can be estimated in several ways, such as by using strength, travel time, or the direction of arrival (using an array of microphones) of the sound signal. Time-based features such as TOA and TDOA, are widely used in solving the SFS problem [16] [17].

The TDOA feature is the basis for the SFS2 algorithm, which is an open-source localization algorithm that can estimate the sound source locations with or without prior knowledge of the microphone locations. Accuracy can be increased by the assumption that the sound source is tracked continuously during smooth motion, which enables using recent estimates as prior location information and temporal smoothness constraints. The SFS2 algorithm consists of the following steps [7]:

- 1) Read synchronized microphone recordings as input.
- 2) Extract initial TDOA estimates for each time instant and each pair of microphones by using Generalized Cross Correlation with Phase Transform (GCC-PHAT).

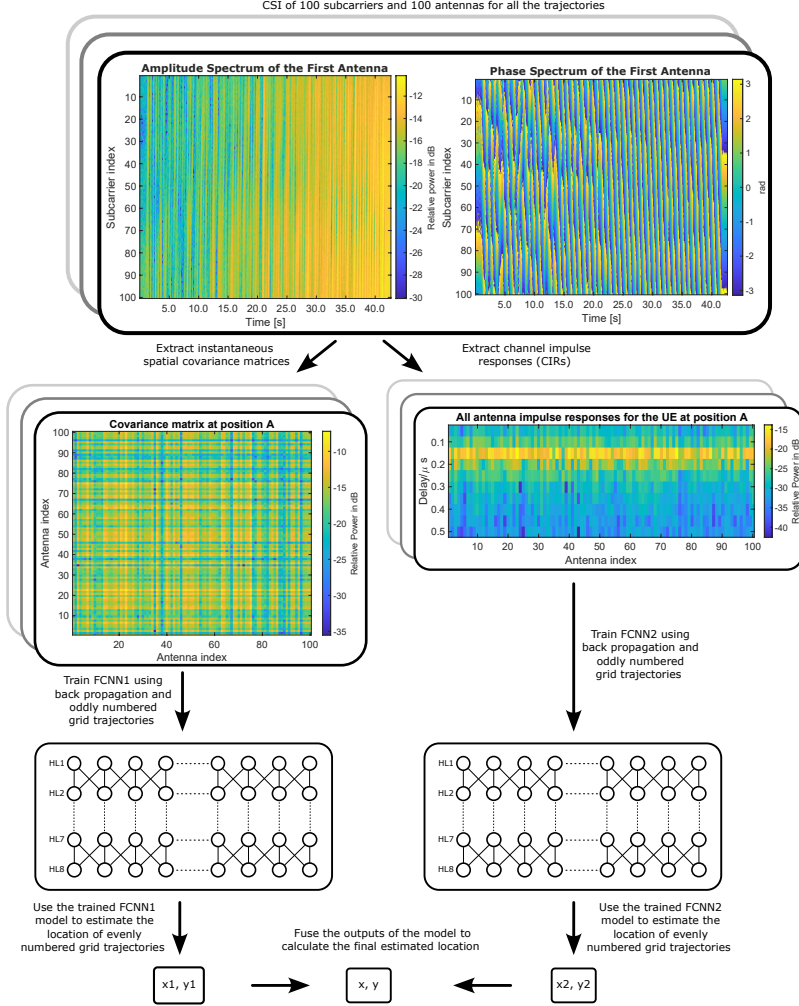


Fig. 3. The ML-based algorithm [6] pipeline for the radio system. The fully connected neural network 1 (FCNN1) extracts angular information and FCNN2 extracts delay information through the respective hidden layers (HL). "Position A" is an example location in the measurement area.

- 3) For each time instant, find and use the most suitable TDOA estimates to calculate an initial location of the microphones and the sound source (using minimal solvers, multilateration, and RANSAC to arbitrate among solutions).
- 4) Use the (initial) location of the microphones and TDOA estimates in a robust multilateration method to calculate a final estimate of the sound source location.

Since GCC-PHAT uses the correlation between microphones to estimate TDOA between each pair, step 2 could be modified to match towards a reference signal, e.g. a known chirp pattern. Such use of prior knowledge is not a part of the current implementation [28]. If the microphone locations are known, step 3

is not required, and finding initial source locations is in step 4. Under certain conditions, e.g., uncorrelated Gaussian noise stochasticity, the sound source location can be estimated with a few centimeters accuracy [28]. Thus, the SFS2 algorithm is selected as the benchmark method for audio data, using known microphone locations and smoothing. Fig. 4 is an overview of how the algorithm is used with the LuViRA dataset.

IV. LUVIRA DATASET

In this section, we describe the realistic measurement data evaluated using the selected algorithms described in the previous section. The LuViRA dataset [3] includes sensor data recorded by a camera, a massive MIMO antenna array, and

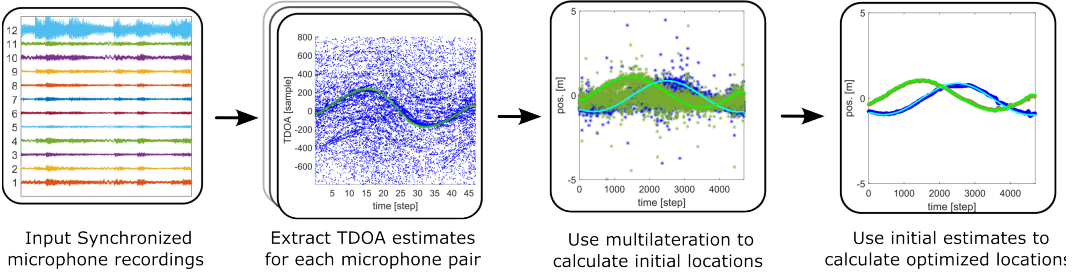


Fig. 4. The SFS2 algorithm [7] pipeline for the audio system using RC2 trajectory.

TABLE I
RANDOM TRAJECTORY ABBREVIATIONS

Trajectory Name	Alias	Trajectory Name	Alias
Random rect1	RR1	Random manual people1	RM1
Random rect2	RR2	Random manual people2	RM2
Random circle1	RC1	Random manual3	RM3
Random circle2	RC2	Random manual4	RM4
		Random manual5	RM5
		Random manual people6	RM6

12 microphones. The camera, a sound source (speaker), and a single-antenna radio transmitter are mounted to a robot so that only a planar movement is involved. During the measurements, a motion capture (mocap) system tracks these objects within 0.5 mm accuracy with the help of markers, placed on the target objects. An overview of the measurement environment can be seen in Fig. 1. The camera, speaker, and antenna are placed at different locations on the robot, as shown in Fig. 1.

The dataset is divided into two parts based on the movement pattern of the robot: “grid” and “random” data. The “grid” data includes trajectories where the robot moves forward and backward, facing the same direction, in a measurement area of $4.2 \times 2.5 \text{ m}^2$. Please note that the relatively small size of the measured area is a limitation of this dataset, which might affect the comparison of different methods.

There are 75 “grid” trajectories in total, approximately 5 cm apart. The speed of the robot is 0.1 m/s for these trajectories. The “random” data includes circular, diagonal, and rectangular trajectories, as well as arbitrary trajectories. Abbreviations used for each trajectory are given in Table I. In RM1, RM2, and RM6 trajectories, people and objects are moving around. Next, the vision, radio, and audio systems used to capture the dataset are briefly introduced.

1) Vision System: The Intel Realsense D435i depth camera is used for obtaining RGB images (30 fps) and depth maps (15 fps) from the environment. The Grid110 and selected “random” trajectories are used as input to RGB-D ORB-SLAM3 for evaluating localization accuracy. The selected trajectories include the “Random manual” trajectories, where the robot makes sharp movements and there are dynamic features in the environment. Automated evaluation tools [29] are used to calculate the absolute trajectory error. The evaluation tool aligns the ground truth labels and the estimated trajectory by

solving for rigid transformation between trajectories. The field-of-view and frame rate of the provided depth maps and RGB images are aligned.

2) Radio System: The measurement setup includes a single user equipment (UE) with one antenna and the LuMaMi testbed with 100 antennas. Both the UE and the LuMaMi testbed have 20 MHz bandwidth and 3.7 GHz center frequency. In the LuViRA dataset, a cable is used to synchronize the UE and LuMaMi testbed to avoid possible signal drops. The “grid” data, which consists of 75 trajectories, is used to train and test the ML-based algorithm since localization estimates of the target object in the “random” trajectories are outside the measurement area defined by the dataset. In the reference paper of the selected radio-based ML algorithm [6], both the training and testing data are randomly selected from all “grid” trajectories, leading to training data that encompasses randomly chosen points from each trajectory. Instead, we separate training and testing trajectories, so we can compare the localization accuracy of each sensor on a completely unused trajectory. To conduct a balanced comparison with localization methods based on other sensors, odd numbered trajectories (number 1, 3, 5, and so on) are used to train the ML algorithm, whereas the accuracy of the algorithm is tested on the evenly numbered trajectories (2, 4, etc.).

3) Audio System: The audio data include recordings from 12 microphones. One of these microphones is placed on the robot to record the sound directly from the speaker. This reference microphone is not used in the scope of this paper. The post-processed recordings are single-channel WAV-files and have a sample rate of 96 kHz and 16-bit resolution. The SFS2 algorithm is evaluated on the Grid110, RR1, and RR2 trajectories where the speaker plays a chirping sound and for the rest of the “random” trajectories where the speaker plays different music clips. In these recordings, there is an audible fan noise from the LuMaMi testbed. The signal-to-noise ratio (SNR) varies naturally as the sound source moves around, and microphone proximity to LuMaMi, but the signal is always audible. Considering that ground truth system calibration might fluctuate between trajectories, the microphone locations are assumed to be fixed for each trajectory separately.

V. RESULTS AND DISCUSSION

Table II summarises the results when applying the selected localization algorithms to the realistic dataset. Some of the

TABLE II
LOCALIZATION RESULTS ON SELECTED TRAJECTORIES

	Vision	Radio	Audio
Sensor	camera	antenna array	microphones
Localization algorithm	ORB - SLAM3 [5]	ML [6]	SFS2 [7]
Estimation rate ^a	15 Hz	100 Hz	100 Hz
Mean error, Grid110	6.5 cm	14 cm	127 cm ^b
SD error, Grid110	3.8 cm	13 cm	74 cm ^b
Mean error, RC1	5.7 cm	N/A ^c	15 cm
SD error, RC1	3.4 cm	N/A ^c	17 cm
Mean error, RM3	10 cm	N/A ^c	6.4 cm
SD error, RM3	7.2 cm	N/A ^c	4.9 cm

^aHow frequent location estimates are completed in this benchmark.

^bTrajectories using chirp audio.

^cSee subsection V-B.

example trajectories can be seen in Fig. 5, 6, 7, 9 and 10. In this section, different aspects of the results are discussed.

A. Localization accuracy

The accuracy of the localization algorithms is calculated based on the mean and SD of the Euclidean distance between the estimated locations and ground truth labels. The vision, radio, and audio systems calculate 6D, 3D, and 3D locations, respectively. Since the orientation is only available for the vision sensor, it is ignored in the accuracy calculations and the objects are located at a constant height over the floor, so the produced 3D locations are projected on the 2D plane to allow comparison between systems. Prior knowledge of the environment affects the localization accuracy of all the sensors. ORB-SLAM3 assumes that the environment is previously unknown, whereas the ML algorithm requires extensive training to learn the radio features in the environment and the SFS2 algorithm assumes the microphone locations to be known. Please note that both ORB-SLAM3 and SFS2 are not fully deterministic algorithms, and exploring the effect of the random seed and its spread is out of the scope of this work.

The selected vision-based localization algorithm successfully localizes the target object in both the “grid” and “random” trajectories. To illustrate the results, we choose the Grid110 and the RC1 trajectories and show the corresponding results in Fig. 5. The localization accuracy of the ORB-SLAM3 algorithm is given in Table III for 11 trajectories.

TABLE III
PART OF THE RESULTS OF THE VISION SYSTEM

Trajectory Name	Mean (cm)	SD (cm)	Median (cm)
Grid110	5.6	2.5	5.4
RR1	9.4	4.0	8.6
RR2	6.1	2.5	5.7
RC1	5.7	3.4	4.6
RC2	5.5	3.7	4.4
RM1	29	12	24
RM2	56	30	50
RM3	10	7.2	8.2
RM4	12	6.4	10
RM5	14	5.2	13
RM6	50	24	51

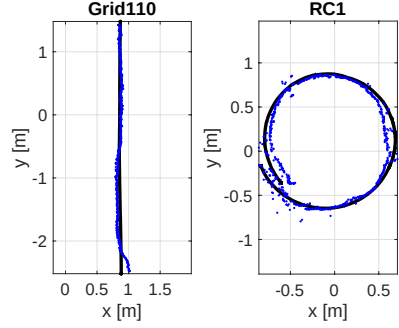


Fig. 5. Ground truth (black) and the result of the ORB-SLAM3 (blue) for Grid110 and RC1.

