

---

# Datenerfassung und Privatsphäre in partizipativen Sensornetzen

Kevin Wiesner

---



Dissertation  
an der Fakultät für Mathematik, Informatik und Statistik  
der Ludwig-Maximilians-Universität München

vorgelegt von  
Kevin Wiesner

Tag der Einreichung: 22. Januar 2015



---

# Datenerfassung und Privatsphäre in partizipativen Sensornetzen

Kevin Wiesner

---



Dissertation  
an der Fakultät für Mathematik, Informatik und Statistik  
der Ludwig–Maximilians–Universität München

vorgelegt von  
Kevin Wiesner

1. Berichterstatter:	Prof. Dr. Claudia Linnhoff-Popien
2. Berichterstatter:	Prof. Dr-Ing. Lars Wolf
Tag der Einreichung:	22. Januar 2015
Tag der Disputation:	18. Mai 2015



## **Eidesstattliche Versicherung**

(siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig,  
ohne unerlaubte Beihilfe angefertigt ist.

Kevin Wiesner



*Danke an alle, die mich auf dem Weg zu dieser Arbeit  
begleitet und unterstützt haben.*



# Zusammenfassung

Partizipative Sensornetze (PSNs) stellen eine neue Art von Sensornetzen dar, die auf Basis von freiwillig zur Verfügung gestellten Mobiltelefonen etabliert werden. Sie ermöglichen eine großflächige Erfassung von Messdaten im direkten Umfeld von Menschen und können für zahlreiche Anwendungsszenarien verwendet werden. Neben ihren Vorzügen bringen PSNs aber auch Schwierigkeiten mit sich. Zwei zentrale Herausforderungen sind die ressourcenschonende Datenerfassung und der Schutz der Privatsphäre – beide resultieren aus der Instrumentalisierung privater Mobiltelefone zur Datenerfassung. Da der primäre Verwendungszweck der Geräte nicht die Aufzeichnung von Messdaten ist, darf diese deren Ressourcen nicht merklich belasten. Außerdem muss sichergestellt werden, dass durch die Erfassung von Messdaten die Privatsphäre der teilnehmenden Nutzer nicht verletzt wird.

Der erste Teil der Arbeit beschäftigt sich mit dem Aspekt der ressourcenschonenden Datenerfassung. Zunächst werden PSNs betrachtet, bei denen punktuell Messungen durchgeführt werden. Bei diesen Netzen müssen die teilnehmenden Geräte über die durchzuführenden Messungen unterrichtet werden. Damit hierbei die Ressourcen der Endgeräte nicht unnötig belastet werden, wird ein Konzept vorgestellt, das einerseits eine robuste Verteilung der Messaufgaben sicherstellt, gleichzeitig jedoch versucht, die Energieressourcen der Mobiltelefone zu schonen. Bei PSNs mit großflächiger und kontinuierlicher Datenerfassung spielt die Verteilung der Messaufgaben keine so entscheidende Rolle. Hier muss vielmehr sichergestellt werden, dass die Energie- und die Übertragungskosten auf Seiten der Nutzer möglichst gering bleiben. Aus diesem Grund wird ein Ansatz zur lokalen Gruppierung von Messknoten beschrieben, der durch eine Aufteilung der anfallenden Aufgaben und eine intelligente Auswahl der Knoten zu einer ressourcenschonenden und weniger redundanten Datenerfassung führt.

Der zweite Teil der Arbeit befasst sich mit dem Schutz der Privatsphäre der Teilnehmer und beinhaltet zwei Themenblöcke. Zum einen wird ein Ansatz zur automatisierten Erzeugung von Privatsphäre-Zonen vorgestellt, der ohne das Eingreifen der Nutzer die Zonen an das jeweilige Umfeld anpasst. Diese Zonen werden um die vom Nutzer häufig besuchten Orte erstellt und verhindern so mögliche, auf der Identifikation dieser Orte basierende Deanonymisierungsangriffe. Zum anderen wird ein Kalibrierungssystem für PSNs beschrieben, dessen Fokus sowohl auf der Verbesserung der Datenqualität als auch auf der Wahrung der Privatsphäre der Nutzer liegt. Hierfür ermöglicht dieses eine rückwirkende Anpassung bereits übertragener Daten, verhindert aber gleichzeitig durch eine Modifikation der Kalibrierungsparameter und der Upload-Zeitpunkte eine direkte Zuordnung zu einem Nutzer.



# Abstract

Participatory Sensing Networks (PSNs) represent a new kind of sensor networks that are established on the basis of voluntarily provided mobile phones. They allow for large-scale data collection in the immediate environment of people and can be used for various application scenarios. However, alongside their advantages, PSNs entail several difficulties. Two key challenges are the resource-efficient data collection and the protection of privacy - both resulting from exploiting private mobile phones for data collection. As the recording of measurement data is not the devices' primary purpose, it must not significantly burden their resources. In addition, it has to be ensured that the privacy of participating users is not violated by the acquisition of measurement data.

The first part of this thesis addresses the issue of resource-efficient data collection. At first, PSNs are examined in which measurements are only conducted at selected places. In these networks, the participating devices have to be informed about the required measurements. To avoid an unnecessary burdening of device resources at this stage, a concept is presented that ensures a robust distribution of measurement tasks and, at the same time, tries to conserve the mobile phones' energy resources. In PSNs with large and continuous data collection, the distribution of measurement tasks is less essential. It is rather necessary to ensure that energy and transmission costs on the user side remain as low as possible. For this reason, an approach for the local clustering of measurement nodes is described that allows for a resource-efficient and less redundant data collection by dividing and intelligently assigning occurring tasks.

The second part of the thesis deals with the protection of the participants' privacy and contains two thematic blocks. Firstly, an approach to automate the generation of privacy zones is presented. It adjusts the zones to the environment without the intervention of the user. These zones are created around the user's frequently visited places and thereby prevent deanonymisation attacks based on the identification of these places. Secondly, a calibration system for PSNs is described that focuses both on improving data quality and on the protection of user privacy. It allows for a retroactive adjustment of already transferred data, but simultaneously prevents an unambiguous attribution to a user by modifying the calibration parameters and the time of their transmission.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen partizipativer Sensornetze</b>	<b>7</b>
2.1	Definitionen und Begriffe . . . . .	7
2.1.1	Partizipative Sensornetze . . . . .	7
2.1.2	Datenerfassung . . . . .	9
2.1.3	Privatsphäre . . . . .	10
2.2	Abgrenzung zu drahtlosen Sensornetzen . . . . .	10
2.3	Anwendungen für partizipative Sensornetze . . . . .	13
2.3.1	Anwendungskategorien . . . . .	13
2.3.2	Menschzentrierte Anwendungen . . . . .	15
2.3.2.1	Überwachung der Gesundheit . . . . .	15
2.3.2.2	Dokumentation der sportlichen Aktivitäten . . . . .	16
2.3.2.3	Erfassung von Preisen . . . . .	16
2.3.3	Umweltzentrierte Anwendungen . . . . .	17
2.3.3.1	Überwachung der Luftqualität . . . . .	17
2.3.3.2	Erfassung von Umgebungslärm und -geräuschen . . . . .	18
2.3.3.3	Bestimmung der Wettersituation . . . . .	19
2.3.3.4	Überwachung der Verkehrssituation . . . . .	20
2.3.4	Entwickelte Anwendungen und prototypische Umsetzungen	21
2.4	Architektur von partizipativen Sensornetzen . . . . .	23
2.5	Herausforderungen partizipativer Sensornetze . . . . .	25
2.5.1	Nicht-technische Herausforderungen . . . . .	25
2.5.2	Technische Herausforderungen . . . . .	26
2.5.3	Bewertung . . . . .	28
2.6	Zusammenfassung . . . . .	29
<b>3</b>	<b>Datenerfassung in partizipativen Sensornetzen</b>	<b>31</b>
3.1	Klassifizierung existierender Ansätze . . . . .	32
3.1.1	Ressourcenschonung in drahtlosen Sensornetzen . . . . .	32
3.1.1.1	Aktivitätszyklen . . . . .	32
3.1.1.2	Datenreduktion und -erfassung . . . . .	35
3.1.1.3	Energieeinsparung durch Knotenmobilität . . . . .	38
3.1.2	Ressourcenschonung in partizipativen Sensornetzen . . . . .	39
3.1.2.1	Positionsbestimmung . . . . .	40
3.1.2.2	Task-Verteilung . . . . .	41

3.1.2.3	Datenreduktion und -erfassung . . . . .	43
3.1.3	Zusammenfassende Bewertung der Ansätze . . . . .	45
3.2	Ansatz zur energieeffizienten Task-Verteilung . . . . .	46
3.2.1	Problemstellung und Anforderungen . . . . .	47
3.2.2	Systemmodell . . . . .	48
3.2.3	Konzept . . . . .	50
3.2.3.1	Zellbasierte Task-Verteilung . . . . .	50
3.2.3.2	Adaptives Monitoring mit Bewegungserkennung . . . . .	52
3.2.3.3	Task-Bündelung . . . . .	53
3.2.4	Evaluation . . . . .	55
3.2.4.1	Simulationsumgebung . . . . .	55
3.2.4.2	Anzahl der Positionsbestimmungen . . . . .	57
3.2.4.3	Anzahl der Kommunikationsverbindungen . . . . .	58
3.2.5	Diskussion und Erweiterungen . . . . .	61
3.3	Ansatz zur lokalen Gruppierung von Messknoten . . . . .	62
3.3.1	Problemstellung und Anforderungen . . . . .	63
3.3.2	Konzept . . . . .	64
3.3.2.1	Aufbau und Erhalt der Cluster . . . . .	64
3.3.2.2	Bestimmung der Messknoten der Cluster . . . . .	67
3.3.3	Evaluation . . . . .	68
3.3.3.1	Simulationsumgebung . . . . .	69
3.3.3.2	Anzahl der Mobilfunk-Übertragungen . . . . .	70
3.3.3.3	Übertragungskosten für Mobilfunk-Übertragungen . . . . .	70
3.3.3.4	Energiekosten . . . . .	72
3.3.3.5	Anzahl der Messungen . . . . .	73
3.3.3.6	Overhead zur Bestimmung der Messknoten . . . . .	74
3.3.4	Diskussion und Erweiterungen . . . . .	75
3.4	Zusammenfassung . . . . .	78
<b>4</b>	<b>Schutz der Privatsphäre</b>	<b>81</b>
4.1	Bedrohungen der Privatsphäre in partizipativen Sensornetzen . . . . .	82
4.1.1	Preisgabe von Kontextdaten . . . . .	82
4.1.2	Sensible Informationen in Messdaten . . . . .	83
4.2	Klassifizierung existierender Ansätze . . . . .	85
4.2.1	Schutz der Kontextdaten . . . . .	85
4.2.1.1	Anonymität . . . . .	86
4.2.1.2	Unscharfe Kontextdaten . . . . .	87
4.2.1.3	Ausblenden sensibler Informationen . . . . .	88
4.2.1.4	Anonyme Task-Verbreitung . . . . .	89
4.2.2	Verarbeitung der Messdaten . . . . .	90
4.2.2.1	Lokale Weiterverarbeitung . . . . .	90
4.2.2.2	Aggregation . . . . .	91
4.2.2.3	Datenperturbation . . . . .	92
4.2.3	Zusammenfassende Bewertung der Ansätze . . . . .	93

---

4.3	Ansatz zur automatisierten Erstellung von Privatsphäre-Zonen . . . .	94
4.3.1	Problemstellung und Anforderungen . . . . .	94
4.3.2	Konzept . . . . .	97
4.3.2.1	Systemübersicht . . . . .	98
4.3.2.2	Random Rect . . . . .	100
4.3.2.3	Random Nearest Neighbor . . . . .	102
4.3.3	Evaluation . . . . .	104
4.3.3.1	Vergleich der Silent-Zone-Algorithmen . . . . .	105
4.3.3.2	Vergleich von Silent Zones mit Ban Zones . . . . .	108
4.3.4	Diskussion und Erweiterungen . . . . .	111
4.4	Ansatz für eine Privatsphäre schützende Kalibrierung . . . . .	112
4.4.1	Problemstellung und Anforderungen . . . . .	113
4.4.2	Kalibrierung in partizipativen Sensornetzen . . . . .	115
4.4.2.1	Messfehler . . . . .	116
4.4.2.2	Kalibrierungsmodell . . . . .	117
4.4.2.3	Existierende Kalibrierungsansätze . . . . .	119
4.4.3	Konzept . . . . .	121
4.4.3.1	Datenerfassung . . . . .	122
4.4.3.2	Vorwärtskalibrierung . . . . .	123
4.4.3.3	Rückwärtskalibrierung . . . . .	125
4.4.4	Evaluation . . . . .	129
4.4.4.1	Simulationsumgebung . . . . .	129
4.4.4.2	Erreichtes $k$ -Anonymitätslevel . . . . .	131
4.4.4.3	Genauigkeitsverlust bei Messdaten . . . . .	133
4.4.5	Diskussion und Erweiterungen . . . . .	138
4.5	Zusammenfassung . . . . .	139
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>141</b>
	<b>Abbildungsverzeichnis</b>	<b>145</b>
	<b>Literaturverzeichnis</b>	<b>147</b>
	<b>Eigene Publikationen</b>	<b>183</b>



# 1 Einleitung

*„The perfect platform for sensing the world is already in our hands.“*

– Nokia Research Center [271]

Der enorme Fortschritt, der in den Bereichen der Mikroelektronik, der Kommunikationstechnik sowie der Informationstechnologie in den letzten Jahren zu beobachten war, hat die Entwicklung und die Bedeutung drahtloser Sensornetze vorangetrieben. Drahtlose Sensornetze (engl. *Wireless Sensor Networks (WSNs)*) bestehen aus einer Vielzahl sogenannter Sensorknoten, d. h. Kleinstrechnern, die neben Prozessoren, Datenspeichern und Energiequellen auch mit entsprechender Sensorik und Modulen zur Funkkommunikation ausgestattet sind. Sensornetze ermöglichen weiträumige und umfassende Beobachtungen von Phänomenen der realen Welt und bilden somit eine Schnittstelle zwischen der digitalen und der physischen Welt. [158, 260] In den vergangenen Jahren wurde intensiv zu diesem Thema geforscht, was sich auch in den zahlreichen Übersichtsartikeln zu diesem Bereich widerspiegelt, vgl. [7–9, 13, 15, 53, 74]. Für jeden Teilbereich wurden spezielle Konzepte und Optimierungen entwickelt, um die Ressourcen der Sensornetze bestmöglich zu nutzen und dem Ziel einer umfassenden Erfassung und Verarbeitung von Informationen der physischen Welt näher zu kommen.

Laut oben genanntem Zitat aus einer Vision des Nokia Research Centers, halten wir die perfekte Plattform zum Erfassen der Welt bereits „in unseren Händen“ [271]. Die Autoren beziehen sich damit auf moderne Mobiltelefone, sogenannte *Smartphones*, die sich durch die technische Entwicklung jüngst zu mobilen „Alleskönnern“ entwickelt haben. Smartphones sind Mobiltelefone mit erweitertem Funktionsumfang. In der Regel zählen hierzu Internet-, Navigations- und Multimedia-Funktionen, wobei die Funktionalität durch die Installation von Applikationen, sogenannten *Apps*, erweitert und individualisiert werden kann [255].

Sehr verbreitet sind integrierte Digitalkameras, die neben Tonaufzeichnungen über das eingebaute Mikrofon auch Bild- bzw. Videoaufnahmen ermöglichen. Darüber hinaus sind Smartphones heutzutage mit einer Vielzahl von Sensoren bestückt. Fast schon standardmäßig gehören Gyroskop, Beschleunigungs- und Magnetfeldsensoren zur Ausstattung. Aber auch Licht- und Abstandsensoren, Messfühler für Temperatur, Luftfeuchtigkeit und Luftdruck, sowie Puls- und Fingerabdrucksensoren sind bereits bei der aktuellen Gerätegeneration verbaut [274, 275], oftmals auch

in Form von externem Zubehör, wie etwa Fitnessbändern [270], die per Funk mit dem Smartphone kommunizieren. Überdies werden immer speziellere und ausgefallenerere Smartphone-Sensoren entwickelt. Es gibt es beispielsweise externe Aufsätze, mit deren Hilfe Windgeschwindigkeit und Windrichtung bestimmt werden können [277]. Im Jahr 2012 wurde in Japan ein Smartphone vorgestellt, das laut Hersteller die radioaktive Strahlenbelastung erfassen und auf einer Karte visualisieren kann [107, 278]. Noch nicht ganz marktreif hingegen ist eine kleines, für Smartphones geeignetes Spektrometer [276], das im Rahmen eines Forschungsprojektes gerade entwickelt wird und mit dem sich die Zusammensetzung von Gasen oder Flüssigkeiten analysieren lassen soll.

Mit diesem Trend entwickeln sich Mobiltelefone, die ursprünglich für die persönliche Kommunikation entworfen wurden, zu allgegenwärtigen und universell einsetzbaren Messgeräten. Mit diesen können neuartige Sensornetze, sogenannte *partizipative Sensornetze (PSNs)* [34, 46, 98], realisiert werden. In PSNs werden die auf freiwilliger Basis zur Verfügung gestellten Mobiltelefone als Messknoten eingesetzt und ermöglichen so eine großflächige Erfassung von Daten, überall dort, wo Menschen mit Mobiltelefonen unterwegs sind und mit ihrer Umwelt interagieren.

Dabei sind nicht nur die Geräte selbst wichtig, sondern auch deren Besitzer. Die Nutzer sind ein wesentlicher Teil partizipativer Sensornetze und können entscheidend zur erfolgreichen Datenerfassung beitragen. So sorgt deren Mobilität für die Möglichkeit, Messungen an unterschiedlichen Orten durchzuführen. Ist außerdem die Erfassung komplexer Sachverhalte, die nicht mit einfachen Sensoren realisierbar ist, das Ziel, können Eingaben der Nutzer als „Messdaten“ verwendet werden. In anderen Fällen, wie z. B. der Aufnahme von Bildmaterial, kann das Eingreifen und Ausrichten der Geräte oder auch die Prüfung von aufgezeichneten Daten durch die Nutzer notwendig sein. Erst die Kombination aus Mobiltelefonen, die mit Sensoren ausgestattet sind, und Nutzern, die bereit sind, sich und ihre Geräte für die Datenerfassung zur Verfügung zu stellen, ermöglicht eine Umsetzung von PSNs.

Die Anwendungsmöglichkeiten für PSNs sind vielfältig. Schon heute werden die Mobiltelefone der Nutzer verwendet, um in Echtzeit Daten zur Verkehrssituation zu erfassen und diese auf der Straßenkarte bzw. bei der Routenplanung anzuzeigen [258]. Weitere mögliche Anwendungsfelder sind die Erfassung der Luftqualität [119] oder der Lärmbelastung [118], die Bestimmung der Wettersituation [293], die Überwachung der Gesundheit [43], die Analyse sportlicher Aktivitäten [70] oder die Erfassung von Preisen [63].

Solche auf Mobiltelefonen basierende Sensornetze haben eine Vielzahl von Vorteilen. Ein wichtiger Aspekt ist die globale Verfügbarkeit von Mobiltelefonen. Laut *International Telecommunication Union (ITU; dt. Internationale Fernmeldeunion)* gibt es weltweit fast 7 Mrd. Mobilfunkteilnehmer [266]. In Deutschland existieren beispielsweise bereits mehr als 116 Mio. Mobilfunkverträge [261]. Zwar ist noch nicht jedes Mobiltelefon für den Einsatz in PSNs geeignet, da viele ältere Geräte weder internetfähig sind noch über relevante Sensoren oder Positionierungstechnologien verfügen. Allerdings nimmt die Verbreitung moderner Smartphones stark zu.

Der Anteil der deutschen Bürger, die ein Smartphone nutzen, stieg innerhalb eines Jahres bis Mai 2014 um 14 Prozentpunkte an und lag damit bereits bei 55 % [259].

Darüber hinaus entstehen durch die Verwendung von privaten Geräten weder Anschaffungs- noch Instandhaltungskosten für die Betreiber der Sensornetze. Gerade bei sehr großflächig angelegten Beobachtungen mit mehreren Tausenden Sensorknoten ist dies ein wichtiger Aspekt, da mit jedem zusätzlichen Knoten die Kosten ansteigen. In PSNs werden diese von den einzelnen Nutzern übernommen. Sie schaffen sich die Geräte für die private Nutzung an und sorgen aus eigenem Interesse dafür, dass diese auch funktionsfähig sind. Hierzu zählt auch, dass die Geräte regelmäßig aufgeladen werden. Die Energiereserven sind zwar dennoch stark limitiert, aber im Vergleich zu traditionellen Sensornetzen ist die Energieversorgung deutlich weniger problematisch.

Ein weiterer Vorteil der Verwendung von Smartphones ist deren nahezu ständige Verbindung mit dem Internet. Die Geräte verfügen über verschiedene technische Möglichkeiten eine Verbindung zum Internet herzustellen, wie z. B. mittels Datenverbindung über das Mobilfunknetz oder durch lokale Funknetze wie *Wireless Local Area Networks (WLANs)*, die auch schon großflächig von den Nutzern angenommen werden: Heute verwenden bereits 54 % der Deutschen mobile Geräte für ihre Internet-Nutzung [265].

Allerdings bringen PSNs auch neue Probleme bzw. Herausforderungen mit sich. Da sie auf die Mitwirkung der Nutzer angewiesen sind und eine hohe Nutzerbeteiligung für den erfolgreichen Betrieb entscheidend ist, sind vor allem zwei Aspekte ausschlaggebend. Zum einen müssen die Ressourcen auf Seiten der privaten Endgeräte geschont werden. Die Geräte werden zwar durch deren Nutzer regelmäßig geladen, allerdings darf die Datenerfassung die Nutzung der Geräte durch den Eigentümer nicht merklich beeinträchtigen. Bei der Datenerfassung sollten daher sowohl die Energie- als auch die Übertragungskosten auf Seiten der Nutzer minimal gehalten werden. Zum anderen muss die Privatsphäre der Nutzer berücksichtigt werden. Da private Geräte verwendet werden, können die Messdaten auch Informationen über den Nutzer und sein Verhalten beinhalten. Hier ist unter anderem darauf zu achten, dass keine Deanonymisierung der Nutzer oder eine Nachverfolgung der Nutzerpfade möglich wird.

PSNs bieten folglich ein großes Potenzial und neue Möglichkeiten. Bevor sie allerdings großflächig eingesetzt werden können, müssen zunächst die genannten Herausforderungen gelöst werden. Der erste Teil dieser Arbeit beschäftigt sich daher mit der *Datenerfassung*. Nach der Betrachtung der Probleme und Herausforderungen in diesem Bereich werden verschiedene Lösungsansätze präsentiert. Zunächst wird der Anwendungsbereich von lokalen und zeitlich beschränkten Messungen betrachtet. Essentiell ist hier, dass die Nutzer und ihre mobilen Endgeräte über die durchzuführenden Messungen informiert werden. Daher wird ein Ansatz beschrieben, der die Verteilung und Überwachung solcher Messaufgaben effizient durchführt.

Des Weiteren wird auch der Anwendungsbereich großflächiger und langfristiger Kampagnen untersucht. Das Ziel ist hier die Reduktion redundanter Messungen und Übertragungen, um die Kosten sowohl für die Nutzer als auch den Betreiber möglichst gering zu halten. Es wird ein Ansatz zur lokalen Gruppierung von Messknoten vorgestellt, bei dem die Nutzer in lokaler Nähe zu Clustern formiert und die Rollen für die Aufnahme und die Übertragung der Messungen auf unterschiedliche Geräte verteilt werden.

Der zweite Teil der Arbeit widmet sich dem Thema der *Privatsphäre*. Zu Beginn werden auch hier die wichtigsten Probleme skizziert und anschließend verschiedene Lösungsansätze dargestellt. Ein Problem entsteht durch die Anhäufung von Messpunkten an den Orten, an denen die Nutzer viel Zeit verbringen, z. B. dem Wohnort oder dem Arbeitsplatz. Eine Identifikation dieser Orte kann zur einer De-anonymisierung bzw. einer Identifizierung der Nutzer führen. Um dieses Problem zu vermeiden, werden Privatsphäre-Zonen eingeführt, in denen keine Messungen durchgeführt werden. Diese Zonen werden um die wichtigen Orte des Nutzers erstellt und verwenden Gebäudeinformationen der Umgebung. Damit wird erreicht, dass sich die Zonen automatisch an das Umfeld anpassen, und somit ein Mindestlevel an Privatsphäre garantieren. Diese Eigenschaft ist für PSNs wichtig, da hierdurch eine systemweite Voreinstellung getroffen werden kann, die ohne Eingreifen der Nutzer ihre Privatsphäre schützt.

Ein weiterer wichtiger Aspekt in PSNs ist die Kalibrierung. Da typischerweise eine möglichst vielseitige Verwendung der erfassten Daten das Ziel ist, sollten diese im Idealfall sowohl umgehend verfügbar als auch möglichst genau sein. Um dies zu erreichen, werden die erfassten Daten unmittelbar nach der Aufnahme übermittelt. Falls im Nachhinein eine Kalibrierung stattfindet, wird eine rückwirkende Korrektur ermöglicht, um genauere Daten zu erhalten. Bei einer rückwirkenden Kalibrierung besteht jedoch die Gefahr, dass durch die Übermittlung der Kalibrierungsparameter eine Nachverfolgung der Nutzerpfade möglich wird. Daher wird ein Ansatz vorgestellt, der eine rückwirkende Kalibrierung bereits übermittelter Daten erlaubt und gleichzeitig die Privatsphäre der Nutzer schützt.

Die vorliegende Arbeit ist folgendermaßen strukturiert:

Kapitel 2 beschäftigt sich mit den Grundlagen partizipativer Sensornetze. Zunächst wird eine Einführung wichtiger Definitionen und Begriffe sowie eine Abgrenzung zu traditionellen Sensornetzen vorgenommen. Anschließend wird ein Überblick über existierende Anwendungen, die typische Architektur von PSNs, sowie deren Herausforderungen gegeben.

Kapitel 3 geht auf eine der bedeutendsten Herausforderungen, die effiziente Datenerfassung, ein. Nach einer ausführlichen Übersicht über die existierenden Ansätze in diesem Bereich, werden zwei konzipierte Ansätze vorgestellt. Der erste stellt ein Verfahren zur energieeffizienten Verteilung von Messaufgaben dar, der zweite ist ein Konzept zur lokalen Gruppierung von Messknoten.

Aspekte dieses Kapitels wurden bereits in [302] und [300] veröffentlicht: die Abschnitte 3.2.1–3.2.4 basieren auf dem in [302] vorgestellten Ansatz und erweitern diesen. Die Konzeption sowie die Evaluation des Ansatzes in [302] stammen dabei vollständig vom Autor der vorliegenden Arbeit. Die Abschnitte 3.3.1–3.3.3 basieren auf [300]. Auch hier sind sowohl die Konzeption als auch die Evaluation in vollem Umfang Kevin Wiesner zuzurechnen.

In Kapitel 4 wird schließlich die zweite wichtige Herausforderung, der Schutz der Privatsphäre der Nutzer, thematisiert. Hierfür werden mögliche Bedrohungen der Privatsphäre betrachtet, bevor bestehende Konzepte für diese vorgestellt werden. Es folgt die Besprechung von zwei neuen Ansätzen zum Schutz der Privatsphäre. Zunächst wird ein Konzept zur automatisierten Erstellung von Privatsphäre-Zonen beschrieben, im Weiteren ein Ansatz für eine die Privatsphäre schützende Kalibrierung.

Auch Teile dieses Kapitels basieren auf bereits veröffentlichten Inhalten [298, 299, 303]: die Abschnitte 4.3.1–4.3.3 basieren auf und erweitern [303]. Das Konzept in [303] wurde dabei überwiegend von Kevin Wiesner erarbeitet (ca. 90%), die Evaluation stammt in vollem Umfang von ihm. Die Abschnitte 4.4.1–4.4.4 gingen aus [298, 299] hervor. Bei beiden Publikationen wurde der überwiegende Teil des Konzeptes (ca. 90%) vom Autor der vorliegenden Dissertation erarbeitet und die Evaluation vollständig durch diesen durchgeführt.

In Kapitel 5 werden die Ergebnisse der Arbeit abschließend zusammengefasst und Möglichkeiten für weiterführende, zukünftige Forschungsarbeiten präsentiert.



## **2 Grundlagen partizipativer Sensornetze**

Dieses Kapitel beschäftigt sich mit den Grundlagen partizipativer Sensornetze. Hierzu zählt neben der Begriffsklärung auch eine Einordnung und Abgrenzung von traditionellen Sensornetzen. PSNs stellen eine neuartige Klasse von Sensornetzen dar, mit deren Einführung sich auch neue Anwendungsmöglichkeiten bieten. Allerdings bringt diese Form der Datenerfassung auch neue Probleme und Schwierigkeiten mit sich. Im folgenden Abschnitt sollen daher sowohl die möglichen Anwendungen für PSNs als auch die zu lösenden Herausforderungen betrachtet werden.

Das Kapitel ist wie folgt aufgebaut: In Abschnitt 2.1 werden zunächst wichtige Begriffe und Bezeichnungen beschrieben, bevor in Abschnitt 2.2 eine Abgrenzung zu traditionellen drahtlosen Sensornetzen erfolgt. Abschnitt 2.3 beschreibt verschiedene Anwendungskategorien in PSNs und gibt zudem einen Überblick über existierende Anwendungen. Abschnitt 2.4 stellt die typische Architektur von PSNs dar, worauf in Abschnitt 2.5 eine abschließende Betrachtung der wichtigsten Herausforderungen folgt.

### **2.1 Definitionen und Begriffe**

Dieser Abschnitt beschäftigt sich mit den zentralen Begriffen dieser Arbeit. Zunächst wird der Begriff der partizipativen Sensornetze erklärt und es werden alternative Bezeichnungen und verwandte Konzepte aufgezeigt. Anschließend wird die Bedeutung der Begriffe Datenerfassung und Privatsphäre im Kontext dieser Arbeit kurz erläutert.

#### **2.1.1 Partizipative Sensornetze**

Bei der Erforschung von Sensornetzen, die auf mobilen Endgeräten basieren, handelt es sich um ein noch relativ junges Forschungsgebiet. Erst die Entwicklung von kompakten, aber dennoch leistungsfähigen mobilen Geräten brachte ab 2006 PSN-Konzepte hervor. Die in den letzten Jahren stark ansteigende Verbreitung von Smartphones, welche mit einer Vielzahl von Sensoren ausgestattet sind, gab diesem Feld einen erneuten Auftrieb. Bisher haben sich jedoch noch keine einheitlichen Begrifflichkeiten durchgesetzt. So existiert eine Vielzahl von Bezeichnungen,

die prinzipiell ähnliche Konzepte beschreiben, sich allerdings durch die konkrete Ausrichtung oder ihren Begriffsumfang unterscheiden. Im Folgenden werden die wichtigsten Bezeichnungen vorgestellt, und im Anschluss die für diese Arbeit verwendete Definition dargelegt.

Eine der ersten Arbeiten in diesem Forschungsgebiet stammt von Burke et al. [34]. Die Autoren führen den Begriff *Participatory Sensing* ein und beschreiben damit die Verwendung und Vernetzung alltäglicher mobiler Geräte zu interaktiven Sensornetzen. Als Beispiel wird hier die Verwendung der in Mobiltelefonen verbauten Mikrofone und Kameras zur Datenerfassung genannt. Ein sehr ähnliches Konzept beschreiben Campbell et al. [37] mit dem Begriff *People-Centric Urban Sensing*. Ihr Fokus liegt auf der Erfassung des städtischen Raums, wofür auch hier Mobiltelefone bzw. andere mit Sensoren ausgestattete mobile Geräte verwendet werden. Die Ergänzung *People-Centric* (dt. menschenzentriert) deutet an, dass dabei auch Eigenschaften und Verhalten der Menschen und ihr direktes Umfeld erfasst werden sollen. In einer anderen Arbeit über *People-Centric Sensing* von Eisenman et al. [69] wird zudem der Begriff *Opportunistic Sensing* geprägt, der widerspiegeln soll, dass die Erfüllung einer Messaufgabe von der nicht steuerbaren Mobilität der teilnehmenden Knoten abhängt. Auch Paulos et al. [183] zweckentfremden die für die Kommunikation gedachten Mobiltelefone als mobile Messinstrumente und bezeichnen das Erfassen, Teilen und Kombinieren der Daten als *Participatory Urbanism*. Abdelzaher et al. [3] hingegen führen den Begriff der *Mobiscopes* ein, welche Zusammenschlüsse verteilter, mobiler Sensorknoten darstellen. Zwar beschränkt sich dieser Begriff nicht nur auf tragbare, mobile Geräte wie Mobiltelefone, er umfasst diese aber, und die Autoren führen Mobiltelefone auch explizit als Beispiel auf.