The radio localization algorithm results in an overall mean error of 13 cm and SD of 12 cm for the testing trajectories. The mean and SD of the trajectory with the performance closest to the overall mean error (medium estimated trajectory or Grid110), are presented in Table II. Fig. 6 gives an overview of all the testing trajectories and the estimation results of the best, worst, and medium trajectories based on the localization accuracy of the ML algorithm. The measurement area and the location of LuMaMi’s antenna array are also given in red dashed lines and the pink line, respectively, as a reference. As seen from Table II, the estimated locations are not valid when the model is trained using the “grid” trajectories and tested using the “random” trajectories. One possible explanation of these results is the sensitivity to dynamic environment, which is discussed in the subsection V-B.

The best and worst performances are achieved for Grid126 and Grid172 trajectories with a mean of 5 cm and 35 cm and SD of 4 cm and 11 cm, respectively. In Fig. 6, the test trajectory furthest away from the antenna array performs the worst while the performance declines when the robot moves away from the antenna array (in the y-direction) for the medium estimated trajectory. This can be explained by the increased distance from the base station and the number of training samples available [30]. On the other hand, the localization accuracy could be improved by adding an extended Kalman filter (EKF), particle filters, or other methods that apply tracking or smoothing [9] [31]. Such methods are used in both ORB-SLAM3 and SFS2 algorithms to correct errors based on the prior location of the tracked object.

A part of the results for the SFS2 algorithm is given in Table IV. The median localization error values are also presented in the table to illustrate the distribution of the results. E.g., the RC1 trajectory shown in Fig. 7, has 15 cm, 17 cm mean and SD error, whereas the median is only 7.2 cm, indicating the results are not normally distributed. As seen in Table IV, trajectories with chirp-based audio signal results in more than a meter localization error, while the music-based trajectories, such as RM3, can achieve up to 6.4 cm mean localization accuracy. The differences between chirp and music-based trajectories will be discussed in the next

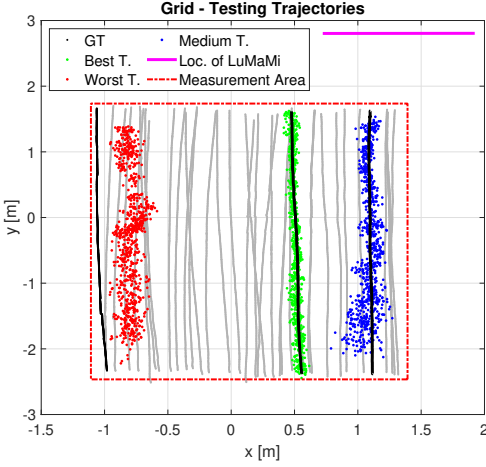


Fig. 6. Overview of the ground truth (gray stripes) and the result of the ML algorithm for the testing trajectories with best (Grid126, in the middle), worst (Grid172, left side), and medium (Grid110, right side) localization accuracy. Their ground truth trajectories are shown in black. The measurement area (dashed red lines) and the antenna array (horizontal pink line) are illustrated too.

TABLE IV
PART OF THE RESULTS OF THE AUDIO SYSTEM

Trajectory Name	Mean (cm)	SD (cm)	Median (cm)
Grid110 ^a	127	74	116
RR1 ^a	152	73	130
RR2 ^a	167	158	111
RC1	15	17	7.2
RC2	9.6	3.1	9.3
RM1	31	44	9.3
RM2	28	38	16
RM3	6.4	4.9	5.2
RM4	21	28	8.7
RM5	49	95	8.7
RM6	34	53	16

^aTrajectories using chirp audio.

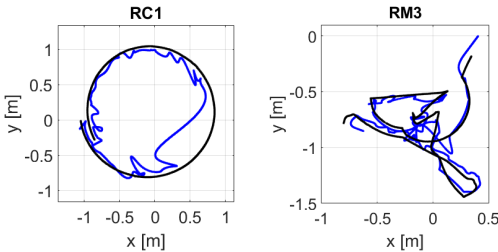


Fig. 7. Ground truth (black) and the result (blue) of SFS2 algorithm given known microphone locations for RC1 and RM3 trajectories.

subsection.

B. Sensitivity to dynamic environment

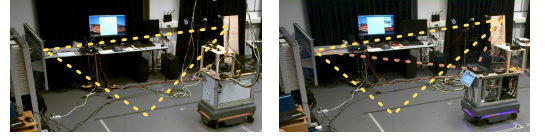


Fig. 8. Conceptual illustration of the change in the propagation environment (dashed, yellow) for different robot orientations in Grid122 (left image) and RC1 trajectories (right image). The difference is highlighted with orange color.

Changes in the propagation environment affect radio-based localization algorithms in terms of reliability. As mentioned above, the ML-based algorithm (in the current configuration) results in invalid estimates for the “random” trajectories. The main difference between the “grid” and the “random” data of the LuViRA dataset is the orientation of the antenna due to the movement of the robot. For example, the location of the robot is always in between the UE and LuMaMi for the “grid” trajectories while for the “random” trajectories, it can be on the side, changing the multipath components in the environment, which results in different radio channel responses for a given location. This change is illustrated in Fig. 8. Hence, we conclude that, when the “grid” data is used for training the selected ML algorithm, localizing the target object in the “random” data is very challenging due to the strong effect of antenna rotation on the channel characteristics.

On the vision system side, having static objects with distinct features in the environment is an important requirement for achieving high accuracy with the ORB-SLAM3 algorithm. If a room is empty and the walls are just white, the algorithm cannot extract features (since there are none) from the given frames and the accuracy of the algorithm drops significantly [32]. Thus, the measurement environment is decorated using posters, a chess board, etc., as seen in Fig. 1 and 2. Moreover, if the camera is moving at a very high speed with many rotations, the frame rate of the camera is not sufficient. E.g., the RM3, RM4, and RM5 trajectories in Table III, have sharp turns which result in a slight increase of the localization error compared to other trajectories that are not labeled “manual”. In contrast to the other sensors, increasing the frame rate to more than 60 fps is challenging due to the associated bandwidth, memory, and power requirements. The RM1, RM2, and RM6 trajectories also have non-static features which result in a sharp decline in the accuracy. The results of all the “Random manual” trajectories are shown in Fig. 9.

For the audio system, the direct path between the speakers and the microphones is obstructed for RM1, RM2, and RM6, which impacts the extraction of the TDOA features. Even though this impact is expected to be minimal due to the distribution of microphones, the results for these trajectories are worse than the RM3, RM4, and RM5 trajectories. Another explanation can be the other sound sources such as footsteps and occasional speech in the audio recordings which only exist in RM1, RM2, and RM6 trajectories. Moreover, RM6 has the music fading and stopping ahead of time, which results in detrimental signal loss, as can be seen in Fig. 10.

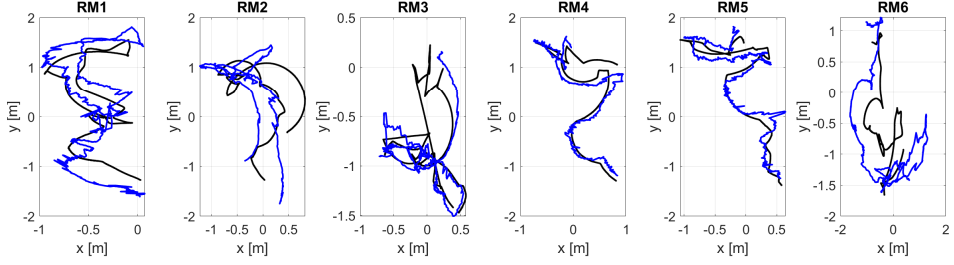


Fig. 9. Ground truth (black) and the result of the ORB-SLAM3 (blue) for the “Random manual” (RM) trajectories. The comparably worse tracking of RM1, RM2, and RM6 are the trajectories with non-static features or dynamic environment where people are moving around.

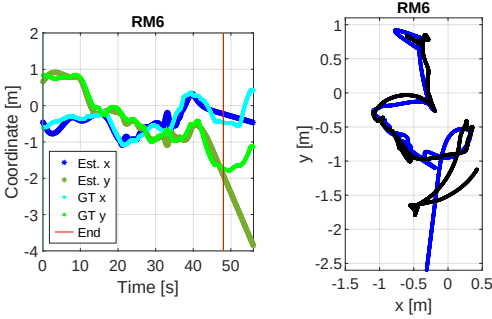


Fig. 10. The left image shows, over time, the estimated X (dark blue) and Y (dark green) coordinates, and ground truth X (light blue) and Y (light green). The vertical red line shows where the audio signal ends. The right image shows the ground truth (black) and the result (blue).

C. Sensitivity to Signal-to-Noise Ratio

The accuracy of vision-based localization algorithms is reduced drastically when there is little to no light (night or darkness), interference with other strong light sources such as the sun, or when the camera is blocked by other objects [33]. To simulate low-light conditions, where the thermal noise in the camera has a greater effect on the image, a zero-mean, Gaussian white noise is added to the RGB images obtained by the camera. Further investigation on adding noise to the depth maps is left as future work. These results are compared to the radio system, where the same noise level is added to both training and testing data with SNR of 10 dB and 20 dB. The noise is applied before any processing is made, including the alignment of images and training of the neural network. The ORB-SLAM3 algorithm is highly affected by the low SNR even when the noise is only added to the RGB images. The radio-based localization algorithm is more robust against low SNR resulting from added white Gaussian noise compared to the vision system. One possible explanation for the difference between these sensors is the redundancy in the 100-antenna matrix and 100-subcarrier channel state information provided by the LuMaMi testbed. Another explanation is how these algorithms process the given inputs. The ORB-SLAM3 algorithm processes its inputs (frames) as a whole and extracts information (such as ORB features) while the

ML-based algorithm learns possible changes in SNR and uses the strength and phase of the radio signal directly.

In many realistic audio-based localization systems, sound sources such as speech, do not transmit continuous signals and may have a limited frequency range. The SFS2 algorithm does not use prior knowledge about the sound signal, instead, the correlation between the microphone pairs is used as a way to separate signal from noise. This works better for wideband signals, such as music, than narrowband signals where a larger share of the spectrum can be dominated by correlated or uncorrelated noise.

TABLE V
RESULTS OF ADDING GAUSSIAN NOISE TO VISION AND RADIO SENSORS FOR GRID110 TRAJECTORY

Sensor	SNR (dB)	Mean (cm)	SD (cm)
Vision	N/A ^a	6.5	3.8
	10	89	48
	20	22	10
Radio	N/A ^a	14	13
	10	15	14
	20	14	13

^aBaseline (no added noise)

Table IV shows that the selected audio system fails for chirp-based trajectories. Fig. 11 shows an example of extracted TDOA estimates for Grid110, with a ground truth overlay. The periodic clusters at zero TDOA may correspond to the silent time periods, but there is no pattern along the ground truth labels. Since the algorithm has no access to the ground truth, it randomly bases most estimates on the noise. This can be compared to a successful localization example, such as Fig. 4 where step 2 extracts well-estimated TDOA features. From the spectrogram analysis of the chirp audio given in Fig. 11, the signal is estimated to be a sine wave rising from 400 Hz to 1400 Hz during 200 ms. It repeats twice per second, which leaves 60 % of the time, silent where the audio system relies on interpolation. In a reverberant room, the challenge is even greater, as echoes may be interpreted as signals and result in misleading estimates. The Fig. 8 illustration of multi-path arrival also in this case. The many reflection surfaces around the speaker result in an ever-present source of correlated noise.

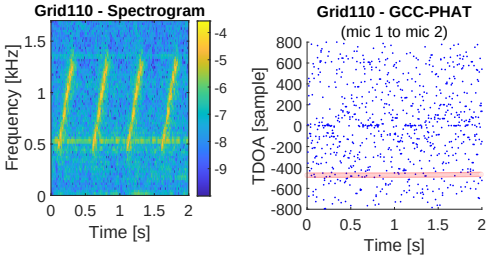


Fig. 11. The left image shows the power spectrogram of the signal from microphone 1. The right image displays the estimated TDOA (blue dots) and the expected TDOA based on the ground truth (red area).

D. Calibration requirements

Each of the discussed sensors requires different intrinsic and extrinsic calibration and time synchronization of different components in the system. Cameras used in the ORB-SLAM3 algorithm need to be calibrated once before the camera starts recording. This calibration configures the intrinsic parameters, such as focal length and distortion of the lens, which are required in the selected algorithm.

The LuMaMi testbed performs the reciprocity calibration automatically. However, the characterization of the antenna array elements is unknown, which is not an obstacle for ML-based algorithms as long as coherency is maintained. Moreover, a cable is used to synchronize the UE and LuMaMi testbed, which is usually not feasible in real-life scenarios.

For the audio system, the synchronization between microphones needs to be very accurate, thus they are connected to a single sound card via cables. In some cases, the smoothing forces the estimates to stray toward an outlier that is perceived as confident. There is a bias around $(-0.5, -0.5)$ which might be related to the position of the fans. Trajectories that pass near this point might achieve better accuracy, partly because these outliers happen to result in a low error and partly because the smoothing can correct some of them.