Wenngleich bei den beschriebenen Bezeichnungen der Fokus unterschiedlich gesetzt ist, stellen sie alle Konzepte dar, bei denen Mobiltelefone oder ähnliche Geräte für die Datenerfassung genutzt werden. Daher werden sie auch unter dem Begriff *Mobile Sensing* [119] oder *Mobile Phone Sensing* [135] zusammengefasst. Demgegenüber führen Sheth et al. [213] mit *Citizen Sensing* einen stark nutzerzentrierten Begriff ein. Bei diesem steht weniger das technische Verfahren als das aktive Mitwirken der Bürger (engl. *Citizen*) im Mittelpunkt, welche durch Beobachtungen und Berichte ihren Beitrag zur Erfassung von Phänomenen der realen Welt leisten. Eine der jüngsten Bezeichnungen ist der Begriff *Mobile Crowdsensing* [80], oft auch *Community Sensing* [135] genannt. Dieser umfasst Szenarien, bei denen das zu beobachtende Phänomen lediglich mit Hilfe einer großen Gruppe (engl. *Crowd*) von Nutzern zu erfassen ist, wie beispielsweise bei großflächigen Luftqualitätsmessungen, und bezeichnet folglich eine Untergruppe der vorherigen Ansätze.

Da der Begriff *Participatory Sensing* eine der ursprünglichen Bezeichnungen ist und inzwischen eine große Verbreitung findet, wird dieser in der vorliegenden Arbeit verwendet. Die deutsche Bezeichnung der partizipativen Sensornetze wurde von Hollick und Christin [98] geprägt. Der Begriff wird dabei sehr allgemein verwendet und beschreibt

*„Sensornetze[,] die durch Nutzerbeteiligung und auf Basis von Mobiltelefonen etabliert werden“ [98].*

Eine etwas ausführlicher Definition liefern Christin et al. in [46]. Dort beschreiben sie den englischen Begriff Participatory Sensing wie folgt:

*„Without loss of generality, we use the generic term participatory sensing to designate applications using mobile phones as sensors (or as data sink for interfaced sensors) where participants voluntarily contribute sensor data for their own benefit and/or the benefit of the community. The notion of participatory sensing therefore includes mobiscopes and opportunistic sensing, and also covers specific terminologies focusing on particular monitoring subjects, such as urban sensing, participatory urbanism, citizen sensing, people-centric sensing, and community sensing.“ [46]*

Der erste Teil der Definition skizziert die Bedeutung des Begriffes. Demnach beschreibt dieser Anwendungen, bei denen Mobiltelefone entweder als Sensoren oder als Datensinke für angeschlossene Sensoren fungieren und Nutzer die so erfassten Sensordaten freiwillig beisteuern. Die Nutzer können dies sowohl für ihren eigenen Nutzen als auch zum Wohle der Gemeinschaft tun. Der zweite Teil greift die zu Beginn des Abschnittes beschriebene Begriffsvielfalt auf und betont, dass die im ersten Teil formulierte Beschreibung somit alle zuvor vorgestellten Bezeichnungen und Konzepte umfasst. Da diese Definition eine sehr umfassende und gleichzeitig sehr konkrete Abgrenzung des Begriffes darstellt, wird sie als Grundlage für diese Arbeit verwendet.

### 2.1.2 Datenerfassung

In der Regel bezeichnet der Begriff Datenerfassung „alle Arbeitsvorgänge, mit denen anfallende Daten in eine maschinenlesbare Form gebracht und auf Datenträgern gespeichert werden“ [48]. Im Kontext der Messtechnik versteht man unter Datenerfassung den Prozess der Erfassung elektrischer oder physikalischer Phänomene mit Hilfe eines Computers [91]. Bei der Messung von physikalischen Phänomenen, z. B. der Umgebungstemperatur, werden diese zunächst mit einem Sensor in ein elektrisches Signal umgewandelt. Die physikalische Größe, der die Messung gilt, wird dabei als Messgröße bezeichnet [61]. Analoge Signale müssen dann mit Hilfe eines Analog-Digital-Wandlers in digitale Signale gewandelt werden, bevor sie von digitalen Geräten verarbeitet werden können.

Bei PSNs bezeichnet die Datenerfassung somit die Aufnahme elektrischer oder physikalischer Phänomene mit Mobiltelefonen und die Umwandlung der erhaltenen Messsignale in digitale Messdaten. In dieser Arbeit wird mit dem Begriff Datenerfassung der komplette Prozess, von der Anweisung zur Datenerfassung und der Verteilung von Messaufgaben bis zur Übermittlung der erfassten Daten zur Speicherung, bezeichnet.

### 2.1.3 Privatsphäre

Da in PSNs die Nutzer mit ihren privaten Mobiltelefonen Daten sammeln, ist der Schutz der Privatsphäre ein zentraler Aspekt zur Erreichung einer hohen Nutzerakzeptanz. Eine allgemeingültige Definition des Begriffes Privatsphäre ist schwierig und es gibt zahlreiche Ansätze zur Begriffsklärung. Eine sehr allgemeine und vollständige Definition bieten Trepte und Dienlin [227]. Sie beschreiben den Begriff basierend auf drei bewährten Definitionen von Westin [236], Altman [12] und Burgoon [33] wie folgt:

*„Privatsphäre ist ein individueller Zustand der Abgeschlossenheit und Intimität (Westin, 1967), der einer stetigen Regulierung von Zuviel und Zuwenig Privatsphäre unterliegt (Altman, 1975), wobei sich zu jedem Zeitpunkt vier verschiedene Privatsphäredimensionen unterscheiden lassen: informationale, soziale, psychische und physische Privatsphäre (Burgoon, 1982).“ [227]*

In PSNs ist vor allem die informationale Privatsphäre von Bedeutung. Diese beschreibt, in welchem Maße die Nutzer die Erhebung von persönlichen Daten kontrollieren können, und bestimmt so auch die Wahrnehmung, inwieweit die Nutzer sich frei von Überwachung fühlen.

Zum Schutz der Privatsphäre der Nutzer müssen in PSNs daher technische Maßnahmen ergriffen werden, welche die Erhebung persönlicher Daten und den Zugriff auf diese regulieren und die Anonymität der Nutzer gewährleisten. Dabei müssen neben personenbezogenen Daten, welche die Identität der Nutzer beschreiben, auch jegliche andere Arten von Daten geschützt werden, die Rückschlüsse auf die Identität der Nutzer oder deren Privatleben erlauben. [66]

## 2.2 Abgrenzung zu drahtlosen Sensornetzen

WSNs sind Zusammenschlüsse von Sensorknoten, die durch Sensoren Daten über ihr Umfeld sammeln, diese gegebenenfalls lokal verarbeiten und mit Hilfe von drahtloser Kommunikation weiter verbreiten, vgl. [221]. Partizipative Sensornetze können als eine spezielle Art drahtloser Sensornetze betrachtet werden, unterscheiden sich allerdings von diesen in einigen Punkten [301]. Durch eine Übersicht der wichtigsten Unterschiede sollen im Folgenden PSNs von traditionellen Sensornetzen abgegrenzt werden:

**Eigentümerschaft:** Im Gegensatz zu traditionellen Sensornetzen, bei denen in der Regel das komplette Netz inklusive aller Hardware im Besitz des Betreibers ist, sind bei PSNs die Nutzer die Eigentümer der Sensorknoten. Da sie aus eigenem Interesse dafür Sorge tragen, dass ihre Geräte funktionsfähig sind, muss der Betreiber sich nicht um die Instandhaltung kümmern. Allerdings ist es ihm dadurch auch nicht möglich, die Anzahl der Knoten zu bestimmen.

Während ein Betreiber eines traditionellen Sensornetzes in der Lage ist, die Anzahl der Sensorknoten im gewünschten Messumfeld zu erhöhen, kann bei PSNs lediglich mit Anreizen für die Nutzer versucht werden, die Anzahl der Teilnehmer zu erhöhen. Zudem muss beim Betrieb die Privatsphäre der Nutzer beachtet werden, wie nachfolgend beschrieben.

**Diversität:** Ein weiteres Unterscheidungsmerkmal ist die größere Hardware-Diversität bei PSNs im Vergleich zu WSNs. Letztere bestehen vorwiegend aus relativ homogenen Hardwarelandschaften, d. h. die verwendeten Sensorknoten sind baulich größtenteils gleich oder sehr ähnlich. Es kann zwar vorkommen, dass die Sensorknoten schrittweise verbessert werden und folglich verschiedene Generationen von Knoten im Einsatz sind, allerdings ist die Anzahl der verschiedenen Knotentypen grundsätzlich klein.

PSNs bestehen hingegen aus sehr heterogenen Hardwarelandschaften, da alle am Markt verfügbaren Mobiltelefone verwendet werden können. Betrachtet man allein den Anteil der auf Android basierenden Smartphones, gibt es bereits mehrere Tausend Modelle [263, 272]. Die unterschiedliche Ausstattung der Geräte muss bei der Verteilung von Messaufgaben und der Datenerfassung beachtet werden, um vergleichbare und qualitativ hochwertige Messdaten gewährleisten zu können.

**Energie:** In WSNs stellt die Beschränktheit der Energieressourcen eine der Hauptherausforderungen dar [53]. Die Batterien der Sensorknoten können meist nicht oder nur mit großem Aufwand geladen werden, weshalb ein umsichtiger Gebrauch der Energiereserven für eine langfristige Nutzung der Netze unabdingbar ist.

Auch in PSNs müssen die Energieressourcen geschont werden, da die Nutzer ihre Mobiltelefone bei einem verschwenderischen Umgang wohl nicht freiwillig zur Verfügung stellen würden. Allerdings werden die Geräte auch regelmäßig durch die Nutzer aufgeladen, wodurch eine langfristige Teilnahme als Messknoten möglich wird. Die Herausforderung in PSNs ist somit nicht allein das technische Problem, sondern auch die Bereitschaft der Nutzer, die vorhandenen Energieressourcen mit dem PSN zu teilen. Bei letzterem können zum einen intelligente Strategien helfen, so dass Messknoten beispielsweise nur aktiv am PSN teilnehmen, solange die Geräte genügend Energiereserven haben. Zum anderen können auch Anreizsysteme angewandt werden, bei denen die Nutzer für das Teilen ihrer Ressourcen belohnt bzw. entschädigt werden, z. B. in finanzieller Form [224].

**Mobilität:** Auch bezüglich der Mobilität der Knoten unterscheiden sich traditionelle und partizipative Sensornetze. Werden bei Ersteren mobile Sensorknoten eingesetzt, ist dies in der Regel eine bewusste Design-Entscheidung [199]. So kann Mobilität genutzt werden, um die Lage der Knoten zu verändern [231] oder um die Messdaten effizienter einzusammeln [233]. Wenngleich es auch Sensornetze gibt, bei denen die Knoten nur passiv mobil sind, beispielsweise

se indem sie an einem bewegten Objekt angebracht sind, das nicht unter der Kontrolle des Sensorknotens steht [111], baut ein großer Teil mobiler Sensornetze auf aktive Mobilität, d. h. auf Knoten, die sich selbst fortbewegen können.

Bei PSNs hingegen sind die Messknoten immer mobil, allerdings können die Nutzer mit ihren Mobiltelefonen nicht gesteuert werden. Messungen in PSNs können folglich nur dort stattfinden, wo die Nutzer mit ihren Geräten vorbeikommen. Es kann jedoch versucht werden, durch monetäre [230] oder spielerische Anreize [159, 201] die Bewegungspfade der Nutzer zu beeinflussen.

**Verwendungszweck:** WSNs werden meist für einen bestimmten Verwendungszweck entworfen und eingesetzt, wie etwa zur Überwachung des Zustandes von Bauwerken [128], der Umwelt [209] oder von Habitaten [154]. Aufgrund der zuvor beschriebenen Energieproblematik sind die Sensorknoten meist für den entsprechenden Einsatzzweck optimiert, d. h. es wird entsprechend abgestimmte Hard- und Software verwendet.

In PSNs sind die Sensorknoten die Mobiltelefone der Nutzer, welche zunächst einmal gar nicht für den Einsatz in solchen Netzen vorgesehen sind. Da diese sich aber inzwischen zu vielseitigen Multimediageräten entwickelt haben, können auch verschiedenste Messdaten erfasst werden. Somit können PSNs für diverse Zwecke genutzt werden, wenngleich die Mehrheit der existierenden PSNs sich zum derzeitigen Stand meist auf eine konkrete Anwendung fokussieren.

**Größe:** Bei traditionellen Sensornetzen kann die Anzahl an Sensorknoten und damit auch die Ausbreitung des überwachten Gebietes direkt gesteuert werden, wobei allerdings die Kosten für die Hardware und Instandhaltung in der Regel die Größe des Netzes begrenzen.

Bei PSNs kann weder die Anzahl der Teilnehmer noch die geografische Ausbreitung direkt gesteuert werden. Es muss folglich auf die Anwerbung entsprechender Teilnehmer gebaut werden. Da die Nutzer ihre Mobiltelefone selbst anschaffen und warten, besteht im Gegensatz zu WSNs keine, aus den Kosten für die Hardware resultierende obere Begrenzung für PSNs, so dass großflächige Netze mit sehr vielen Messknoten entstehen können.

**Privatsphäre:** Bei den meisten traditionellen Sensornetzen ist der Schutz der Privatsphäre kein Thema. Die Sensorknoten sind im Besitz der Betreiber und erfassen gewöhnlich keine Daten, welche die Privatsphäre der Beteiligten gefährden könnten. Lediglich bei personenbezogener Datenerhebung oder sogenannten Multimedia-Sensornetzen [8], die beispielsweise mit Kameras die Bewegungen von Personen aufzeichnen, muss auf die Wahrung der Privatsphäre geachtet werden.

Bei PSNs werden die Sensorknoten von den Nutzern mitgeführt und können somit in mehrfacher Hinsicht die Privatsphäre gefährden: Zum einen können

die Messdaten selbst sensible Daten enthalten, zum anderen können die Kontextdaten der Messungen, wie Ort und Zeit, sensible Informationen über den Nutzer preisgeben.

## 2.3 Anwendungen für partizipative Sensornetze

Im Folgenden soll ein Überblick über das breite Spektrum existierender Anwendungen für PSNs gegeben werden. Dabei soll die Vielfalt an Möglichkeiten, die PSNs bieten, illustriert werden, weswegen die Beschreibungen der jeweiligen Anwendungen kurz gefasst sind. Ausführliche Betrachtungen sind in den Übersichtsarbeiten von Lane et al. [135], Kanhere [117], Christin et al. [46] und auch Khan et al. [125] zu finden. Vor der eigentlichen Übersicht über die Anwendungen, werden im folgenden Abschnitt zunächst die verschiedenen Kategorien von Anwendungen beschrieben.

### 2.3.1 Anwendungskategorien

Die mit Hilfe von PSNs realisierten Anwendungen können anhand zweier Dimensionen kategorisiert werden: dem Beobachtungsziel einerseits und der Art der Datenerfassung andererseits.

Das Beobachtungsziel beschreibt den Fokus der Anwendung und gibt somit an, was und weshalb gemessen werden soll. Bei der Klassifizierung der Beobachtungsziele gibt es verschiedene Betrachtungsweisen. Campbell et al. [38] unterteilen diese in drei Kategorien: *persönliche*, *soziale* und *öffentliche* Anwendungen. Diese Kategorien beinhalten sowohl das Beobachtungsziel als auch die Zielgruppe der Messdaten. Bei persönlichen Anwendungen liegt der Fokus auf der Erfassung von Messdaten einer Person, die gemeinhin auch nur für diese Person zugänglich sind. Soziale Anwendungen hingegen erfassen Informationen über das Sozialverhalten der Nutzer, welche dann mit bestimmten sozialen Gruppen geteilt werden. Öffentliche Anwendungen beschreiben Szenarien, bei denen die Erfassung von Messdaten für die Allgemeinheit erfolgt. Da in diesem Fall die Messdaten öffentlich sind, handelt es sich in der Regel um Daten ohne Bezug zum Nutzer, wie beispielsweise Umweltinformationen.

Hollick und Christin [98] unterscheiden nur zwei Kategorien: *menschzentrierte* und *umweltzentrierte* Anwendungen. Erstere sind Anwendungen, bei denen Informationen über die Nutzer erfasst werden. Umweltzentrierte Anwendungen hingegen erfassen Informationen über das Umfeld der Nutzer mit dem Ziel einer Umweltüberwachung. Betrachtet man nur das Beobachtungsziel, können die zuvor beschriebenen persönlichen und sozialen Anwendungen als menschzentriert und öffentliche Anwendungen als umweltzentriert klassifiziert werden. Da Hollick und Christin das

	Partizipative Datenerfassung	Opportunistische Datenerfassung
Mensch-zentrisch	I. z. B. aktives Erfassen der Ernährung zur Gesundheitsüberwachung [58]	II. z. B. automatische Erfassung der persönlichen Schadstoffbelastung [119]
Umwelt-zentrisch	III. z. B. aktives Verfassen von Wetterberichten [293]	IV. z. B. kontinuierliche Überwachung der Luftqualität [286]

Abbildung 2.1: Klassifizierung von Anwendungen partizipativer Sensornetze

Beobachtungsziel deutlicher von der Zielgruppe abgrenzen, wird im Rahmen dieser Arbeit diese Klassifizierung verwendet.

Bei der Datenerfassung wird zwischen zwei Arten unterschieden: der *partizipativen* und der *opportunistischen* Datenerfassung [134]. Bei der partizipativen Datenerfassung entscheidet sich der Nutzer bewusst dazu, eine Anfrage der Anwendung zu beantworten, d. h. der Nutzer greift bewusst in den Prozess der Datenerfassung ein und trägt aktiv Daten bei. Dies kann beispielsweise durch Aufnehmen eines Fotos [63] oder durch Absenden eines Reports [293] geschehen. Bei der opportunistischen Datenerfassung werden die Daten automatisch und ohne Zutun des Nutzers gesammelt, wenn die Situation dafür geeignet ist. Der Nutzer willigt zwar ein, am PSN teilzunehmen, wird aber für die eigentliche Erfassung der Messdaten nicht mehr aktiv. Es ist daher möglich, dass dem Nutzer die gegenwärtige Datenerfassung gar nicht bewusst ist. Eine solche Art der Datenerfassung bietet sich für kontinuierliche Messungen an, wie beispielsweise eine Überwachung der Luftqualität [286]. Wichtig ist hierbei, dass der Begriff der partizipativen Sensornetze nicht die Art der Datenerfassung vorgibt. Wie in Abschnitt 2.1.1 erläutert, umfassen PSNs sowohl Anwendungen mit partizipativer als auch mit opportunistischer Datenerfassung.

Anhand der zwei Dimensionen lassen sich die Anwendungen, wie in Abbildung 2.1 dargestellt, in vier Kategorien einteilen. In den folgenden Abschnitten wird nun eine Übersicht über existierende Anwendungen gegeben. Dabei werden die Anwendungen nach ihrem Beobachtungsziel aufgelistet, d. h. es wird nach menschenzentrierten und umweltzentrierten Anwendungen unterschieden, da so thematisch ähnliche Anwendungen gebündelt vorgestellt werden können.

## 2.3.2 Menschzentrierte Anwendungen

Bei menschenzentrierten Anwendungen steht, wie bereits erläutert, der Mensch im Mittelpunkt. Diese Art von Anwendungen wird häufig zur Dokumentation von Aktivitäten, z. B. sportlichen Leistungen, oder Verhaltensweisen, wie etwa Ernährungsverhalten [117], genutzt. Ein besonderes Beispiel stellt hier *CenceMe* dar [162, 163], bei dem, ohne speziellen Anwendungsfokus, möglichst viele Daten über den Nutzer, wie seine physische Aktivität, seine soziale Aktivität und sein Standort, aufgezeichnet und mit anderen Nutzern geteilt werden. Die meisten Anwendungen erheben die Daten jedoch mit einem speziellen Ziel. Eine große Gruppe untersucht und analysiert die Gesundheit der Nutzer, eine weitere Gruppe erfasst die sportliche Aktivität der Teilnehmer und eine dritte Gruppe erfasst die von Produkt- oder Dienstleistungsanbietern festgelegten Preise.

### 2.3.2.1 Überwachung der Gesundheit

Bei der Gruppe der Gesundheitsanwendungen gibt es zwei große Untergruppen: Zum einen Anwendungen, die eine gesunde Lebensweise der Nutzer fördern wollen. Zum anderen Applikationen, welche die gesundheitliche Situation der Nutzer überwachen und gegebenenfalls Alarm schlagen, falls sich ein Nutzer in einer ungesunden oder gefährlichen Situation befindet. Bei ersteren liegt der Fokus meistens auf der Analyse von Ernährung und Bewegung. Bei *DietSense* [196] werden Mobiltelefone zur Aufzeichnung von Essgewohnheiten verwendet. Hierzu werden die Geräte den Nutzern mit einem Band umgehängt und nehmen neben Fotos der Mahlzeiten auch den Kontext, d. h. Ort, Zeit und Ton, auf. Bevor die aufgezeichneten Daten mit Ernährungsexperten geteilt werden, können die Teilnehmer diese einsehen und gegebenenfalls zensieren. Eine Reihe von Anwendungen versucht darüber hinaus auch die Bewegung und den daraus resultierenden Kalorienverbrauch zu bestimmen. Einige Systeme verwenden zur Bewegungs- und Aktivitätserkennung Beschleunigungssensoren, wie *HealthAware* [81], welches mit einem „Aktivitätszähler“ die Nutzer zu mehr Bewegung animieren will. Consolvo et al. [49] verfolgen mit ihrem *UbiFit Garden* einen ähnlichen Ansatz, verwenden jedoch ein zusätzliches Barometer. *BALANCE* [58] hat das gleiche Ziel, erfasst aber für eine genauere Aktivitätserkennung noch Informationen über die Umgebung, d. h. Licht, Druck, Feuchtigkeit, Ton, sowie den Standort des Nutzers, der mit Hilfe des *Global Positioning Systems (GPS)*; dt. „Globales Positionsbestimmungssystem“ bestimmt wird. Für eine noch genauere Bestimmung der Aktivitäten und des Energiebedarfs der Nutzer werden bei vielen Ansätzen spezielle Sensoren integriert. So werden bei *KNOWME* [14] und auch bei *Jog Falls* [172] Herzfrequenz-Sensoren eingesetzt, die z. B. in die Kopfhörer [186] der Nutzer verbaut werden können. Der Fokus der zweiten Untergruppe liegt auf dem aktuellen Gesundheitszustand der Nutzer. Bei *Mobicare* [43] werden neben Beschleunigungsdaten auch kontinuierlich Elektrokardiogramm-Daten (EKG) aufgezeichnet und per Bluetooth an das Mobiltelefon des Nutzers gesendet. Erkennt dieses Unregelmäßigkeiten oder kritische

Werte, wird über eine entsprechende Warnmeldung der zuständige Arzt informiert. Auch *Spa* [210] überwacht den Gesundheitszustand der Nutzer, in dem Herzfrequenz, Blutdruck, Bewegung und Lärmbelastung aufgezeichnet werden, und meldet Auffälligkeiten an medizinische Fachkräfte. Im Gegensatz zu direkten Messungen der Körperfunktion zur Analyse des Gesundheitszustandes, gibt es auch Anwendungen, die einen indirekten Weg wählen und beispielsweise messen, wie stark die Nutzer Schadstoffen ausgesetzt sind. Die mit *Personal Environmental Impact Report (PEIR*, dt. „persönlicher Umweltverträglichkeitsbericht“) [170] bezeichnete Anwendung ermittelt dafür die GPS-Position und mit Hilfe der Beschleunigungssensoren auch das verwendete Fortbewegungsmittel. Diese Informationen werden mit den entsprechenden Wetter- und Verkehrsdaten kombiniert und sowohl die Schadstoffmenge, die der Nutzer verursacht hat, als auch die Schadstoffbelastung, der er ausgesetzt war, bestimmt.

### 2.3.2.2 Dokumentation der sportlichen Aktivitäten

Die zweite Gruppe von Anwendungen befasst sich mit der Dokumentation von sportlichen Aktivitäten und Leistungen. Eines der ersten Systeme in diesem Bereich war *SkiScape* [67], ein PSN für Skigebiete. Die mobilen Sensorknoten werden auf Skiern montiert und erfassen Temperatur- und Beschleunigungsdaten, sowie kurze Tonaufnahmen. Hauptziel von *SkiScape* ist die Erfassung des Pistenzustandes, der den Skifahrern nach der Analyse in aggregierter Form über Hinweistafeln wieder zur Verfügung gestellt wird. Nutzer können das System aber auch verwenden, um ihre gefahrenen Routen zu protokollieren oder ihre Freunde im Skigebiet zu lokalisieren. *BikeNet* [70] ist ein System für Fahrradfahrer, welches sowohl Daten zur Leistung und Fitness des Fahrers als auch zur Umwelt aufzeichnet. Dabei werden unter anderem die Geschwindigkeit, die zurückgelegte Strecke, die Herzfrequenz und die elektrodermale Aktivität, also die Veränderungen der bioelektrischen Eigenschaften der Haut [30], aufgenommen. Auch *Biketastic* [197] ist ein auf Fahrrädern basierendes Sensornetz, bei dem der Fokus allerdings weniger auf der Leistung des Fahrers als auf der gefahrenen Strecke liegt. *Biketastic* zeichnet dafür die Fahrbahnbeschaffenheit und die Lärmbelastung von gefahrenen Routen auf. Nutzer können zusätzlich, durch das Aufnehmen von Fotos oder Videos, ihre Fahrten dokumentieren und diese mit anderen Nutzern teilen.

### 2.3.2.3 Erfassung von Preisen

Bei der dritten Anwendungsgruppe geht es um die Erfassung von Preisen für Produkte oder Dienstleistungen. Diese erfassen zwar nicht direkt menschliche Aktivitäten oder Verhaltensweisen, da die Preise jedoch von Menschen – den Anbietern – festgelegt werden, kann dies auch als Resultat menschlichen Verhaltens gewertet werden. Im Folgenden werden diese, der Kategorisierung von Kanhere [117] sowie Christin et al. [46] folgend, als menschenzentrierte Anwendungen bezeichnet. Bei

der Gruppe der Preisvergleichs-Anwendungen liefern die Teilnehmer die benötigten Daten, in der Regel sind dies der Preis des Produktes oder der Dienstleistung, sowie der Standort des Anbieters. Bei *MobiShop* [208] müssen die Nutzer aktiv werden und ein Foto ihrer Rechnung machen. Mit Hilfe automatischer Texterkennung (engl. *Optical Character Recognition (OCR)*) werden die Produkte und Preise erfasst. Diese Informationen werden zusammen mit der über GPS bestimmten Position an den Anwendungsserver übertragen, so dass andere Nutzer die Preise vergleichen können. Ein ähnlicher Ansatz wird auch bei *LiveCompare* [57] verfolgt, nur dass statt der Rechnung die Preisschilder des Anbieters fotografiert werden. *PetrolWatch* [63] ist ein System zur Bestimmung der Benzinpreise mit Hilfe von Smartphones und deren Kameras, das jedoch ohne Eingreifen der Nutzer funktioniert. Die Geräte der Teilnehmer müssen hierfür im Auto auf dem Armaturenbrett montiert und mit GPS ausgestattet sein. Kommt ein Nutzer mit seinem Auto in die Nähe einer Tankstelle, wird die *PetrolWatch*-Anwendung automatisch gestartet und es wird versucht, ein Foto der Preistafel aufzunehmen. Gelingt dies, wird der aktuelle Preis mit OCR erfasst und an den Server übermittelt.

### 2.3.3 Umweltzentrierte Anwendungen

Bei den umweltzentrierten Anwendungen gibt es vier Hauptanwendungsszenarien: Die Überwachung und Erfassung der Luftqualität, der Lärmbelastung, der Wittersituation, sowie der Verkehrsbedingungen im Untersuchungsgebiet.

#### 2.3.3.1 Überwachung der Luftqualität

Die Überwachung der Luftqualität wurde bereits in einem der ersten Artikel zu PSNs von Burke et al. [34] als mögliches Anwendungsfeld erwähnt. Im darauf folgenden Jahr wurde dann von Paulos et al. [183] die erste Feldstudie durchgeführt. Hierzu verteilten die Autoren speziell für diesen Einsatz entwickelte Sensorknoten, die mit einem GPS-Empfänger und Sensoren für Kohlenstoffmonoxid, Schwefeldioxid und Stickstoffdioxid ausgestattet waren, an Taxifahrer und Studenten und ließen sie damit zwei Wochen lang die Luftqualität messen. Auch bei *Pollution-Spy* [119] soll die Luftqualität erfasst und auf einer Karte visualisiert werden. Allerdings sind hier die entsprechenden Gassensoren mittels Bluetooth mit den Mobiltelefonen der Nutzer gekoppelt. Hierdurch können zum einen die Nutzer die erfassten Werte direkt auf ihren Endgeräten betrachten, zum anderen können die Geräte die gesammelten Daten auch direkt an den Anwendungsserver übermitteln. Neben der von Paulos et al. durchgeführten Feldstudie [183] gibt es einige Projekte, die auf die Mobilität von Fahrzeugen setzen. Dadurch wird zum einen erreicht, dass schon mit einer relativ geringen Anzahl an Messknoten eine große örtliche Abdeckung erreicht werden kann, da die Knoten sich in der Regel schneller bewegen und größere Strecken zurücklegen als Nutzer ohne Fahrzeug. Zum anderen können an Fahrzeugen auch leichter Sensoren angebracht werden, da z. B. Größe und Gewicht nicht so

entscheidend sind. Bei *Haze Watch* [219, 262] werden die Sensorknoten an Autos montiert und die erfassten Messdaten per Bluetooth an die Smartphones der Fahrer übertragen. Über das Mobilfunknetz werden die Daten an den zentralen Server übermittelt. Im Rahmen des *OpenSense*-Projektes in Zürich [140] werden die mobilen Sensorknoten auf Straßenbahnen angebracht. Die Erhebung erfolgt somit auf festgelegten Strecken und zu bekannten Zeiten. Eine weitere Gruppe von Arbeiten versucht die Mobilität von Fahrradfahrern für die Messungen auszunutzen. Das schon im vorherigen Abschnitt beschriebene *BikeNet* [70] nimmt, neben den Informationen zum Fahrerlebnis, auch Daten zur Luftverschmutzung und Lärmbelastung auf. Beim *Copenhagen Wheel* [177] werden Sensoren zur Messung von Kohlenmonoxid, Stickoxiden, Temperatur, Geräuschpegel und Feuchtigkeit in das Hinterrad eingebaut. Die so erfassten Umweltinformationen werden auch per Bluetooth an das Smartphone des Fahrers übertragen, über das die Daten mit der Gemeinschaft geteilt werden können. Auch *NoxDroid* [256] ist ein auf Fahrrädern basierendes Sensorsystem zur Überwachung der Luftqualität in Städten. Die Funktionsweise ist dieselbe, wie bei den zuvor beschriebenen Systemen, nur dass *NoxDroid* speziell für Android-Geräte entwickelt wurde. Durch die Entwicklung kompakter und günstiger Sensorknoten ist die ursprüngliche Idee von der Verwendung vorhandener Mobiltelefone und deren Nutzer zur Datenerfassung wieder stärker in den Fokus gerückt. Im *CitiSense*-Projekt [20, 175] wurde eine Sensorbox entwickelt, die an Taschen oder Rucksäcken befestigt werden kann und die Messdaten an die Mobiltelefone der jeweiligen Nutzer sendet. Auch Hasenfratz et al. [88] konzentrieren sich auf Smartphone-basierte Messungen. Das von den Autoren entwickelte *Gas-Mobile* ist eine kleine Messeinheit mit einem Ozonsensor, welche per USB direkt an das Smartphone angeschlossen wird.

### 2.3.3.2 Erfassung von Umgebungslärm und -geräuschen

Die Erfassung der Lärmbelastung war eines der ersten Anwendungsfelder partizipativer Sensornetze. Da bei diesen typischerweise das eingebaute Mikrofon zur Bestimmung der Umgebungslautstärke verwendet wird, eigneten sich hierfür schon die ersten, im Vergleich zu heutigen Geräten relativ einfachen, internetfähigen Mobiltelefone. Neben der Erfassung der Umgebungslautstärke muss jedoch auch eine örtliche Einordnung der Messung erfolgen, weshalb auch dieses Anwendungsfeld erst mit dem Aufkommen GPS-fähiger Mobiltelefone wirklich relevant wurde. Das von Bilandzic et al. vorgestellte *Laermometer* [29] war eine der ersten Anwendungen in diesem Bereich. Bei dieser wird in festgelegten Intervallen die Lautstärke automatisch analysiert, der Ort bestimmt und die Daten an den Server übermittelt. Zusätzlich können die Nutzer Kommentare zu den Messungen abgeben, so dass z. B. besondere Vorkommnisse erklärt werden können. Es handelt sich um eine opportunistische Datenerfassung mit optionalen, partizipativen Elementen, vgl. Abschnitt 2.3.1. Die gesammelten Daten werden schließlich in Form einer Lärm-Karte visualisiert. In der Folge wurde eine Vielzahl von Anwendungen entwickelt, die Mobiltelefone zur Erfassung der Lärmbelastung verwenden und prinzipiell sehr ähnlich zu

dem beschriebenen Laermometer sind. Zu diesen zählen *MobGeoSen* [120], *NoiseSpy* [118, 119], *NoiseTube* [155], *Ear-Phone* [193], *NoiseMap* [206] und auch *NoizCrowd* [240]. Letzteres unterscheidet sich dadurch von dem zuvor beschriebenen System, dass statt der tatsächlich erfassten Messdaten nur daraus abgeleitete Modelle als Ergebnis dargestellt werden, um das Fehlen von Messdaten und die hohe Fluktuation aufzufangen. Neben der reinen Bestimmung der Umgebungslautstärke können PSNs auch genutzt werden, um Lärmquellen zu lokalisieren und gegebenenfalls zu verfolgen. Ein Beispiel hierfür ist *MetroTrack* [6]. Bei diesem bestimmen die Mobiltelefone gemeinsam den Ort der Lärmquelle und versuchen auch ihre weitere Bewegung vorherzusagen.

Eine weitere Einsatzmöglichkeit der Mikrofone besteht in der Bestimmung des aktuellen Nutzerkontextes. Obwohl dies im Grunde eine menschenzentrierte Anwendung ist, können die Kontextinformationen bei einer nutzerübergreifenden Erfassung auch verwendet werden, um Ereignisse und Veranstaltungen in dem Untersuchungsgebiet zu erfassen. Mit *SoundSense* [146] wird exemplarisch ein Audio-Klassifizierungssystem vorgestellt, das Töne und Geräusche lernt bzw. erkennt und durch diese den Kontext der Nutzer folgert. Das et al. [54] stellen ein *Party Thermometer* vor, bei dem geprüft wird, ob in der Umgebung laute Musik gespielt wird. Ist dies der Fall, geht die Anwendung davon aus, dass sich der Nutzer auf einer Party befindet und befragt ihn zur Stimmung der Veranstaltung. So kann beispielsweise eine Karte mit Stimmungsbarometer aller Partys einer Stadt erstellt werden.

### 2.3.3.3 Bestimmung der Wettersituation

Ein immer wichtiger werdendes Anwendungsfeld ist die Bestimmung der aktuellen Wettersituation im Untersuchungsgebiet. Frühe Generationen von Smartphones hatten noch keine Sensoren, die wirklich hilfreich bei der Bestimmung des Wetters waren. Daher wurden verschiedene Ansätze entwickelt, um anhand der verfügbaren Informationen dennoch auf die jeweilige Wettersituation schließen zu können. Eine der ersten Anwendungen in diesem Bereich ist *Ikarus* [112], ein PSN für Gleitschirmpiloten. Bei diesem werden die GPS-fähigen Mobiltelefone genutzt, um Ort und Höhe der Piloten zu bestimmen. Durch die Aggregation der Daten können Steig- und Sinkwerte sowie die Dichte an Gleitschirmfliegern in einer Region bestimmt werden. Somit kann auf indirektem Wege eine Karte mit thermischen Bedingungen erstellt werden. Overeem et al. [178] präsentieren einen Ansatz zur Bestimmung der Temperatur in Städten. Statt jedoch eigens dafür bestimmte Thermometer zu verwenden, schätzt das System die Lufttemperatur mit Hilfe der Batterietemperatur der teilnehmenden Smartphones. Durch die Mittelung über eine Vielzahl an Messungen kann die reale Temperatur relativ gut angenähert werden, obwohl die Geräte nicht für solche Messungen konzipiert wurden. Neuere Mobiltelefone haben eine Vielzahl an Sensoren integriert, von denen sich auch einige für die Erfassung des Wetters eignen. Das Samsung GALAXY S4 [275] ist sowohl mit einem Barometer als auch mit einem kombinierten Temperatur- und Luftfeuchtigkeitssensor

ausgestattet. Mit Geräten dieser Art sind folglich PSNs, welche die Wettersituation auf direktem Wege bestimmen, möglich. Einen hybriden Ansatz stellen Niforatos et al. [174] mit ihrer *Atmos*-Anwendung vor. Dabei werden zum einen die verfügbaren Sensoren des Gerätes verwendet und Luftdruck, Luftfeuchtigkeit, Temperatur und Helligkeit bestimmt. Zum anderen wird der Nutzer gebeten, Eingaben zum aktuellen Wetter an seinem Standort zu machen. Über die Benutzeroberfläche der Anwendung kann dieser angeben, wie er das Wetter wahrnimmt, wie warm und wie windig es sich für ihn anfühlt. Die kombinierten Daten der Sensoren und des Nutzers werden dann an den Anwendungsserver übertragen. Im Gegenzug erhält der Nutzer eine Kurzzeitprognose für seinen Aufenthaltsort.

#### 2.3.3.4 Überwachung der Verkehrssituation

Das vierte Anwendungsfeld sind Sensornetze zur Überwachung der Verkehrssituation im Untersuchungsgebiet. Hierzu zählen einerseits Anwendungen, die den Verkehr erfassen, andererseits aber auch Anwendungen, die Informationen zum Umfeld des Verkehrs, z. B. den Straßenzustand, aufnehmen. Eines der ersten Systeme in diesem Bereich war das von Hull et al. [105] vorgestellte *CarTel*. Bei diesem wurden Autos als mobile Sensorknoten verwendet, um Daten zum aktuellen Verkehr zu erfassen und Geschwindigkeiten, Verzögerungen und kritische Verkehrspunkte zu bestimmen. Neben den Verkehrsdaten führt CarTel auch WLAN-Messungen durch und sammelt die Daten des On-Board-Diagnosesystems. Darüber hinaus gibt es zahlreiche weitere Ansätze, welche die aktuelle Verkehrssituation bestimmen. Dazu zählen unter anderem das Konzept der *Virtual Trip Lines* [94, 97], das *VTrack*-System [226] und auch die Verkehrsschätzung des *Mobile-Millennium*-Projektes [241]. Inzwischen sind auch kommerzielle Anbieter in diesem Bereich vertreten. So verwendet Google die GPS-Informationen der Nutzer um eine Ebene zur Verkehrslage in ihren Karten anzuzeigen [258]. Das von Monhan et al. vorgestellte *Nericell* bzw. *TrafficSense* [165–167] versucht gefährliche oder kritische Bereiche zu identifizieren und analysiert, an welchen Stellen stark gebremst oder viel gehupt wird. Eine weitere Gruppe von Anwendungen hat das Ziel, den Straßenzustand zu erfassen und Schäden wie Schlaglöcher zu melden. Zu diesen zählt neben Nericell auch *Pothole Patrol* [72], der *Road Bump Monitor* [54] und die von Mednis et al. [160] vorgestellte Schlagloch-Erkennung. Allerdings beschränken sich die Anwendungen nicht nur auf die Verkehrs- und die Straßensituation. *GreenGPS* [78] erfasst über das On-Board-Diagnosesystem den Benzinverbrauch der Autos und bildet diesen auf die Straßen des Gebietes ab. Über einen Navigationsdienst können die Nutzer dann, basierend auf diesen Daten, die kraftstoffsparendste Route zu ihrem Ziel bestimmen. Thiagarajan et al. [225] stellen einen Ansatz vor, bei dem Nutzer des öffentlichen Nahverkehrs die aktuelle Position ihres Fahrzeugs bestimmen und veröffentlichen, so dass andere Nutzer diese auf einer Karte mitverfolgen und unter anderem die Ankunftszeit an ihrer Haltestelle genauer abschätzen können. Ferreira et al. [76] stellen weitere potentielle Anwendungsfälle vor. Einerseits nutzen sie die per GPS bestimmten Pfade der Nutzer, um automatisch Karten zu erstellen. Ander-



(a) Visualisierung der interpolierten Messdaten für das Untersuchungsgebiet (b) Bestimmung der Route mit (rosa) und ohne (blau) Eco Routing

Abbildung 2.2: Screenshots der Plattform für Umweltbeobachtungen

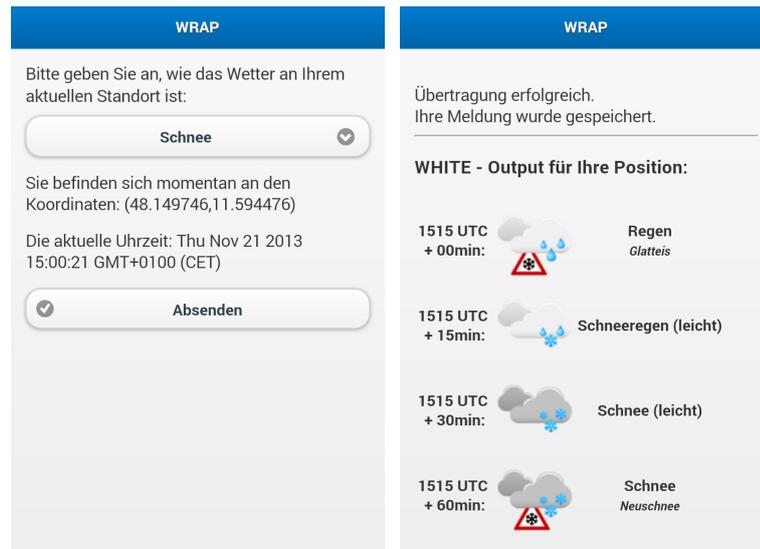
seits verwenden sie die in einigen Autos vorhandenen *Dashcams* zur Erkennung der Kennzeichen vorbeifahrender Autos, um diese Informationen beispielsweise für die Strafverfolgung zur Verfügung zu stellen.

### 2.3.4 Entwickelte Anwendungen und prototypische Umsetzungen

Neben den vorgestellten Arbeiten wurden auch eigene Anwendungen und prototypische Umsetzungen partizipativer Sensornetze realisiert, die im Folgenden kurz beschrieben werden.

In Kooperation mit Siemens wurde eine Plattform zur Erfassung, Anreicherung und Visualisierung von Umweltbeobachtungen entwickelt [286]. Die Idee der entwickelten Plattform ist es, dass Nutzer mit ihren mobilen Endgeräten Umweltmessungen durchführen, wie beispielsweise Messungen der Luftqualität, und die Plattform aus diesen Daten durch Interpolation flächendeckende Visualisierungen für das Untersuchungsgebiet bestimmt, siehe Abbildung 2.2(a). Darüber hinaus ermöglicht die Plattform zusätzliche Dienste, wie das vorgestellte *Eco Routing* (dt. „ökologische Routenplanung“). Der Eco-Routing-Dienst bestimmt statt der kürzesten oder schnellsten Route die schadstoffärmste Route, d. h. die Strecke, bei der die Luft am wenigsten verschmutzt ist. Ein solcher Dienst ist für alle Verkehrsteilnehmer sinnvoll, die den Luftschadstoffen direkt ausgesetzt sind, also beispielsweise Fahrradfahrer und Fußgänger. Abbildung 2.2(b) zeigt an einem Beispiel zwei unterschiedliche Routen. Die Standardroute (blau) führt entlang der Hauptverkehrsader und somit auch durch Gegenden mit schlechter Luftqualität. Im Gegensatz dazu wählt das Eco Routing eine Strecke (rosa), bei der die Nutzer weniger Schadstoffen ausgesetzt sind.

In einer weiteren Arbeit [291] wurde ein System zur Erkennung von Stress entwickelt. Das Ziel hierbei war, möglichst unaufdringlich in Echtzeit das Stresslevel



(a) Ansicht zum Erfassen einer Wettermeldung (b) Ausgabe der erhaltenen lokalen Wetterprognosen

Abbildung 2.3: Screenshots der mobilen WRAP-Anwendung

der Nutzer zu erfassen. Hierzu wurde ein System zur Aktivitätserkennung mit einem Brustgurt zur Messung der Herzfrequenz gekoppelt, wobei die Messdaten an ein Smartphone übermittelt und von diesem ausgewertet wurden. Mit Hilfe dieses Prototyps konnte eine Unterscheidung zwischen den Zuständen „Stress“ und „Kein Stress“ bei den Nutzern erkannt werden. Ein mögliches Anwendungsszenario für diese Anwendung stellen Notfall- und Rettungseinsätze dar, bei denen das eingesetzte Personal oftmals an die körperlichen Belastungsgrenzen kommt. Durch eine kontinuierliche Überwachung des Stresszustandes könnte das Personal beispielsweise bei erheblicher Überlastung gewarnt und vor daraus eventuell resultierenden Fehlentscheidungen geschützt werden.

Eine dritte Anwendung wurde in Kooperation mit dem *Deutschen Zentrum für Luft- und Raumfahrt (DLR)* im Bereich des *Nowcasting* entwickelt. Nowcasting bezeichnet in der Meteorologie die Beschreibung des aktuellen Wetters und Kurzzeitprognosen, d. h. Vorhersagen für die nächsten Stunden [156, 281]. Da jedoch die örtliche und zeitliche Auflösung von Wettermeldungen mittels stationärer Wetterstationen meist relativ gering ist, ist eine flächendeckende und verlässliche Kurzzeitprognose nur schwer möglich. Ziel des in [293] vorgestellten Systems ist es, mit Hilfe eines PSNs zusätzliche Wetterinformationen zu sammeln und diese zu nutzen, um die erstellten Prognosen des Nowcasting-Systems *WHITE* zu validieren. Hierfür wurde die *Weather Reporting Application (WRAP)*; dt. „Wettermeldungsapplikation“ entwickelt. WRAP ermöglicht es den Nutzern, ohne großen Aufwand die aktuelle Wettersituation zu melden. Hierzu wählen die Nutzer lediglich aus einer Liste von Wetterarten das passende aus und schicken diese Informationen zusammen mit ihrem automatisch bestimmten Standort an das System. Abbildung 2.3(a) zeigt die

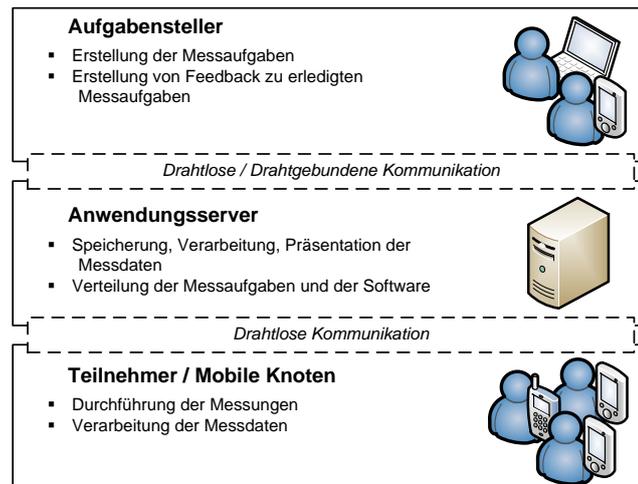


Abbildung 2.4: Architektur von PSNs (angelehnt an Wang et al. [176])

entsprechende Benutzerschnittstelle. Nach der Übertragung der Wettermeldung erhalten die Nutzer im Gegenzug die Prognosen des Nowcasting-Systems für ihren Standort für die nächste Stunde, siehe Abbildung 2.3(a). Die Anwendung wurde in einer Testphase zunächst nur an fachkundige Nutzer mit meteorologischem Grundwissen verteilt. Dazu zählten neben den Mitarbeitern des DLR am *Institut für Physik der Atmosphäre* unter anderem auch Beschäftigte des *Deutschen Wetterdienstes (DWD)* und des *Meteorologischen Instituts* der LMU München, sowie Angestellte des *Flughafens München* und der *Deutschen Flugsicherung (DFS)*. Langfristig soll die Anwendung auch der Allgemeinheit zur Verfügung gestellt werden.

## 2.4 Architektur von partizipativen Sensornetzen

Die Architektur von PSNs kann je nach Einsatzzweck variieren. So kann z. B. für manche Anwendungen ein dezentrales System [75] sinnvoller als der im Allgemeinen verwendete zentrale Ansatz sein. Bei anderen Architekturen wird die Speicherung der Messdaten und die Informationsgewinnung aus diesen zum Schutze der Privatsphäre der Nutzer getrennt [79].

Im Gegensatz zu den spezialisierten Architekturen, wird in der vorliegenden Arbeit eine sehr typische und allgemeine Architektur von PSNs angenommen, wie sie unter anderem in [46, 135, 176] beschrieben wird. Abbildung 2.4 zeigt die verschiedenen Komponenten eines solchen Netzes und deren Aufgaben. Dabei wird in der Regel zwischen drei Komponenten unterschieden:

**Aufgabensteller:** Die Aufgabensteller sind Nutzer, die durch die Messdaten des PSNs Informationen erhalten möchten. Hierfür erstellen sie Messaufgaben, die von einzelnen Messungen bis hin zu dauerhaften Kampagnen reichen können, und übermitteln diese an den Anwendungsserver. Aufgabensteller

müssen dabei nicht notwendigerweise die Aufgaben selbst erstellen, sondern können sich auch für die Ergebnisse laufender, öffentlicher Kampagnen interessieren. In diesem Fall abonnieren sie lediglich die Ergebnisse für eine schon existierende Aufgabe. Daneben können die Aufgabensteller typischerweise auch die erhaltenen Messdaten bewerten und Feedback geben.

**Anwendungsserver:** Die zentrale Komponente in PSNs ist der Anwendungsserver, der sowohl Dienstanbieter als auch Vermittler zwischen den Aufgabenstellern und den Teilnehmern ist. In der Rolle des Ersteren übernimmt der Anwendungsserver die Speicherung der erfassten Messdaten, die eventuell nötige Verarbeitung und auch die Präsentation der Ergebnisse, z. B. in Form von Visualisierungen. Damit aber überhaupt Messdaten erfasst werden, muss der Anwendungsserver zwischen Aufgabenstellern und Teilnehmern vermitteln. Daher ist der Anwendungsserver auch für die Verteilung der entsprechenden Software zum Aufzeichnen der Daten verantwortlich. Zudem muss der Server, wenn ein Aufgabensteller Messaufgaben erstellt hat, diese an entsprechende Teilnehmer übermitteln und von diesen erledigen lassen, vgl. Abschnitt 3.2.

**Teilnehmer / Mobile Knoten:** Die Teilnehmer sind die Nutzer, die mit ihren Geräten die Messungen durchführen. Im Folgenden werden die Begriffe Teilnehmer und mobile Knoten synonym verwendet und beinhalten jeweils den Verbund von einem Nutzer und seinem, mit Sensoren ausgestatteten mobilen Endgerät. Die Hauptaufgabe ist die Erfassung der Messdaten. Zusätzlich kann aber, aus Privatsphäre- oder Effizienzgründen, auch eine Verarbeitung der Daten auf Seiten der mobilen Knoten notwendig sein.

De Cristofaro und Soriente [55] zählen zudem noch die Mobilfunk- und Netzbetreiber als vierte Komponente zu einem PSN. Da diese aber keine aktive Rolle haben und ihnen unter Umständen die Existenz eines solchen Netzes gar nicht bewusst ist, werden sie bei der Architektur in Abbildung 2.4 nur als notwendige Kommunikationsinfrastruktur und nicht als eigene Komponente dargestellt. Gleichwohl sind die Netze für die Realisierung eines PSNs sehr wichtig. Vor allem das drahtlose Kommunikationsnetz, über das die mobilen Knoten angebunden sind und ihre Messdaten an den Anwendungsserver kommunizieren, beeinflusst die Leistung eines PSNs [200]. Geschieht diese Kommunikation über ein Mobilfunknetz mit hoher Abdeckung, erhält der Anwendungsserver quasi in Echtzeit Messdaten und kann infolgedessen nahezu Live-Dienste anbieten. Wird für die Kommunikation der Messdaten hingegen auf weniger flächendeckende Netze, wie etwa WLAN-basierte Funknetze, zurückgegriffen, kann es zu erheblichen Verzögerungen zwischen der Aufzeichnung und der Übertragung der Messdaten kommen.

Demgegenüber ist die Kommunikation zwischen den Aufgabenstellern und dem Anwendungsserver unkritisch. Da die Aufgabensteller nicht im Untersuchungsgebiet unterwegs sein müssen, können diese ihren Aufenthaltsort und damit auch ihre Kommunikationsverbindung zum Anwendungsserver frei wählen. Zudem müssen

sie auch nicht notwendigerweise mobil sein und können sich daher auch über drahtgebundene Kommunikation mit dem Anwendungsserver austauschen.

## 2.5 Herausforderungen partizipativer Sensornetze

PSNs stellen ein neues Paradigma für die Erfassung von Messdaten dar. Sie bauen auf einer existierenden Infrastruktur auf, zweckentfremden private Mobiltelefone und sind von der freiwilligen Teilnahme der Nutzer abhängig. Hieraus resultiert eine Vielzahl von Herausforderungen, sowohl technischer als auch nicht-technischer Art. Im Folgenden sollen diese in einem kurzen Überblick dargestellt werden, wobei dieser Abschnitt sich, nach einer kurzen Betrachtung nicht-technischer Probleme, schwerpunktmäßig mit den technischen Herausforderungen befassen wird.

### 2.5.1 Nicht-technische Herausforderungen

Die wohl größte nicht-technische Herausforderung resultiert aus der Freiwilligkeit der Teilnahme. Damit überhaupt Messdaten erfasst werden, müssen die Nutzer motiviert werden, ihre Geräte zur Verfügung zu stellen und somit zur Erfüllung einer Messaufgabe beizutragen. Zudem muss auch eine möglichst große Anzahl an Nutzern daran teilnehmen, um aussagekräftige Daten aus dem PSN zu erhalten. Aufgrund der Wichtigkeit dieses Themas haben sich bereits viele Wissenschaftler damit auseinandergesetzt und verschiedene Lösungsansätze konzipiert. Bei einigen Anwendungen, wie *EpiCollect* [1], wird auf altruistisch motivierte Nutzer gebaut. In dem genannten Fall sind es Nutzer, deren persönliches Interesse es ist, die Natur und das Ökosystem zu erhalten und durch die Datenerfassung ihren Teil zum Umweltschutz beizutragen. Andere Ansätze versuchen die Nutzer durch die Bereitstellung von Zusatzinformationen zur Mitwirkung zu animieren. So werden den Teilnehmern bei BikeNet [70] oder Biketastic [197] persönliche Analysen ihrer sportlichen Leistung bereitgestellt, in der Hoffnung, dass die Nutzer im Gegenzug die erfassten Daten auch mit der Gemeinschaft teilen. Bei LiveCompare [57] wird ein Preisvergleich zu einem Produkt nur dann angezeigt, wenn der Nutzer zu diesem Produkt auch selbst einen Preis beigetragen hat. Eine weitere Gruppe von Ansätzen versucht die Nutzer mit Belohnungen zur Teilnahme zu motivieren. Dies kann sowohl durch spieltypische Elemente, wie Ranglisten oder Leistungsabzeichen, als auch durch Geld- oder Sachpreise geschehen. Bei *Waze*, einer Verkehrs- und Navigationsanwendung, können die Nutzer z. B. Gewinne erhalten, wenn sie bestimmte, virtuell markierte Orte mit ihrem Fahrzeug anfahren [280]. Durch dieses Konzept können einerseits neue Teilnehmer für die Anwendung gewonnen werden, andererseits kann die Mobilität der teilnehmenden Nutzer in gewissem Maße gesteuert werden. Eine Belohnung ist natürlich auch in monetärer Form möglich. Reddy et al. [194] schlagen sogenannte *Micro-Payments* vor, also die Zahlung von Kleinstbeträgen für den Beitrag zur Erfüllung einer Messaufgabe.

Eine weitere Herausforderung ist die Aufklärung der Gesellschaft über PSNs. Gerade nach Bekanntwerden großflächiger Abhöraktionen im Rahmen der NSA-Affäre [204] stehen viele Bürger dem Erfassen von Messdaten mit Hilfe privater Mobiltelefone kritisch gegenüber. Shilton [215] fordert daher, dass die Gesellschaft über Gefahren und Möglichkeiten unterrichtet werden müsse. Es solle für die Nutzer möglich sein, vertrauenswürdige Anwendungen zu erkennen und zu verstehen, worauf sie sich mit der Teilnahme an einem PSN einließen. Hierzu zählt Shilton unter anderem, dass es für die Nutzer klar erkenntlich sein müsse, wie lange Daten gespeichert und ob diese an Dritte weitergegeben werden.

## 2.5.2 Technische Herausforderungen

Auch auf der technischen Seite gibt es eine Vielzahl von Herausforderungen die es zu lösen gilt, bevor PSNs großflächig etabliert und sinnvolle Informationen aus diesen extrahiert werden können. In Anlehnung an Kanhere [117] sollen im Folgenden die wichtigsten davon dargestellt werden.

**Lückenhafte Abdeckung des Untersuchungsgebietes:** Damit eine Messung an einem Ort erfolgen kann, muss ein Nutzer nicht nur dort vorbeikommen, sondern sein Endgerät muss zu diesem Zeitpunkt auch für eine Messung bereit sein, d. h. es muss eingeschaltet sein, genügend Energieressourcen besitzen und die entsprechende Anwendung oder den Dienst zur Datensammlung gestartet haben. Da die Mobilität der Nutzer nicht gesteuert werden kann, führt dies in der Regel zu einer lückenhaften Abdeckung des Untersuchungsgebietes. Auch eine große Zahl an Teilnehmern kann diese Problematik nicht vollständig beheben, weswegen PSNs Lösungsansätze hierfür bereitstellen müssen.

Rana et al. [193] haben für die Erstellung von Lärm-Karten einen Algorithmus entwickelt, der Ansätze aus dem Bereich des *Compressive Sensing* (dt. „komprimiertes Abtasten“) verwendet, um die reale Lärmbelastung anhand relativ weniger Messungen zu rekonstruieren. Ein orthogonaler Ansatz hierzu ist es, Messungen in seltener besuchten Gegenden höher zu belohnen [230]. Ist der daraus resultierende Nutzen für die Teilnehmer hoch genug, weichen diese eventuell von ihren normalen Mobilitätspfaden ab und lassen sich vom PSN so leiten, dass die Lücken in der Abdeckung verkleinert werden.

**Kontext- und Aktivitätserkennung:** Viele der in Abschnitt 2.3 vorgestellten Anwendungen versuchen den Kontext oder die Aktivität der Nutzer zu erfassen. So zielt CenceMe [162] darauf ab, sowohl die physische als auch die soziale Aktivität der Nutzer zu bestimmen. Hierfür werden verschiedene Sensoren und verfügbare Informationen verwendet, um mit Hilfe von Algorithmen aus dem Bereich des maschinellen Lernens die gewünschte Information zu extrahieren. Viele der bisherigen Entwicklungen in diesem Bereich sind entweder sehr speziell für den jeweiligen Anwendungsfall optimiert oder noch nicht

wirklich robust genug, um eine zuverlässige, flächendeckende Erfassung ermöglichen zu können.

Neben dem Kontext des Nutzers ist auch der des Gerätes von entscheidender Bedeutung. So sind Schadstoff- oder Lärmmessungen nur dann sinnvoll, wenn das messende Mobiltelefon sich im richtigen Kontext und nicht in einer Tasche des Nutzers befindet. Eine Herausforderung ist demnach, verlässlich zu erkennen, ob der Kontext eines Gerätes den Anforderungen der Anwendung des PSNs genügt [134].

**Vertrauenswürdigkeit der Daten:** In PSNs werden Messungen mit privaten Mobiltelefonen durchgeführt. Dieser Umstand führt zu verschiedenen Unsicherheiten bei den erfassten Daten, im Vergleich zu traditionellen Sensornetzen, vgl. Abschnitt 2.2. Auf der einen Seite führt die Heterogenität der Geräte [3] zu Ungenauigkeiten bei den Messungen. Diese Unterschiede können aber durch regelmäßige Kalibrierungen, siehe Abschnitt 4.4, abgeglichen werden. Wie schon beim vorherigen Punkt beschrieben, muss zudem sichergestellt werden, dass nur unter sinnvollen Kontextbedingungen durchgeführte Messungen beigetragen werden.

Neben der durch die Geräte eingeführten Ungenauigkeit können auch die Nutzer selbst ein Problem für die Vertrauenswürdigkeit der Daten darstellen. Da prinzipiell jeder an einem PSN teilnehmen kann, können auch Nutzer Messdaten beitragen, die auf Grund von eigenen Interessen die Daten mutwillig manipulieren. Kanhere [117] führt hierfür das Beispiel eines Maklers an, der beispielsweise die Lärmbelastung nach unten manipulieren könne, um die von ihm angebotenen Wohnobjekte zu höheren Preisen verkaufen oder vermieten zu können. Daher müssen PSNs die erhaltenen Daten prüfen, z. B. mittels einer Ausreißerererkennung, und im Falle von inkorrekten Daten die Reputation der entsprechenden Nutzer heruntersetzen bzw. diese für die zukünftige Teilnahme sperren [103, 150].

**Schonung der Energieressourcen:** Eine wichtige Herausforderung in PSNs ist die Schonung der Energieressourcen. Wengleich die Geräte durch die Nutzer regelmäßig aufgeladen werden, ist ein sparsamer Umgang mit vorhandenen Energieressourcen essentiell. Die teilnehmenden Nutzer haben ihre Mobiltelefone nicht für die Verwendung in PSNs angeschafft, sondern möchten diese primär für die Kommunikation nutzen.

Daher müssen die Entwickler von PSNs darauf achten, dass die stattfindende Datenerfassung die eigentliche Nutzung der Mobiltelefone nicht beeinträchtigt. Bei PSNs wird Energie für die Erfassung, die Verarbeitung und die Übermittlung der Messdaten verbraucht. Daher ist es sinnvoll, Ansätze in PSNs zu integrieren, die in jeder dieser Phasen die benötigte Energie reduzieren.

**Schutz der Privatsphäre:** Auch der Schutz der Privatsphäre der Nutzer ist eine essentielle Herausforderung [3]. Da die privaten Geräte der Nutzer verwendet werden, enthalten die Messdaten auch, teils explizit, teils implizit, In-

formationen über den Nutzer und sein Verhalten. Daher sind Mechanismen notwendig, welche die teilnehmenden Nutzer davor schützen, überwacht und ausspioniert zu werden, wie etwa eine verifizierte, aber dennoch anonyme Meldung der erfassten Daten [217].

Bei der Umsetzung der Privatsphäre-Mechanismen fordern Christin et al. [46] die Nutzer stärker in den Fokus zu stellen. So ist die Benutzerfreundlichkeit auch bei den Privatsphäre-Einstellungen ein zentraler Aspekt. Außerdem soll der Schutz der Privatsphäre messbar werden, so dass den Nutzern ein bestimmtes Privatsphäre-Level garantiert werden kann.

### 2.5.3 Bewertung

Die zuvor beschriebenen Herausforderungen, sowohl technischer als auch nicht-technischer Art, sind allesamt wichtig und müssen gelöst werden, bevor ein großflächiger Einsatz partizipativer Sensornetze möglich ist. Wie zu Beginn des Abschnittes schon erwähnt, stehen im Rahmen dieser Arbeit vor allem die technischen Probleme im Fokus.

Die technischen Herausforderungen unterscheiden sich deutlich in der Dringlichkeit. Die drei zuerst genannten Herausforderungen gehen alle davon aus, dass PSNs etabliert sind und Messdaten erfasst werden. Eine lückenhafte Abdeckung bedeutet, dass zumindest für Teile des Untersuchungsgebietes Messungen erfolgt sind. Das Problem der Kontext- bzw. der Aktivitätserkennung wird auch erst relevant, wenn Nutzer wirklich Daten beitragen. Und auch das Problem der Vertrauenswürdigkeit tritt erst auf, wenn PSNs wirklich im Einsatz sind.

Die beiden zuletzt genannten Herausforderungen, die Schonung der Energieressourcen und der Schutz der Privatsphäre, sind hingegen elementare Herausforderungen, die über eine erfolgreiche Etablierung von PSNs entscheiden können. Nehmen die Nutzer die Anwendungen des PSNs als energieverschwendend wahr, werden diese nicht lange teilnehmen und eine langfristige Etablierung wird nicht erfolgen. Genauso ist der Schutz der Privatsphäre eine notwendige Voraussetzung für den Aufbau und den Erhalt von PSNs. Erst wenn die Privatsphäre gewahrt bleibt, werden die Nutzer bereit sein, ihre Geräte für ein PSN zur Verfügung zu stellen.

Lösungsansätze für diese beiden Herausforderungen stellen somit die Grundlage für die erfolgreiche Etablierung von PSNs dar. Daher liegt der Fokus dieser Arbeit auch auf diesen zwei Herausforderungen, welche in den folgenden Abschnitten adressiert werden:

- Abschnitt 3 befasst sich mit Verfahren für eine effiziente Datenerfassung, um die Energieressourcen der Geräte zu schonen.
- Abschnitt 4 widmet sich der Privatsphäre der Teilnehmer und stellt Konzepte zum Schutze dieser vor.

## 2.6 Zusammenfassung

Dieses Kapitel befasste sich mit den Grundlagen partizipativer Sensornetze und lieferte einen Überblick über relevante Begriffe, Anwendungen und Herausforderungen.

Anfangs wurde ein Überblick über verschiedene alternative Bezeichnungen für PSNs gegeben, bevor die im Kontext dieser Arbeit relevanten Definitionen und Begriffe erläutert wurden. Anschließend erfolgte eine Abgrenzung gegenüber traditionellen drahtlosen Sensornetzen.

Der darauf folgende Abschnitt lieferte einen ausführlichen Überblick über Anwendungen für PSNs. Dabei wurden zunächst die verschiedenen Anwendungskategorien beschrieben, bevor eine thematisch sortierte Beschreibung existierender Anwendungen erfolgte. Als Abschluss wurden noch eigene Anwendungen und Prototypen skizziert.

Nach einer Darstellung der typischen Architektur von PSNs und den involvierten Parteien und Komponenten, wurden abschließend die Herausforderungen partizipativer Sensornetze beleuchtet. Dabei wurde sowohl auf technische als auch auf nicht-technische Herausforderungen und Problemstellungen eingegangen. Schließlich folgte eine Bewertung der herausgearbeiteten Herausforderungen, welche als Grundlage für die im Rahmen dieser Arbeit vorgestellten Ansätze dient.