The changes in humidity and temperature of the room might affect all the sensors differently. Two works examining the effect of temperature change on cameras are presented in [34] [35]. The temperature impacts the circuitry inside the radio system. According to our analysis, the temperature change given in the LuViRA dataset does not have a quantitative impact on the vision and radio systems. On the other hand, the SFS2 algorithm calculates the distance using the speed of sound, which varies with temperature. According to the description of the LuViRA dataset, the temperature changed from 22°C to 28°C which corresponds to a 3.6 m/s change in the speed of sound. For reference, the speed of sound at the initial room temperature is 344 m/s . Thus, a 1% error in the speed of sound will have a below-centimeter impact in the given measurement area of $4.2 \times 2.5\text{ m}^2$.

The processed ground truth follows virtual points based on the markers. In reality, these points may have an offset against the target objects, e.g. the speaker. For the audio system, Fig. 7 may indicate that the ground truth labels are further away than the estimated trajectory. However, no bias is confirmed,

given the large SD. A similar bias could affect the other sensor estimates (e.g. Fig. 5), although with an unrelated bias since they track different points. These systems may compensate for such bias. The radio system is trained to predict the ground truth (including any consistent bias) and the vision system gives relative locations that are aligned to the ground truth. The audio system estimates absolute locations (relative to the microphones) without any reference to ground truth labels.

VI. CONCLUSION & FURTHER WORK

In this paper, we evaluated vision, radio, and audio based localization methods, using state-of-the-art algorithms for each sensor type and the LuViRA dataset. Aspects including localization accuracy, reliability, calibration requirements, and potential system complexity are discussed to analyze the advantages and limitations of using different sensors for indoor localization tasks. To sum up, each sensor has its advantages and limitations. It is challenging to find a sensor that is suitable for all situations. The vision-based localization algorithm can localize all the trajectories in the LuViRA dataset while the audio system has lower accuracy in trajectories with a chirp signal as the sound source and the radio-based localization algorithm cannot localize the “random” trajectories. However, when objects or people are moving around, the audio-based localization algorithm reaches higher accuracy and may be the most robust (as long as there is no additional sound in the environment from the movement). The current implementation of the radio-based localization algorithm cannot successfully localize the “random” data in the dataset but, it is more robust against low SNR conditions.

To leverage the strengths of each medium and overcome the weaknesses, a fusing strategy that jointly processes the information and increases algorithm robustness should also be developed. One of the challenges to tackle in such a scenario is to determine how to combine the data from different sensors. In LuViRA dataset, the vision data is captured from the onboard computer on the robot, whereas radio and audio signals are captured using an external MIMO antenna and microphone array. This could trigger a research problem regarding where the processing should take place, especially considering the processing capabilities of different platforms. One idea would be to conduct thorough power analyses of each localization algorithm and devise a strategy to offload or replace the most power-intensive components from the onboard computer.

ACKNOWLEDGMENT

The authors would like to thank Henrik Garde from Lund University Humanities Lab, Alexander Dürr and Volker Kruger from the Department of Computer Science, Martin Larsson from the Centre for Mathematical Sciences, Anders Robertsson from the Department of Automatic Control, and Michiel Sandra, Sirvan Abdollah Poor, Jesús Rodríguez Sánchez and Sara Willhammar from the Department of Electrical and Information Technology at Lund University for providing resources, technical support, and assistance in our project. We’re also grateful to the reviewers and editors who helped us with the paper. In Memoriam of Anders Robertsson.

REFERENCES

- [1] G. Kia, D. Plets, B. Van Herbruggen, E. De Poorter, and J. Talvitie, "Toward Seamless Localization: Situational Awareness Using UWB Wearable Systems and Convolutional Neural Networks," *IEEE Journal of Indoor and Seamless Positioning and Navigation*, vol. 1, pp. 12–25, 2023.
- [2] M. Säily *et al.*, "Positioning Technology Trends and Solutions Toward 6G," in *IEEE Inter. Symp. on Personal, Indoor and Mobile Radio Comm. (PIMRC)*, 2021.
- [3] I. Yaman *et al.*, "The LuViRA Dataset: Measurement Description," 2023. [Online]. Available: <https://arxiv.org/abs/2302.05309>
- [4] S. Malkowsky *et al.*, "The World's First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation," *IEEE Access*, 2017.
- [5] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," *IEEE Transactions on Robotics*, 2021.
- [6] G. Tian *et al.*, "High-Precision Machine-Learning Based Indoor Localization with Massive MIMO System," in *IEEE International Conf. on Communications (ICC)*, 2023.
- [7] K. Åström, M. Larsson, G. Flood, and M. Oskarsson, "Extension of Time-Difference-of-Arrival Self Calibration Solutions Using Robust Multilateration," in *European Sig. Proc. Conf. (EUSIPCO)*, 2021.
- [8] A. Kendall, M. K. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [9] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015.
- [10] Y. Tian *et al.*, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2022–2038, 2022.
- [11] R. Liu *et al.*, "Exploiting radio fingerprints for simultaneous localization and mapping," *IEEE Pervasive Computing*, vol. 22, no. 3, pp. 38–46, 2023.
- [12] H. Wymeersch *et al.*, "6G Radio Requirements to Support Integrated Communication, Localization, and Sensing," 2022. [Online]. Available: <https://arxiv.org/abs/2205.10783>
- [13] J. R. Sánchez, O. Edfors, and L. Liu, "Positioning for Distributed Large Intelligent Surfaces using Neural Network with Probabilistic Layer," in *2021 IEEE Globecom Workshops (GC Wkshps)*, 2021.
- [14] S. De Bast, A. P. Guevara, and S. Pollin, "CSI-based Positioning in Massive MIMO systems using Convolutional Neural Networks," 2019. [Online]. Available: <https://arxiv.org/abs/1911.11523>
- [15] J. Vieira, E. Leitingner, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Comm. (PIMRC)*, 2017.
- [16] M. Pollefeys and D. Nister, "Direct computation of sound and microphone locations from time-difference-of-arrival data," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.
- [17] S. Zhayida, F. Andersson, Y. Kuang, and K. Åström, "An automatic system for microphone self-localization using ambient sound," in *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2014.
- [18] X. Su, I. Ullah, X. Liu, and D. Choi, "A review of underwater localization techniques, algorithms, and challenges," *Journal of Sensors*, vol. 2020, no. 1, p. 6403161, 2020.
- [19] Y. Liu *et al.*, "Simultaneous localization and mapping related datasets: A comprehensive survey," *arXiv preprint arXiv:2102.04036*, 2021. [Online]. Available: <https://arxiv.org/abs/2102.04036>
- [20] K. Åström, M. Larsson, G. Flood, and M. Oskarsson, "Extension of Time-Difference-of-Arrival Self Calibration Solutions Using Robust Multilateration," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 870–874.
- [21] T. Ishihara, K. M. Kitani, C. Asakawa, and M. Hirose, "Deep radio-visual localization," in *IEEE Winter Conf. App. of Computer Vision*, 2018.
- [22] A. Chau, K. Sekiguchi, A. A. Nugraha, K. Yoshii, and K. Funakoshi, "Audio-Visual SLAM towards Human Tracking and Human-Robot Interaction in Indoor Environments," in *28th IEEE International Conference on Robot and Human Interactive Communication*, 2019.
- [23] T. Yang, A. Cabani, and H. Chafouk, "A survey of recent indoor localization scenarios and methodologies," *Sensors*, 2021.
- [24] X. Gao and T. Zhang, "Introduction to Visual SLAM: From Theory to Practice," *Springer Nature Singapore*, 2021.
- [25] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, 2019.
- [26] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-Based Multi-Antenna and Multi-Point Indoor Positioning Using Probability Fusion," *IEEE Transactions on Wireless Communications*, 2022.
- [27] S. Thrun, "Affine structure from sound," in *Int. Conf. on Neural Information Processing Systems*. MIT Press, 2005, pp. 1353–1360.
- [28] M. Larsson, G. Flood, M. Oskarsson, and K. Åström, "Fast and robust stratified self-calibration using time-difference-of-arrival measurements," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2012.
- [30] G. Tian *et al.*, "Deep-Learning-Based High-Precision Localization With Massive MIMO," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, 2024.
- [31] P. Kowalski and R. Smyk, "Review and comparison of smoothing algorithms for one-dimensional data noise reduction," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018.
- [32] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up slam: Semantic monocular plane slam for low-texture environments," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [33] W. Cheng, Y. Zhang, Y. Qi, J. Liu, and F. Liu, "A fast global adaptive solution to low-light images enhancement in visual SLAM," in *IEEE Inter. Conference on Computer and Communications (ICCC)*, 2020.
- [34] M. Elias, A. Eltner, F. Liebold, and H.-G. Maas, "Assessing the Influence of Temperature Changes on the Geometric Stability of Smartphone- and Raspberry Pi Cameras," *Sensors*, 2020.
- [35] M. Adamczyk, P. Liberadzki, and R. Sitnik, "Temperature compensation method for digital cameras in 2d and 3d measurement applications," *Sensors*, vol. 18, no. 11, 2018. [Online]. Available: <https://www.mdpi.com/1424-8220/18/11/3685>



İlayda Yaman (Student Member, IEEE) completed her bachelor's degree at Istanbul Technical University in 2018 and her master's degree in Embedded Electronics Engineering at Lund University in 2020. During her master's degree, she received the LU Global Scholarship. Currently, she is a Ph.D. student at Lund University (Main Supervisor: Liang Liu). Her current research area is low-power ML hardware for vision and radio based systems.



Guoda Tian (Student Member, IEEE) finished his bachelor's degree in automation and control engineering at Northeastern University in 2016 and his master's degree in wireless communication at Lund University in 2018 where he received the LU Global Scholarship. Currently, he is a Ph.D. student at Lund University (Main Supervisor: Fredrik Tufvesson). His research topic is applying machine learning to wireless localization, sensing, and communication systems.



Erik Tegler (Student Member, IEEE) completed his master's degree in Engineering Physics at Lund University in 2021. He is currently a third-year Ph.D. student at Lund University under the supervision of Kalle Åström, Magnus Oskarsson, Bo Bernhardsson and Fredrik Tufvesson. His current research area is machine learning and its application in localization systems.



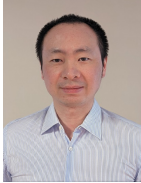
Jens Gulin (Student Member, IEEE) received his Master's degree in Computer Science and Engineering from Lund University in 2001. After many years in the mobile communication industry, the WASP initiative gave an opportunity to reconnect with the university to pursue a Ph.D. (Main Supervisor: Kalle Åström) while employed by Sony in Lund. His research interest is systems able to learn from little data, here specifically for audio localization.



Steffen Malkowsky received his bachelor's degree in electrical engineering and information technology from Pforzheim University in 2011, his master's degree in electronic design from Lund University, in 2013, and his Ph.D. degree from Lund University, in 2019. Between 2019 and 2023, he worked as a Post-Doctoral Researcher at Lund University. Currently, he is the founder and CTO of Inceptiontron.



Nikhil Challa (Student Member, IEEE) completed his bachelor's degree at Visvesvaraya National Institute of Technology in 2008 and his master's degree in Electrical Engineering at Virginia Tech in 2011. Afterwards, he worked at Qualcomm for 10 years. In 2023, he completed another master's program in Machine Learning, Systems and Control at Lund University. He is currently an ML engineer at Scania.



Liang Liu (Member, IEEE) received his Ph.D. degree from Fudan University in 2010. He joined Lund University as post-doc. Since 2024, he has been a Professor at Lund University. His research interests include wireless systems and digital integrated circuit design. He is a member of the Technical Committee of VLSI Systems and Applications and CAS for Communications of the IEEE Circuit and Systems Society.



Fredrik Tufvesson (Fellow, IEEE) received the Ph.D. degree from Lund University in 2000. He is a professor of radio systems at the Department of Electrical and Information Technology, Lund University. His main research interest is the interplay between the radio channel and the rest of the communication system with various applications in 5G/B5G systems, such as massive MIMO, mmWave, vehicular communication, and radio-based positioning.



Ove Edfors (Senior Member, IEEE) is Professor of Radio Systems at the Department of Electrical and Information Technology, Lund University, Sweden. His research interests include statistical signal processing and low complexity algorithms with applications in wireless communications. In the context of Massive MIMO and large intelligent surfaces, his main research focus is on how realistic propagation characteristics influence system performance and baseband processing complexity.



Kalle Åström (Senior Member, IEEE) received his B.Sc. in Mathematics in 1990, M.Sc. degree in Engineering Physics in 1991 and Ph.D. in Mathematics in 1996 from Lund University, Sweden. His thesis was awarded Best Nordic Ph.D. Thesis in pattern recognition and image analysis 1995-1996 at the Scandinavian Conference in Image Analysis, 1997. Currently, he is a professor at the Centre for Mathematical Sciences, LU.

Paper III

Paper III

Reproduced, with permission, from: IEEE

G. TIAN, I. YAMAN, M. SANDRA, X. CAI, L. LIU AND F. TUFVESSON, "High-Precision Machine-Learning Based Indoor Localization with Massive MIMO System," *IEEE International Conference on Communications (ICC)*, Rome, Italy, Oct 2023, pp. 3690-3695, doi:10.1109/ICC45041.2023.10278664.