# 3 Datenerfassung in partizipativen Sensornetzen

Wie im vorherigen Kapitel identifiziert, ist die Schonung der Ressourcen ein entscheidender Aspekt in PSNs. Da diese auf freiwillig zur Verfügung gestellten Mobiltelefonen von Nutzern basieren, müssen auch die Interessen der Gerätebesitzer im Vordergrund stehen.

Ein zentraler Punkt bei der Verwendung von Mobiltelefonen ist, dass diese vom Nutzer nicht für die Verwendung in PSNs angeschafft werden. Primärer Nutzungszweck ist in der Regel die Kommunikation, d. h. das Führen von Telefonaten und der Austausch von Textnachrichten über SMS, E-Mail oder Instant-Messaging-Dienste. Dieser primäre Nutzungszweck darf durch die Datenerfassung eines PSNs nicht merklich beeinträchtigt werden, da die Nutzer sonst ihre freiwillige Teilnahme widerrufen würden. Hinzu kommt, dass die Nutzer in der Regel keinen direkten Nutzen aus der von ihnen durchgeführten Datenerfassung ziehen, sondern meist nur indirekt, von den durch die Gemeinschaft der Teilnehmer erfassten Daten profitieren.

Auf der anderen Seite ist eine effiziente Datenerfassung auch für die Betreiber der PSNs von entscheidender Bedeutung. Da PSNs auf eine möglichst große Teilnehmerzahl abzielen, müssen die Betreiber dafür sorgen, dass sie die vielen aus den großen Nutzerzahlen resultierenden Anfragen bedienen können und müssen zudem für die großen Datenmengen, die durch die Messungen entstehen, gerüstet sein. Da dies natürlich Kosten für die Betreiber verursacht, streben auch diese nach einer möglichst effizienten Datenerfassung, um unnötige Kosten, z. B. durch redundante Messdaten, zu vermeiden. Für eine erfolgreiche Etablierung von PSNs ist es daher unumgänglich, Ansätze zur effizienten Datenerfassung in diesen umzusetzen. Da die Ressourcenschonung ein so essentieller Aspekt ist, sollten in allen Bereichen und Phasen der Datenerfassung in PSNs entsprechende Ansätze verwendet werden.

Dieses Kapitel ist wie folgt aufgebaut: In Abschnitt 3.1 werden zunächst existierende Ansätze zur Schonung der Ressourcen dargestellt, wobei sowohl Konzepte für traditionelle als auch für partizipative Sensornetze betrachtet werden. Darauf folgen zwei neue Lösungsansätze für PSNs: Als erstes wird in Abschnitt 3.2 ein Ansatz zur energieeffizienten Verteilung von Messaufgaben vorgestellt. Danach folgt die Beschreibung eines Ansatzes zur lokalen Gruppierung von Messknoten in Abschnitt 3.3. Abschließend wird in Abschnitt 3.4 das Kapitel zusammengefasst.

## 3.1 Klassifizierung existierender Ansätze

Im Folgenden soll ein Überblick über existierende Ansätze zur effizienten Datenerfassung in PSNs gegeben werden. Wie schon zuvor dargestellt, stellen PSNs eine spezielle Form von drahtlosen Sensornetzen dar. Da Letztere schon länger Gegenstand der Forschung sind und auch als Vorläufer von PSNs gesehen werden können, sollen zunächst Arbeiten aus diesem Bereich vorgestellt werden, bevor anschließend speziell für PSNs entwickelte Ansätze vorgestellt werden.

Die Ansätze für WSNs unterscheiden sich merklich von denen für PSNs, da der hauptsächliche Fokus von ersteren auf der Schonung der Energieressourcen liegt. Typischerweise ist das Ziel der Ansätze die Verlängerung der Lebensdauer der Sensornetze. In PSNs sind die Probleme etwas vielschichtiger, da die Endgeräte nicht dediziert für die Erfassung von Messdaten im Einsatz sind, sondern vielmehr private Geräte sind, die nebenher noch Messdaten erfassen sollen.

Dabei treten neue Herausforderungen auf, wie die Instruktion der Knoten über die Messaufgaben, die nicht steuerbare Mobilität der Knoten und die zu beachtende Privatsphäre der Nutzer. Letztere wird zunächst in dem folgenden Abschnitt ausgelassen und detailliert in Abschnitt 4 betrachtet.

### 3.1.1 Ressourcenschonung in drahtlosen Sensornetzen

Die ressourcenschonenden Ansätze in WSNs können nach Anastasi et al. [13] in drei Gruppen unterteilt werden, siehe Abbildung 3.1: Aktivitätszyklen, bei denen die Sensorknoten zwischen Aktiv- und Schlafmodus wechseln; Verfahren zur effizienten Datenreduktion und -erfassung, bei denen versucht wird, möglichst wenige Daten zu erfassen und zu übermitteln; sowie Ansätze, welche die Mobilität von Knoten ausnutzen, um Energie einzusparen.

#### 3.1.1.1 Aktivitätszyklen

Die erste Gruppe von Verfahren verwenden sogenannte *Aktivitäts-* oder auch *Arbeitszyklen*. Im Englischen wird dieses Vorgehen in der Regel mit *Duty Cycling* [13] oder *Activity Scheduling* [10] bezeichnet. Hierbei werden die Sensorknoten, um Energie zu sparen, regelmäßig in einen Schlafmodus geschickt. Im Schlafmodus werden alle Funktionseinheiten, die nicht für den Betrieb benötigt werden, vorübergehend abgeschaltet, was zu einer Verlängerung der Batterielaufzeit führt. Die Verfahren können nochmals in zwei Untergruppen unterteilt werden: Einerseits Verfahren, die versuchen trotz dieser Aktivitätszyklen die Topologie und die Konnektivität des Sensornetzes zu erhalten. Andererseits Verfahren mit dem primären Ziel, die Energiereserven der Knoten zu schonen.

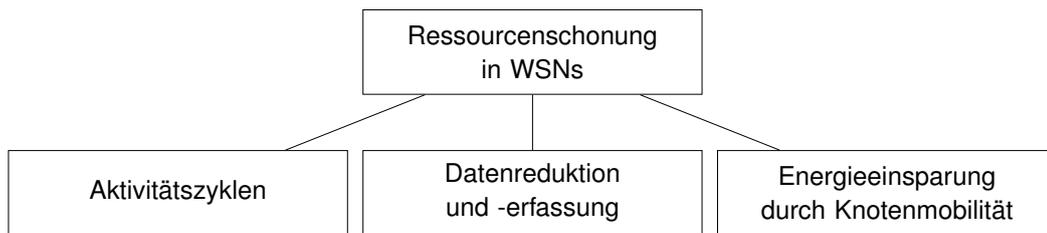


Abbildung 3.1: Klassifizierung ressourcenschonender Ansätze für WSNs in Anlehnung an Anastasi et al. [13]

Ein Beispiel für die erste Untergruppe stellt der Ansatz *Geographic Adaptive Fidelity (GAF)* [243] dar. Das Ziel von GAF ist es, möglichst viele Knoten in den Schlafmodus zu versetzen, ohne dabei die Effizienz des Routings im Sensornetz zu beeinflussen. Erreicht wird dies durch die Einteilung des Netzes in ein virtuelles Raster, ein sogenanntes *Grid*, in dem jeder Knoten einer Gridzelle mit jedem Knoten der Nachbarzellen kommunizieren kann. Haben die Knoten die Kommunikationsreichweite  $R$ , muss für die Seitenlänge  $r$  des Grids gelten:  $r \leq \frac{R}{\sqrt{5}}$ . Pro Zelle ist dann immer nur ein Knoten aktiv, während die restlichen Knoten im Schlafmodus ihre Ressourcen schonen können. Durch dieses Vorgehen wird erreicht, dass zu jeder Zeit über jede Gridzelle geroutet werden kann. Allerdings müssen die Knoten über ihre Position Bescheid wissen. Dafür müssen diese entweder mit GPS ausgestattet sein, was jedoch aus Kosten- und Energiegründen meist nicht sinnvoll ist, oder es wird nur eine Teilmenge der Knoten damit ausgestattet. Im zweiten Fall müssen die Knoten ohne GPS spezielle Verfahren verwenden, um ihre Position zu approximieren [136].

Ein anderer Ansatz, der lediglich Konnektivitäts- an Stelle der Ortsinformationen verwendet, ist *Span* [42]. Bei diesem wird eine Teilmenge der Knoten zu sogenannten *Backbone*-Knoten, die dauerhaft aktiv sind. Die restlichen Knoten sind im Schlafmodus, prüfen jedoch periodisch, ob sie als Backbone-Knoten benötigt werden. Letzteres ist der Fall, wenn zwei oder mehr Nachbarknoten weder direkt noch über Backbone-Knoten kommunizieren können. Damit die Knoten dies prüfen können, tauschen sie regelmäßig Konnektivitätsnachrichten aus. Diese Nachrichten enthalten den Status des Knotens, eine Liste der Backbone-Knoten, mit denen dieser Knoten verbunden ist, und die Liste der Nachbarn.

Bei der zweiten Untergruppe gibt es Ansätze, bei denen Knoten nur bei Bedarf, d. h. wenn Daten versendet werden sollen, aufgeweckt werden. Ein Beispiel hierfür ist *Sparse Topology and Energy Management (STEM)* [205]. Bei STEM befinden sich die Knoten die meiste Zeit im Schlafmodus, wachen aber periodisch für kurze Zeit auf und prüfen, ob Daten an sie gesendet werden sollen. Falls ein Knoten Daten übertragen will, versendet dieser hintereinander mehrere Beacons, so dass andere Knoten diese während ihrer Zeit im Wachzustand empfangen können. STEM geht dabei von Knoten aus, die mit zwei Funkmodulen ausgestattet sind, und eines die-

ser Module nur für den Aufweck-Vorgang genutzt wird. Die Energieeinsparungen gehen in diesem Fall aber auf Kosten der Latenz, wodurch dieser Ansatz vor allem für Sensornetze mit wenigen und unkritischen Daten geeignet ist. Neben diesen bedarfsorientierten Ansätzen gibt es auch Verfahren, bei denen die Aktivitätszyklen mittels eines Plans festgelegt werden. Eines der einfachsten Konzepte ist hier das *Fully Synchronized Wakeup Pattern* [124], bei dem sich alle Knoten gleichzeitig im Aktiv- bzw. Schlafmodus befinden. Je nach Einsatzszenario gibt es von diesem Konzept verschiedene Optimierungen, wie das *Ladder Wakeup Pattern* [124]. Dieses Schema reduziert die Latenz in Netzen, die hauptsächlich in eine Hierarchie-richtung kommunizieren, z. B. von den Messknoten zur Basisstation. Bei beiden Ansätzen müssen allerdings die Uhren der Knoten synchronisiert sein, vgl. [71].

Alternativ wurden verschiedene Ansätze konzipiert, die keine Synchronisierung erfordern. In [228] beschreiben die Autoren verschiedene asynchrone Aktivitäts-schemata für den IEEE-Standard 802.11. Auch das *Asynchronous Wakeup Protocol (AWP)* [250] stellt sicher, dass benachbarte Knoten sich in endlicher Zeit erkennen und miteinander kommunizieren können, ohne dass ein zeitlicher Abgleich stattfinden muss. Bei *Random Asynchronous Wakeup (RAW)* [181] ist dies ebenfalls nicht nötig. RAW ist gedacht für Netze mit hoher Knotendichte und basiert darauf, dass Daten in der Regel über mehrere Pfade übertragen werden können. Die Knoten können zu zufälligen Zeiten aufwachen, da das Routing-Protokoll dafür sorgt, dass Pakete einen Weg zu ihrem Ziel über die jeweils aktiven Knoten finden.

Neben diesen eigenständigen Verfahren, gibt es auch Ansätze, bei denen das Konzept der Aktivitätszyklen schon auf der Ebene des *Medium Access Control (MAC)*, also der Koordination des Medienzugriffs, integriert wird. Bei MAC-Protokollen unterscheidet man zwischen wettbewerbsfreien und wettbewerbsbasierten Verfahren. Ein Beispiel für erstere sind Ansätze, bei denen Knoten in Clustern gruppiert werden und der sogenannte *Clusterhead* Zeitslots für Übertragungen an die restlichen Knoten des Clusters vergibt [2, 249]. Ein weiteres Beispiel für einen wettbewerbsfreien Ansatz ist *Traffic-Adaptive Medium Access (TRAMA)* [192]. Bei TRAMA werden die Zeitpunkte von Übertragungen über lokal bestimmte Pläne festgelegt, welche an die jeweiligen Nachbarknoten kommuniziert werden. Dadurch ist jedem Knoten bekannt, wann er sich in den Schlafmodus begeben kann. Beispiele für wettbewerbsbasierte Ansätze sind *Sensor-MAC (S-MAC)* [247] und *Berkeley-MAC (B-MAC)* [187]. Bei S-MAC wachen die Knoten einer Region des Netzes immer synchron auf, wofür zu Beginn der aktiven Zeit Synchronisations-Nachrichten verschickt werden. Übertragungen werden mittels *RTS-CTS-Handshake* initiiert, wodurch Kollisionen vermieden werden sollen. Nicht beteiligte Nachbarknoten können sich, sobald sie den Handshake mithören, in den Ruhemodus versetzen. Bei B-MAC befinden sich die Knoten die meiste Zeit im Schlafmodus, prüfen aber periodisch, ob Aktivität auf dem Kanal stattfindet. Knoten, die Daten senden möchten, müssen dies den schlafenden Knoten anzeigen, indem sie eine Präambel voraus schicken, die mindestens einen Schlaf-Wach-Zyklus lang ist. Dadurch ist sichergestellt, dass die Nachbarknoten bei der periodischen Prüfung Aktivität erkennen und

somit bei der eigentlichen Übertragung wach sind. Einen hybriden Ansatz stellt *Zebra-MAC (Z-MAC)* [198] dar. Bei diesem werden Zeitslots für Knoten reserviert, können von diesen aber auch wieder für die anderen Knoten freigegeben werden. Hierdurch wird erreicht, dass sich Z-MAC bei großem Datenaufkommen wie ein Zeitmultiplexverfahren (engl. *Time Division Multiple Access (TDMA)*) verhält, bei wenigen Datenverkehr sich jedoch einem klassischen wettbewerbsbasierten Verfahren annähert.

### 3.1.1.2 Datenreduktion und -erfassung

Eine Möglichkeit der Datenreduktion besteht darin, die in einem Sensornetz anfallenden Rohdaten auf Seiten der Sensorknoten bereits zu verarbeiten und zu aggregieren. Dieses Vorgehen bezeichnet man als *In-Network Processing* [13] bzw. *In-Network Aggregation* [74]. *Tiny AGgregation (TAG)* [153] ist ein Ansatz, der diese Technik verwendet und speziell für Monitoring-Netze mit baumartigen Topologien konzipiert wurde. TAG reduziert in diesen die übertragene Datenmenge durch die Aggregation der Daten bei jedem Zwischenknoten auf dem Pfad zum Empfänger der Daten. Fasolo et al. [74] bieten einen guten Überblick über weitere Ansätze und Techniken in diesem Bereich.

Eine zweite Möglichkeit die Datenmenge zu reduzieren ist die Verwendung von Kompressionsalgorithmen. Viele der existierenden Ansätze sind aufgrund der Komplexität nicht für Sensorknoten mit ihren begrenzten Ressourcen geeignet [22, 129]. *Sensor LZW (S-LZW)* [202] ist eine speziell für Sensornetze entwickelte Variante des *Lempel-Ziv-Welch-Algorithmus (LZW)* [234], welcher die Daten mit Hilfe eines selbst erzeugten Wörterbuchs komprimiert. Um bei fehlerhaften Übertragungen die Energiekosten für wiederholte Übermittlung der Daten gering zu halten, wurden bei S-LZW sowohl die Datenblöcke als auch das erzeugte Wörterbuch in der Größe begrenzt. Neben der Kompression der einzelnen Knoten kann durch *Distributed Source Coding (DSC)* auch die Korrelation der Messdaten mehrerer Sensorknoten für die Kompression ausgenutzt werden [242].

Eine dritte Gruppe von Ansätzen verringert die übertragenen Daten mittels Vorhersagen. Typischerweise verwenden die Sensorknoten ein Modell, das sie mit dem Empfänger der Daten, der Datensinke, teilen. Passen die erfassten Messwerte nicht zu diesem Modell, senden die Knoten die Messdaten an die Senke oder bestimmen ein neues Modell und übermitteln dies. Bei diesen Ansätzen kann man zwischen stochastischen-, Zeitreihen-basierten oder algorithmischen Prognosen unterscheiden [13]. Bei Ken [47] werden für die Vorhersagen dynamische probabilistische Modelle verwendet, so dass nach einer Lernphase eine Wahrscheinlichkeitsdichtefunktion für die Wertebelegung der Variablen bestimmt werden kann. Ein interessanter Zeitreihen-basierter Ansatz wurde von Le Borgne et al. [137] vorgestellt. Bei diesem können die Knoten selbstständig das für ihre Messungen passendste Modell aus einer Menge möglicher Modelle auswählen, wodurch die Komplexität je nach aktuellem Verhalten der Messgröße angepasst wird. PREMON [84] ist ein

Beispiel für einen algorithmischen Ansatz. Bei diesem werden die erfassten Messdaten zu einem Zeitpunkt als „Pixel“ eines Bildes interpretiert. Die Überwachung eines Gebietes ist somit eine Folge von Bildern und wird als „Sensorfilm“ betrachtet. Auf diesen werden dann, den Konzepten der MPEG-Videokompression [138] sehr ähnliche Verfahren angewandt, welche die zeitliche und örtliche Korrelation benachbarter Pixel zur Kompression ausnutzen.

Neben der Möglichkeit die übertragene Datenmenge zu reduzieren, gibt es auch die Option, die erhobene Datenmenge, d. h. die Anzahl der durchgeführten Messungen, zu reduzieren. Wie auch bei den vorherigen Ansätzen soll hierbei ein möglichst geringer Informationsverlust entstehen. Diese Verfahren werden in der Regel unter dem Begriff „energieeffiziente Datenerfassung“ zusammengefasst und werden von Raghunathan et al. [190] in drei Untergruppen unterteilt: adaptives, hierarchisches und Model-basiertes Sampling.

Beim adaptiven Sampling werden die Messungen nicht wie gewöhnlich in periodischen Zeitintervallen an bestimmten Orten durchgeführt, sondern die zeitliche bzw. örtliche Messauflösung wird auf Basis der erfassten Messwerte und deren Entwicklung angepasst. Soll die zeitliche Auflösung angepasst werden, werden Kriterien für die Erhöhung bzw. die Reduzierung dieser benötigt. Alippi et al. [11] verwenden in ihrem Ansatz eine modifizierte Variante der kumulativen Summe (engl. *cumulative sum (CUSUM)*) [25], um die Sampling-Frequenz der Messungen anzupassen, wohingegen Jain und Chang [108] die Messhäufigkeit basierend auf dem Schätzfehler des eingesetzten Kalman-Filters [113] einstellen. Ein Ansatz zur Anpassung der örtlichen Auflösung stellen Willet et al. mit ihrem *Backcasting*-Konzept vor [239]. Dabei wird in zwei Schritten vorgegangen: In einem initialen Durchgang wird nur eine bestimmte Teilmenge der Sensorknoten für die Messungen aktiviert, wodurch ein grobes Abbild des zu messenden Phänomens konstruiert werden kann. Im zweiten Schritt aktiviert der Server zusätzliche Knoten in den Regionen, in denen die Korrelation der Messwerte gering ist. Somit erhält man in Regionen, in denen sich die Messgröße stark verändert, eine feinere örtliche Auflösung, wohingegen in Gegenden mit konstanten Werten die Auflösung geringer ist. Bei dem für mobile Messroboter entwickelten Ansatz von Rahimi et al. [191] wird ähnlich vorgegangen: Bewegt sich der mobile Knoten in eine Region mit großen räumlichen Schwankungen der Messwerte, steigert dieser die Messfrequenz, wodurch bei gleichbleibender Geschwindigkeit die örtliche Auflösung erhöht wird.

Um Energie zu sparen, werden beim hierarchischen Sampling Sensoren verschiedenen Typs genutzt, welche die zu beobachtende Messgröße in unterschiedlicher Qualität erfassen. Einfache Sensoren, die wenig Energie verbrauchen, haben in der Regel eine begrenzte Auflösung bzw. Genauigkeit. Dagegen benötigen komplexere und genauere Sensoren meist deutlich mehr Energie für ihre Messungen. Beim hierarchischen Sampling wird daher mit den einfachen Sensoren ein grobes Bild der Umgebung erfasst und bei Bedarf werden komplexere Sensoren mit einer genaueren Untersuchung beauftragt. Kijewski-Correa et al. [126] haben ein Konzept mit hierarchischem Sampling für die Überwachung struktureller Schäden einer Brücke

entwickelt. Dabei werden zwei Arten von Sensoren verwendet: (1) *MESO*-Knoten mit Beschleunigungssensoren und (2) *MICRO*-Knoten mit Dehnungssensoren. Mittels der *MESO*-Knoten wird das Bauwerk regelmäßig auf Schäden geprüft, während die *MICRO*-Knoten sich im Schlafmodus befinden. Wird ein mögliches Problem erkannt, wird zunächst eine Gegenprobe mit den umliegenden *MESO*-Knoten durchgeführt. Falls der Verdacht eines Schadens sich erhärtet, werden schließlich die *MICRO*-Knoten der entsprechenden Region aktiviert, mit deren Hilfe ein Schaden sicher erkannt werden kann. Auch Tseng et al. haben mit *iMouse* [229] ein Überwachungssystem entwickelt, welches aus zwei Arten von Sensorknoten besteht. Zum einen wird eine Vielzahl einfacher statischer Knoten eingesetzt, um Lichtintensität, Lautstärke und Temperatur zu messen. Zum anderen steht eine kleine Anzahl mit Kameras und WLAN ausgestatteter mobiler Sensorknoten zur Verfügung. Diese werden dann aktiv, wenn die statischen Knoten ein ungewöhnliches Ereignis erkennen. Ist dies der Fall, werden die mobilen Sensoren zu der Gefahrenstelle geschickt und können mit ihrer Kamera ein detaillierteres Bild der Situation an die Basis übertragen. Analog wird bei dem von Sing et al. [218] vorgeschlagenen Ansatz vorgegangen. Zur Bestimmung der Intensität des einfallenden Lichts am Boden eines Waldes wird in der ersten Stufe eine „globale“ Kamera verwendet, mit der die ungefähre Lichtstärke im Untersuchungsgebiet erfasst wird. Die Ausgabe der Kamera wird schließlich genutzt, um mobile Knoten, die mit Sensoren zur Erfassung photosynthetisch nutzbarer Strahlung ausgestattet sind, an die relevanten Orte zu schicken und dort exakte Messungen durchzuführen.

Modell-basierte Sampling-Verfahren gehen sehr ähnlich wie die zuvor beschriebenen Vorhersage-basierten Übertragungsverfahren vor. Bei letzteren liegt der Fokus jedoch auf der Reduzierung der kommunizierten Daten, d. h. die Sampling-Frequenz wird nicht angepasst, wodurch auch die Energie, die für die Messungen benötigt wird, unverändert bleibt [13]. Modell-basierte Sampling-Verfahren zielen jedoch darauf ab, auch die Anzahl der Messungen auf ein Minimum zu reduzieren. Dafür wird bei diesen Ansätzen versucht, die Messwerte mit einem gewissen Konfidenzintervall durch Messungen an anderen Orten, zu anderen Zeitpunkten und/oder mit anderen Sensortypen vorherzusagen [190]. In [60] stellen die Autoren *Barbie-Q (BBQ)* vor, ein System zur effizienten Beantwortung von Messanfragen mit Hilfe eines probabilistischen Modells. Anhand dieses Modells entscheidet die Planungskomponente, wie eine Anfrage beantwortet werden soll. Wird z. B. die Temperatur einer Region angefragt, kann ein Plan zur Energiereduktion sein, dass gewisse Knoten nur die Batteriespannung messen, welche von der Umgebungstemperatur abhängt, und nur eine kleine Menge an Knoten einen wirklichen Temperatursensor verwenden. Ein anderes Konzept ist das von Padhy et al. vorgestellte *Utility-based Sensing and Communication protocol (USAC)* [179]. Bei diesem bestimmt jeder Knoten ein Modell basierend auf den zuletzt durchgeführten Messungen. Die Sampling-Frequenz wird immer dann angepasst, wenn eine neue Messung nicht in das vorhergesagte Konfidenzintervall fällt. In diesen Fällen wird angenommen, dass ein sogenannter Phasenwechsel stattgefunden, d. h. dass die Messgröße sich deutlich verändert hat. USAC setzt die Sampling-Frequenz dabei auf die Maximal-

frequenz, um genügend Messwerte der neuen Phase zu erhalten und daraufhin ein neues Vorhersagemodell erstellen zu können.

### 3.1.1.3 Energieeinsparung durch Knotenmobilität

Die bisher vorgestellten Ansätze basieren mehrheitlich auf Sensornetzen mit statischen Sensorknoten. Sobald sie einmal postiert wurden, können sie ihre Lage nicht mehr verändern. Neben diesen gibt es aber auch mobile Sensorknoten, die sich selbstständig fortbewegen und somit ihre Lage anpassen können. Die Mobilität dieser Knoten kann genutzt werden, um Energie einzusparen. Soll etwa vermieden werden, dass die Energiereserven der Knoten in der Nähe der Senke zu schnell aufgebraucht werden, kann die Mobilität verwendet werden, um diese Knoten durch das Fortbewegen der Senke in eine andere Region des Netzes zu entlasten, vgl. [44]. Prinzipiell kann man zwei Untergruppen von Ansätzen unterscheiden: Einerseits Ansätze, die wie zuvor beschrieben, eine mobile Senke verwenden, um Energie einzusparen; andererseits Ansätze, die mobile Knoten als „Übermittler“ nutzen. Die Übermittler-Knoten sammeln die Daten der anderen Knoten ein und leiten diese an die eigentliche Senke weiter.

Bei Ansätzen, die auf einer oder mehreren mobilen Senken basieren, ist eine entscheidende Frage, wie und wohin diese bewegt werden sollen. Sowohl Gandham et al. [77] als auch Wang et al. [233] nutzen in ihren Ansätzen lineare Optimierung, um zu entscheiden, welche Knoten von der mobilen Senke besucht und wie lange diese bei den verschiedenen Knoten verweilen soll. Das Ziel beider Ansätze ist es, die Lebensdauer des Gesamtnetzes zu maximieren. Die von Basagni et al. entwickelte Heuristik *Greedy Maximum Residual Energy (GMRE)* [24] bewegt die mobile Senke in die Gegenden, in denen die Knoten die höchsten verbleibenden Energiereserven haben. Im Gegensatz zu den vorherigen zentral bestimmten Optimierungen handelt es sich bei GMRE um einen verteilten Ansatz.

Daneben gibt es auch Ansätze, welche die Mobilität der Senke für neue Routing- und Verteilungsverfahren nutzen. Luo et al. [148] haben ein Routing-Konzept für kreisförmige Sensornetze mit mobiler Senke entwickelt. In diesen soll die mobile Senke  $S$  sich auf einem Kreis um den Mittelpunkt  $M$  des Sensornetzes durch das Untersuchungsgebiet bewegen. Sensorknoten im Inneren des durch die Senke geformten Kreises leiten ihre Pakete auf dem kürzesten Weg zur Senke (*Shortest Path Routing*), wohingegen Knoten außerhalb des Kreises die Pakete zunächst kreisförmig in Richtung der mobilen Senke senden (*Round Routing*). Dies geschieht solange, bis diese auf die Verlängerung der Strecke  $\overline{MS}$  treffen. Ab dieser Stelle wird wieder die kürzeste Route zur Senke gewählt. Die Erfahrungen aus dieser theoretischen Arbeit wurden schließlich bei der Konzeption des Routing-Verfahrens *MobiRoute* [149] genutzt. *Two-Tier Data Dissemination (TTDD)* [246] ist ein hierarchisches Konzept für mehrere Quellen und mobile Senken. Für eine effiziente Datenverteilung bauen die Quellen eine Rasterstruktur auf, bei der die Knoten, die den Gitterpunkten des Rasters am nächsten liegen, die Rolle von sogenannten Verteiler-Knoten über-

nehmen. An diese Verteiler-Knoten verschicken die Quellen proaktiv Nachrichten, wenn bei ihnen Informationen vorhanden sind. Will eine mobile Senke diese Informationen erhalten, versendet sie eine Broadcast-Anfrage innerhalb ihrer aktuellen Rasterzelle. Der nächstgelegene Verteiler-Knoten leitet die Anfrage über das Netz von Verteiler-Knoten zur Quelle, welche die Daten über den gleichen Weg an die Senke übermittelt.

Die zweite Untergruppe nutzt die Mobilität einzelner Knoten, um die Daten der anderen Knoten einzusammeln und an die Datensenke weiterzuleiten. Ein Beispiel ist die in [109, 211] vorgestellte Architektur, welche aus drei Ebenen besteht. Zu der obersten Ebene gehören die drahtlosen Zugangspunkte, die über einen direkten Anschluss an das Weitverkehrsnetz verfügen und so Daten an den Anwendungsserver übertragen können. Zu der mittleren Schicht gehören mobile Knoten, die von den Autoren mit *Mobile Ubiquitous LAN Extensions (MULEs)* bezeichnet werden. Diese sammeln die Daten der statischen Sensoren aus der untersten Schicht ein und geben sie bei der Begegnung mit einem Zugriffspunkt an diesen weiter. Bei den MULEs setzen die Autoren dabei auf Objekte, die ohnehin im Untersuchungsgebiet unterwegs sind, wie Personen, Fahrzeuge oder Tiere. Die Mobilität der MULEs kann folglich nicht gesteuert oder vorhergesagt werden. Andere Ansätze verwenden hingegen speziell eingesetzte mobile Knoten, deren Mobilität gesteuert werden kann [121, 220]. Bei diesen wird ein optimaler Pfad bestimmt, so dass die mobilen Knoten an möglichst vielen statischen Knoten vorbeikommen und letztere durch die kurzen Übertragungsstrecken möglichst viel Energie einsparen können.

### 3.1.2 Ressourcenschonung in partizipativen Sensornetzen

Die Ansätze zur Ressourcenschonung in PSNs unterscheiden sich von denen für WSNs. So sind Aktivitätszyklen, bei denen sich Knoten in den Ruhemodus versetzen, in PSNs nicht praktikabel, da die Mobiltelefone der Nutzer für eine durchgehende Erreichbarkeit der Nutzer ununterbrochen aktiv sein müssen. Auch die Ansätze, die auf die Mobilität einiger Knoten setzen, sind nicht auf PSNs übertragbar, da diese von einem Netz von statischen Knoten ausgehen, bei denen nur einige spezielle Knoten mobil sind. Da die Nutzer ihre Geräte in PSNs mit sich führen, sind alle Knoten mobil, allerdings ist deren Mobilität nicht steuerbar.

Bei der Betrachtung partizipativer Sensornetze sind drei Aspekte von zentraler Bedeutung, siehe Abbildung 3.2. Erstens eine effiziente Positionsbestimmung, da alle Knoten mobil sind und deren Positionen für die Erfüllung der Messaufgaben essentiell sind. Zweitens eine effiziente Verteilung der Messaufgaben, um die Knoten über die durchzuführenden Messungen zu instruieren. Drittens wie bei traditionellen Sensornetzen eine effiziente Datenerfassung, um die Erfassung und Übermittlung unnötiger Daten zu vermeiden.

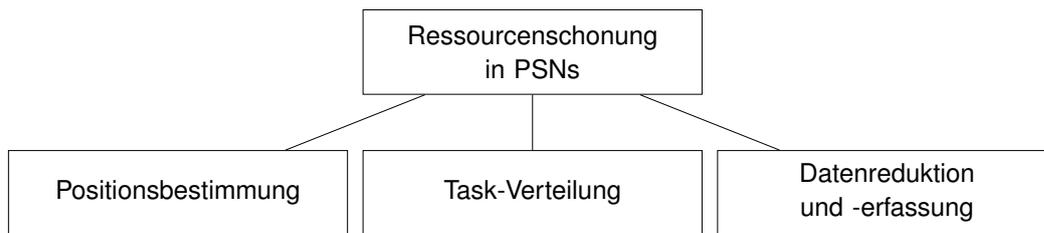


Abbildung 3.2: Klassifizierung ressourcenschonender Ansätze für PSNs

### 3.1.2.1 Positionsbestimmung

Aufgrund der Mobilität der Knoten ist die effiziente Positionsbestimmung ein zentraler Aspekt in PSNs. Die Position der Geräte ist zum einen wichtig für die Verteilung der Messaufgaben, zum anderen zur geografischen Einordnung der erfassten Messwerte.

Ein mögliches Vorgehen besteht darin, die verwendeten Sensoren für die Positionsbestimmung der jeweiligen Situation anzupassen. Bei *V-Track* [226], einem Konzept für die Bestimmung von Verkehrsstörungen mit Hilfe von Mobiltelefonen, kann anstatt der energieintensiven GPS-Positionierung dauerhaft auf WLAN-basierte Ortung umgestellt werden. Hierfür erstellen Endgeräte, die GPS aktiviert haben, eine Datenbank mit verfügbaren WLAN-Zugriffspunkten und speichern deren Position. Andere Geräte können mit dieser Datenbank ihre ungefähre Position bestimmen und mit Hilfe eines *Hidden Markov Model (HMM)* den zurückgelegten Pfad relativ genau bestimmen. Mit *a-Loc* [142] schlagen die Autoren ein Konzept vor, bei dem, basierend auf probabilistischen Modellen für den Ort des Nutzers und den aktuellen Messungenauigkeiten, dynamisch die jeweils energieeffizientesten Sensoren aktiviert werden, welche für die von der Applikation geforderte Genauigkeit benötigt werden.