High-Precision Machine-Learning Based Indoor Localization with Massive MIMO System

Guoda Tian, Ilayda Yaman, Michiel Sandra, Xuesong Cai, Liang Liu, Fredrik Tufvesson
Dept. of Electrical and Information Technology, Lund University, Sweden
email: firstname.lastname@eit.lth.se

Abstract—High-precision cellular-based localization is one of the key technologies for next-generation communication systems. In this paper, we investigate the potential of applying machine learning (ML) to a massive multiple-input multiple-output (MIMO) system to enhance localization accuracy. We analyze a new ML-based localization pipeline that has two parallel fully connected neural networks (FCNN). The first FCNN takes the instantaneous spatial covariance matrix to capture angular information, while the second FCNN takes the channel impulse responses to capture delay information. We fuse the estimated coordinates of these two FCNNs for further accuracy improvement. To test the localization algorithm, we performed an indoor measurement campaign with a massive MIMO testbed at 3.7 GHz. In the measured scenario, the proposed pipeline can achieve centimeter-level accuracy by combining delay and angular information.

Index Terms—massive MIMO, machine learning, radio-based localization, measurement, pre-processing

I. INTRODUCTION

Cellular-based localization opens the door for numerous location-aware applications such as navigation, autonomous driving, robot mapping, intelligent transportation [1]–[5], etc. Since cellular-based localization has a crucial function in those applications, efforts have been made by standardization organizations, such as the third generation partnership project (3GPP) to further investigate this technology. Motivated by the growing demand for high-quality localization services, cellular-based localization technology has rapidly evolved in the last ten years. In the current 3GPP specification (Release 17 and 18), cellular-based localization is one of the fundamental features of the new radio standard.

Traditional cellular-based localization methods include time-of-arrival (ToA), time-of-flight (ToF), time-difference-of-arrival (TDoA), angle-of-arrival (AoA) based methods [6]. These methods estimate the user equipment (UE) coordinates directly based on relevant propagation channel characteristics. To further enhance localization precision, advanced radio technologies such as ultra-wideband (UWB) [7] or massive

multiple-input multiple-output (MIMO) [8]–[10] have been considered. A UWB system can inherently capture the propagation channels with high delay resolution. Similarly, by scaling up the number of antennas at the base station (BS) side, one can achieve a higher angular resolution and thus better AoA estimation quality. Therefore, massive MIMO enables excellent localization accuracy, even though available bandwidth is limited [8]. Despite the good positioning accuracy delivered by traditional localization algorithms, it is still challenging to use those advanced signal processing techniques because of the computational complexity of the algorithms [6]. In addition, AoA estimation algorithms usually require array calibration. Since radio-based localization is in itself a regression task (from channel state information to UE positions), data-driven algorithms such as machine learning (ML) can also be utilized [11]–[19]. The performance of ML-based localization tasks hinges to large extent on two factors: selecting appropriate channel fingerprints as training data and choosing suitable learning algorithms. Training data can be either the raw transfer function itself [11], [12], or other channel fingerprints such as received signal strength, power delay profile (PDP), angle-delay spectrum, etc., [13]–[17]. Several ML-based localization algorithms can be found in literature, which are categorized into the Kernel family [14], the neural network family [11]–[13], [15]–[18] and the Bayesian family [19].

Applying ML to massive MIMO systems for localization purposes is a promising yet immature research topic since it is challenging to process the rich spatial channel information captured by a massive antenna array. Early research work [15] proposed an approach by first extracting the delay-angle spectrum as a channel fingerprint and feeding this fingerprint to a convolutional neural network (CNN). This work was validated by simulation. The work in [18] proposed an auto-encoder-based calibration algorithm to first calibrate the antenna array before estimating the AoA. Measurement results show that this approach requires fewer training samples.

In this paper, we analyze a new ML-based localization algorithm for massive MIMO systems and we evaluate our algorithm using real measurement data. The first step of the algorithm is to extract both the instantaneous spatial covariance matrix and channel impulse responses (CIRs) of all antennas as fingerprints. These fingerprints contain channel information in both the angle and the delay domains. Calibration of

This work has been supported by Ericsson AB, Horizon Europe Framework Programme under the Marie Skłodowska-Curie grant agreement No., 101059091, the Swedish Research Council (Grant No. 2022-04691), and the strategic research area ELLIT, Excellence Center at Linköping — Lund in Information Technology. The authors would like to thank Lund University Humanities Lab, and the LTH robotic lab for the help with equipment. Special thanks to Henrik Garde, Alexander Dürr, Steffen Malkowsky, Sara Willhammar and Sirvan Abdollah Poor, Nikhil Challa, and Ove Edfors for their assistance throughout the measurements.

all RF chains is not necessary for the generation of those fingerprints. We then feed the fingerprints to two separate processing branches consisting of fully connected neural networks (FCNN), to estimate independent UE coordinates. The measurements indicate that those two processing branches deliver similar localization performance and their positioning errors have a weak correlation, given that we have enough training samples. Therefore, we can achieve an accuracy gain by averaging those two coordinates compared to only using the raw transfer function itself. In addition, the measurement results illustrate that pre-processing is crucial to improve localization accuracy.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a single-input multiple-output (SIMO) orthogonal frequency-division multiplexing (OFDM) system, where signals are transmitted by a single-antenna UE to an M -antenna BS. At the BS side, each antenna is connected to an individual radio frequency (RF) chain so that the signals from antenna can be processed coherently. Furthermore, we consider an indoor scenario where the UE moves at a normal walking speed, and the 2-D coordinates of UE at position $\mathbf{p}_i \in \mathbb{R}^2$ are denoted as $\mathbf{p}_i(x_i, y_i)$. We can then write the received channel transfer function matrix $\mathbf{Y}_{p_i} \in \mathbb{C}^{M \times N}$ with respect to all subcarriers at position $\mathbf{p}_i$ as,

$$\mathbf{Y}_{p_i} = \mathbf{H}_{p_i} \odot \mathbf{\Gamma} + \mathbf{N}, \quad (1)$$

where the propagation channel between UE and BS at position $\mathbf{p}_i$ is denoted as $\mathbf{H}_{p_i} \in \mathbb{C}^{M \times N}$. $\mathbf{\Gamma} \in \mathbb{C}^{M \times N}$ collects all coefficients which contain amplitudes and phase drifts with respect to all M RF chains and N sub-carriers, while $\mathbf{N} \in \mathbb{C}^{M \times N}$ illustrates an additive noise matrix. In addition, $\odot$ represents the Hadamard product. While the UE moves, we capture in total $\mathcal{T}$ different channel snapshots. Thus, $\mathcal{T}$ different channel matrices $\mathbf{Y}_{p_i}$ and their corresponding UE coordinates $\mathbf{p}_i(x_i, y_i)$ are recorded. Our target is to establish a functional relationship between $\mathbf{Y}_{p_i}$ and coordinates $\mathbf{p}_i$.

III. PROPOSED LOCALIZATION METHOD

First, we calculate instantaneous spatial covariance matrices as well as truncated CIRs as two separate fingerprints to reflect angular properties and delay properties, respectively. We then train two neural networks, which learn functional relationships between those two extracted fingerprints and their corresponding 2-D UE coordinates. As a final step, the two estimated UE coordinates are averaged to obtain the final estimated UE positions.

A. Generate Fingerprint

The performance of ML-based localization algorithms can be significantly enhanced if adequate channel fingerprints are extracted from the raw received channel matrices (1). In accordance with (1), it is a non-trivial task to directly attain calibrated fingerprints such as angle spectrum because of the influence of $\mathbf{\Gamma}$. Considering this, we first compute the instantaneous spatial correlation matrices $\mathbf{C}_{Y,p_i} \in \mathbb{C}^{M \times M}$, by correlating across all subcarriers as

$$\mathbf{C}_{Y,p_i} = \mathbf{Y}_{p_i} \mathbf{Y}_{p_i}^H. \quad (2)$$

Note that $\mathbf{C}_{Y,p_i}$ is a Hermitian matrix according to its definition. Therefore, the information embedded in the upper minor diagonal elements is identical to the information of the lower minor diagonal elements. To reduce algorithm complexity, we define a matrix $\hat{\mathbf{C}}_{Y,p_i} \in \mathbb{R}^{M \times M}$ and a vector $\hat{\mathbf{c}}_{Y,p_i} \in \mathbb{R}^{M^2}$ as

$$\begin{aligned} \hat{\mathbf{C}}_{Y,p_i} &= \text{ltril}\{\Re[\mathbf{C}_{Y,p_i}]\} + \text{sltril}\{\Im[\mathbf{C}_{Y,p_i}]\} \\ \hat{\mathbf{c}}_{Y,p_i} &= \text{vec}\{\hat{\mathbf{C}}_{Y,p_i}\}, \end{aligned} \quad (3)$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ represent real and imaginary parts of a matrix. In addition, $\text{ltril}\{\cdot\}$ denotes a matrix operation that replaces all the values above the diagonal with zero and keeps the rest elements, while $\text{sltril}\{\cdot\}$ maintains all elements below the diagonal and replaces all other elements (including the diagonal) as zero, and the $\text{vec}\{\cdot\}$ operation vectorizes a matrix as a vector.

Channel information in the delay domain is not presented in $\hat{\mathbf{c}}_{Y,p_i}$ while this information is still useful for enhancing positioning accuracy. Therefore, we generate a CIR matrix $\mathbf{D} \in \mathbb{C}^{M \times L}$ by performing inverse discrete Fourier transform along each row of $\mathbf{Y}_{p_i}$ and then selecting the first L delay bins. We then define a vector $\mathbf{d} \in \mathbb{R}^{2ML}$ that contains all elements of $\mathbf{D}$. Specifically, $\mathbf{d} = [\text{vec}\{\Re(\mathbf{D})\}^T, \text{vec}\{\Im(\mathbf{D})\}^T]^T$. Both vectors $\mathbf{d}$ and $\hat{\mathbf{c}}_{Y,p_i}$ are utilized by our proposed ML-based localization pipeline to estimate UE positions.

B. Proposed processing pipeline

We use a supervised learning pipeline based on pure FCNN for indoor localization. The framework of our pipeline is illustrated in Fig. 1.

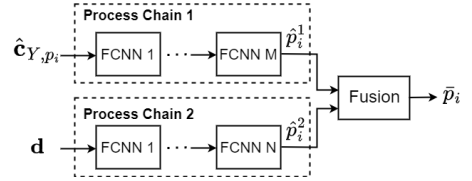


Fig. 1: Proposed localization pipeline.

The pipeline has two separate sub-chains and each chain individually learns the functional relationship between its own input channel fingerprint and UE coordinates. During the training phase, the upper chain takes fingerprint $\hat{\mathbf{c}}_{Y,p_i}$ as input and generates $\hat{\mathbf{p}}_i^1 \in \mathbb{R}^2$ as the estimated UE coordinates. In addition, the lower processing chain utilizes fingerprint $\mathbf{d}$ and generates instead $\hat{\mathbf{p}}_i^2 \in \mathbb{R}^2$ as estimated UE coordinates. We define $\mathbf{f}_1(\cdot)$ and $\mathbf{f}_2(\cdot)$ as the functions to represent the upper and lower FCNN-based processing chain, respectively, while their individual hyperparameters are contained in vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. In addition, all available training datasets are represented by Ω while the UE groundtruth label is denoted as a 2-D vector $\mathbf{p}_i$. Based on this information, we can express two loss functions Ψ_1 and Ψ_2 , which measure the mean

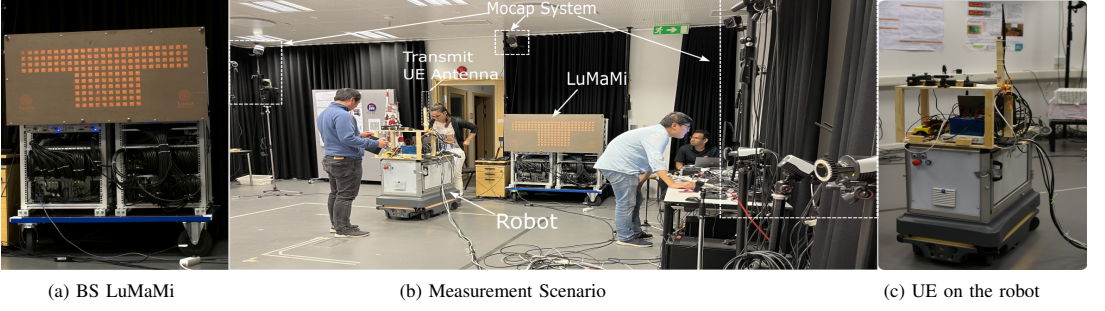


Fig. 2: Components in the indoor measurement campaign.

square errors (MSE) between the predicted labels $\hat{\mathbf{p}}_i^1$, $\hat{\mathbf{p}}_i^2$ and groundtruth $\mathbf{p}_i$, as

$$\Psi_1 = \sum_{\hat{\mathbf{c}}_{Y, \mathbf{p}_i} \in \Omega} \|\mathbf{f}_1(\hat{\mathbf{c}}_{Y, \mathbf{p}_i}, \boldsymbol{\theta}_1) - \mathbf{p}_i\|_F^2, \quad (4)$$

$$\Psi_2 = \sum_{\mathbf{d} \in \Omega} \|\mathbf{f}_2(\mathbf{d}, \boldsymbol{\theta}_2) - \mathbf{p}_i\|_F^2, \quad (5)$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a vector, which is used to measure the Euclidean distance between estimated positions and groundtruth labels. The hyperparameter vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are adjusted through a backpropagation procedure to minimize Ψ_1 and Ψ_2 during the training phase. Due to page limitations, we refrain from showing the full-length mathematical derivations, however, readers can find relevant materials in [20]. Next, we fuse two individual estimates $\hat{\mathbf{p}}_i^1$ and $\hat{\mathbf{p}}_i^2$. Considering the algorithm complexity, we combine $\hat{\mathbf{p}}_i^1$ and $\hat{\mathbf{p}}_i^2$ by taking their average as the final estimated UE coordinate $\bar{\mathbf{p}}_i$, i.e., $\bar{\mathbf{p}}_i = (\hat{\mathbf{p}}_i^1 + \hat{\mathbf{p}}_i^2)/2$. We show through the measurement results that an apparent localization accuracy enhancement can readily be achieved by this combination.

IV. MASSIVE MIMO MEASUREMENT CAMPAIGN

To verify the localization pipeline, an indoor measurement campaign was conducted inside Lund University Humanities Lab's motion capture (MoCap) studio. During this measurement campaign, the Lund University massive MIMO testbed (LuMaMi) [21] acted as a BS while a robot carried the UE equipped with a dipole antenna. While the robot was moving, the 3-D coordinates of the UE antenna were tracked by the MoCap system continually with millimeter accuracy. The environment in MoCap studio, LuMaMi, and the robot are shown in Fig. 2. We briefly introduce the measurement devices and scenarios, for a detailed description of the measurement campaign please see [22].

A. Measurement devices

During our measurement campaign, the UE moved along a pre-defined trajectory and the BS was equipped with 100 individual RF chains. To extract more information from the *azimuth* compared to the *elevation* domain, the upper 4×25 antenna elements are selected and each antenna is connected to an individual RF chain. Those antenna elements were separated by approximately half wavelength (4 cm) in both the vertical and horizontal directions. Antenna indexes are labeled

as follows: antennas on the first row are labeled in turn from 1 to 25 (from left to right), the second row from 26 to 50, the third row from 51 to 75 and the last row from 76 to 100. Both UE and BS operated at a carrier frequency of 3.7 GHz and occupied 20 MHz bandwidth. An OFDM transmission scheme was used, and $N = 100$ sub-carriers were allocated for uplink pilots. The UE transmitted pilot signals every 10 ms, which were then received by the BS to estimate the channel transfer function in (1). While the UE was moving, the coordinates of the UE antenna were continually computed by the MoCap system, which consisted of 18 networked high-speed infrared (IR) cameras. The MoCap system updated the coordinates of the interesting object (UE antenna) at a frame rate of 100 Hz, which was the same as the channel snapshot rate.

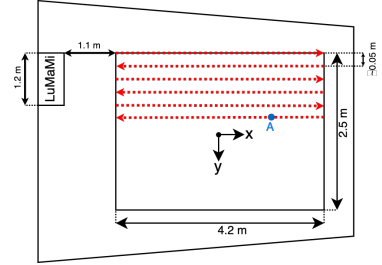


Fig. 3: Measurement arrangement in the MoCap studio and the selected point A (0.216, -1.335) in the effective area which is by $4.2 \text{ m} \times 2.5 \text{ m}$. The red dotted arrow shows the trajectory of the UE.

B. Measurement scenarios and Procedures

The measurement route consisted of 75 parallel UE trajectories, as illustrated in Fig. 3, which resulted in 75 individual channel measurements. For the i -th trajectory, $\mathcal{T}_i$ channel snapshots were recorded. We denote a complex tensor $\mathcal{A}_i \in \mathbb{C}^{\mathcal{T}_i \times M \times N}$ to store those channel snapshots. The whole measurement campaign collects $\mathcal{T} = \sum_i \mathcal{T}_i = 302500$ channel snapshots. We define a complex tensor $\mathcal{A}' \in \mathbb{C}^{\mathcal{T} \times M \times N}$ that combines all $\mathcal{A}_i$, which represents the overall dataset.

In the beginning of the measurement campaign, the robot was placed at the border of our measurement area. After finishing each trajectory, the robot was moved around 5 cm along the y -direction, while its orientation remained unchanged.

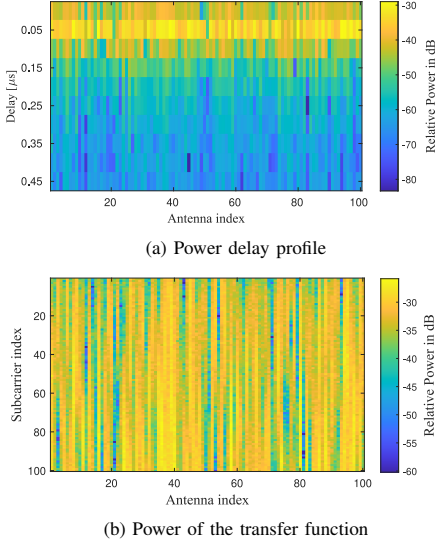


Fig. 4: Power delay profile and power of the transfer function at position A.

During each measurement, the robot moved straight in the x-direction, with a speed of 0.1 m/s. The same measurement procedure was repeated 75 times so that the whole measurement area was densely scanned with a resolution of around 5 cm in the y-direction and 1 mm in the x-direction. Throughout all measurements, all objects (except the robot) inside the room remained static and humans were not present in the direct vicinity.

V. RESULTS AND DISCUSSIONS

A. Measured channel response

To visualize the measured indoor massive MIMO channel characteristics, one UE position is selected and labeled in Fig. 3 as position A. We illustrate some key channel characteristics at this position, such as the PDP and the power over the different subcarriers for all antennas in Fig. 4. This figure shows typical indoor channel characteristics since the majority of power in the delay domain is concentrated in the first few delay bins. The same characteristics are also inherent in the transfer functions, with much higher variations of signal power among different antennas while the power among different subcarriers at every single antenna varies more smoothly and slowly. In this propagation environment, it is expected to have a higher frequency correlation compared to spatial correlation.

To display the change of channel response when the UE is moving, we select the first antenna and plot channel transfer functions with respect to all subcarriers along one robot moving trajectory in Fig. 5. As seen from the phase and amplitude, the UE remains static during the first and last second. We can also observe that received signal power

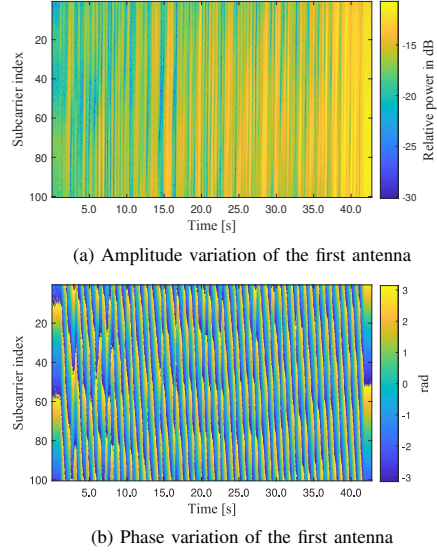


Fig. 5: Amplitude and phase of the transfer function at the first antenna along the UE trajectory.

increases, which indicates that the UE approaches the BS. In addition, we investigate the channel spatial correlation for different UE positions. We denote the received channel transfer function at position p_x as $\tilde{\mathbf{H}}_{p_x} \in \mathbb{C}^{M \times N}$ and we define a vector $\tilde{\mathbf{h}}_{p_x} \in \mathbb{C}^{MN}$ by vectorizing $\tilde{\mathbf{H}}_{p_x}$. Suppose that we consider in total T' positions, the spatial correlation coefficient $\rho(\Delta_d)$, is estimated as

$$\rho(\Delta_d) = \frac{1}{T'} \sum_{p_x} \left\{ \frac{\tilde{\mathbf{h}}_{p_x}^H \tilde{\mathbf{h}}_{p_x + \Delta_d}}{\sqrt{\|\tilde{\mathbf{h}}_{p_x}\|_F^2 \|\tilde{\mathbf{h}}_{p_x + \Delta_d}\|_F^2}} \right\}, \quad (6)$$

where Δ_d represents the separation distances of two UE positions. We select the first UE trajectory and plot the absolute value of $\rho(\Delta_d)$ with respect to Δ_d from 0 to 2λ according to (6) in Fig. 6, where λ represents the wavelength. As illustrated in Fig. 6, the spatial correlation is still strong if $\Delta_d \leq \frac{1}{8}\lambda$. However, a significantly weaker correlation is expected for larger separations.

B. Preprocessing and Network Parameter setting

After collecting the measurement results as tensor $\mathcal{A}'$, the entire tensor is normalized by multiplying a scaling factor $\alpha = \sqrt{\frac{TMN}{\|\mathcal{A}'\|_F^2}}$ so that the average power of each element of the new tensor $\tilde{\mathcal{A}} \in \mathbb{C}^{T \times M \times N} = \alpha \mathcal{A}'$ equals 1. As a next step, the total T samples are divided into a training set with $\mathcal{X}$ training samples and a testing set with $T - \mathcal{X}$ testing samples.

To compare our proposed method with the baseline, in total three networks are trained, corresponding to 3 different training inputs, namely, (1) the transfer functions $\mathbf{Y}_{p_i}$ itself (baseline), (2) the covariance matrix for the whole array, (3) the channel impulse response at first $L = 10$ delay bins. We illustrated the structures of those three FCNNs in Table I.

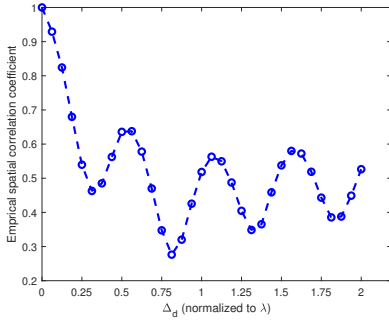


Fig. 6: Empirical spatial correlation function with respect to different UE separation distances. $\lambda \approx 8.1$ cm and Δ_d represents the physical distance, which is normalized to λ , between two positions.

TABLE I: Parameter settings of the proposed neural network.

	Network 1	Network 2	Network 3
Input layer	$2MN \times MN$	$M^2 \times M^2$	$2ML \times ML$
Hidden layer 1	$MN \times MN/2$	$M^2 \times M^2/2$	$ML \times ML$
Hidden layer 2	$MN/2 \times MN/4$	$M^2/2 \times M^2/4$	$ML \times ML$
Hidden layer 3	$MN/4 \times MN/4$	$M^2/4 \times M^2/4$	$ML \times 512$
Hidden layer 4	$MN/4 \times 1024$	$M^2/4 \times 1024$	512×512
Hidden layer 5	1024×512	1024×512	512×256
Hidden layer 6	512×128	512×128	256×128
Hidden layer 7	128×32	128×32	128×32
Hidden layer 8	32×4	32×4	32×4
Output Layer	4×2	4×2	4×2
Network Size	1.01 GB	349 MB	440 MB
Batch Size	64	64	64
Epoch	200	200	200
Complexity	$O(M^2N^2)$	$O(M^4)$	$O(M^2L^2)$

For all those three FCNNs, Leaky Relu is chosen as a non-linear activation function so that the gradient vanish problem [23] can be avoided. We initially set learning rates for all 3 networks as 0.0001 and they are reduced 20% every 10 epoch. As shown in Table I, compared to Network 1, 65% and 55% less storage resources are required by Network 2 and Network 3, respectively. We also investigate the algorithm complexities of those three networks. Compared to the baseline, a reduction of time complexity can be achieved by either using the covariance matrix or the truncated CIR, because $L \leq N$ and most of the commercial systems fulfill $M \leq N$. Furthermore, if we input both the covariance matrix and the truncated CIR, the time complexity is $O(M^4 + M^2L^2) = O(M^2 \max(M^2, L^2))$, which is still better than feeding the raw transfer function as the input.

C. Localization Result

We finally compare localization results of four localization methods, including *i)* training on the instantaneous covariance matrix, *ii)* training on the truncated CIRs, *iii)* training on the raw transfer functions, *iv)* average combining the estimated outputs of *i)* and *ii)*. Fig. 7 illustrates the localization performances of the four methods when 10% and 20% of all available channel data are selected for training and the rest samples are used for testing and evaluation purposes. The average distance between two adjacent training samples is smaller than $\frac{1}{8}\lambda$, i.e., the propagation channel is sufficiently

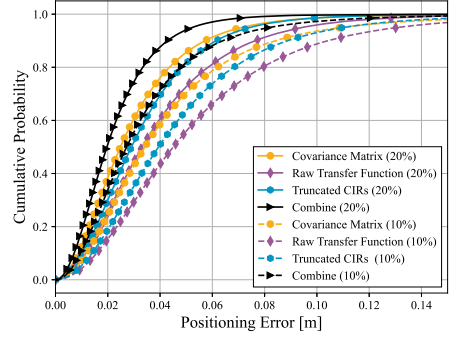


Fig. 7: Positioning error cumulative distribution function with 10% and 20% training data, respectively.