*Jigsaw* [147] ist ein System für kontinuierliche Messungen mit Mobiltelefonen, welches darauf abzielt, eine Balance zwischen Leistungsanforderungen und Ressourcenschonung zu finden. Dafür verwendet es sensorspezifische Pipelines, mit welchen den jeweiligen Herausforderungen begegnet wird. Für die GPS-Positionierung bestimmt *Jigsaw* mit Hilfe der Beschleunigungssensoren die Aktivität des Nutzers und legt zusammen mit der Messdauer und dem Hardware-Zustand, wie z. B. der noch verfügbaren Batterielaufzeit, eine Sampling-Strategie fest. Ein ähnliches Vorgehen schlagen auch Zhuang et al. [254] vor und beschreiben verschiedene Strategien für eine effiziente Positionsbestimmung. Eine Positionierung wird nur durchgeführt, wenn die Beschleunigungssensoren eine Bewegung melden. Zudem wird, wann immer möglich, GPS durch GSM-Ortung ersetzt. Außerdem wird die Messfrequenz je nach Batterielevel angepasst, d. h. bei geringen Energiereserven wird seltener eine Positionsbestimmung durchgeführt. Bareth und Küpper [21] ha-

ben einen hierarchischen Positionierungsalgorithmus für proaktive standortbezogene Dienste entwickelt, bei dem die Positionierungsmethode dynamisch ausgewählt wird, die am wenigsten Energie benötigt und gleichzeitig noch die benötigte Genauigkeit liefert. Will ein Dienst erkennen, sobald ein Nutzer eine Zielregion erreicht, wird dessen Position zunächst nur auf der Funkzelle basierend bestimmt. Bewegt sich der Nutzer in die Zelle, in der die Zielregion liegt, wird auf WLAN-Positionierung umgeschaltet. Diese wird solange genutzt, bis der Nutzer sich in einem WLAN befindet, das die Zielregion schneidet. Erst im letzten Schritt wird dann auf die sehr energieintensive GPS-Positionierung umgestellt.

Baier et al. [18] haben mit *PSense* ein Positionierungsverfahren für PSNs entwickelt. Bei diesem wird die nächste Position erst dann bestimmt, wenn der Nutzer sich seit der letzten Positionierung so weit bewegt haben kann, dass er sich in der Zielregion befinden könnte. *PSense* nimmt dabei an, dass Nutzer sich auf Straßen bewegen und verwendet daher für die Berechnung der Zeitdauer bis zur Zielregion einen entsprechenden Straßengraphen.

Daneben gibt es auch einige Verfahren, die mit Hilfe von Vorhersagen die Anzahl der Positionsbestimmungen reduzieren. Constandache et al. [50] bestimmen hierfür aus den Bewegungsprofilen der Nutzer Gewohnheiten und typische Pfade. Bewegt sich ein Nutzer auf einem dieser Pfade, wird GPS nur verwendet, um die Vorhersage zu bestätigen oder eine Abweichung von dieser zu erkennen. Bei dem von Kjær-gaard et al. [130] vorgestellten Positionierungssystem *EnTracked* wird die Anzahl der Positionsbestimmungen mit Hilfe einer Bewegungserkennung und einer adaptiven Sampling-Rate reduziert. Anwendungen, die den Pfad des Nutzers verfolgen wollen, stellen eine Anfrage an *EnTracked*, bei der auch ein zulässiges Fehlerlimit angegeben wird. Daraufhin bestimmt das System mittels Beschleunigungssensoren, ob der Nutzer sich bewegt, und falls dies der Fall ist, wird eine Positionierung mit GPS durchgeführt. Der nächste Positionierungszeitpunkt wird dann basierend auf der mit GPS bestimmten Geschwindigkeit ermittelt, d. h. es wird eine Schätzung abgegeben, wann die Position des Nutzers nicht mehr innerhalb des zulässigen Fehlerlimits liegt. Auch Paek et al. [180] verwenden Beschleunigungssensoren zur Bewegungserkennung und Bewegungsprofile für Vorhersagen. Darüber hinaus wird eine Funkturm-Negativliste geführt, mit der die Funktürme von den Regionen markiert werden, in denen GPS eine unzuverlässige Genauigkeit liefert und somit ausgeschaltet bleiben kann. Außerdem findet bei jeder Positionsbestimmung ein Bluetooth-basierter Positionsaustausch statt. Dazu verbreiten die Geräte die neu bestimmte Position per Broadcast an alle Nachbarn, so dass diese sich eine Positionsbestimmung sparen können.

### 3.1.2.2 Task-Verteilung

WSNs werden in der Regel mit einem bestimmten Einsatzzweck aufgebaut und die Sensorknoten entsprechend programmiert. Eine Umprogrammierung der Knoten ist zwar oftmals möglich, allerdings geschieht dies eher selten, vgl. [235]. In PSNs

werden jedoch private Mobiltelefone genutzt, die nicht speziell für die Durchführung von Messungen verteilt wurden. Daher müssen Geräte, die von ihren Nutzern für die Erfassung von Daten im Rahmen eines PSNs zu Verfügung gestellt werden, über die durchzuführenden Messaufgaben informiert werden. Für die Auswahl der zu informierenden Knoten und die Verteilung der Tasks wurden daher verschiedene Ansätze entwickelt. Einen pragmatischen Weg bei der Verteilung von Messanfragen wählen Gaonkar et al. [82]: Anfragen werden an alle mobilen Geräte gesendet, die sich innerhalb einer spezifizierten Region befinden. Dafür muss jedes Gerät, sobald der Nutzer sich bewegt, fortlaufend seinen aktuellen Standort melden.

Daneben gibt es Ansätze, die eine Auswahl der Nutzer anhand verschiedener Kriterien treffen. Bei Cardone et al. [39] werden die Nutzer in einem mehrstufigen Verfahren bestimmt. Zunächst wird die Liste aller Nutzer, zufällig oder basierend auf ihrer Anwesenheit in der Anfrageregion, geordnet. Im nächsten Schritt werden die Nutzer aussortiert, deren Geräte zu geringe Energiereserven aufweisen. Im letzten Schritt wird ein spezifizierter Anteil der Nutzer aus der verbleibenden Liste als Empfänger selektiert. Reddy et al. [195] schlagen zwei verschiedene Ansätze zur Rekrutierung von Teilnehmern in PSNs vor. Bei einem der Ansätze wird die erreichbare Abdeckung durch die Teilnehmer maximiert. Dafür werden die Bewegungsprofile der Nutzer analysiert, dann wird eine Teilmenge bestimmt, so dass die größtmögliche Abdeckung bei gleichzeitig kleinstmöglichen Kosten erreicht wird. Ein zweiter Ansatz nimmt die Auswahl der Nutzer basierend auf der Erfüllung vorheriger Aufgaben vor. Dabei werden sowohl kampagnenübergreifende als auch kampagnenspezifische Metriken genutzt, welche die Qualität und Quantität der Messdaten der Nutzer bewerten.

Andere Ansätze setzen auf eine modellbasierte Auswahl von Nutzern, um die Messaufgaben möglichst effizient zu verteilen. Bei Philipp et al. [184] bestimmt das System ein Modell, das die Korrelationen der Messwerte nahe gelegener Knoten widerspiegelt. Dieses kann genutzt werden, um Messwerte von Knoten abzuleiten und somit die Anzahl der benötigten Messungen zu reduzieren. Baier et al. [17] schlagen ein Konzept vor, bei dem die Empfängermenge mit Hilfe eines probabilistischen Modells ausgewählt wird. Hierzu wird für jeden Nutzer die Wahrscheinlichkeit bestimmt, dass sein Pfad innerhalb des gewünschten Zeitfensters durch die Region des Tasks führt. Mit diesen Wahrscheinlichkeiten wird eine Menge an Nutzern bestimmt, so dass der Task mit hoher Wahrscheinlichkeit erfüllt, gleichzeitig aber möglichst wenig Energie verbraucht wird, d. h. möglichst wenige Nutzer diesen Task erhalten und ausführen.

Einen weiteren, dezentralen Ansatz stellt das von Lu et al. [145] entwickelte *Bubble-Sensing* dar. Bei diesem werden die Aufgaben durch Broadcasts innerhalb der Zielregion verteilt. Knoten erfahren somit nur von existierenden Tasks, wenn sie auch vor Ort sind und die Messung durchführen können. Nur wenn kein Knoten vor Ort ist, der den Broadcast übernehmen kann, müssen die Knoten sich die Tasks vom Anwendungsserver herunterladen.

### 3.1.2.3 Datenreduktion und -erfassung

Wie bei traditionellen Sensornetzen kommt es auch bei PSNs auf eine energieeffiziente Datenreduktion und -erfassung an. Um die zu übertragenen Daten zu reduzieren, können diese, wie in Abschnitt 3.1.1.2 skizziert, weiterverarbeitet, komprimiert oder vorhergesagt werden. Da die Kompressionstechniken jedoch nicht spezifisch für PSNs sind, werden im Folgenden nur Ansätze für die Weiterverarbeitung und die Vorhersage von Messdaten betrachtet.

Rachuri et al. [189] stellen ein Konzept vor, bei dem die mobilen Endgeräte für die Weiterverarbeitung der Messdaten entscheiden, ob diese lokal auf dem Gerät oder auf Seiten des Servers erfolgen soll. Dafür können Regeln spezifiziert werden, anhand derer die Gewichte für Energieschonung, Latenz und der zu übertragenen Datenmenge festgelegt werden. So kann z. B. die Latenz die höchste Priorität erhalten, solange die Batteriereserven ausreichend groß sind, und sobald diese sich dem Ende zuneigen, können die Gewichte zugunsten der Energieschonung umverteilt werden. Bei dem von Lu et al. [146] vorgestellten System SoundSense, das den Kontext der Nutzer auf Basis von Tonaufnahmen bestimmt, wird versucht, die Menge der zu verarbeitenden Daten zu reduzieren. Da die Analyse der Audio-Daten aufwendig ist, wird zunächst geprüft, ob die erfassten Daten relevant sind oder nicht. Dazu werden die Daten blockweise untersucht. Datenblöcke, deren Tonpegel zu gering und bei denen die Entropie im Frequenzbereich zu groß ist, werden ohne Verarbeitung verworfen.

Motegi et al. [169] stellen einen Ansatz vor, der durch Vorhersage von Messwerten Übertragungen einsparen soll. Dabei prüft jedes Gerät zwei Aspekte: Zuerst wird untersucht, ob der neueste Messwert zu der bisherigen Vorhersage passt. Ist dies nicht der Fall, wird der neue Messwert übermittelt und ein neues Vorhersagemodell bestimmt. Liegt der Messwert innerhalb der Toleranzgrenzen der Vorhersage, wird in einem zweiten Schritt geprüft, ob sich durch den Messwert eine Statistik, wie etwa der Mittelwert oder das Maximum der Gesamtwerte, ändern kann. Falls die Wahrscheinlichkeit hierfür hoch ist, wird der Messwert an den Server übermittelt, wenngleich dieser zu dem Vorhersagemodell passt.

Musolesi et al. [171] stellen verschiedene Strategien zur Kommunikation von Messergebnissen an den Server vor, wobei die Ergebnisse aus einer endlichen Menge von Zuständen stammen. Die Autoren entwickeln Strategien für Szenarien mit kontinuierlicher Konnektivität, wie beispielsweise den einer Abstimmung ähnelnden Upload (engl. *Voting based uploading strategy*). Bei diesem Vorgehen wird nur der häufigste Zustand innerhalb eines spezifizierten Zeitfensters übertragen und das auch nur, wenn sich dieser gegenüber dem vorherigen Zeitfenster geändert hat. Des Weiteren wurde auch eine Strategie für Szenarien mit gewollten und nicht gewollten Konnektivitätsunterbrechungen konzipiert. In diesen Fällen verwendet der Server Markov-Ketten [31] zur Vorhersage der Zustände, wozu die Endgeräte den Server regelmäßig über die nötigen Übergangsmatrizen, also die Übergangswahrscheinlichkeiten zwischen den Zuständen, informieren.

Neben den Ansätzen, welche die Übertragung der Daten optimieren, gibt es auch für PSNs verschiedene Konzepte um die Datenerfassung effizienter zu gestalten. Ein Ansatz für adaptives Sampling wird von Rachuri et al. [189] vorgestellt. In ihrem *SociableSense*-System wird zwischen Events, die das System verpassen kann (engl. *missable*), und solchen, die es nicht verpassen sollte (engl. *unmissable*), unterschieden. Basierend auf dieser Unterscheidung erfolgt auch die Anpassung der Sampling-Rate: Wird ein als missable klassifizierter Event erfasst, wird die Messfrequenz reduziert. Wird hingegen ein unmissable Event erkannt, wird die Häufigkeit der Messungen erhöht. Bei Yan et al. [244] wird für jeden möglichen Zustand eine optimale Sampling-Rate definiert. Wird ein Zustand erkannt, wird die Messfrequenz angepasst und so lange beibehalten, bis die Messwerte einen Zustandswechsel andeuten. Ist der Zustand dem System nicht bekannt, werden die Daten mit der maximalen Frequenz erfasst, bis wieder ein bekannter Zustand erreicht ist.

Auch das zuvor vorgestellte Konzept des hierarchischen Samplings, vgl. Abschnitt 3.1.1.2, kann bei PSNs sinnvoll angewandt werden, da Smartphones typischerweise mit unterschiedlichen Sensortypen ausgestattet sind. Daher bietet es sich bei kontinuierlichen Messungen an, diese mit einfacheren Sensoren durchzuführen und energieintensivere Sensoren solange im Ruhemodus zu lassen, bis deren Genauigkeit wirklich benötigt wird. Diese Technik wird bei Nericell [165], einem System zur Überwachung des Straßen- und Verkehrszustandes, eingesetzt. So bestimmt Nericell die Position grundsätzlich mittels GSM-Ortung und verwendet nur dann die energieintensivere Standortbestimmung mittels GPS, wenn sich das Gerät in einen relevanten Bereich bewegt. Auch die aufwendige Erkennung von Hupsignalen wird nur aktiviert, wenn durch die Beschleunigungssensoren ein abrupter Bremsvorgang ermittelt wurde.

Das von Wang et al. [232] entwickelte *Energy Efficient Mobile Sensing System (EEMSS)* ist ein Framework zur automatisierten Erkennung von Zustandsänderungen des Nutzers. Auch hier wird hierarchisches Sampling eingesetzt, d. h. es wird jeweils eine minimale Menge an Sensoren verwendet, um die relevanten Änderungen zu erkennen. Hierfür können in Zustandsbeschreibungen die benötigten Sensoren und auch die Aktivitätszyklen für die jeweiligen Zustände festgelegt werden, wodurch das Framework für verschiedenste Szenarien einsetzbar ist. Einen ähnlichen Ansatz wählen auch Kang et al. [116] für ihr *SeeMon*-System. Auch bei diesem wird Energie gespart, indem die minimale Menge an benötigten Sensoren zu Beantwortung aller Kontextanfragen, das sogenannte *Essential Sensor Set (ESS)*, bestimmt wird.

Daneben gibt es auch Konzepte, die darauf abzielen, Mobiltelefone für den Einsatz als Messknoten zu optimieren. Priyantha et al. [188] schlagen mit *LittleRock* eine spezielle Architektur vor, bei der die Messungen und die Verarbeitung der Messdaten an einen dedizierten, stromsparenden Prozessor ausgelagert werden. Der Stromverbrauch solcher Prozessoren ist sehr gering und vergleichbar mit dem von Sensoren. Zudem können durch die einfachere Hardware-Architektur auch schnellere Wechsel zwischen Aktiv- und Ruhemodus vollzogen werden, wodurch zusätzlich

Energie eingespart werden kann. Auch bei *SpeakerSense* [144] wird eine solche heterogene Multiprozessor-Architektur für die kontinuierliche Aufzeichnung von Daten verwendet.

### 3.1.3 Zusammenfassende Bewertung der Ansätze

In diesem Abschnitt wurden Ansätze zur effizienten Datenerfassung in drahtlosen und partizipativen Sensornetzen vorgestellt. Die Breite des Spektrums existierender Ansätze spiegelt die Bedeutung und die Relevanz des Themas wider. Ohne intelligente und effiziente Konzepte zur Datenerfassung ist eine Umsetzung von Sensornetzen nicht, bzw. nur mit hohen Kosten und Kompromissen, realisierbar. Wenn-gleich die für WSNs vorgestellten Ansätze als Inspiration für PSNs dienen können, sind sie oftmals nicht direkt übertragbar. Wie bereits in Abschnitt 3.1.2 beschrieben, sind weder die auf Aktivitätszyklen basierenden Ansätze noch die auf steuerbaren mobilen Sensorknoten stützende Konzepte in PSNs praktikabel, da die Einsatzbedingungen durch die Verwendung von Mobiltelefonen zu stark divergieren.

Bei den für PSNs entwickelten Konzepten wurden drei verschiedene Themenfelder betrachtet: die Positionsbestimmung, die Task-Verteilung und die Datenerfassung. Bei der Positionsbestimmung gibt es bereits eine Vielzahl von Ansätzen, welche die verschiedensten Einsatzszenarien und Anwendungsfelder gut abdecken. Ein Grund hierfür ist, dass diese Thematik nicht nur für PSNs interessant ist, sondern auch bei anderen Anwendungen von großer Bedeutung ist, wie etwa bei ortsbezogenen Diensten (engl. *Location-based Services (LBS)*) [52].

Bei den anderen beiden Themenfeldern, der Task-Verteilung und der Datenerfassung, gibt es zwar diverse Ansätze, allerdings besteht hier im Vergleich zu der Positionsbestimmung ein größerer Optimierungsbedarf. Neben den sehr rudimentären Ansätzen zur Task-Verteilung gibt es eine Reihe von Ansätzen, die zwar auf eine effizientere Verteilung der Aufgaben abzielen, die aber im Gegenzug nicht die Effektivität garantieren können. Das bedeutet, sie können nicht sicherstellen, dass immer die relevanten Knoten die Tasks erhalten, da diese z. B. auf Vorhersagen mit Hilfe von Bewegungsprofilen [195] oder auf probabilistischen Modellen [17] basieren. Bei der Datenerfassung gibt es zahlreiche vielversprechende Ansätze, wie beispielsweise neue, für PSNs optimierte Architekturen für Endgeräte [188]. Viele bisherige Konzepte sind aber hauptsächlich auf die Optimierung der Datenerfassung eines einzelnen Gerätes fokussiert. Zudem wurden auch diverse Verfahren vorgestellt, die für spezielle Anwendungsfelder, wie etwa für die Aufnahme von Audio-Daten [146], konzipiert wurden. Auch bei den auf hierarchischem Sampling basierenden Konzepten sind für jedes Anwendungsszenario optimierte Regeln nötig.

Hieraus wurden zwei Themen identifiziert, die durch die bisherigen Ansätze nur unzureichend abgedeckt sind und die mit den in den folgenden Abschnitten vorgestellten Konzepten adressiert werden:

- Eine effiziente Task-Verteilung für örtlich und zeitlich beschränkte Messungen, welche die Energiekosten minimiert, aber gleichzeitig die Effektivität des Systems nicht beeinflusst.
- Ein Konzept zur lokalen Gruppierung von Messknoten mit dem Ziel, durch Einsparung redundanter Messungen Nutzer und Betreiber bei großflächigen Messkampagnen zu entlasten.

## 3.2 Ansatz zur energieeffizienten Task-Verteilung

In diesem Abschnitt soll ein Ansatz zur energieeffizienten Task-Verteilung vorgestellt werden. Der Anwendungsbereich dieses Ansatzes umfasst Szenarien mit lokal und zeitlich beschränkten Messungen, d. h. es sollen nur an spezifizierten Orten und zu spezifizierten Zeiten Messdaten erfasst werden. Ein Beispiel hierfür wäre die Aufnahme von Fotos einer kritischen Kreuzung zur Bestimmung der aktuellen Verkehrssituation oder auch die Bestimmung der Geräuschbelastung in bestimmten Gebieten, wie z. B. in der direkten Nachbarschaft eines Krankenhauses.

Bei solchen Anwendungen ist eine effiziente Task-Verteilung sehr wichtig. Wie bereits erläutert, ist eine Steuerung der Mobilität der Knoten nicht möglich. Daher kann nicht garantiert werden, dass eine Messanfrage auch wirklich beantwortet wird. Kommt beispielsweise kein Teilnehmer des PSNs auf die Kreuzung, von der ein Nutzer ein Foto wünscht, ist eine Beantwortung der Anfrage logischerweise nicht möglich. Aus diesem Grund ist es wichtig, dass genügend bzw. die „richtigen“ Nutzer die Tasks erhalten, so dass die Wahrscheinlichkeit der Erledigung der Aufgabe sehr hoch ist. Gleichzeitig muss aber darauf geachtet werden, dass nicht schon durch die Task-Verteilung zu viel Energie auf Seiten der Endgeräte verbraucht wird. Energie wird auf der einen Seite für den Empfang der Aufgaben benötigt, auf der anderen Seite beansprucht auch die Überwachung der erhaltenen Tasks die Energiereserven. Bei letzterem prüft das Gerät systematisch, ob es sich in einer der Regionen befindet, die in den erhaltenen Tasks spezifiziert sind.

Als Lösung für die beschriebene Problematik wird im Folgenden ein robuster und skalierbarer Ansatz für die Task-Verteilung beschrieben. Bei diesem verteilt das PSN die Tasks an alle Knoten, die in die Nähe der jeweiligen Region kommen, versucht aber gleichzeitig durch verschiedene Optimierungen, die benötigte Energie auf Seiten der Endgeräte zu reduzieren.

Dieser Abschnitt basiert auf dem in [302] vorstellten Ansatz und ist wie folgt strukturiert: Zunächst werden in Abschnitt 3.2.1 Problemstellung und Anforderungen beleuchtet. Anschließend beschreibt Abschnitt 3.2.2 das zugrunde liegende Systemmodell, bevor in Abschnitt 3.2.3 das entwickelte Konzept zur Task-Verteilung dargestellt wird. In Abschnitt 3.2.4 wird dieses evaluiert, worauf in Abschnitt 3.2.5 abschließend eine Diskussion des Ansatzes und möglicher Erweiterungen folgt.

### 3.2.1 Problemstellung und Anforderungen

Die Besonderheit partizipativer Sensornetze ist die Verwendung von Mobiltelefonen zur Erfassung von Messdaten. Im Gegensatz zu Sensorknoten eines WSNs, werden die Mobiltelefone von den Nutzern nicht für die Datenerfassung angeschafft. Daher wissen die Geräte per se nicht, welche Messdaten sie erheben sollen.

Sollen die Mobiltelefone in PSNs als Sensorknoten fungieren, müssen sie somit über zu erledigende Tasks informiert werden. In PSNs werden Tasks durch andere Nutzer erzeugt, indem sie eine Task-Anfrage stellen. Will ein Nutzer etwa die Geräuschbelastung einer bestimmten Region untersuchen, kann er eine entsprechende Anfrage stellen. Eine Task-Anfrage ist dabei ein Tupel der Form  $a = \langle i, r, d, m \rangle$ . Dabei beschreibt

- $i$  die Art der Information, die erfasst werden soll, z. B. den Geräuschpegel;
- $r$  die Region, in der die Messungen stattfinden sollen;
- $d$  das Enddatum der Anfrage und somit die Gültigkeitsdauer;
- $m$  die gewünschte Anzahl an Nutzern, die entsprechende Messungen durchführen sollen.

Task-Anfragen können von Nutzern jederzeit erzeugt werden und bleiben solange bestehen, bis entweder das Enddatum überschritten oder die gewünschte Anzahl an Messungen durchgeführt wurde. Sowohl  $d$  und  $m$  sind damit Endkriterien der Anfrage, von denen allerdings nur eines angegeben werden muss. Sind beide angegeben, wird der Endzeitpunkt einer Task-Anfrage nach dem zuerst eintretenden Kriterium bestimmt. Die Menge aktiver Task-Anfragen  $A$  unterliegt, aufgrund des kontinuierlichen Hinzukommens und Wegfallens, einem ständigen Wandel, was wiederum zur Folge hat, dass die Endgeräte auch kontinuierlich über die neuen Aufgaben informiert werden müssen.

Bei der Beantwortung von Tasks wird auf Seiten der mobilen Endgeräte in folgenden Phasen Energie verbraucht:

1. *Empfang von Tasks*: Zunächst müssen die Endgeräte über die zu erledigenden Aufgaben informiert werden, d. h. sie müssen die Tasks erhalten. Die Kommunikation der Tasks kann dabei entweder durch den Server, durch eine sogenannte *Push*-Nachricht, oder durch das Endgerät, durch einfaches Herunterladen der Aufgaben, initiiert werden.
2. *Überwachung der Tasks*: Sobald die Endgeräte Tasks erhalten haben, müssen diese kontinuierlich überprüfen, ob sie eine der Aufgaben erledigen und beantworten können, d. h. ob sie sich in der mit  $a[r]$  spezifizierten Region befinden. Dabei wird die meiste Energie für die Positionsbestimmung, z. B. mittels GPS, benötigt.
3. *Bearbeitung der Tasks*: Sobald ein Knoten sich in der spezifizierten Region einer Task-Anfrage  $a \in A$  befindet, soll der Task bearbeitet, d. h. die in der

Anfrage angegebene Information  $a[i]$  erfasst werden. In dieser Phase hängt die verbrauchte Energie von den zu erfassenden Daten ab, also was und wie lange Daten aufgezeichnet werden sollen.

4. *Versand der Ergebnisse*: Wurden Messdaten für einen Task aufgezeichnet, müssen diese Daten noch übermittelt werden. Auch in dieser letzten Phase hängt die benötigte Energie stark von den erfassten Daten ab. Soll nur ein Geräuschpegel bestimmt werden, müssen deutlich weniger Daten übertragen werden, als wenn ein Foto der Verkehrssituation versendet wird.

Da die in Phase drei und vier benötigte Energie stark von den zu erfassenden Messdaten abhängt, ist bei diesen eine anwendungsspezifische Optimierung am besten geeignet. In dem folgenden Ansatz liegt der Fokus auf den ersten beiden Phasen. Diese werden prinzipiell in allen PSNs benötigt und eignen sich für die Konzeption eines generischen Ansatzes für die Task-Verteilung.

Das Ziel des Ansatzes ist es, die benötigten Positionsbestimmungen und die erforderliche Kommunikation zu reduzieren. Gleichzeitig soll aber die Effektivität des Systems nicht unter diesen Optimierungen leiden, d. h. jeder Nutzer, der in die von einer aktiven Task-Anfrage spezifizierten Region kommt, soll den entsprechenden Task auch erhalten.

### 3.2.2 Systemmodell

In diesem Abschnitt wird das zugrunde liegende Systemmodell beschrieben, auf dem das Konzept für die Task-Verteilung aufbaut. In Abbildung 3.3 sind die Akteure eines PSNs und deren Interaktionen dargestellt. Bei den Akteuren wird zwischen drei Rollen unterschieden, vgl. Abschnitt 2.4:

- *Aufgabensteller*: Aufgabensteller sind Nutzer, die Tasks durch das PSN erledigt haben möchten und daher Task-Anfragen an dieses stellen.
- *Anwendungsserver*: Der Anwendungsserver nimmt die Task-Anfragen der Aufgabensteller entgegen und verteilt die Tasks an die Teilnehmer.
- *Teilnehmer*: Die Teilnehmer des PSNs sind Nutzer, die ihre Geräte als Messknoten zur Verfügung stellen und Tasks erledigen und beantworten.

Der erste Schritt bei der Interaktion dieser Akteure ist das Abschicken von Task-Anfragen durch die Aufgabensteller. Das Verfassen solcher Anfragen kann über eine Web-Schnittstelle oder über eine mobile App erfolgen.

Der Server nimmt diese Anfragen entgegen, führt gegebenenfalls eine Vorverarbeitung durch und verteilt diese an die Teilnehmer des PSNs. Die Verteilung der Task kann auf zwei Weisen geschehen: Zum einen bietet der Server eine Schnittstelle, über welche die Teilnehmer eine Liste aktueller Task abfragen können. Dieses Vorgehen, bei dem die Empfänger den Informationsaustausch initiieren, wird auch mit dem Begriff *Pull-Kommunikation* (von engl. *to pull*, ziehen) bezeichnet. Zum

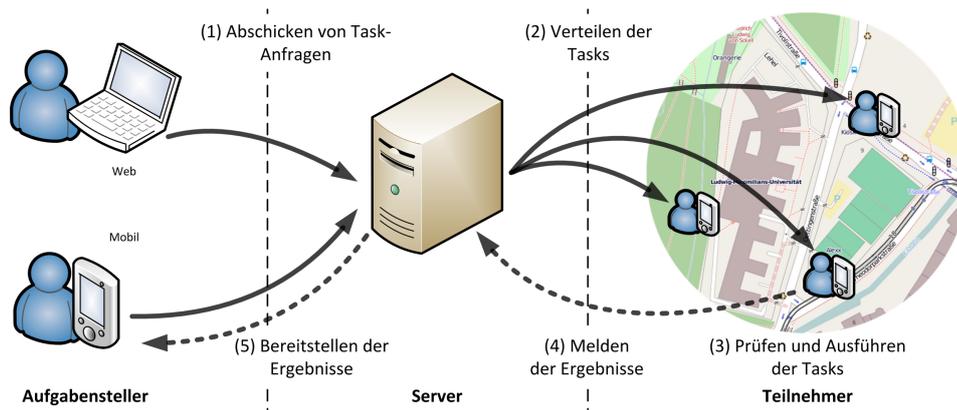


Abbildung 3.3: Akteure und deren Interaktionen innerhalb eines PSNs

anderen kann der Server die Teilnehmer auch direkt kontaktieren, d. h. ihnen direkt Nachrichten schicken. Eine solche, vom Absender initiierte Kommunikation bezeichnet man analog als *Push*-Kommunikation (von engl. *to push*, drücken/schieben/stoßen).

Die Teilnehmer sind Nutzer, die ihre mobilen Geräte für die Erfassung von Messdaten zur Verfügung stellen. Teilnehmer können dabei gleichzeitig auch Aufgabensteller sein, wobei jedoch anzunehmen ist, dass Nutzer nicht ihre eigenen Tasks erledigen und somit nur jeweils eine der beiden Rollen pro Task annehmen. Sei  $U$  die Menge der Teilnehmer, wobei die Anzahl der Teilnehmer  $|U|$  über die Zeit variieren kann. Es wird angenommen, dass jeder Teilnehmer ein mobiles Endgerät hat, welches er dem PSN zur Verfügung stellt. Aufgrund dieser 1:1-Relation werden im Folgenden „der Teilnehmer“ und „das Endgerät des Teilnehmers“ gleichbedeutend verwendet. Des Weiteren wird angenommen, dass jedes Endgerät seine Position, z. B. mittels GPS, bestimmen kann und dass die mögliche Maximalgeschwindigkeit  $v_{max}$  für jeden Teilnehmer bekannt ist. Letztere kann entweder für alle Nutzer auf einen Standardwert eingestellt werden oder über die Zeit automatisch pro Nutzer gelernt werden.

Die Sensoren, die in einem Endgerät des Teilnehmers  $u$  verbaut oder mit diesem verbunden sind, werden mit  $S_u$  bezeichnet. Jeder Sensor  $s \in S_u$  wird dabei mit einem Tupel  $\langle i, r \rangle$  beschrieben, wobei  $i$  die Art der Information angibt, die  $s$  aufzeichnet, und  $r$  die Region, in der diese erfasst werden kann. Ein Task-Anfrage  $a \in A$  wird folglich genau dann vom Teilnehmer  $u$  bearbeitet, wenn  $\exists s \in S_u : (a[i] = s[i] \wedge a[r] \cap s[r] \neq \emptyset)$ .

Wird ein Task durch einen Teilnehmer erledigt, meldet dieser die erfassten Messdaten an den Server. Im letzten Schritt stellt der Server diese zusammengefassten Ergebnisse dem Aufgabensteller zur Verfügung.

### 3.2.3 Konzept

Im Folgenden soll nun ein Konzept vorgestellt werden, dass für die zuvor beschriebene Problemstellung der energieeffizienten Task-Verteilung entwickelt wurde. Das Konzept ist speziell für PSNs konzeptioniert und instrumentalisiert daher zur Erledigung von Messaufgaben die Mobiltelefone der Teilnehmer.

Das Konzept besteht aus drei Komponenten: der zellbasierten Task-Verteilung, dem adaptivem Monitoring mit Bewegungserkennung und der Task-Bündelung. In den folgenden Abschnitten wird jede dieser Komponenten detailliert beschrieben.

#### 3.2.3.1 Zellbasierte Task-Verteilung

Die erste Komponente ist die zellbasierte Task-Verteilung. Das Grundprinzip bei dieser ist, dass nicht alle Tasks an alle Nutzer verteilt werden, da die Kommunikation einen großen Anteil der benötigten Energie darstellt. Stattdessen sollen die Tasks nur an die Nutzer übermittelt werden, die sich in der Nähe der Task-Region befinden. Wird die Anzahl der Tasks reduziert, verringert sich gleichzeitig auch der Aufwand zur Überwachung der erhaltenen Tasks. Nutzer, die während der Lebenszeit einer Messaufgabe nicht in die Nähe der Task-Region kommen, sparen somit Energie durch das Wegfallen der Kommunikation und der Überwachung der Messaufgabe.

Zur Bestimmung der relevanten Aufgaben für die Nutzer baut das Konzept auf den vorhandenen Mobilfunkzellen auf. Die Information über die aktuelle Mobilfunkzelle ist auf Seiten der Endgeräte ohnehin verfügbar und kann bei den meisten Plattformen ohne wirkliche Zusatzkosten bestimmt werden.

Die vorgeschlagene Task-Verteilung ist ein hybrides *Pull-Push*-Konzept, bei dem eine zellbasierte Pufferung (engl. *Caching*) durchgeführt wird. Das bedeutet, dass eine Teilmenge der Tasks durch eine vom Nutzer initiierte Pull-Kommunikation ausgetauscht wird, vgl. Abschnitt 3.2.2, eine andere Teilmenge wird durch eine vom Server ausgelöste, sogenannte Push-Kommunikation ausgetauscht. Dafür wird, basierend auf der Mobilfunkzelle  $c_t$ , in der sich ein Nutzer zum Zeitpunkt  $t$  befindet, eine Menge an Zellen  $C_t$ , mit  $c_t \in C_t$ , bestimmt, für welche die aktuellen Messaufgaben vom Nutzer heruntergeladen werden. Eine Messaufgabe  $a$  wird hierzu allen Zellen  $c_j$  zugeordnet, für die gilt  $c_j \cap a[r] \neq \emptyset$ .

Für die Auswahl von  $C_t$  wurden verschiedene Strategien entwickelt, welche auch in Abbildung 3.4 illustriert sind:

- *Single*: Nutzer laden nur die Task-Liste für die Zelle herunter, in der sie sich aktuell befinden. Es gilt also  $C_t = \{c_t\}$ . Jeder Zellwechsel resultiert folglich auch in einem neuem Task-Download, vgl. Abbildung 3.4(a).
- *Neighbors*: Nutzer laden die Messaufgaben für  $c_t$  und alle Nachbarzellen  $N_{c_t}$  herunter, somit gilt  $C_t = \{c_t\} \cup N_{c_t}$ . In Abbildung 3.4(b) umfasst  $C_t$  jeweils

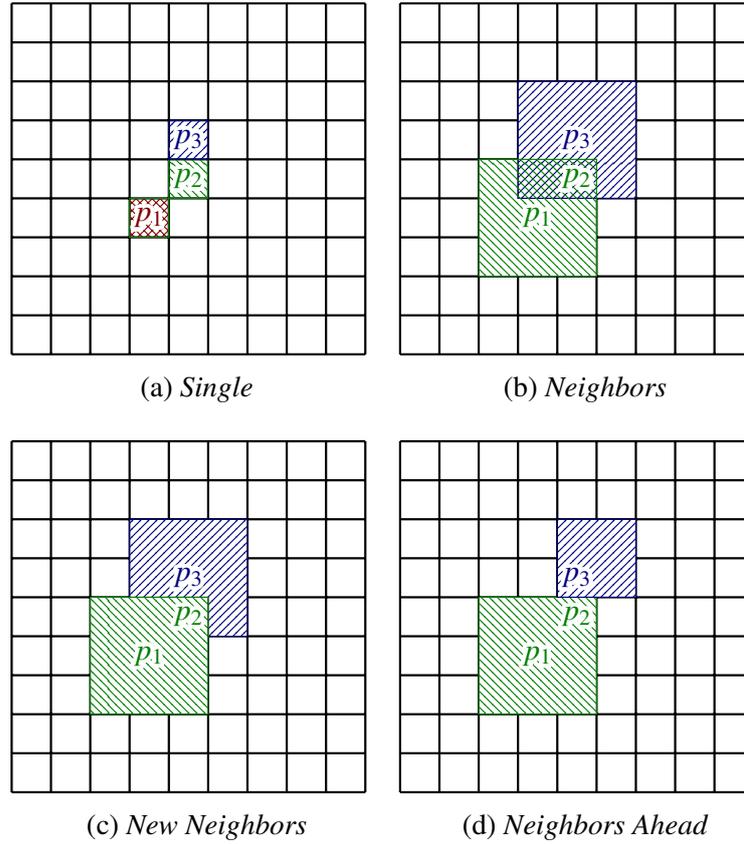


Abbildung 3.4: Zellbasierte Caching-Strategien für einen Nutzer, der sich von Position  $p_1$  über  $p_2$  zu  $p_3$  bewegt. Grün schraffierte Zellen stellen die zum Zeitpunkt  $p_2$ , blau schraffierte Zellen die zum Zeitpunkt  $p_3$  ausgewählten Zellen  $C_t$  dar.

neun Zellen, wobei die gleichmäßige Gitterstruktur natürlich nur eine stark vereinfachte Repräsentation darstellt.

- *New Neighbors*: Nutzer laden die Messaufgaben von  $c_t$  und allen Nachbarzellen  $N_{c_t}$ , die nicht in der Auswahl des vorherigen Downloads  $C_{prev}$  enthalten waren:  $C_t = \{c_t\} \cup N_{c_t} \setminus C_{prev}$ .
- *Neighbors Ahead*: Nutzer laden die Task-Listen für  $c_t$  und alle Nachbarzellen, die in der Bewegungsrichtung des Nutzers liegen. Die Bewegungsrichtung wird durch den Vergleich mit Hilfe der vorherigen Auswahl  $C_{prev}$  bestimmt. Die Nachbarzellen werden in diesem Fall wie folgt bestimmt:  $N_a = \{n \in N_{c_t} | (\Delta x_{c_{prev}, c_t} \leq \Delta x_{c_{prev}, n}) \wedge (y_{c_{prev}, c_t} \leq \Delta y_{c_{prev}, n})\}$ . Für  $C_t$  gilt folglich:  $C_t = \{c_t\} \cup N_a$ .

Solange sich der Nutzer in Zellen bewegt, die in  $C_t$  enthalten sind, wird kein neuer Task-Download vom Nutzer initiiert, es findet somit keine Pull-Kommunikation statt. Bei den Abbildungen 3.4(b) bis 3.4(d) ist dies bei dem Zellwechsel von  $p_1$  zu  $p_2$  der Fall, daher bleibt  $C_t$  unverändert. Erst bei Verlassen der in  $C_t$  enthalte-

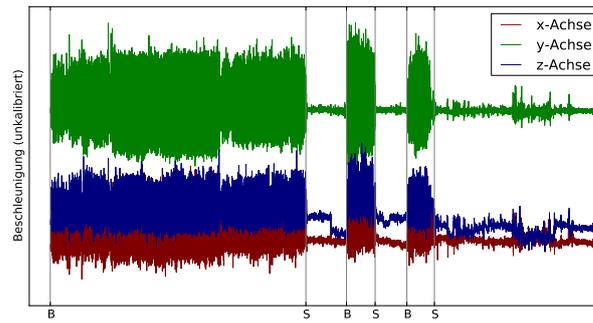


Abbildung 3.5: Beschleunigungsdaten eines Nutzers (Datensatz aus [40]). Phasen, in denen sich der Nutzer in Bewegung befindet sind mit *B* gekennzeichnet, Phasen des Stillstands mit *S*.

nen Zellen wird eine neue Task-Liste für eine neu ausgewählte Menge an Zellen angefordert.

Werden in  $C_t$  während des Aufenthaltes eines Nutzers neue Messaufgaben erstellt oder bestehende gelöscht, muss der Nutzer hierüber unterrichtet werden. Dies erfolgt mittels Push-Kommunikation, d. h. der Server initiiert die Übertragung. Wann diese Nachrichten vom Server gesendet werden, hängt von der Task-Bündelung ab, die in Abschnitt 3.2.3.3 beschrieben wird.

### 3.2.3.2 Adaptives Monitoring mit Bewegungserkennung

Die im vorherigen Abschnitt vorgestellte zellbasierte Task-Verteilung zielt darauf ab, die Anzahl der übermittelten Tasks zu verringern und eine energieeffizientere Verteilung zu erreichen. Daneben sollen auch die durch die Positionsbestimmungen entstehenden Energiekosten reduziert werden. Dafür wurde in das Gesamtkonzept ein adaptives Monitoring mit Bewegungserkennung integriert.

Dieses baut auf dem allgemeinen adaptiven Monitoring auf, bei dem die Nutzer über alle Tasks informiert werden. Statt aber periodisch die Position zu bestimmen und zu prüfen, ob einer der Task-Regionen erreicht wurde, wird das Intervall der Überprüfung an die aktuelle Situation des Nutzers angepasst. Dafür bestimmt jeder Nutzer die Distanz zum nächstgelegenen Task, im Folgenden mit  $\min(D)$  bezeichnet, und ermittelt, wann die entsprechende Task-Region mit Maximalgeschwindigkeit  $v_{max}$  erreicht werden könnte. Dadurch erhält der Nutzer das kürzest mögliche Zeitfenster  $t_s$ , in dem ein Task erreicht werden kann:

$$t_s = \frac{\min(D)}{v_{max}} \quad (3.1)$$

Innerhalb dieses Zeitfensters ist somit keine Positionsbestimmung notwendig. Falls der Nutzer einen neuen Task erhält, der näher gelegen ist, muss dieses Zeitfenster neu bestimmt werden.

Bei PSNs führt dieses Vorgehen jedoch zu vielen unnötigen Positionsbestimmungen, da Nutzer sich in der Regel nicht mit der Maximalgeschwindigkeit oder oftmals auch gar nicht fortbewegen. Daher wurde bei diesem Konzept eine Bewegungserkennung integriert. Dabei soll ausgenutzt werden, dass moderne Mobiltelefone heutzutage vielfach mit Beschleunigungs- und Lagesensoren ausgestattet sind. Mit deren Hilfe ist eine Bewegung des Nutzers relativ leicht erkennbar. Abbildung 3.5 zeigt beispielhaft die Werte der Beschleunigungssensoren eines Nutzers und die dazugehörige Aktivität, wobei ein deutlicher Unterschied zwischen den Bewegungs- und den Ruhephasen erkennbar ist. In anderen Arbeiten wird mit Hilfe dieser Sensoren sogar versucht, auf genaue Aktivitäten des Nutzers zu schließen, vgl. [28, 45, 133]. Da dies aber noch sehr fehleranfällig ist und typischerweise eine Trainingsphase des Systems benötigt, soll im Rahmen dieses Konzeptes nur erkannt werden, ob ein Nutzer sich bewegt oder nicht.

Wie beim adaptiven Monitoring wird auch  $t_s$  bestimmt, allerdings wird dieser Wert nicht direkt für die Ermittlung der nächsten Positionsbestimmung verwendet. Stattdessen wird ein Zähler  $t_m$  mit  $t_s$  initialisiert und periodisch alle  $m$  Sekunden eine Bewegungserkennung durchgeführt. Wird eine Bewegung erkannt, werden  $m$  Sekunden von  $t_m$  abstrahiert. Sobald  $t_m = 0$ , wird eine neue Positionsbestimmung angestoßen. Der ungünstigste Fall tritt ein, wenn der Nutzer sich ununterbrochen bewegt. In diesem Fall verhält sich der um die Bewegungserkennung erweiterte Ansatz wie der einfache adaptive Ansatz. In allen anderen Fällen, werden weniger Positionsbestimmungen durchgeführt. Wie bei  $t_s$ , muss auch  $t_m$  neu bestimmt werden, sobald ein Nutzer einen neuen Task erhält, der näher gelegen ist als die zuvor nächstgelegene Messaufgabe. Die Positionsbestimmung findet jedoch nur statt, wenn innerhalb der Zellmenge  $C_t$  überhaupt Tasks vorhanden sind. Ist dies nicht der Fall, wird bis zum Verlassen von  $C_t$  bzw. dem Eintreffen einer neuen Aufgabe keine Positionsbestimmung durchgeführt.

### 3.2.3.3 Task-Bündelung

Wie schon in Abschnitt 3.2.3.1 beschrieben, werden Tasks nur dann von den Nutzern heruntergeladen, wenn diese die Mobilfunkzellen verlassen, für die sie zuletzt die Task-Liste geladen haben, d. h. sobald  $c_t \notin C_t$ . Alle Tasks, die nach dem Herunterladen der Task-Liste durch den Nutzer in  $C_t$  erstellt oder gelöscht werden, müssen folglich durch Push-Nachrichten vom Server an den Nutzer kommuniziert werden. Wird dabei jeder Task unverzüglich an alle Nutzer  $U_{C_t}$  der entsprechenden Zellen übermittelt, fallen pro Task jeweils  $|U_{C_t}|$  Push-Nachrichten an.

Das Ziel der Task-Bündelung besteht darin, die Anzahl der übertragenen Push-Nachrichten zu reduzieren, da bei jedem Aufbau einer Kommunikationsverbindung

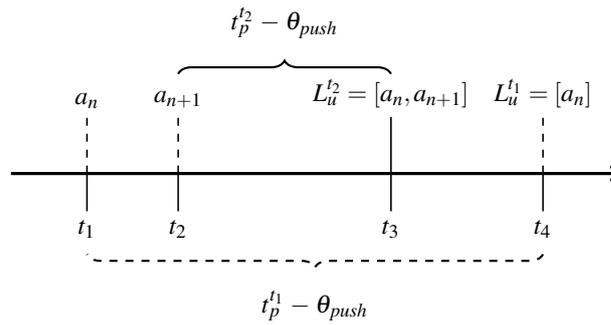


Abbildung 3.6: Schematische Darstellung der Task-Bündelung für zwei Messanfragen  $a_n$  und  $a_{n+1}$

ein gewisser Overhead anfällt, der unabhängig von den zu übertragenen Daten ist. Durch die Bündelung der Tasks und das gesammelte Versenden kann folglich Energie eingespart werden.

Um möglichst viele Tasks in einer Nachricht zu bündeln, sollten diese solange zurück gehalten werden, bis der Nutzer wirklich in die Nähe der Task-Region kommt. Für diese Entscheidung benötigt der Server jedoch den aktuellen Standort des Nutzers. Um auch hier unnötige Übertragungen zu vermeiden, meldet der Nutzer seine Position nur, wenn ein Task erledigt und beantwortet wird, oder wenn der Nutzer  $C_t$  verlässt und eine neue Task-Liste anfordert. Die Standortinformationen werden in diesen Fällen nicht separat versendet, sondern mit ohnehin versendeten Daten gebündelt übertragen. Falls jedoch beide genannten Fälle nicht eintreten, wird nach einem festgelegten Zeitintervall ein sogenannter *GPS-Ping* vom Nutzer gesendet, der nur aus den Standortinformationen des Nutzers besteht. In allen Fällen werden die neusten Standortinformationen übermittelt, die auf Seiten des Nutzers vorhanden sind, jedoch keine, eigens für diesen Zweck angestoßene Positionsbestimmung durchgeführt.

Der Server hält für jeden Nutzer  $u \in U$  eine Liste  $L_u$  vor, die alle Messaufgaben enthält, die dem Nutzer noch nicht per Push-Nachricht übermittelt wurden. Dabei wird jeder Task, der in einer der Zellen  $C_t$  des Nutzers  $u$  erstellt wird, zu  $L_u$  hinzugefügt. Zudem wird ein Zeitintervall  $t_p$  bestimmt, das angibt, wie lange  $L_u$  noch zurückgehalten werden kann:

$$t_p = \min_{a \in L_u} \frac{d(a[r], r_{pos})}{v_{max}} - \Delta t_{r_{pos}, t_{now}}, \quad (3.2)$$

wobei  $r_{pos}$  den Standort des Nutzers, der dem Server zuletzt mitgeteilt wurde, und  $t_{r_{pos}}$  den Zeitpunkt der entsprechenden Positionsbestimmung bezeichnet. Mit voranschreitender Zeit wird  $t_p$  dekrementiert, so dass  $t_p$  zu jedem Zeitpunkt noch das verbleibende Zeitintervall angibt. Sobald  $t_p$  unter einen festgelegten Schwellwert  $\theta_{push}$  fällt, wird  $L_u$  an den Nutzer übertragen und anschließend geleert. Der

Schwellwert  $\theta_{push}$  dient dazu, dass Nutzer die Tasks schon kurz vor Erreichen der Task-Region erhalten, um möglichen Latenzen vorzubeugen.

Da  $t_p$  auf dem minimalen Abstand zu den Task-Regionen basiert, wird das Zeitintervall bei jedem Hinzufügen eines Tasks neu bestimmt. Abbildung 3.6 illustriert dies an einem Beispiel. Zu den Zeitpunkten  $t_1$  und  $t_2$  werden die Task-Anfragen  $a_n$  und  $a_{n+1}$  erzeugt und sollen an den Nutzer  $u$  gesendet werden. Zum Zeitpunkt  $t_1$  beinhaltet  $L_u$  nur  $a_n$ . Daher wird der spätestmögliche Sendezeitpunkt  $t_4$  anhand des Abstandes zur Task-Region  $a_n[r]$  bestimmt. Zum Zeitpunkt  $t_2$  kommt nun mit  $a_{n+1}$  ein weiterer Task hinzu. In diesem Beispiel sei dieser Task deutlich näher zum Standort des Nutzers gelegen, weshalb der neu berechnete Sendezeitpunkt  $t_3$  vor  $t_4$  liegt. Kommen keine weiteren Aufgaben hinzu, werden beide Tasks schließlich zum Zeitpunkt  $t_3$  gebündelt an den Nutzer übertragen.

### 3.2.4 Evaluation

Ziel des vorgestellten Ansatzes ist es, die Task-Verteilung in PSNs effizienter zu gestalten. Dabei wird sowohl die Anzahl der benötigten Positionsbestimmungen als auch die Menge der zu übertragenen Nachrichten reduziert. In diesem Abschnitt sollen nun die Effekte des Ansatzes auf diese zwei Aspekte mit Hilfe von Simulationen untersucht werden.

Zunächst wird daher die Simulationsumgebung beschrieben, anschließend folgen die Unterabschnitte für die Positionsbestimmung und die Kommunikation.

#### 3.2.4.1 Simulationsumgebung

Für die Simulationen wurde die *JiST/SWANS*-Umgebung [282] verwendet. *JiST* (*Java in Simulation Time*) ist eine Simulationsumgebung für ereignisorientierte Simulationen und verwendet die Java Virtual Machine. *SWANS* (*Scalable Wireless Ad hoc Network Simulator*) ist eine Simulationsumgebung für drahtlose Netze und basiert auf JiST. Für die Untersuchung des vorgestellten Ansatzes wurde für JiST/SWANS eine Erweiterung entwickelt, welche die Verteilung und Beantwortung von Tasks ermöglicht.

Es wurden Simulationen für ein  $5 \times 5$  km großes Gebiet mit bis zu 100 Knoten durchgeführt. Die Knoten repräsentierten Nutzer mit Mobiltelefonen und wurden zufällig in diesem Gebiet verteilt. Deren Mobilität wurde mit Hilfe des *Random Waypoint Model* [110] modelliert. Dabei bewegen sich die Nutzer zu zufällig ausgewählten Zielpunkten, pausieren dort kurz und wählen schließlich einen neuen Zielpunkt. Die Gehgeschwindigkeiten wurden in den Simulationsdurchläufen mit 0,8 bis 2,0 m/s angenommen, die Pausen wurden mit 10 Sekunden kurz gehalten, um den vorgestellten Ansatz nicht zu stark zu bevorteilen.

Task-Anfragen, dargestellt durch Tupel der Form  $\langle i, r, d, m \rangle$ , vgl. Abschnitt 3.2.1, wurden mit kreisförmigen Task-Regionen mit einem Radius von 50 Metern modelliert. Die Gültigkeitsdauer  $d$  war unbefristet, die gewünschte Anzahl an Nutzern  $m$  wurde für alle Tasks auf 10 gesetzt. Da man PSNs mit verschiedenartigen Sensortypen auch als mehrere Sensornetze mit gleichartigen Sensortypen betrachten kann, wurde in den Simulationen angenommen, dass alle Knoten die gewünschte Information  $i$  aufzeichnen können.

Task-Anfragen werden in der Simulationsumgebung periodisch für zufällige Punkte erzeugt, wobei die Frequenz der Erzeugung für die Analyse variiert wurde. Es wurden drei Szenarien untersucht: (1) Szenario „Geringe Last“ (GL), bei dem alle 5 Minuten eine neue Anfrage erzeugt wurde; (2) „Mittlere Last“, mit einer neuen Anfrage alle 30 Sekunden; (3) „Hohe Last“, mit einer eingehenden Task-Anfrage alle 3 Sekunden.

Um die Mobilfunkzellen zu modellieren, wurde das Simulationsgebiet gleichmäßig in Zellen unterteilt. Da die Größe von Mobilfunkzellen stark von der Umgebung abhängt – Zellen in ländlichen Gebieten sind typischerweise deutlich größer als solche in städtischen Bereichen – wurden auch hier verschiedene Szenarien definiert, um unterschiedlich große Zellgrößen zu untersuchen: Große Zellen (L) wurden mit einer Seitenlänge von 500 m, mittlere Zellen (M) mit 250 m und kleine Zellen (S) mit 100 m modelliert.

Zudem wurde der in Abschnitt 3.2.3.3 eingeführte GPS-Ping auf 5 Minuten festgesetzt. Es wurde jeweils eine Simulationszeit von 6 Stunden simuliert und jeder Durchlauf wurde 20 Mal wiederholt. Eine Zusammenfassung der Simulationsparameter ist in Tabelle 3.1 zu finden.

Knoten	1–100
Simulationszeit	6 Stunden
Simulationsgebiet	5 × 5 km
Mobilität	Random Waypoint Model
Geschwindigkeit	0,8–2,0 m/s
Pause	10 Sek.
Task-Anfragen	$\langle -, \{X   \overline{XP} \leq 50m\}, \infty, 10 \rangle$
Taskerstellung	Hohe Last (HL): 1/3s Mittlere Last (ML): 1/30s Geringe Last (GL): 1/300s
Zellgrößen	Klein (S): 100 × 100 m Mittel (M): 250 × 250 m Groß (L): 500 × 500 m
GPS-Ping	5 Min.

Tabelle 3.1: Simulationsparameter für die Evaluation der Task-Verteilung

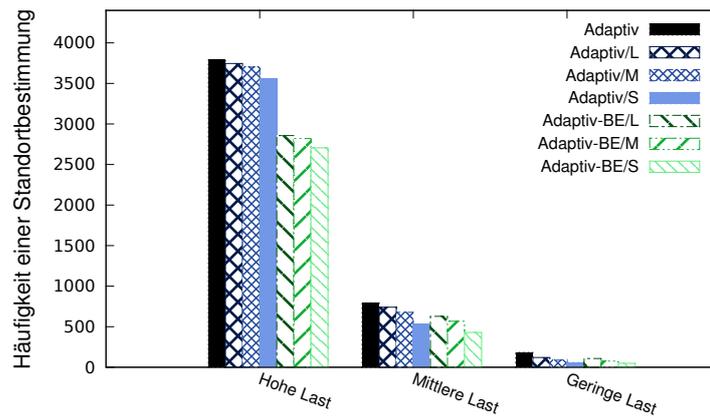


Abbildung 3.7: Anzahl der Positionsbestimmungen pro Knoten für verschiedene Last-Szenarien und Zellgrößen

### 3.2.4.2 Anzahl der Positionsbestimmungen

Da die Bestimmung der Position, z. B. mittels GPS, sehr viel Energie benötigt, wurde in einem ersten Schritt die Anzahl der nötigen Standortbestimmungen untersucht. Dafür wurde der vorgestellte Ansatz in den Simulationen mit jeder Zellgröße und jedem Last-Szenario getestet. Als Vergleichsgröße diente der in Abschnitt 3.2.3.2 beschriebene einfache adaptive Ansatz. Letzterer wird in den Ergebnissen mit „Adaptiv“ bezeichnet, wohingegen das vorgestellte Konzept als „Adaptiv-BE“ (Adaptiv mit Bewegungserkennung) aufgeführt wird.

Die Ergebnisse sind in Abbildung 3.7 dargestellt. An diesen ist zu erkennen, dass in allen Fällen weniger Positionsbestimmungen durchgeführt wurden als bei dem adaptiven Referenzansatz. Die Höhe der Reduzierung hängt jedoch von den Last-Szenarien und der gewählten Zellgröße ab.

In dem Szenario mit hoher Last konnte eine signifikante Reduzierung durch die Bewegungserkennung erreicht werden. Eine Positionsbestimmung wird nur durchgeführt, wenn in der aktuellen Zelle  $c_t$  überhaupt Messaufgaben vorhanden sind. Aufgrund der großen Anzahl an Tasks war die Wahrscheinlichkeit dafür sehr hoch. Folglich war auch die Chance durch die integrierte Bewegungserkennung Positionsbestimmungen einzusparen sehr hoch. Im Vergleich zu den Durchläufen ohne Bewegungserkennung, konnte die Anzahl der Positionsbestimmungen mittels GPS um 23,54 % bis 24,04 % reduziert werden. Diese Reduzierung wurde erreicht, obschon die Nutzer in den Simulationen vergleichsweise kurze Pausen machten.

In dem Szenario mit mittlerer Last war der Einfluss der Bewegungserkennung deutlich geringer, es wurde aber immer noch eine Reduzierung um 15,19 % bis 19,18 % erreicht. Da bei diesem Szenario weniger Tasks erstellt wurden, sank auch die Wahrscheinlichkeit, dass in der aktuellen Zelle  $c_t$  eines Nutzers auch ein Task vorhanden

war. Ist in  $c_t$  kein Task vorhanden, findet überhaupt keine Standortbestimmung statt, folglich ist die Bewegungserkennung in diesen Fällen obsolet und es kann keine Verbesserung erzielt werden.

Neben der Bewegungserkennung hatte hier auch die Zellgröße einen entscheidenden Einfluss. Fiel diese bei hoher Last noch nicht wirklich stark ins Gewicht, konnten bei mittlerer Last bei einer Verkleinerung der Zellen von mittleren auf kleine Zellen ohne Bewegungserkennung ca. 21,44 % der Positionsbestimmungen eingespart werden, bei Verwendung der Bewegungserkennung waren es sogar 24,2 %.

Der bei mittlerer Last auftretende Trend ließ sich auch bei geringer Last beobachten. Die Bewegungserkennung hatte einen noch geringeren Einfluss auf die Reduktion der Positionsbestimmungen, konnte die Anzahl dieser aber immerhin noch um 10 % bis 17,6 % reduzieren, wohingegen der Einfluss der Zellgröße noch weiter stieg. Wurden mittlere Zellen eingesetzt, konnte die Anzahl der Standortbestimmungen gegenüber der bei großen Zellen um 27,32 % bzw. 29,71 % gesenkt werden, je nachdem ob die Durchläufe ohne oder mit Bewegungserkennung betrachtet wurden. Bei der Verwendung kleiner Zellen konnte nochmals eine Reduktion um 33,35 % bzw. 38,9 % gegenüber der mittleren Zellen erreicht werden. Insgesamt war die Anzahl an Positionsbestimmungen in diesem Szenario zwar auch relativ klein, aber im Referenzansatz wurde im Schnitt noch 182,86 mal die Position bestimmt, wohingegen Adaptiv-BE bei kleinen Zellen im Mittel mit nur 47,9 Standortbestimmungen auskam.

### 3.2.4.3 Anzahl der Kommunikationsverbindungen

In einem zweiten Schritt wurden die Kommunikationskosten analysiert. Hierfür wurde die Anzahl der benötigten Kommunikationsverbindungen in den verschiedenen Szenarien untersucht. Die Ergebnisse sind in Abbildung 3.8 dargestellt, wobei jeder Balken für eine Kombination von Zellgröße, Lastszenario und Caching-Strategie steht. Jeder dieser Ergebnisbalken wiederum repräsentiert die vom Nutzer initiierten (Pull), die vom Server initiierten (Push) und die aufgrund des GPS-Pings versendeten Nachrichten.

Bei großen Zellen kann man erkennen, dass der Einsatz der Caching-Strategien, vgl. Abschnitt 3.2.3.1, nur bei geringer Last (*GL*) sinnvoll war. In diesem Szenario konnten im Schnitt über 38,64 % der Nachrichten eingespart werden, wenn eine der entsprechenden Strategien gewählt wurde. Sowohl bei mittlerer als auch bei hoher Last war die Single-Strategie, bei der nur die Tasks der aktuellen Zelle  $c_t$  geladen werden, am effizientesten. Da durch die Caching-Strategien das beobachtete Gebiet vergrößert wird, wurden in diesen Fällen sehr viele Tasks per Push-Kommunikation übermittelt.

Bei dem Szenario mit mittelgroßen Zellen war die Verwendung der Caching-Strategien schon bei mittlerer Last sinnvoll. Da die Zellen in diesem Fall kleiner waren, traten öfter Zellwechsel auf als bei großen Zellen, wodurch die Anzahl der Pull-

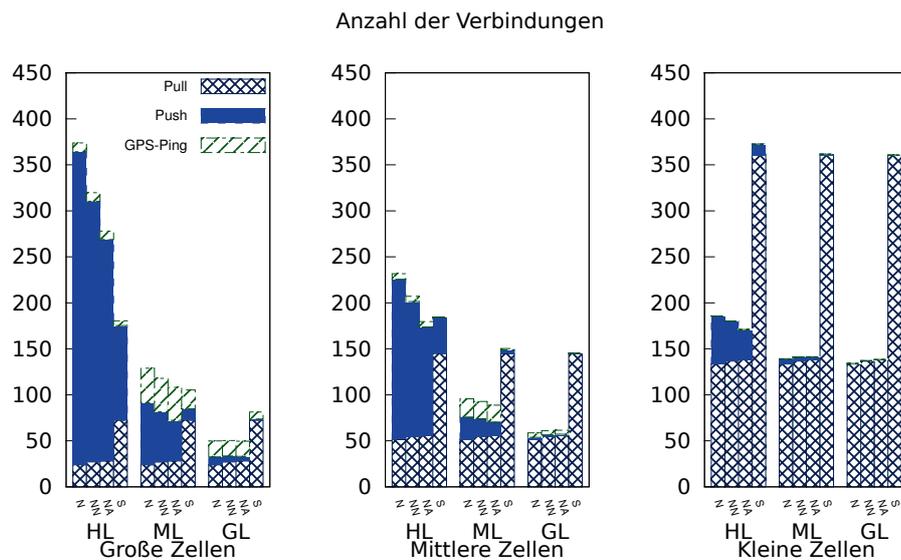


Abbildung 3.8: Anzahl der Kommunikationsverbindungen für verschiedene Zellgrößen, Lastszenarien und Caching-Strategien

Nachrichten deutlich stieg. Daher konnte die Anzahl der Nachrichten durch den Einsatz der genannten Strategien um mehr als 36,36 % reduziert werden. Bei dem Szenario mit geringer Last konnten sogar mehr als 57,68 % der Nachrichten eingespart werden, da hier kaum Push-Nachrichten versendet wurden.

Im dritten Szenario, in dem kleine Zellen verwendet wurden, ist in allen Lastszenarien ein deutlich positiver Effekt der Caching-Strategien erkennbar. Da hier die Anzahl der Pull-Nachrichten aufgrund der vielen Zellwechsel stark anstieg, war eine „virtuelle“ Vergrößerung der Zelle durch eine der drei Strategien sinnvoll. Selbst bei hoher Last wurden hier über 50,12 % weniger Nachrichten übermittelt.

Interessant ist auch der durch die GPS-Pings anfallende Overhead. GPS-Pings werden nur versandt, wenn eine Position bestimmt wurde und darüber hinaus dem Server eine gewisse Zeit keine Position mehr mitgeteilt wurde. Bei geringer Last wurde nur selten eine Position bestimmt, weswegen die Anzahl der GPS-Pings geringer war als bei mittlerer Last. Bei hoher Last bearbeiteten die Knoten hingegen sehr oft existierende Tasks und übermittelten dabei ihre Position, so dass auch in diesem Fall seltener GPS-Pings notwendig wurden. Betrachtet man die Zellgrößen, so wird ersichtlich, dass bei größeren Zellen mehr GPS-Pings benötigt wurden. Dies ergab sich aus den seltener stattfindenden Zellwechseln, bei denen jeweils die zuletzt bekannte Position des Knotens übermittelt wurde.