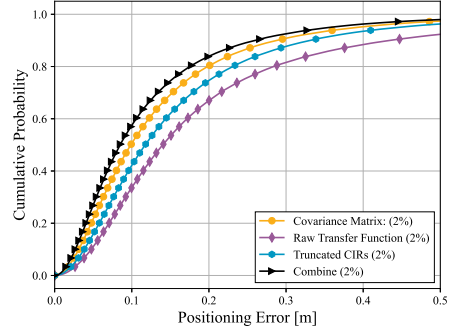


Fig. 8: Positioning error cumulative distribution function with 2% training data.

sampled. As shown in Fig. 7, although the truncated CIR and the raw transfer function contain the same channel information, localization performance can be improved by feeding truncated CIR into the ML pipeline. We postulate that it is because feeding truncated CIR into the pipeline can reduce the size of the network, which expedites the training process. Meanwhile, the signal-to-noise ratio (SNR) can be increased by truncating the tail part of CIRs, since this part contains only noise and no useful channel information. It can also be observed that the performance of using the instantaneous covariance matrix is better than that of the raw transfer function or truncated CIR, even though the delay information has been removed by averaging over all sub-carriers. The possible reasons are *i)* the channels are LoS-dominant and the bandwidth is too small to resolve multipath components (MPCs), and *ii)* the pre-processing helps the network to better exploit the angle information for localizing the UE. *iii)* The network size is also decreased, which facilitates the training process. Moreover, it can be seen that the combining strategy has the best performance. This is because both delay and angle information have been utilized for training, also with the decreased network size and the increased SNR. This shows that delay information is useful for positioning, even though the bandwidth is limited. We believe that localization accuracy

will be further enhanced in a rich-MPC scenario for a system with a larger bandwidth.

To explain the accuracy improvement from another perspective, we measure the correlation coefficient ρ between the positioning errors of localization methods *i*) and *ii*) as $\rho = \frac{\text{cov}(e_1, e_2)}{\sqrt{\text{var}(e_1)\text{var}(e_2)}}$, where e_1 and e_2 denote the Euclidean distances between $\hat{\mathbf{p}}_i^1$, $\hat{\mathbf{p}}_i^2$ and the groundtruth $\mathbf{p}_i$, respectively. The covariance between e_1 and e_2 is represented as $\text{cov}(e_1, e_2)$ while $\text{var}(e_1)$ represents the variance of e_1 . Measurement results show that when 20% of data is used for training, $\rho \approx 0.23$; when 10% of data is used, $\rho \approx 0.32$. Owing to the weak correlation between e_1 and e_2 , localization accuracy can be further improved by averaging $\hat{\mathbf{p}}_x^1$ and $\hat{\mathbf{p}}_x^2$. As shown in Fig. 8, if only 2% of channel samples are used for training, the localization accuracy of the four methods has the same ranking order as in Fig. 7. However, their performances decreased significantly. We conjecture that this training density (2%) is insufficient to represent the overall propagation channel properties since the average distance between adjacent training samples is larger than $\frac{1}{2}\lambda$, see Fig. 6. The degraded positioning accuracy can also be explained from the viewpoint of the sampling theorem, since lower-than-Nyquist sampling, i.e. too large distances between training samples, results in spectrum aliasing. Based on the above observations, we conclude that both data pre-processing and density of training samples play a major role in ML-based localization tasks.

VI. CONCLUSIONS

In this paper, we investigate the potential of ML for indoor localization with a massive MIMO system. We show through a measurement campaign that pre-processing plays an important role in enhancing the ML-based localization performance. We found that in our case the two parallel networks, using instantaneous covariance matrices and truncated impulse responses as input, respectively, give two uncorrelated position estimates with similar variance, which means that they can be averaged to achieve better localization accuracy. By our approach, centimeter-level accuracy can be achieved assuming we have enough training data and sufficient channel information. To achieve this accuracy, it is not required to calibrate the whole array. If only 2% of samples are used, median positioning accuracy is still comparable to a wavelength. In addition, it should be noted that the channel remained static and people were not present in this measurement campaign. In future works, we will investigate the influence of people and other dynamic scatters on localization accuracy, as well as other fingerprints not being so sensitive to sampling density of the training data.

REFERENCES

- [1] R. Whiton, "Cellular Localization for Autonomous Driving: A Function Pull Approach to Safety-Critical Wireless Localization," *IEEE Veh. Technol. Mag.*, pp. 2–11, 2022.
- [2] A. A. Abdallah, C.-S. Jao, Z. M. Kassas, and A. M. Shkel, "A Pedestrian Indoor Navigation System Using Deep-Learning-Aided Cellular Signals and ZUPT-Aided Foot-Mounted IMUs," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5188–5198, 2022.
- [3] M. Maaref and Z. M. Kassas, "Ground Vehicle Navigation in GNSS-Challenged Environments Using Signals of Opportunity and a Closed-Loop Map-Matching Approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2723–2738, 2020.
- [4] X. Mu, Y. Liu, L. Guo, J. Lin, and R. Schober, "Intelligent Reflecting Surface Enhanced Indoor Robot Path Planning: A Radio Map-Based Approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4732–4747, 2021.
- [5] R. Whiton, J. Chen, T. Johansson, and F. Tufvesson, "Urban Navigation with LTE using a Large Antenna Array and Machine Learning," in *IEEE 95th Vehicular Technology Conference (VTC2022-Spring)*, pp. 1–5.
- [6] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of Cellular Mobile Radio Localization Methods: From 1G to 5G," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 2, pp. 1124–1148, 2018.
- [7] E. Leitinger, P. Meissner, C. Rüdiger, G. Dumphart, and K. Witrisal, "Evaluation of Position-Related Information in Multipath Components for Indoor Positioning," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2313–2328, 2015.
- [8] X. Li, E. Leitinger, M. Oskarsson, K. Åström, and F. Tufvesson, "Massive MIMO-Based Localization and Mapping Exploiting Phase Information of Multipath Components," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4254–4267, 2019.
- [9] L. Lian, A. Liu, and V. K. N. Lau, "User Location Tracking in Massive MIMO Systems via Dynamic Variational Bayesian Inference," *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5628–5642, 2019.
- [10] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct Localization for Massive MIMO," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2475–2487, 2017.
- [11] S. D. Bast, A. P. Guevara, and S. Pollin, "CSI-based positioning in massive mimo systems using convolutional neural networks," in *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.
- [12] S. De Bast and S. Pollin, "Mamimo CSI-Based Positioning using CNNs: Peeking inside the Black Box," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.
- [13] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-based Multi-Antenna and Multi-Point Indoor Positioning Using Probability Fusion," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2162–2176, 2022.
- [14] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Machine Learning Methods for RSS-Based User Positioning in Distributed Massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8402–8417, 2018.
- [15] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6.
- [16] X. Sun, C. Wu, X. Gao, and G. Y. Li, "Fingerprint-based Localization for Massive MIMO-OFDM System With Deep Convolutional Neural Networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10846–10857, 2019.
- [17] P. Ferrand, A. Decurninge, and M. Guillaud, "Dnn-based Localization from Channel Estimates: Feature Design and Experimental Results," in *IEEE Global Communications Conference*, 2020, pp. 1–6.
- [18] S. De Bast, E. Vinogradov, and S. Pollin, "Expert-knowledge-based Data-Driven Approach for Distributed Localization in Cell-Free Massive MIMO Networks," *IEEE Access*, vol. 10, pp. 56427–56439, 2022.
- [19] J. Fan, X. Dou, W. Zou, and S. Chen, "Localization Based on Improved Sparse Bayesian Learning in mmWave MIMO Systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 354–361, 2022.
- [20] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1998.
- [21] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. C. Wong, F. Tufvesson, V. Öwall, and O. Edfors, "The World's First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation," *IEEE Access*, vol. 5, pp. 9073–9088, 2017.
- [22] I. Yaman et al., "The LuViRA Dataset: Measurement Description," 2023. [Online]. Available: <https://arxiv.org/abs/2302.05309>
- [23] M. Khalid, J. Baber, M. K. Kasi, M. Bakhtyar, V. Devi, and N. Sheikh, "Empirical evaluation of activation functions in deep convolution neural network for facial expression recognition," in *International Conference on Telecommunications and Signal Processing (TSP)*, 2020, pp. 204–207.

Paper IV

Paper IV

Reproduced, with permission, from: IEEE

I. YAMAN, G. TIAN, D. PJANIC, F. TUFVESSON, O. EDFORS, Z. ZHANG, AND L. LIU,
“Adaptive Attention-Based Model for 5G Radio-Based Outdoor Localization,”
IEEE Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA,
USA, Oct 2025, pp. 192-197, doi: 10.1109/IEEECONF67917.2025.11443733.

Adaptive Attention-Based Model for 5G Radio-Based Outdoor Localization

Ilayda Yaman*, Guoda Tian[‡], Dino Pjanić[‡], Fredrik Tufvesson*, Ove Edfors*, Zhengya Zhang[†], Liang Liu*

*Dept. of Electrical and Information Technology, Lund University, Sweden

[‡]Ericsson AB, Lund, Sweden

[†]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA

ilayda.yaman@eit.lth.se, guoda.tian@ericsson.com, dino.pjanic@ericsson.com, fredrik.tufvesson@eit.lth.se, ove.edfors@eit.lth.se, zhengya@umich.edu, liang.liu@eit.lth.se

Abstract—Radio-based localization in dynamic environments, such as urban and vehicular settings, requires systems that efficiently adapt to varying signal conditions and environmental changes. Factors like multipath interference and obstructions introduce different levels of complexity that affect the accuracy of the localization. Although generalized models offer broad applicability, they often struggle to capture the nuances of specific environments, leading to suboptimal performance in real-world deployments. In contrast, specialized models can be tailored to particular conditions, enabling more precise localization by effectively handling domain-specific variations, which also results in reduced execution time and smaller model size. However, deploying multiple specialized models requires an efficient mechanism to select the most appropriate one for a given scenario. In this work, we develop an adaptive localization framework that combines shallow attention-based models with a router/switching mechanism based on a single-layer perceptron. This enables seamless transitions between specialized localization models optimized for different conditions, balancing accuracy and computational complexity. We design three low-complex models tailored for distinct scenarios, and a router that dynamically selects the most suitable model based on real-time input characteristics. The proposed framework is validated using real-world vehicle localization data collected from a massive MIMO base station and compared to more general models.

Index Terms—Vehicle localization, radio-based localization, adaptive models, attention-based models

I. INTRODUCTION

Reliable and accurate location information with low latency is essential to ensure efficient and reliable workflows in various tasks, including vehicle navigation, intelligent traffic management, and autonomous driving. Localization in urban environments presents significant challenges due to multipath propagation, signal blockages, and dynamic environmental conditions. Traditional Global Navigation Satellite System (GNSS)-based positioning often suffers from poor accuracy in dense urban areas, where buildings and other structures obstruct satellite signals. A robust and continuously available localization solution is needed that works seamlessly across different environments and platforms [1]–[3]. For example, in [4], the authors implement random forest and gradient boosting algorithms to effectively use multipath information to improve outdoor location. When tested on real-world data,

machine learning (ML) models achieve a mean localization error of approximately 100 m in an area of 580,000 m².

A key challenge in radio-based localization is designing computationally efficient models capable of adapting to dynamic signal conditions. Several works in the literature have addressed this issue [5]–[7]. A widely adopted approach is fingerprinting, where channel state information or channel impulse response (CIR) measurements serve as unique signatures of specific locations within an environment. By collecting and storing these measurements in a database, ML models can be trained to map real-time channel responses to their corresponding spatial coordinates (x, y) . Deep learning-based fingerprinting methods have demonstrated high localization accuracy by taking advantage of the spatial and temporal characteristics of the wireless channel [8]–[10].

Specialized fine-tuned models can achieve higher accuracy with smaller model sizes and less computational complexity compared to generalized models. To accommodate multiple conditions, general models are more complex, use more parameters, and have higher test times and computational costs. However, to fully leverage the advantages of specialized models across diverse environments, a router is needed to dynamically determine which model is best suited for a given scenario. By analyzing input characteristics and environmental conditions, the router selects the most appropriate specialized model, ensuring that only the necessary parameters are activated at each time step. Due to the continuous nature of the application, switching between different models would happen rarely, and the weights would be reused extensively. Thus, computational overhead is reduced and high localization accuracy is maintained under different conditions. By integrating adaptive models, localization systems dynamically adjust to varying urban scenarios, ensuring reliable and efficient positioning for applications such as autonomous driving, intelligent transportation, and urban navigation.