Vergleicht man die Caching-Strategien untereinander, wird deutlich, dass bei wenig Last möglichst viele Zellen bei der Caching-Strategie eingeschlossen werden sollten, damit wenig Pull-Requests anfallen. In diesen Fällen ist die Neighbors-Strategie, die alle Nachbarzellen mit beobachtet, am effizientesten. Werden viele

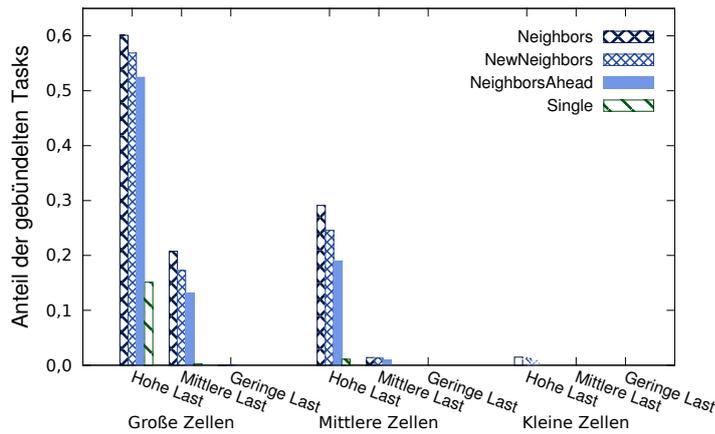


Abbildung 3.9: Anteil der gebündelt versendeten Tasks

Messaufgaben erzeugt und somit viele Push-Nachrichten verschickt, zeigen Strategien wie New Neighbors oder Neighbors Ahead die besten Ergebnisse. Diese reduzieren durch Betrachtung der Mobilität des Nutzers die Menge der beobachteten Zellen und verringern so die Anzahl der erhaltenen Push-Nachrichten.

In einem letzten Schritt wurde noch der Anteil der Tasks untersucht, der durch die Task-Bündelung zusammengefasst übertragen wurde. Der Anteil der gebündelten Tasks  $q_{push}$  wird dabei aus der Anzahl der Messaufgaben, die mittels Push-Nachricht übermittelt werden sollen,  $|A_{push}|$ , und der Anzahl der dafür verwendeten Push-Nachrichten  $|N_{push}|$  wie folgt ermittelt:

$$q_{push} = \frac{|A_{push}| - |N_{push}|}{|A_{push}|} \quad (3.3)$$

Werden 10 zu übertragende Messaufgaben in 6 Push-Nachrichten zusammengefasst, erhält man folglich  $q_{push} = 40\%$ , d. h. für 40 % der Aufgaben wurde keine separate Kommunikationsverbindung benötigt.

Abbildung 3.9 zeigt die Ergebnisse für die verschiedenen Szenarien. Je mehr Zellen in der Caching-Strategie enthalten waren und je höher die Frequenz eingehender Messanfragen war, desto mehr Tasks konnten sichtlich gebündelt versendet werden. Die Ergebnisse zeigen auch, dass durch die Task-Bündelung in einigen Szenarien die Anzahl der übertragenen Nachrichten deutlich reduziert werden konnte. Die größten Einsparungen wurden bei großen Zellen und hoher Last erreicht: Bei Verwendung der Neighbors-Strategie konnten hier bis zu 60 % der Task gebündelt verschickt werden.

Gute Ergebnisse konnten auch bei mittlerer Last erreicht werden. In den Simulationen konnten bis zu 20,8 % der Tasks gebündelt werden. Auch bei mittleren Zellen und hoher Last zeigte die Task-Bündelung bei allen Caching-Strategien, die Nachbarzellen mit beobachten, positive Effekte. So konnten hier bis zu 29,1 % der Nachrichten eingespart werden. In allen anderen Fällen lag  $q_{push}$  bei 1 % oder darunter,

was vor allem an dem großen Zeitintervall zwischen den Erstellungszeitpunkten zweier Tasks lag.

### 3.2.5 Diskussion und Erweiterungen

Die Ergebnisse der Simulationen zeigen, dass das vorgeschlagene Konzept für den Anwendungsbereich von örtlich und zeitlich begrenzten Messungen sinnvoll ist und durch die Reduzierung der Positionsbestimmungen und der kommunizierten Nachrichten einen positiven Effekt auf die Energiereserven hat. Da das Konzept aus drei Komponenten besteht, der zellbasierten Task-Verteilung, dem adaptivem Monitoring mit Bewegungserkennung und der Task-Bündelung, werden diese im Folgenden kurz separat diskutiert.

Die Task-Bündelung zeigte in Szenarien, in denen vielen Tasks versendet wurden, gute Effekte. In den restlichen Fällen führte sie aber auch zu keiner Verschlechterung. Somit ist ein Einsatz der Task-Bündelung durchweg sinnvoll und verhindert, dass bei einer größer werdenden Task-Frequenz die Kommunikationslast proportional mit ansteigt.

Bei dem adaptiven Monitoring wurde bislang bei der Bewegungserkennung lediglich zwischen zwei Zuständen, *In Bewegung* und *Stillstand*, unterschieden, um die Erkennung möglichst robust und fehlerfrei zu gestalten. Bei einer Erweiterung um mehrere Bewegungszustände, wie z. B. eine Unterscheidung zwischen *Gehen* und *Laufen*, könnte die Anzahl der Positionsbestimmungen weiter reduziert werden, da für jeden Bewegungszustand eine spezielle Maximalgeschwindigkeit  $v_{max}$  verwendet werden kann.

Bei der zellbasierten Task-Verteilung sind zwei Aspekte zu beachten. Zum einen ist die Neighbors-Ahead-Strategie nicht ohne weiteres umsetzbar, da für sie Informationen zur Lage und Ausbreitung der Zellen vorhanden sein müssen. Die Annahme ist hier, dass diese Informationen, wenn nicht durch die Mobilfunkbetreiber veröffentlicht, mit der Zeit gelernt werden können, d. h. die Zellwechsel der Nutzer werden aufgezeichnet und die relative lokale Ordnung wird sukzessive bestimmt. Zum anderen wurden die verschiedenen Caching-Strategien zwar analysiert, es wurde jedoch keine direkte Empfehlung abgegeben, da die Auswahl der Strategie an die jeweilige Situation angepasst werden muss. Die Zellgröße kann im realen Umfeld nicht angepasst werden, sondern hängt von der Struktur des vom Nutzer verwendeten Mobilfunknetzes ab [26, 264]. Somit sollte basierend auf der Größe der Zellen im Umfeld des Nutzers und der jeweiligen Last-Szenarien die passende Caching-Strategie ausgewählt werden. Befindet sich der Nutzer in großen Zellen mit hoher Last, kann die Single-Strategie angewandt werden. Bewegt sich der Nutzer dann aber in kleinere Zellen, kann entsprechend auf eine der Strategien, welche die Nachbarzellen mit beobachten, umgestellt werden.

Eine zusätzliche Erweiterungsmöglichkeit besteht in der Integration von Wegenetzen zur Bestimmung der frühestmöglichen Ankunft des Nutzers in der Anfrageregion.

on. Bisher wird hier die „Luftlinien“-Distanz verwendet, was jedoch nur selten dem für den Nutzer kürzest möglichen Pfad entspricht. Durch die Hinzunahme von Wegenetzinformationen könnte dies verbessert werden. Da Menschen sich aber nicht nur auf vorgesehenen Wegen bewegen, muss bei einer solchen Erweiterung darauf geachtet werden, dass alle möglichen Bewegungspfade berücksichtigt werden, um die Effektivität des Systems weiterhin zu gewährleisten.

### **3.3 Ansatz zur lokalen Gruppierung von Messknoten**

Auch in diesem Abschnitt soll ein Ansatz zur Datenerfassung in PSNs vorgestellt werden. Im Gegensatz zum vorherigen Verfahren, das für örtlich und zeitlich beschränkte Messungen konzeptioniert wurde, soll nun der Anwendungsbereich großflächiger und langfristiger Messkampagnen betrachtet werden. Ein mögliches Beispielszenario hierfür ist die Messung der Luftqualität in Städten. Es sind also nicht nur einzelne Gebiete zu bestimmten Zeitpunkten von Interesse, sondern das Ziel ist vielmehr, möglichst umfassende Informationen zu erhalten. In diesem Szenario führen die Nutzer mit ihren Mobiltelefonen quasi kontinuierlich und überall Messungen durch.

Bei der Realisierung solcher großflächiger PSNs ist die Schonung der Ressourcen ein zentraler Aspekt. Die naive Umsetzung, bei der jeder Nutzer kontinuierlich, also wann immer es möglich ist, Messungen durchführt, führt sowohl auf Nutzer- als auch auf Betreiberseite zu einer nicht optimalen Nutzung der Ressourcen und damit zu Problemen. Für die Nutzer entstehen neben den Energiekosten, die bei der Aufnahme der Messung und dem Versand der Messdaten anfallen, auch monetäre Übertragungskosten für die Übermittlung der Daten an den Betreiber. Der Betreiber hingegen muss mit einer sehr großen Anzahl an Nutzern und damit mit vielen eingehenden Daten zurechtkommen, wenngleich viele von diesen redundante Informationen enthalten.

Mit dem folgenden Ansatz zur lokalen Gruppierung von Messknoten sollen diese Probleme gelöst werden. Die Gruppierung von Messknoten soll dazu führen, dass die Aufgaben unter lokal benachbarten Knoten aufgeteilt werden, so dass bei jedem Knoten ein geringerer Ressourcenverbrauch anfällt und gleichzeitig auch weniger redundante Daten an den Anwendungsserver des Betreibers übermittelt werden.

Dieser Abschnitt basiert auf dem in [300] vorstellten Ansatz und ist wie folgt strukturiert: Der Abschnitt beginnt in 3.3.1 wieder mit der Beschreibung der Problemstellung und der zu erfüllenden Anforderungen. Darauf folgt in Abschnitt 3.3.2 die Darstellung des Konzeptes zur lokalen Gruppierung von Messknoten. In Abschnitt 3.3.3 wird das Konzept evaluiert und abschließend in Abschnitt 3.3.4 diskutiert.

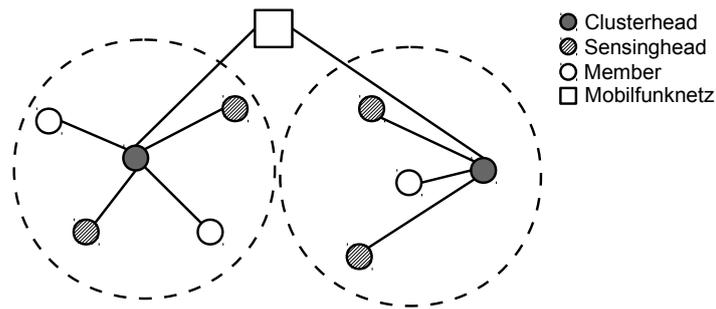


Abbildung 3.10: Schematische Darstellung eines Clusters

### 3.3.1 Problemstellung und Anforderungen

Die mobilen Knoten eines PSNs sollen lokal gruppiert werden, um die Anzahl redundanter Messungen zu verringern. Solche lokalen Gruppierungen von Knoten werden als Cluster bezeichnet. Ziel des Ansatzes ist es, geografisch benachbarte Messknoten zu Clustern zusammenzufassen und die anfallenden Aufgaben auf die Knoten eines Clusters aufzuteilen. Bei der Aufgabenverteilung soll zwischen drei verschiedenen Rollen unterschieden werden:

- **Clusterhead:** Ein Knoten wird zum sogenannten *Clusterhead*, der für das Aggregieren und die Übermittlung der Messdaten an den Server des Betreibers verantwortlich ist. Letzteres soll über die mobile Datenverbindung der Endgeräte erfolgen.
- **Sensinghead:** Eine Teilmenge der Knoten bekommt die Rolle eines sogenannten *Sensingheads*, d. h. die eines Messknotens. Messknoten führen die gewünschten Messungen durch und übermitteln die erhobenen Messdaten an den Clusterhead. Da die Knoten sich in lokaler Nachbarschaft befinden, kann hierfür eine Ad-hoc-Kommunikationsverbindung genutzt werden.
- **Member:** Die restlichen Knoten werden als *Member*-Knoten bezeichnet. Sie haben keine spezifische Aufgabe zu erfüllen, können aber jederzeit als zusätzliche Messknoten angefordert werden.

Ein Cluster soll sich aus einem Clusterhead, einem oder mehreren Sensingheads und beliebig vielen Member-Knoten zusammensetzen. Abbildung 3.10 stellt dies schematisch dar. Falls zu wenige Knoten in einem Cluster enthalten sind, kann der Clusterhead zusätzlich die Rolle eines Sensingheads annehmen. Ein minimales Cluster besteht folglich aus einem Knoten, der gleichzeitig Clusterhead und Sensinghead ist.

Durch die Cluster soll die Anzahl redundanter Messungen verringert werden, indem nur eine Teilmenge der Knoten in lokaler Nachbarschaft Messungen durchführt. Um die Informationsverluste zu begrenzen, soll die geografische Ausdehnung der Cluster begrenzt werden, indem nur *1-Hop-Cluster* gebildet werden. D. h. von

jedem Knoten eines Clusters besteht eine direkte Ad-hoc-Verbindung zum Clusterhead. Dies impliziert, dass Messungen innerhalb des Kommunikationsradius der Knoten redundante Informationen liefern. Bis zu welchem Abstand zwei Messungen noch als redundant angesehen werden können, hängt von der Messgröße ab. So können Messungen der Außentemperatur, die mehrere hundert Meter auseinander liegen, beispielsweise als redundant betrachtet werden, Messungen der Luftqualität, die entsprechend weit auseinander liegen, dürften jedoch in den meisten Fällen unterschiedliche Werte erfassen. Daher wird angenommen, dass die Messknoten die Kommunikationsreichweite an die räumlichen Eigenschaften der Messgröße anpassen, z. B. durch Anpassung der Übertragungsstärke oder durch Filterung von Paketen bestimmter Signalstärken.

Außerdem soll das Verfahren zur lokalen Gruppierung folgende Anforderungen erfüllen:

- **Unterstützung der Mobilität:** Da die Knoten eines PSNs kontinuierlich in Bewegung sein können, muss Mobilität von dem Verfahren unterstützt werden.
- **Dezentrale Formierung der Cluster:** Aus Gründen der Skalierbarkeit soll die Gruppierung der Knoten nicht mit Hilfe einer zentralen Instanz, sondern dezentral erfolgen.
- **Energieeffiziente Umsetzung:** Aufgrund der limitierten Energieressourcen der mobilen Endgeräte soll das Verfahren möglichst energieeffizient sein.

### 3.3.2 Konzept

Für die zuvor beschriebene Problemstellung wurde das Konzept des *Distributed Multi-Head Clustering (DMHC)* entwickelt. DMHC ist ein speziell für PSNs konzipierter verteilter Ansatz zur Gruppierung von Messknoten. Die Bezeichnung „Multi-Head“ deutet darauf hin, dass es mehrere „Hauptknoten“ mit speziellen Rollen gibt: die Clusterheads und die Sensingheads.

Die Funktionsweise von DMHC gliedert sich in zwei Teilbereiche: Der erste Bereich sorgt für den Aufbau und den Erhalt der Cluster. In diesem Rahmen werden auch die Clusterheads bestimmt. Der zweite Bereich ist für die Bestimmung der Sensingheads zuständig. Beide Bereiche werden im Folgenden detailliert beschrieben.

#### 3.3.2.1 Aufbau und Erhalt der Cluster

Das Vorgehen zur Gruppierung der Knoten basiert auf dem von Basagni vorgestellten *Distributed Mobility-Adaptive Clustering (DMAC)* [23], das ein Clustering-Verfahren für Ad-hoc-Netze mit mobilen Knoten darstellt. DMHC ist eine auf den

Einsatz in PSNs zugeschnittene Erweiterung von DMAC, bei der die folgenden Nachrichten, Knotengewichte und Prozeduren zum Einsatz kommen.

**Nachrichten** Damit sich Knoten zu Clustern gruppieren können, müssen sie Kenntnis von benachbarten Knoten haben. Aus diesem Grund senden Knoten bei DMHC periodische Broadcast-Nachrichten, sogenannte *Periodic Clustering Messages (PCM)*. PCM-Nachrichten enthalten die Knoten-ID, die noch vorhandenen Energiereserven (engl. *remaining energy (RE)*) und die aktuellen Kommunikationskosten. Letztere setzen sich auch den Kosten pro Übertragung (engl. *costs per transmission (CPT)*) und den akkumulierten Kosten für bisherige Übertragungen (engl. *total costs (TC)*) zusammen. Durch den Empfang bzw. das Ausbleiben von PCM-Nachrichten kann der Status der Verbindungen zu benachbarten Knoten erkannt werden.

Zudem werden noch zwei weitere Nachrichtentypen verwendet. *CH*-Nachrichten zeigen an, dass der Sender die Rolle des Clusterheads übernommen hat und werden per Broadcast an alle Knoten in der Umgebung verteilt. Knoten, die eine CH-Nachricht empfangen, können dem Cluster dieses Clusterheads beitreten.

Dies kann mittels einer *JOIN*-Nachricht angezeigt werden. Auch diese werden per Broadcast verteilt, enthalten aber den gewünschten Empfänger, d. h. die Knoten-ID des Clusterheads, um kenntlich zu machen, welchem Cluster der Absender beitreten möchte. Falls der beitretende Knoten zuvor Teil eines anderen Clusters war, wird die JOIN-Nachricht auch vom Clusterhead des vorherigen Clusters verarbeitet. Neben der Knoten-ID des Clusterheads beinhalten JOIN-Nachrichten außerdem eine Liste der auf dem absendenden Gerät verfügbaren Sensoren, z. B.  $\{Temp, Hum\}$ , falls dieses mit einem Temperatur- und einem Luftfeuchtigkeitssensor ausgestattet ist.

Empfängt ein Knoten mehrere CH-Nachrichten, tritt dieser dem Cluster bei, dessen Clusterhead das größere Knotengewicht hat.

**Knotengewichte** DMHC verwendet wie DMAC Knotengewichte für die dezentrale Gruppierung der Knoten. Da für eine erfolgreiche Verbreitung partizipativer Sensornetze besonderer Wert auf geringe Übertragungs- und Energiekosten gelegt werden sollte, werden diese Aspekte bei DMHC zur Bestimmung der Knotengewichte genutzt.

$$HNW(n, m) = \begin{cases} \min_{TC+CPT}(n, m), & \text{falls } (TC + CPT)_n \neq (TC + CPT)_m \\ \max_{RE}(n, m), & \text{falls } (TC + CPT)_n = (TC + CPT)_m \wedge RE_n \neq RE_m \\ \min_{ID}(n, m), & \text{falls } (TC + CPT)_n = (TC + CPT)_m \wedge RE_n = RE_m \end{cases} \quad (3.4)$$

Für die Bestimmung des Clusterheads werden die Knotengewichte mittels der Funktion *HigherNodeWeight (HNW)* paarweise verglichen, vgl. Gleichung 3.4. Um die Übertragungskosten der Nutzer gering zu halten, wird der Knoten mit den geringsten Übertragungskosten als Clusterhead ausgewählt. Sind

diese für zwei Knoten identisch erhält der Knoten mit den größeren Energie-reserven die Rolle des Clusterheads. Ist auch hierdurch keine Entscheidung möglich, fällt die Wahl auf den Knoten mit der geringeren Knoten-ID.

**Prozeduren** Um auf die oben beschriebenen Nachrichten reagieren zu können, sind verschiedene Prozeduren notwendig. Diese sind stark an die DMAC-Prozeduren angelehnt [23] und unterscheiden sich größtenteils nur durch das Bearbeiten der neu eingeführten Nachrichtentypen:

- *Init()*: Die Init-Prozedur wird aufgerufen, wenn ein Knoten keinem Clusterhead zugeordnet ist. Dies kann in zwei Situationen der Fall sein: (1) Wenn der Knoten gerade erst dem Sensornetz beitrifft oder (2) wenn ein Knoten die Verbindung zu seinem Clusterhead verliert. Die Init-Prozedur überprüft, ob sich in der Umgebung ein Clusterhead befindet, dessen Knotengewicht größer als das eigene ist. Ist dies der Fall, tritt er diesem Cluster bei, andernfalls wird dieser Knoten selbst zum Clusterhead.
- *LinkFailure(u)*: Die LinkFailure-Prozedur wird aktiv, wenn die Verbindung zu einem Knoten  $u$  abbricht. Dies wird durch das Ausbleiben der periodischen PCM-Nachrichten erkannt. Die Prozedur prüft, ob  $u$  oder der Knoten selbst vor dem Verbindungsabbruch Clusterhead gewesen sind. Falls  $u$  ein Clusterhead gewesen ist, führt der Knoten die Init-Prozedur zur Bestimmung eines neuen Clusterheads aus. Falls der Knoten selbst der Clusterhead ist, entfernt dieser  $u$  einfach aus der Menge der Cluster-Knoten. War keiner der beteiligten Knoten ein Clusterhead, hat der Verbindungsabbruch keinen direkten Einfluss auf den Knoten und wird ignoriert.
- *NewLink(u)*: Analog zu LinkFailure setzt sich die NewLink-Prozedur mit neu erkannten Verbindungen auseinander. Empfängt der Knoten eine PCM-Nachricht eines Knotens  $u$ , der sich zuvor nicht in der Nachbarschaft befunden hat, wird lediglich geprüft, ob  $u$  ein Clusterhead mit größerem Knotengewicht als dem des aktuellen Clusterheads ist. Ist dies der Fall, tritt der Knoten dem Cluster von  $u$  bei.
- *OnReceivingCH(u)*: Die Prozedur wird bei Empfang einer CH-Nachricht aufgerufen. Wie bei NewLink wird dann geprüft, ob diese Nachricht von einem Clusterhead mit größerem Knotengewicht als dem des aktuellen Clusterheads kommt. Ist dies der Fall, tritt der Knoten dem entsprechenden Cluster bei.
- *OnReceivingJoin(u,z)*: Die letzte Prozedur behandelt JOIN-Nachrichten. Diese enthalten neben dem Absender  $u$  auch die Knoten-ID des Clusterheads  $z$ , dessen Cluster der Absender beitreten möchte. Ist der aktuelle Knoten selbst ein Clusterhead, muss dieser prüfen, ob  $u$  gerade sein Cluster verlässt oder diesem beitreten möchte. Wenn  $z$  kein Clusterhead

ist, ignoriert dieser das eingehende JOIN, was  $u$  durch die nächste von  $z$  versendete PCM-Nachricht erkennen kann.

### 3.3.2.2 Bestimmung der Messknoten der Cluster

Neben dem Aufbau der Cluster müssen die Messknoten, also die Sensingheads ( $SH$ ) der Cluster, bestimmt werden. Um eine hohe Zuverlässigkeit und Qualität der Messwerte gewährleisten zu können, sollten mehrere Messknoten pro Cluster vorhanden sein, da so fehlerhafte Messungen einzelner Knoten erkannt und eventuell ausgeglichen werden können. Auf der anderen Seite sollten jedoch die Energiekosten möglichst gering gehalten werden. Aus diesem Grund sollte die Durchführung zu vieler redundanter Messungen vermieden werden. Folglich ist ein Kompromiss zwischen dem Erlangen einer gewünschten Redundanz, zum Zwecke der Qualitätssicherung, und der Vermeidung überflüssiger Redundanz, zum Ziele des Energiesparens, zu finden.

**Auswahl der Messknoten** Dieser Kompromiss spiegelt sich bei DMHC in zwei Parametern wider, welche die Anzahl der Messknoten ( $|SH|$ ) eines Clusters festlegen: Einerseits kann der Anteil der Sensingheads  $\rho$  festgelegt werden. Dieser Parameter bestimmt, wie groß der Anteil der Messknoten in Relation zu der Gesamtanzahl  $n$  der Knoten eines Clusters sein soll. Andererseits kann auch eine minimale Anzahl an Sensingheads  $v$  pro Cluster angegeben werden. Wenn ein Knoten einem Cluster beitrifft oder diesen verlässt, wird folglich eine neue Bestimmung von Messknoten durchgeführt, wenn der Ausdruck 3.5 wahr ist.

$$(|SH| < v) \vee \left( \frac{|SH|}{n} < \rho \right). \quad (3.5)$$

Für die Auswahl der Messknoten wurden zwei Verfahren entwickelt: *Sequential Sensinghead Election* (SSE; dt. „sequenzielle Messknotenwahl“) und *Highest Remaining Energy* (HRE; dt. „größte verbleibende Energiereserven“).

SSE beschreibt ein Verfahren, das auf der Sequenz von bei- und austretenden Knoten basiert. Wird ein neues Cluster geformt, werden alle neu beitretenden Knoten zu Messknoten bis der gewünschte Anteil an Sensingheads  $\rho$  vorhanden ist. Daraufhin wird bei jedem neuen Knoten geprüft, ob durch dessen Beitritt der Anteil der Messknoten unter  $\rho$  fällt. Ist das der Fall, wird dieser neue Knoten zu einem Sensinghead. Verlässt ein Messknoten ein Cluster, z. B. weil er sich einem anderen Cluster anschließt, und wird in Folge dessen  $\rho$  nicht mehr eingehalten, wird der Member-Knoten als Sensinghead gewählt, der schon am längsten Teil dieses Clusters ist.

Das zweite Verfahren basiert auf den verbleibenden Energiereserven der jeweiligen Knoten. Tritt ein neuer Knoten einem Cluster bei, wird dessen verbleibendes Energielevel mit denen der aktuellen Sensingheads verglichen. Ist

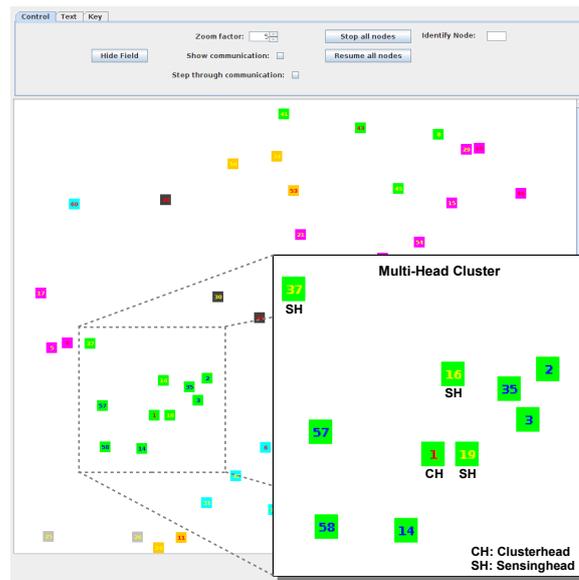


Abbildung 3.11: Visualisierung der Simulationsumgebung mit vergrößertem Cluster

es größer als bei einem der aktuellen Messknoten, wird der neue Knoten zu einem Sensinghead und der Knoten mit den geringsten Energiereserven verliert seine Sensinghead-Rolle und wird ein Member-Knoten.

**Nachrichten** Die zuvor beschriebene Auswahl der Messknoten findet beim Clusterhead statt, da nur diesem Anzahl und Eigenschaften aller Knoten des Clusters bekannt sind. Damit die Knoten wissen, ob sie Messungen durchführen sollen oder nicht, müssen sie über ihre Rolle informiert werden. Aus diesem Grund wurden zwei neue Nachrichtentypen eingeführt: *MeasurementRequest* (*MR*; dt. „Messanfrage“) und *RevokeMeasurement* (*RM*; dt. „Zurücknahme/Widerruf der Messung“). Bei Beitritt eines Clusters nehmen Knoten zunächst immer an, dass sie Messungen durchführen sollen. Ist dies nicht nötig, informiert der Clusterhead durch das Senden einer *RM*-Nachricht, dass der Adressat kein Sensinghead ist. *MR*-Nachrichten werden hingegen genutzt, um Member-Knoten mitzuteilen, dass sie ab sofort Sensingheads sind und Messungen durchführen sollen. Durch die beim Beitritt implizite Annahme der Sensinghead-Rolle, werden *MR*-Nachrichten bei SSE nur bei Austritten von Knoten versendet.

### 3.3.3 Evaluation

Ziel von DMHC ist es, die Datenerfassung bei großflächigen Kampagnen in PSNs effizienter zu gestalten, d. h. es sollen weniger redundante Messungen durchgeführt und die Kosten für Betreiber und Nutzer gesenkt werden. Im folgenden Abschnitt

soll das Verhalten von DMHC und die Effekte in Bezug auf dieses Ziel analysiert werden. Zu diesem Zweck wurden mit einer eigens entwickelten Simulationsumgebung verschiedene Simulationen durchgeführt. Im Folgenden wird daher zuerst die Simulationsumgebung beschrieben, bevor in den darauf folgenden Abschnitten die Ergebnisse präsentiert und diskutiert werden.

### 3.3.3.1 Simulationsumgebung

Für die Simulationsumgebung kam wieder die JiST/SWANS-Umgebung [282] zum Einsatz, wobei für die Visualisierung die Erweiterung swans++ [257] verwendet wurde. Die darauf aufbauend entwickelte DMHC-Erweiterung setzt die Gruppierung der Knoten und die Vergabe der Rollen innerhalb der Cluster um.

Abbildung 3.11 zeigt einen Screenshot der Simulationsumgebung, bei dem zur Veranschaulichung ein Cluster vergrößert ist. Das Cluster bestand in diesem Fall aus zehn Knoten, was durch die gemeinsame Hintergrundfarbe der Knoten gekennzeichnet ist. Die Rolle der Knoten wurde durch die Farbe der Knoten-ID markiert. Die ID des Clusterheads wurde rot, die der Sensingheads wurde gelb eingefärbt. Member-Knoten wurden mit blauen IDs dargestellt.

Mit dieser Umgebung wurden verschiedene Simulationen durchgeführt. Wie in Tabelle 3.2 angegeben, wurden Szenarien mit bis zu 100 Knoten simuliert, wobei jeder Durchlauf 50-mal wiederholt wurde. Die Verteilung der mobilen Knoten im Simulationsgebiet war dabei zufällig und die Geschwindigkeit dieser wurde auf 6 m/s, d. h. umgerechnet 21,6 km/h, begrenzt, da die Knoten Fußgänger und Fahrradfahrer repräsentierten. Die Ad-hoc-Kommunikation der Knoten untereinander wurde durch WLAN implementiert, wohingegen für die Kommunikation der Clusterheads eine ständig verfügbare Datenverbindung zum Anwendungsserver angenommen wurde.

Knoten	1–100
Simulationszeit	6 Stunden
Simulationsgebiet	5×5 km
Mobilität	Random Waypoint Model
Geschwindigkeit	1–6 m/s
Pause	10s
Kommunikation	WLAN mit Freiraumdämpfung
Sendeleistung	15 dBm
Antennengewinn	1 dB

Tabelle 3.2: Simulationsparameter für die Evaluation von DMHC

### 3.3.3.2 Anzahl der Mobilfunk-Übertragungen

In einem ersten Schritt wurde die Anzahl der Mobilfunk-Übertragungen untersucht. Mobilfunk-Übertragungen werden von den Clusterheads genutzt, um die aggregierten Messdaten an den Anwendungsserver zu übertragen. Für die Analyse wurde DMHC mit dem naiven Ansatz verglichen, bei dem alle Knoten Messungen durchführen und diese direkt an den Anwendungsserver übermitteln. Bei diesem Vergleich ist klar, dass DMHC die Anzahl der Mobilfunk-Übertragungen deutlich senkt, da jeweils nur die Clusterheads Daten an den Server übertragen. Bei zu geringen Knotendichten treten jedoch vermehrt 1-Knoten-Cluster auf, wodurch sich DMHC dem naiven Ansatz annähert. Demzufolge war bei dieser Untersuchung vor allem von Interesse, ab welcher Knotendichte eine signifikante Reduzierung dieser mobilen Datenübertragungen zu erkennen ist.

Abbildung 3.12(a) zeigt die Ergebnisse dieser ersten Analyse. Dargestellt ist hier die durchschnittliche Anzahl der Mobilfunk-Übertragungen pro Knoten in Abhängigkeit der Knotenanzahl. Die Simulationen wurden mit unterschiedlichen Messintervallen ( $MI = \{15s, 30s, 120s\}$ ) durchgeführt. MI ist ein entscheidender Faktor für die Anzahl der Übertragungen: Je kleiner das Intervall, desto häufiger werden Messungen aufgezeichnet und desto häufiger müssen die Messwerte an den Anwendungsserver übertragen werden. Dies ist auch an den Werten des naiven Ansatzes ohne Clustering zu erkennen. Bei einer Simulationszeit von 6 Stunden und einer Messung alle 30 Sekunden ( $MI=30$ ) wurden 720 Messungen pro Knoten durchgeführt, wohingegen bei einem viermal so großen Intervall ( $MI=120$ ) nur ein Viertel der Messungen durchgeführt wurden.

Insgesamt kann man erkennen, dass DMHC die Anzahl der Übertragungen deutlich reduzierte. Bei 10 Knoten wurde schon eine Reduzierung der Last von ca. 70 % erreicht, ab 40 Knoten sogar eine Reduzierung von mehr als 95 %. Diese Ergebnisse zeigen, dass der Einsatz von DMHC auch bei geringen Knotendichten sinnvoll ist und dabei helfen kann, die Netzlast in PSNs deutlich zu reduzieren. Bei der Analyse wurde zwar lediglich die reine Anzahl der Übertragungen nicht die übertragene Datenmenge betrachtet, allerdings ist die Annahme, dass durch die auf Seiten der Clusterheads stattfindende Aggregation auch die Datenmenge in vergleichbarem Maße reduziert werden kann.

### 3.3.3.3 Übertragungskosten für Mobilfunk-Übertragungen

In einem zweiten Schritt wurden die monetären Übertragungskosten analysiert. Das Szenario in diesem Fall war, dass ein gewisser Anteil der Nutzer eine Daten-Flatrate hat, d. h. dass für diese Nutzer keine zusätzlichen monetären Kosten für eine Übertragung entstehen. Der Idealfall ist, wenn in jedem Cluster mindestens ein Knoten mit einer solchen Daten-Flatrate existiert, der die aggregierten Messwerte ohne Zusatzkosten übermitteln kann.