Fig. 1 visualizes three methods, with Method 1 being the most general and Method 3 the most specialized. Method 1 uses the same trainable parameters and architecture in all data subsets. Method 2 consists of N sets of trainable parameters that are loaded into the same model depending on the selected subset of data. Switching between trainable parameters is done

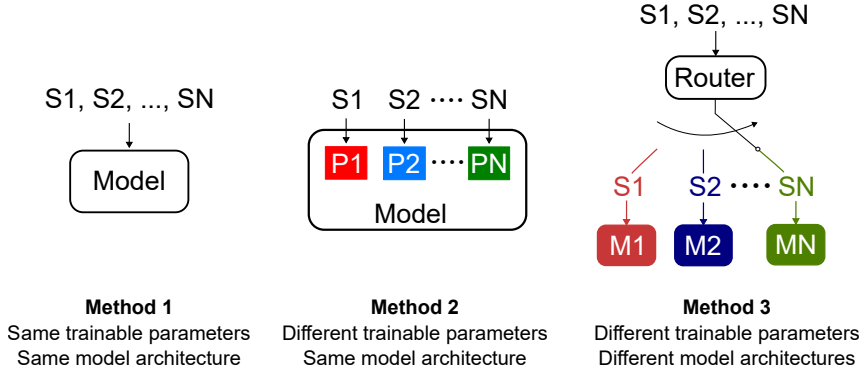


Fig. 1: Overview of the adaptive model, where S, N, P, and M represent the subset of data, the total number of subsets, the trainable parameters (weights and biases), and the model, respectively

manually. The proposed adaptive model, Method 3, includes specialized trainable parameters and a model architecture with run-time adaptation between different data subsets. The router in Method 3 determines which scenario the given input belongs to and selects the appropriate model accordingly, ensuring that only a subset of all parameters is active.

By tuning the model architecture parameters, the specialized models provide better localization accuracy than the other methods. The number of active trainable parameters and the training and testing times are also reduced. The main contributions of the paper are:

- We establish specialized attention-based models to minimize computational complexity, localization error, and model size for different scenarios. We then compare the results with more general models.
- We develop an adaptive localization framework capable of dynamically adjusting to environmental changes and signal propagation conditions. Measurement data from a massive MIMO base station (BS) and a moving user equipment (UE) on a vehicle is used for verification.

II. FUNDAMENTALS OF ATTENTION MODELS AND EXPERIMENTAL VALIDATION

This section provides an overview of the key components of attention-based models and describes the measurement campaign conducted to validate the proposed model.

A. Building Blocks of Attention-based Models

The architecture of the shallow encoder-only attention-based model is shown in Fig. 2. The model consists of positional encoding, multihead attention, and feedforward layers, each followed by residual connections and optional layer normalization. Additionally, a pooling layer is applied before the fully connected neural network (FCNN) to reduce the size of the model.

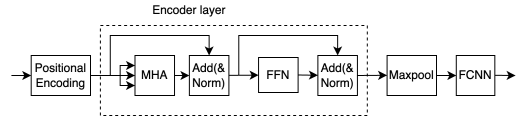


Fig. 2: The attention-based algorithm pipeline with one encoder layer and pooling layer added.

1) *Positional Encoding*: is used to incorporate position information into the model, using sinusoidal functions to generate continuous and differentiable positional vectors. This enables the model to capture relative positional relationships effectively. The model is tested with and without positional encoding, showing a significant drop in accuracy when it is removed.

2) *Multi-Head Attention (MHA)*: is the fundamental mechanism in attention-based architectures, enabling efficient parallel processing of input data [11]. It is based on Scaled Dot-Product Attention, which computes attention scores based on query, key, and value matrices derived from the input tensor $\mathbf{X} \in \mathbb{R}^{d_k \times d_{\text{model}}}$ at a given timestep

$$\mathbf{Q} = \mathbf{X} \mathbf{W}_q, \mathbf{K} = \mathbf{X} \mathbf{W}_k, \mathbf{V} = \mathbf{X} \mathbf{W}_v. \quad (1)$$

The attention mechanism is computed as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q} \mathbf{K}^T}{\sqrt{d_k}} \right) \mathbf{V}, \quad (2)$$

where d_k is the dimensionality of the key vectors, and the scaling factor $\sqrt{d_k}$ prevents large softmax values that could hinder training. The MHA extends this by applying multiple attention mechanisms in parallel, each with different learned projections

$$\text{head}_i = \text{Attention}(\mathbf{Q} \mathbf{W}_i^Q, \mathbf{K} \mathbf{W}_i^K, \mathbf{V} \mathbf{W}_i^V), \quad (3)$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ are learned projection matrices for the i -th attention head. The outputs of all heads are concatenated and projected to obtain the final representation:

$$\text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}^O, \quad (4)$$

where h is the number of attention heads, and $\mathbf{W}^O$ is a learned output projection matrix. By allowing the model to attend to different representation subspaces simultaneously, MHA enhances its ability to capture complex dependencies, e.g. in challenging environments where noise and signal multipath effects complicate traditional methods.

3) *Layer Normalization (LN)*: normalizes activations across the feature dimension for each input independently, making it effective for stabilizing deep networks.

4) *Position-wise Feed-Forward Networks (FFN)*: applies two linear transformations with an activation function in between. Each token in the sequence is processed independently through a position-wise transformation. The dimensionality of the inner layer is represented as d_{ff} . The activation function is the rectified linear unit (ReLU).

5) *Pooling layer*: is used to reduce the model size and computational complexity while retaining key feature representations. A one-dimensional max pooling layer is introduced between the encoder and the fully connected network. Max pooling is preferred over alternatives such as mean pooling due to its hardware efficiency, as it requires only comparison operations rather than arithmetic computations. Padding can be added to adjust alignment. This approach effectively reduces the number of parameters and computations in subsequent layers, resulting in a more compact and efficient model.

6) *Dropout*: is a regularization technique that helps prevent overfitting by randomly deactivating neurons during training. The dropout is added after the positional encoding, softmax, eq. 4 and pooling, and before the residual connections seen in Fig. 2. Different dropout rates were explored (0, 0.05, 0.1, 0.2, 0.3 and 0.5) and 0.05 was chosen based on the accuracy and training/validation loss of the model.

B. Experiments

The measurement campaign is based on a commercial self-contained massive MIMO BS and a single mobile user in a vehicle. The vehicle is equipped with a UE and a GNSS receiver, which is used for ground truth. The BS operates at 3.85 GHz center frequency and 100 MHz bandwidth. The uplink (UL) sounding reference signal (SRS) channel estimates span 273 Physical Resource Blocks (PRBs) across the full bandwidth, with each snapshot capturing all 64 beams every 20 ms. To reduce complexity during data collection, adjacent PRBs are averaged and downsampled into 137 subgroups (SGs), which are then further reduced to 46 PRBSGs (subcarriers) by interleaving every third SG. The SRS measurements represent the angular delay spectrum of the radio channel in the beam space after the FFT transformation from the antenna space.

The BS is equipped with 32 vertically and 32 horizontally polarized antenna ports, while the UE includes 4 antenna ports. The elements of the antenna array are used to form

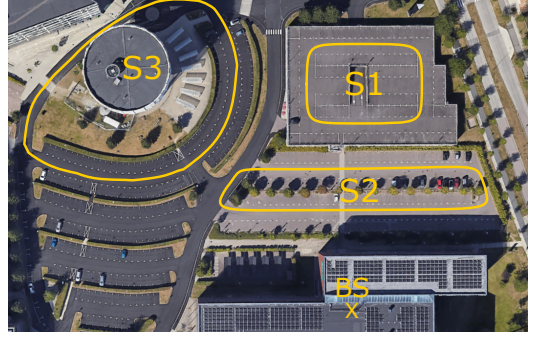


Fig. 3: Bird-eye view of the measurement environment and trajectories labeled S1, S2, and S3.

64 beams in both the UL and the downlink. The UL SRS data sounded from 2 paired UE antenna ports at the time are recorded across 46 subcarriers at the BS. The 32 horizontal and vertical beam space matrices of the channel transfer function (CTF) from the first and second antenna pairs at time t and across 46 subcarriers are denoted as $\mathbf{H}_{V1,t}, \mathbf{H}_{H1,t}, \mathbf{H}_{V2,t}$ and $\mathbf{H}_{H2,t}$, respectively. The combined channel matrix corresponds to $\mathbf{H}_t \in \mathbb{C}^{128 \times 46} = [\mathbf{H}_{H1,t}^T, \mathbf{H}_{V1,t}^T, \mathbf{H}_{H2,t}^T, \mathbf{H}_{V2,t}^T]^T$ for each channel snapshot.

In the pre-processing step, invalid data is filtered out, and a 46-point Hann window is applied across the rows of the CTF to suppress sidelobes. Subsequently, an IFFT is performed along the x-axis (i.e., across rows) to transform CTF into a CIR beam matrix that captures angular information. The amplitude values of the CIR beam matrix are then used as input to the attention-based model.

The UE is mounted on a vehicle following three predefined trajectories, forming three distinct scenarios, shown by yellow lines in Fig. 3. The BS location is marked by a yellow cross. The UE antenna maintains a consistent orientation as it is fixed to the vehicle roof and follows the same orientation for each lap. The average speed of the vehicle is 15 km/h and the BS is mounted on a 20-meter-high building. The vehicle follows the given trajectories in the same direction for five laps. For more detailed information on the measurement campaign and pre-processing of the measurement data, see [5], [12].

In the first scenario (S1), the vehicle drives on the roof of a garage, approximately 10 meters above ground level, following a trajectory that predominantly captures line-of-sight (LoS) conditions. The second scenario (S2) takes place at ground level beneath the base station (BS), where the vehicle is mostly not in direct LoS with the BS. The third scenario (S3), also at ground level, features intermittent LoS along the trajectory, with obstructions introduced by the tower visible in Fig. 3. These three scenarios constitute our data subsets: S1, S2, and S3. From the BS's perspective, users typically follow predefined urban mobility patterns. However, we acknowledge the repetitive nature of the selected UE trajectory and the fixed

antenna orientation toward the BS, as the UE was securely mounted on the vehicle roof, thereby reducing variability in the radio channel.

C. Evaluation Method

Of the five laps, the first four laps are used for training and validation, whereas the fifth lap of each scenario is reserved for testing. For the generalized model, the training/validation datasets of different scenarios are randomly mixed to create a dataset that consists of different scenarios, whereas the specialized models are only trained on the specific trajectories.

The accuracy of the localization algorithms is calculated using the mean of the Euclidean distances between the estimated locations and the ground truth labels. We calculate the mean error (ME) as

$$\text{ME} = \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_F \quad (5)$$

where N is the number of samples, $\hat{\mathbf{y}}_i$ is the predicted 2D location, and $\mathbf{y}_i$ is the corresponding ground truth label. The norm $\|\cdot\|_F$ denotes the Frobenius norm, which is equivalent to the Euclidean norm in this context.

III. PROPOSED METHOD

The model architecture selection of specialized models and the router that is used to dynamically switch between different scenarios are described in this section. The hyperparameter configuration of the model is summarized in Table I. For more information on the selection of the hyperparameters, please see [5]. The input of the model is the CIR beam matrix for each single snapshot described in the previous section.

TABLE I: Hyperparameters of the model.

Parameter	Value
Epochs	200
Batch Size (b)	64
d_{model}	46
d_k	128
d_{ff}	64
Learning Rate	0.0006
Number of heads	2
Dropout Rate	0.05
Loss function	MSE

A. Specialized models

The complexity of the model design is closely linked to the characteristics of the channel state information, which is highly influenced by multipath propagation. In an non LoS (NLoS) environment, the CIR and angular characteristics would differ significantly from a scenario where there is a dominant LoS component. In LoS environments, the energy is typically concentrated in the LoS path; therefore, the power delay profile changes slowly with UE movement. For example, in Fig. 4, the power is concentrated in the few early delay bins. The relative power of each beam is calculated by summarizing the power among all delay bins.

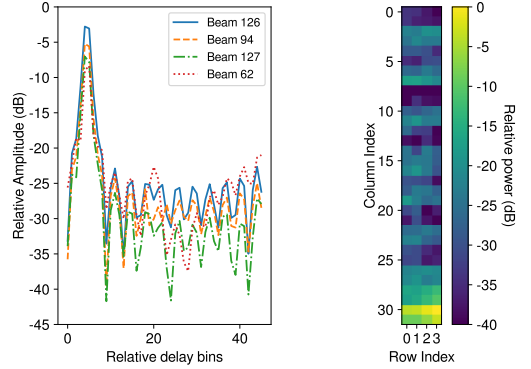


Fig. 4: Power delay profile of the 4 dominant beams and relative power of all 128 beams in a LoS scenario.

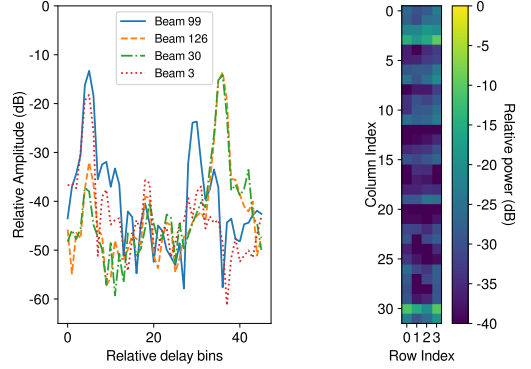


Fig. 5: Power delay profile of the 4 dominant beams and relative power of all 128 beams in an NLoS scenario.

In rich scattering environments (e.g., dense urban areas, indoor clutter), multipath reflections originate from numerous uncorrelated scatterers, resulting in less structured and more random components. Figure 5 illustrates this with a power delay profile and the corresponding beam matrix in an urban NLoS scenario. Unlike LoS conditions, the power delay profile in urban NLoS environments changes rapidly with UE position, motivating the design of specialized models to better capture such dynamics. Deeper models are more effective at modeling complex multipath interactions in these scenarios, whereas simpler models suffice under LoS conditions.

The following model architecture parameters are explored to select specialized models for different scenarios with different complexities: the effect of increasing the number of encoder layers from 1 to 5, removing/adding layer normalization, and adding a pooling layer before the FCNN layer. The pooling layer downsamples the input to the FCNN layer by selecting

the maximum value in non-overlapping segments of size 4. Padding is added to adjust the alignment. The output dimension becomes $(b, d_k, (d_{\text{model}} + 2)/4)$, resulting in a reduced number of trainable parameters. By exploring these parameters and comparing them to Method 1, the optimal specialized models are selected that improve the accuracy while keeping the size of the model similar for each scenario.

The results are shown in Table II, comparing the generalized (Method 1) and specialized (Method 3) models. The highest localization accuracy for Method 1 is achieved with 3 encoder layers, resulting in ME of (0.47m, 0.83m, 0.81m) for S1, S2, and S3, respectively. The best results are achieved when layer normalization is added for S3 and removed for S1 and S2. For Model 3, a lower ME for S1 can be achieved using 1 encoder layer, with or without the addition of pooling and layer normalization. For S2 and S3, 2 encoder layers are needed to achieve better performance with specialized models. Although adding more encoder layers can further reduce ME, computational complexity and model size increase accordingly.

TABLE II: Localization accuracy comparison between Method 1 and Method 3 across various model architectures (EL = Encoder Layers, LN = Layer Normalization, MP = MaxPool). Bold values indicate architectures achieving an optimal balance between Mean Error (ME), computational complexity, and parameter count.

EL	LN	MP	ME (m), Generalized			ME (m), Specialized		
			S1	S2	S3	S1	S2	S3
1	-	-	0.85	1.14	1.11	0.41	0.96	0.99
2	-	-	0.53	0.95	1.00	0.40	0.80	0.93
3	-	-	0.47	0.83	0.86	0.35	0.81	0.83
4	-	-	0.51	1.02	0.86	0.33	0.80	0.60
5	-	-	1.47	3.10	2.28	0.49	6.16	0.59
1	+	-	0.69	1.18	0.97	0.45	1.06	0.97
2	+	-	0.54	1.00	1.05	0.37	0.88	0.74
3	+	-	0.55	0.86	0.81	0.37	0.71	0.77
4	+	-	0.60	0.98	0.76	0.34	0.80	0.65
5	+	-	0.68	1.08	0.97	1.60	5.76	1.01
1	-	+	0.93	1.80	1.62	0.40	1.10	1.26
2	-	+	0.64	1.14	1.24	0.38	0.78	0.77
3	-	+	0.57	1.14	1.11	0.38	0.81	0.81
4	-	+	0.88	1.30	1.13	0.40	0.93	1.55
5	-	+	2.59	4.39	3.91	9.68	0.79	2.62
1	+	+	1.13	1.57	1.84	0.61	1.10	1.25
2	+	+	0.76	1.20	1.03	0.37	0.89	0.75
3	+	+	0.70	1.08	1.10	0.38	0.92	0.82
4	+	+	0.75	1.15	0.88	0.33	0.99	0.97
5	+	+	1.13	1.93	1.74	0.86	7.84	3.00

B. Router

In the proposed model, the router determines the type of scenario based on the input, a task known as classification. ML classifiers such as the Single-Layer Perceptron (SLP), Multi-layer Perceptron (MLP), and Convolutional Neural Networks (CNN) are widely used for classification tasks. Among SLP, MLP, and CNN, the SLP is the simplest and least computationally expensive, making it suitable for linearly separable

problems. It consists of an input layer directly connected to an output layer, with no hidden layers in between. The SLP performs a linear transformation of the input and applies an activation function to produce the output, and defined as

$$\mathbf{y} = \sigma(\mathbf{x}W + b) \quad (6)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the input vector, W is the weight vector, b is the bias term, $\sigma(\cdot)$ is the activation function (typically sigmoid or softmax), and $\mathbf{y}$ is the predicted output. The training and test data split used for the generalized attention-based model is also applied in this section. This strategy ensures that the training data were constructed by randomly sampling from all environments, ensuring that each batch contained a diverse set of propagation conditions.

When the router is implemented as an MLP or CNN, it achieves a test accuracy of 100%. To investigate the effects of overfitting, the dropout is increased to 0.7, and a weight decay of 1×10^{-5} is introduced. As a result, the test accuracy decreases to 99%. These findings indicate that the router model can be further reduced in complexity, potentially to an SLP.

With the complete input matrix, the training and validation losses of the SLP model converge to zero, and the test accuracy remains at 100%. To explore further model compression, trials are conducted using a smaller model that trains on the beam power matrix of one relative delay bin. The corresponding input is fed into a compact model with 387 parameters, achieving a test accuracy of 98.87%. Additional trials in different relative delay bins reveal that the minimum test accuracy observed is 97.76%. Given that there are 46 relative delay bins in total, this shows that using a subset of input features can still produce high classification accuracy.

IV. RESULTS

The final proposed architecture is depicted in Fig. 6. The diagram illustrates the active model with black lines, while the inactive models appear in gray. The sub-blocks of the "Encoder" layer in Fig. 6 are given in Fig. 2.

Table III summarizes the final results obtained for the three methods: Method 1 (the generalized model), Method 2 (different trainable parameters of each model but the same model architecture [5]), and Method 3 (the proposed adaptive model). The comparison is based on various metrics, including ME in three scenarios (S1, S2, and S3), the number of model parameters, adaptivity, and test time for S1. The test time is provided as a reference and may vary depending on the underlying hardware, software environment, and implementation details. All experiments are carried out in a Linux-based CPU-only environment with 64 GB of RAM.

In [5], the accuracy of Method 2 is given as 0.99 m, 2.00 m, and 1.01 m for S1, S2, and S3, respectively. In comparison, the proposed method shows improved accuracy, achieving ME of 0.40 m, 0.78 m, and 0.77 m for S1, S2, and S3, respectively. Method 1, which is the generalized model, achieves an ME of 0.47 m, 0.83 m, and 0.81 m for S1, S2, and S3, respectively, showing a slightly lower performance than the proposed method. The router system enables the activation

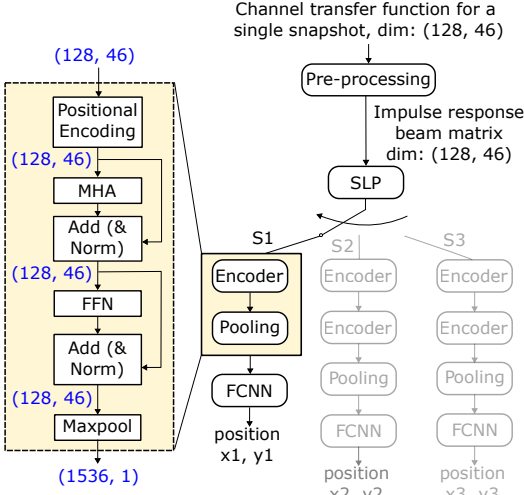


Fig. 6: The final block diagram of the proposed method after selecting the specialized models and the SLP router. The positional encoding is integrated into the first encoder layer for simplicity in the diagram. Input data dimensions at each sub-layer are indicated in blue.

TABLE III: Final result comparison.

	Method 1	Method 2 [5]	Method 3
S1, ME (m)	0.47	0.99	0.40
S2, ME (m)	0.83	2.00	0.78
S3, ME (m)	0.81	1.01	0.77
number of parameters ^a	303k	227k + 227k + 227k	87k + 126k + 126k
adaptive	+	-	+
test time, S1 (s)	4.40	1.83	1.48

^a Only one subset of the parameters is active at a given time for Method 2 and 3.

of a single model at any moment, reducing the number of active parameters by 58% to 71% and the test time by 66%, compared to the generalized model. The test time of the router is negligible. Please note that these results are specific to the given scenarios, and adaptations may be needed for different deployment scenarios. For example, the deployment in new base station scenarios requires retraining of the model; however, this limitation is beyond the scope of this study.

V. CONCLUSION

In this work, we present an adaptive localization model that dynamically selects specialized sub-models based on the given input scenario. We compare our approach against two methods: (i) a manually switched model where each scenario is assigned to a specialized model, and (ii) a generalized model that attempts to handle all scenarios with a single model but suffers from degraded localization accuracy. The proposed

model uses a routing mechanism to determine whether an input belongs to S1, S2, or S3, activating the corresponding sub-model accordingly. This approach enables the use of only a subset of the total parameters at any given time, reducing computational complexity compared to the generalized model, while achieving higher accuracy. To further improve efficiency, we introduce a pooling layer and remove layer normalization, reducing the model size and computational cost. The results highlight the potential of adaptive computation in radio-based localization, balancing accuracy and computational complexity across varying environmental conditions.

ACKNOWLEDGMENT

The authors would like to thank Jens Gulin and Axel Berg from Lund University, as well as Wei Tang, Sangbu Yun, and the Lab for Efficient Application Processors (LEAPs) at the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, for their valuable suggestions regarding our paper. This work is funded by the Swedish Research Council, Ericsson AB, and ELLIIT (Excellence Center at Linköping-Lund in Information Technology).

REFERENCES

- [1] S. Djosic, I. Stojanovic, M. Jovanovic, and G. L. Djordjevic, "Multi-algorithm UWB-based localization method for mixed LOS/NLOS environments," *Computer Communications*, vol. 181, pp. 365–373, 2022.
- [2] M. Malmström, I. Skog, S. M. Razavi, Y. Zhao, and F. Gunnarsson, "5g positioning - a machine learning approach," in *2019 16th Workshop on Positioning, Navigation and Communications (WPNC)*, pp. 1–6, 2019.
- [3] J. Chen, X. Li, R. Whiton, E. Leitingner, and F. Tufvesson, "Robust localization in modern cellular networks using global map features," 2025. arXiv preprint arXiv:2505.18160.
- [4] M. N. de Sousa, R. Sant'Ana, R. P. Fernandes, J. C. Duarte, J. A. Apolinário, and R. S. Thomä, "Improving the performance of a radio-frequency localization system in adverse outdoor applications," *EURASIP J. Wirel. Commun. Netw.*, vol. 2021, May 2021.
- [5] G. Tian, D. Pjanić, X. Cai, B. Bernhardtsson, and F. Tufvesson, "Attention-Aided Outdoor Localization in Commercial 5G NR Systems," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 1678–1692, 2024.
- [6] A. Sellami, L. Nasraoui, and L. N. Atallah, "Multi-Stage Localization for Massive MIMO 5G Systems," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6, 2020.
- [7] A. Decurninge, L. G. Ordóñez, P. Ferrand, H. Gaoning, L. Bojie, Z. Wei, and M. Guillaud, "CSI-based Outdoor Localization for Massive MIMO: Experiments with a Learning Approach," in *International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–6, 2018.
- [8] A. Al-Tahmeesschi, J. Talvitie, M. López-Benítez, and L. Ruotsalainen, "Deep Learning-based Fingerprinting for Outdoor UE Positioning Utilising Spatially Correlated RSSs of 5G Networks," in *International Conference on Localization and GNSS (ICL-GNSS)*, pp. 1–7, 2022.
- [9] G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu, and F. Tufvesson, "Deep-Learning-Based High-Precision Localization With Massive MIMO," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, 2024.
- [10] R. Whiton, J. Chen, T. Johansson, and F. Tufvesson, "Urban Navigation with LTE using a Large Antenna Array and Machine Learning," in *IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–5, 2022.
- [11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, (Red Hook, NY, USA), pp. 6000–6010, Curran Associates Inc., 2017.
- [12] D. Pjanić, G. Tian, A. Reial, X. Cai, B. Bernhardtsson, and F. Tufvesson, "Illuminating the path: Attention-assisted beamforming and predictive insights in 5g nr systems," 2025. arXiv preprint arXiv:2505.18160.

Paper V

Paper V

I. YAMAN, S. CHENG, O. EDFORS, AND L. LIU, "Implementation of an Adaptive Transformer Accelerator for Accurate Outdoor Localization with Massive MIMO," *Submitted for review*, 2026. Available at arXiv:2605.13507.