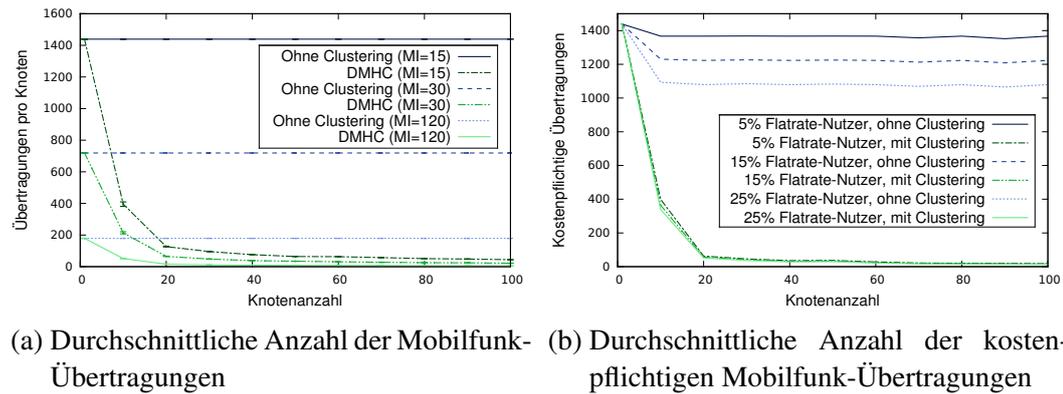


Abbildung 3.12: Analyse der Mobilfunk-Übertragungen und der daraus resultierenden Kosten

Bei den Simulationen dieser Analyse wurde der Anteil der Flatrate-Nutzer mit 5 %, 15 % und 25 % festgesetzt. Für die restlichen Knoten wurde jede Datenübertragung über das Mobilfunknetz als „kostenpflichtige“ Übertragung gezählt. Die Übertragungskosten können durch Einbeziehung der Kosten pro Übertragung (CPT) bestimmt werden. Bei den Simulationsdurchläufen wurde das Messintervall auf  $MI=15s$  gesetzt und PCM-Nachrichten wurden mit einem Intervall (PI) von 3 bis 4 Sekunden per Broadcast verteilt.

Die Ergebnisse sind in Abbildung 3.12(b) dargestellt. Bei den Durchläufen ohne Clustering war eine Reduzierung der Anzahl der kostenpflichtigen Übertragungen zu erkennen, die dem Anteil der Flatrate-Nutzer entspricht. Statt der eigentlichen 1440 Übertragungen reduzierte sich beispielsweise diese Anzahl bei einem Anteil von 25 % Flatrate-Nutzern um 25 % auf 1080 Übertragungen. Bei DMHC war der Einfluss des Anteils an Flatrate-Nutzern relativ gering. In allen drei Szenarien fiel die Anzahl der kostenpflichtigen Übertragungen mit zunehmender Knotendichte stark ab. Somit kann man schließen, dass die größte Reduktion auf das Clustering zurückzuführen ist. Vergleicht man die Werte allerdings mit den Übertragungen pro Knoten aus der vorherigen Analyse, vgl. Abbildung 3.12(a), wird eine deutliche Reduktion bei der Anzahl kostenpflichtiger Übertragungen erkennbar. Betrachtet man wieder das Szenario mit einem Anteil von 25 % von Flatrate-Nutzern, so reduzierte sich die Anzahl der Übertragungen um knapp 70 %. Bei einer höheren Knotendichte stieg die Anzahl der Knoten pro Cluster und somit auch die Wahrscheinlichkeit, dass ein Flatrate-Nutzer in diesem Cluster vorhanden war. DMHC näherte sich somit dem Idealfall an, bei dem keine der Mobilfunk-Übertragungen kostenpflichtig ist.

Die Ergebnisse zeigen, dass DMHC durch die Gruppierung und die Auswahl der Clusterheads, die auf den Übertragungs- und Energiekosten der Knoten basiert, die Anzahl der kostenpflichtigen Übertragungen verringert. Damit werden die Kosten der Nutzer deutlich reduziert, was ein ausschlaggebender Aspekt für eine hohe Nutzerbeteiligung in PSNs ist.

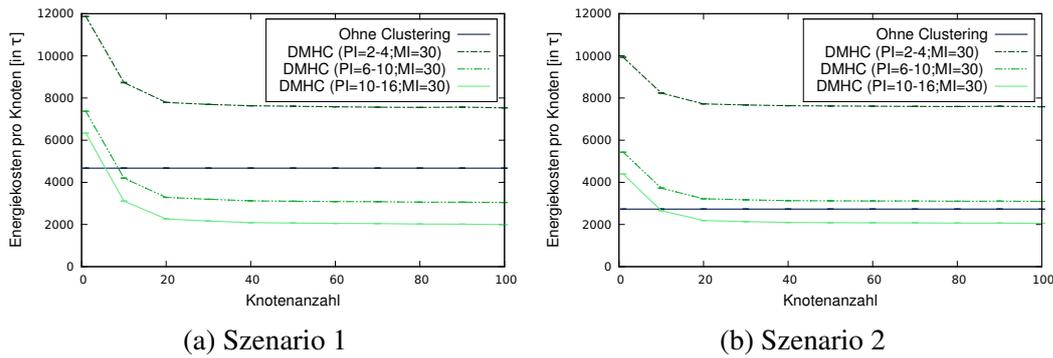


Abbildung 3.13: Energiekosten für die in Tabelle 3.3 spezifizierten Szenarien

### 3.3.3.4 Energiekosten

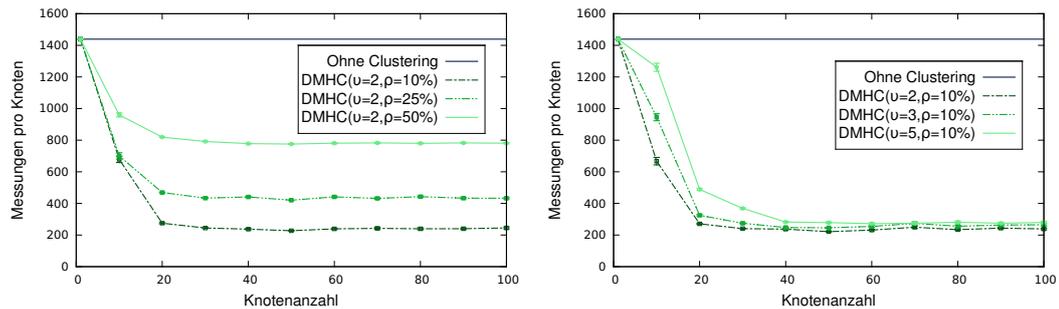
Im Folgenden sollen die Energiekosten untersucht werden. In diesem Kontext wird mit den Energiekosten die Summe der für die Messungen und Übertragungen benötigten Energie bezeichnet.

	Szenario 1	Szenario 2
Mobilfunk-Übertragungen	$6\tau$	$3\tau$
Ad-hoc-Übertragungen	$1\tau$	$1\tau$
Messungen	$0.5\tau$	$0.8\tau$

Tabelle 3.3: Relative Energiekosten für die definierten Szenarien

Für die Bestimmung der Energiekosten wurden zwei Szenarien mit unterschiedlichen Annahmen für die relativen Energiekosten definiert. Letztere werden im Folgenden in  $\tau$  angegeben. Das erste Szenario wurde auf den in [19] präsentierten Untersuchungsergebnissen spezifiziert. Diese zeigen, dass die Energiekosten für eine Mobilfunk-Übertragung im Schnitt ca. sechsmal so hoch sind, wie die für eine Ad-hoc-Übertragung. Des Weiteren wurde für dieses Szenario angenommen, dass die benötigte Energie für die Durchführung der Messungen im Vergleich zu den Übertragungen relativ gering ist, weshalb die Kosten für eine Messung auf die Hälfte der Energiekosten für eine Ad-hoc-Übertragung festgelegt wurden. Im zweiten Szenario sollte untersucht werden, wie sich geringe relative Kosten für die Mobilfunk-Übertragungen auf die Ergebnisse auswirken. Daher wurden diese um den Faktor 2 reduziert. Zusätzlich wurden auch die Energiekosten für die Messungen leicht erhöht, so dass sie nur noch 20 % geringer im Vergleich zu den Ad-hoc-Übertragungen sind. Tabelle 3.3 fasst die relativen Energiekosten für beide Szenarien zusammen.

In den Simulationen wurden zudem drei verschiedene Intervalle für den Versand von PCM-Nachrichten ( $PI$ ) bei gleichbleibenden Messintervall  $MI=30s$  untersucht. Im ersten Durchgang wurde  $PI$  auf 2–4 Sekunden gesetzt, so dass die PCM-Nachrichten sehr häufig versendet wurden. In diesem Fall wurden folglich sehr viele Ad-



(a) Bei variierendem Anteil an Sensingheads  $\rho$  pro Cluster (b) Bei variierender Minimalanzahl an Sensingheads  $v$  pro Cluster

Abbildung 3.14: Anzahl der Messungen pro Knoten

hoc-Nachrichten pro durchgeführter Messung ausgetauscht. Im zweiten und dritten Fall wurde PI mit 6–10 Sekunden bzw. mit 10–16 Sekunden gewählt, so dass deutlich weniger Ad-hoc-Übertragungen pro Messung anfielen.

Die Ergebnisse für das erste Szenario sind in Abbildung 3.13(a) dargestellt. Die Energiekosten bei der Verwendung von DMHC waren mit mittlerem und großem PI deutlich geringer als die des naiven Ansatzes ohne Clustering. Lediglich bei sehr häufigem Versenden von PCM-Nachrichten, d. h. mit  $PI=2-4s$ , überstiegen die Energiekosten die des Referenzansatzes. In diesem Fall wurden im Schnitt zehn Ad-hoc-Nachrichten pro Messintervall versendet, was durch die Einsparung von Mobilfunk-Übertragungen und Messungen nicht kompensiert werden konnte.

Abbildung 3.13(b) zeigt die Ergebnisse für das zweite Szenario. Hier überstiegen die Energiekosten die des Referenzansatzes schon beim Durchlauf mit mittlerem PI, allerdings nur in geringem Maße. Selbst diese höheren Energiekosten können noch akzeptabel sein, wenn dadurch wie zuvor gezeigt die Übertragungskosten deutlich reduziert werden können. Bei der Verwendung eines großen PI sind die Energiekosten aber auch in diesem Szenario noch unter denen des naiven Ansatzes.

### 3.3.3.5 Anzahl der Messungen

In einem weiteren Schritt soll der Einfluss der Parameter für die Anzahl der Messknoten,  $\rho$  und  $v$ , auf die effektive Anzahl an Messungen untersucht werden. Wie in Abschnitt 3.3.2.2 beschrieben, bestimmt  $\rho$  den Anteil der Sensingheads in den Clustern, wohingegen  $v$  die minimale Anzahl an Sensingheads festlegt.

Diese Analyse untersuchte dabei zwei Varianten. Bei der ersten wurde der Anteil der Sensingheads mit  $\rho=\{10\%, 25\%, 50\%\}$  bei den Simulationen variiert, wogegen die minimale Anzahl an Messknoten mit  $v=2$  konstant belassen wurde. Bei der zweiten Variante wurde dagegen die minimale Anzahl an Sensingheads mit  $v=\{2,3,5\}$  variiert und der Anteil der Messknoten konstant bei  $\rho=10\%$  belassen. In beiden Fällen wurde zudem  $MI=15s$  und  $PI=3-4s$  festgesetzt.

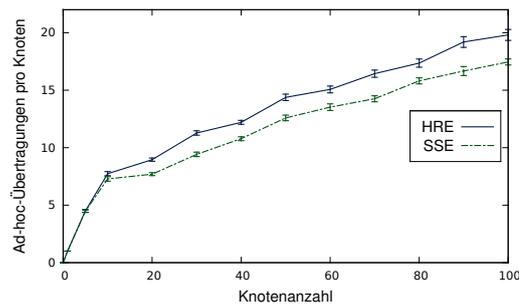


Abbildung 3.15: Ad-hoc-Overhead zur Bestimmung der Messknoten

Die Ergebnisse der ersten Variante sind in Abbildung 3.14(a) dargestellt. Diese zeigen, dass die durchschnittliche Anzahl an Messungen pro Knoten relativ schnell konvergierte. Schon ab 30 Knoten blieb die Anzahl an Messungen stabil, was zeigt, dass die Einführung von Sensingheads auch schon bei geringen Knotendichten sinnvoll ist. Dabei konvergierte die Anzahl immer leicht über dem mit  $\rho$  angegebenen Anteil an Messungen des naiven Ansatzes. Bei letzterem wurden insgesamt 1 440 Messungen pro Knoten durchgeführt. Bei DMHC mit  $\rho=10\%$  erfolgten bei 100 Knoten im Schnitt 244,89 Messungen, was ca. 70 % über der 10 %-Marke von 1 440 Messungen liegt. Mit  $\rho=25\%$  und  $\rho=50\%$  erfolgten bei 100 Knoten im Schnitt ca. 432,87 bzw. 780,92 Messungen, was den jeweiligen Anteil an Messungen des naiven Ansatzes nur leicht übersteigt. Dass die Anzahl der Messungen immer etwas höher war als durch  $\rho$  spezifiziert, hat zwei Gründe: Zum einen bestimmt  $\rho$  den minimalen Anteil an Sensingheads. Der Anteil der Messknoten eines Clusters ist somit immer größer oder gleich dem Wert von  $\rho$ . Zum anderen hat die Festlegung einer Mindestanzahl  $v$  von Messknoten zur Folge, dass bei kleinen Clustern der Anteil an Sensingheads immer deutlich über dem spezifizierten Wert liegen wird. Bei  $\rho=10\%$  und  $v=2$  kann erst bei einer Anzahl von 20 Knoten pro Cluster der effektive Anteil an Messknoten bei 10 % liegen.

Die Ergebnisse in Abbildung 3.14(b) führen zu vergleichbaren Schlüssen. Der Einfluss von  $v$  ist bei geringer Knotenanzahl zwar deutlich erkennbar, so führten größere Werte von  $v$  bei kleineren Knotendichten zu einer geringeren Reduktion. Allerdings wurde ab 40 Knoten ein stabiler Durchschnitt erreicht und die Menge der Messungen variierte nicht mehr signifikant.

### 3.3.3.6 Overhead zur Bestimmung der Messknoten

Im letzten Schritt wurde der Ad-hoc-Overhead der zwei Verfahren zur Auswahl der Messknoten, SSE und HRE, untersucht. Der größte Anteil des generellen Ad-hoc-Overheads entsteht durch den periodischen Versand der PCM-Nachrichten. Die Höhe dieses Overheads hängt nur von dem Intervall  $PI$  ab und ist zudem für SSE und HRE identisch. Außerdem werden die PCM-Nachrichten nicht direkt für die Bestimmung der Sensingheads benötigt. Vielmehr soll die Anzahl der Nachrichten

untersucht werden, die für die Gruppierung und Bestimmung der Messknoten direkt relevant sind, d. h. CH-, JOIN-, RM- und MR-Nachrichten.

Die Ergebnisse sind in Abbildung 3.15 dargestellt. Sie zeigen, dass SSE etwas besser abschneidet als HRE. Das liegt daran, dass HRE bei jedem Bei- oder Austritt eines Knotens die Sensingheads neu bestimmt und somit auch öfter entsprechende MR- oder RM-Nachrichten versenden muss. Allerdings war die absolute Anzahl bei beiden Verfahren relativ gering, vor allem im Vergleich zu dem gesamten Ad-hoc-Overhead, so dass auch der Einsatz von HRE sinnvoll sein kann, da bei diesem die Verteilung auf den verbleibenden Energiereserven basiert und somit die Aufgabenverteilung aus Sicht der teilnehmenden Nutzer fairer ist.

### 3.3.4 Diskussion und Erweiterungen

Die Ergebnisse zeigen, dass das Konzept von DMHC prinzipiell für PSNs sinnvoll ist. In diesem Abschnitt sollen aber kritische Aspekte und Erweiterungsmöglichkeiten hervorgehoben werden.

In PSNs ist neben der Datenerfassung der Schutz der Privatsphäre ein zentraler Aspekt, vgl. Abschnitt 2.5. Letzterer wurde bei DMHC ausgeklammert, da der Fokus des Ansatzes die Schonung der Ressourcen ist. Bei einem Einsatz in einem realen PSN muss die Privatsphäre der Nutzer natürlich auch mit beachtet werden. So sollte DMHC auf jeden Fall einen Ansatz zum anonymen Datenaustausch bereitstellen und gegebenenfalls auch sicherstellen, dass durch die Messdaten keine sensiblen Daten bekannt werden. Eine Möglichkeit ist die Integration des entwickelten *Private-Pooling*-Konzeptes [301]. Dieses ermöglicht eine gesicherte Kommunikation in Gruppen und entkoppelt die geteilten Informationen von den beitragenden Nutzern.

Eine weitere Verbesserungsmöglichkeit stellt die Kombination von DMHC mit Datenkompressions- oder Model-basierten Verfahren, wie etwa in [137], dar. Dadurch kann die Anzahl der Übertragungen und die dabei übermittelte Datenmenge nochmals reduziert werden.

Nicht betrachtet wurde zudem, ob verschiedene Sensorarten für die Erfassung benötigt werden. Hier wäre z. B. die Einführung von Prioritätsklassen möglich, um Geräte mit seltener verfügbaren Sensortypen zu schonen. Sei  $S_{Req}$  die Menge der Sensortypen, die für die Messungen verwendet werden sollen, und  $S_j$  die Menge der verfügbaren Sensoren des Knotens  $j$ . Bei einem Cluster mit  $n$  Knoten kann dann die Prioritätsklasse  $P_j$  wie folgt bestimmt werden:

$$P_j = \max_{s \in (S_j \cap S_{Req})} \left( n - \sum_{k=1}^n |\{s\} \cap S_k| \right). \quad (3.6)$$

Falls ein Knoten keine der geforderten Sensoren besitzt, d. h. wenn  $S_j \cap S_{Req} = \emptyset$ , kann dieser bei der Bestimmung der Messknoten ignoriert werden. Für die anderen

Knoten sortiert Gleichung 3.6 die Knoten so, dass die über seltene Sensoren verfügenden Knoten in höhere Klassen eingeordnet werden. Die Idee dabei ist, diese Knoten nur für Messungen zu verwenden, bei denen die seltenen Sensoren benötigt werden, damit diese ihre Energiereserven nicht durch Messungen verschwenden, die auch von anderen Knoten durchgeführt werden können. Die Einordnung soll an einem kurzen Beispiel verdeutlicht werden: Sei ein Cluster mit vier Knoten gegeben und die Messungen sollen mit den Sensoren des Typs T1-3 durchgeführt werden. Den Knoten sollen dabei die in Tabelle 3.4 aufgelisteten Sensoren zur Verfügung stehen.

Knoten	Verfügbare Sensoren	Prioritätsklasse
1	T1, T2, T3	3
2	T1	0
3	T1, T4	0
4	T1, T2	2

Tabelle 3.4: Prioritätsklassen der Knoten eines Clusters für  $S_{Req}=\{T1,T2,T3\}$

In diesem Fall hat der Knoten 1 die höchste Prioritätsklasse, da dieser als einziger Knoten des Clusters einen Sensor vom Typ  $T3$  besitzt. Die Knoten 2 und 3 haben die niedrigste Prioritätsklasse, da beide von den gewünschten Sensortypen  $S_{Req}$  nur  $T1$  integriert haben und  $T1$  bei allen Knoten des Clusters vorhanden ist. Somit können bei Messungen, die den Typ  $T1$  benötigen, die Knoten 2 und 3 bevorzugt verwendet werden, wohingegen Knoten 1 und 4 für Messaufgaben des Typs  $T2$  bzw.  $T3$  geschont werden.

Ein weiterer kritischer Punkt ist die Wahl der Parameter  $\rho$  und  $\nu$ . Diese haben einen entscheidenden Einfluss auf die Effizienz von DMHC, allerdings ist die Festlegung dieser nicht trivial. Des Weiteren kann durch eine statische Belegung dieser Parameter nicht auf Umstände der realen Welt reagiert werden. Daher wurden als Erweiterung von DMHC sechs Strategien zur adaptiven Auswahl der Messknoten entwickelt. Diese können in zwei Gruppen unterteilt werden: Zum einen Strategien, bei denen die Anzahl an Sensingheads basierend auf den gemessenen Werten angepasst wird. Zum anderen Strategien, die für eine solche Anpassung den Ort und die Abdeckung der Knoten betrachten.

**Wertebasierte Anpassung** Bei diesen Strategien erfolgt eine Erhöhung oder Reduzierung der Anzahl der Messknoten basierend auf den aktuell gemessenen Werten. Messen die Sensingheads sehr ähnliche Werte, wird davon ausgegangen, dass die Messgröße im Bereich dieses Clusters relativ konstant ist und somit weniger Messknoten benötigt werden, um ohne große Informationsverluste ein realistisches Abbild der physischen Welt zu erhalten. Messen die Sensingheads sehr unterschiedliche Werte, ist jeder einzelne Messwert sehr wichtig und trägt zu einem besseren Abbild der Realität bei. Folglich wird die Auflösung der Messpunkte erhöht.

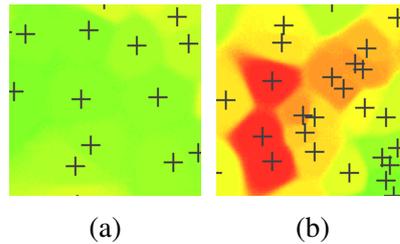


Abbildung 3.16: Wertebasierte Anpassung der Anzahl an Messknoten in einem Gebiet (a) mit relativ konstanten Messwerten bzw. (b) mit stark variierenden Messwerten

Abbildung 3.16 soll dies verdeutlichen. Die Farben repräsentieren die Höhe des Messwertes und die Kreuze die Messknoten. Abbildung 3.16(a) stellt ein Gebiet dar, in dem die Messwerte nur sehr geringfügig variieren. In solchen Fällen kann die Anzahl der Sensingheads ohne große Informationsverluste reduziert werden. Bei Abbildung 3.16(b) unterscheiden sich die Messwerte innerhalb des Gebietes stark. Da in diesen Fällen der Informationsverlust bei Weglassen von Sensingheads sehr groß sein kann, wird die Anzahl daher erhöht.

Für diese wertebasierte Anpassung wurden drei Strategien entwickelt:

- *Linear*: Es erfolgt eine lineare Anpassung der Anzahl. Dafür wird die Abweichung der Messwerte untereinander bestimmt. Unter- bzw. überschreitet diese eine bestimmte Grenze, wird der Anteil der Messknoten um einen festgelegten Betrag reduziert bzw. erhöht.
- *Adaptiv*: In diesem Fall erfolgt die Anpassung basierend auf der Höhe der Abweichung. Weichen die Messwerte sehr stark ab, wird der Anteil der Messknoten um einen größeren Betrag erhöht als bei kleineren Abweichungen.
- *Mobilität*: Die Anpassung erfolgt wie bei der zuvor beschriebenen adaptiven Strategie, nur mit zusätzlicher Berücksichtigung der Geschwindigkeit des Clusterheads. Ist diese gering, d. h. das Cluster ist verhältnismäßig konstant an einem Ort, wird die Anzahl der Messknoten schneller reduziert als bei Clustern mit hoher Geschwindigkeit.

**Abdeckungs-basierte Anpassung** Bei der zweiten Gruppe von Strategien erfolgt die Anpassung der Messknotenanzahl basierend auf der Abdeckung der Knoten. Die Abdeckung eines Knotens bezeichnet den Bereich, in dem kein weiterer Messknoten nötig ist, vgl. Gosh und Das [83]. Bei den Strategien wird geprüft, ob ein Messknoten durch die Präsenz anderer Knoten überflüssig ist. Gegebenenfalls wird dieser von seiner Rolle als Sensinghead befreit.

In Abbildung 3.17 ist ein Szenario mit drei Messknoten schematisch dargestellt. A, B und C repräsentieren die Messknoten, die gestrichelten Kreise de-

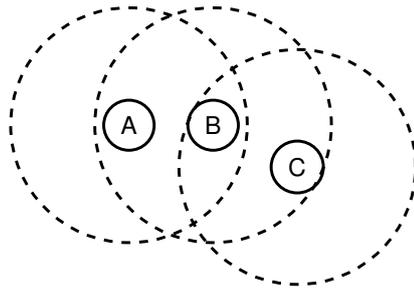


Abbildung 3.17: Drei Messknoten mit überlappender Abdeckung

ren Abdeckung. In diesem Fall könnte Knoten B seine Messungen einstellen, da dieser relativ gut durch die Knoten A und C abgedeckt ist.

Auch für diese abdeckungsbasierte Anpassung wurden drei Strategien entwickelt:

- *Ist-Zustand*: Diese Strategie betrachtet die aktuelle Abdeckung der Knoten und entfernt überflüssige Knoten.
- *Prognose*: Bei dieser Strategie wird anhand der aktuellen Bewegung der Knoten eine Vorhersage über deren Ort und der daraus folgenden Abdeckung zum Zeitpunkt der nächsten Messung gemacht. Wird prognostiziert, dass ein Messknoten bei der nächsten Messung durch die Abdeckung anderer Messknoten überflüssig wird, wird dieser von seiner Rolle als Sensinghead befreit.
- *Gewichtete Prognose*: Das Vorgehen erfolgt analog zur Prognose-Strategie, nur mit zusätzlicher Gewichtung der Vorhersage basierend auf der Geschwindigkeit der Knoten. Dabei werden Vorhersagen von sich langsam bewegenden Knoten höher gewichtet, da angenommen wird, dass diese genauer sind als Prognosen sehr schneller Knoten.

### 3.4 Zusammenfassung

Dieses Kapitel beschäftigte sich mit dem Thema der Ressourcenschonung durch eine effiziente Datenerfassung in PSNs. Diese ist von entscheidender Bedeutung, um einerseits eine große Akzeptanz bei den Nutzern zu erreichen und andererseits die Kosten für Betreiber möglichst gering zu halten. Nach einem Überblick über existierende Arbeiten wurden in diesem Kapitel zwei neue Konzepte vorgestellt, die diese Thematik adressieren.

Im ersten Teil des Kapitels wurde eine ausführliche Übersicht über existierende Arbeiten im Bereich der Datenerfassung gegeben. Da viele Ansätze von Konzepten traditioneller drahtloser Sensornetze inspiriert sind, wurde mit einer Betrachtung

dieser, für WSNs entwickelten, Arbeiten begonnen. Anschließend wurden speziell für PSNs entwickelte Ansätze aus den Bereichen der Positionsbestimmung, der Task-Verteilung und der Datenreduktion bzw. der Datenerfassung beschrieben.

Im zweiten Teil des Kapitels folgte die Präsentation eines entwickelten Konzeptes für eine effiziente Task-Verteilung. Der Ansatz wurde speziell für lokal und zeitlich beschränkte Messungen konzipiert und basiert auf drei Komponenten: der zellbasierten Task-Verteilung, dem adaptivem Monitoring mit Bewegungserkennung und der Task-Bündelung. Nach der Beschreibung des Konzeptes wurde dieses anhand von Simulationen evaluiert und diskutiert.

Im dritten Teil des Kapitels wurde schließlich mit DMHC ein Ansatz zur lokalen Gruppierung von Messknoten vorgestellt, welcher, im Gegensatz zu dem zuvor beschriebenen Konzept, für großflächige und langfristige Messkampagnen entwickelt wurde. Zunächst wurde der Aufbau und Erhalt der Cluster sowie die Bestimmung der Messknoten beschrieben, bevor das Konzept simulativ evaluiert wurde. Abgeschlossen wurde der Abschnitt durch eine Diskussion des Ansatzes und eine ausführliche Betrachtung von Erweiterungen.



## 4 Schutz der Privatsphäre

Neben der effizienten Datenerfassung spielt der Schutz der Privatsphäre der Nutzer eine entscheidende Rolle und ist essentiell für die erfolgreiche Etablierung partizipativer Sensornetze. Der Fokus partizipativer Sensornetze liegt auf der Erhebung und Sammlung von Messdaten. Im Gegensatz zu traditionellen Sensornetzen verwenden PSNs jedoch die Endgeräte der Nutzer als Sensorknoten. Erhobene Daten erhalten folglich, direkt oder indirekt, auch immer Informationen über den jeweiligen Nutzer und erlauben somit auch Rückschlüsse auf das Privatleben der Nutzer. Ohne Kontroll- und Sicherheitsmechanismen könnten PSNs folglich auch genutzt werden, um die Nutzer großflächig und kontinuierlich auszuspionieren.

Ist die Privatsphäre der Nutzer nicht ausreichend geschützt oder haben die Nutzer auch nur das Gefühl, dass nicht ausreichend Wert auf deren Schutz gelegt wird, kann dies zu einer ablehnenden Haltung gegenüber dem PSN führen. In der Regel profitieren die Nutzer selbst nicht direkt von der Erhebung und Übermittlung der Messdaten, sondern tun dies zum Nutzen der Gemeinschaft. Wenn sie dadurch ihre Privatsphäre aufs Spiel setzen, wird dies zur Folge haben, dass Nutzer nicht bereit sind eigene Messdaten beizutragen. PSNs sind jedoch auf eine große Nutzerbeteiligung angewiesen. Nur durch eine hohe Beteiligung können genügend Messungen durchgeführt werden, so dass die erhobenen Daten nicht nur einzelne Stichproben darstellen, sondern relevante und aussagekräftige Ergebnisse liefern können. Daher sind Mechanismen zum Schutz der Privatsphäre der Nutzer unerlässlich.

Dem entgegen steht das Ziel von PSNs, möglichst viele und genaue Daten zu sammeln. Für die Qualität der Ergebnisse wäre es von Vorteil, wenn sowohl die Kontextdaten, wie Ort und Zeit, als auch die Messdaten so präzise wie möglich aufgenommen und übertragen werden. Hinsichtlich der Quantität wäre es wünschenswert, wenn jeder Nutzer kontinuierlich messen würde. Beide Punkte sind jedoch für die Privatsphäre als kritisch einzustufen. Schutzmechanismen müssen daher sehr effizient sein, d. h. sie müssen die Privatsphäre so gut wie möglich schützen, und gleichzeitig möglichst geringe Verluste hinsichtlich Qualität und Quantität der Messdaten erzeugen.

Dieses Kapitel ist wie folgt aufgebaut: Zunächst werden in Abschnitt 4.1 die möglichen Bedrohungen für die Privatsphäre der Nutzer analysiert. Bisherige Ansätze und verwandte Arbeiten werden in Abschnitt 4.2 dargestellt. Anschließend werden zwei eigene Lösungsansätze präsentiert: Als Erstes wird in Abschnitt 4.3 ein Ansatz zur automatisierten Erstellung von Privatsphäre-Zonen präsentiert. Danach folgt die Beschreibung eines Konzeptes zur Bewahrung der Privatsphäre bei Kalibrierungs-

vorgängen in Abschnitt 4.4. Abschließend werden zentrale Aspekte des Kapitels in Abschnitt 4.5 zusammengefasst.

## 4.1 Bedrohungen der Privatsphäre in partizipativen Sensornetzen

Im Folgenden sollen mögliche Bedrohungen der Privatsphäre in PSNs dargestellt werden. Dabei werden nur Bedrohungen betrachtet, die entweder speziell für PSNs sind, d. h. nur in diesen auftreten, oder durch diese stark begünstigt werden. Allgemeine Angriffsszenarien, wie z. B. das Abhören des gesamten Datenverkehrs eines Nutzers, werden hier vernachlässigt. Der Fokus liegt vielmehr auf Bedrohungen, die durch die Interaktion mit dem Anwendungsserver und die an diesen übermittelten Daten entstehen.

Bei den übermittelten Daten kann man zwei Arten unterscheiden: Einerseits Kontextdaten, wie Ort und Zeit, die zur Einordnung der Messungen notwendig sind, andererseits die erfassten Messdaten, z. B. Umweltdaten. Anhand dieser Unterscheidung lassen sich auch die möglichen Bedrohungen für die Privatsphäre untergliedern: Zunächst werden Bedrohungen skizziert, die auf den Kontextdaten basieren. Anschließend werden mögliche Gefahren beschrieben, die durch sensible Informationen in den Messdaten entstehen.

### 4.1.1 Preisgabe von Kontextdaten

In den meisten Anwendungsszenarien partizipativer Sensornetze werden neben den Messdaten auch Kontextinformationen erhoben, vor allem Ort und Zeitpunkt der jeweiligen Messung. Diese werden benötigt, um die Messdaten in den zeitlichen und örtlichen Kontext einordnen zu können. Für den persönlichen Gebrauch können Nutzer zwar auch Messungen ohne Ortsinformationen aufnehmen, beispielsweise Luftqualitätsmessungen zur Berechnung der persönlichen Schadstoffbelastung. Der Informationsgehalt solcher Messungen ohne jeglichen Zeit- und Ortsbezug ist für die Nutzergemeinschaft jedoch sehr begrenzt. In einigen Fällen werden sogar nur Orts- und Zeitinformationen erhoben und stellen folglich auch gleichzeitig die Messdaten dar. Dies ist der Fall in Anwendungen zur Überwachung bzw. der Messung des Verkehrsflusses [94, 226, 258].

Neben dem Zeitpunkt der Messung, für den die lokale Uhr des Gerätes oder auch der Upload-Zeitpunkt verwendet werden kann, ist auch eine relativ präzise Bestimmung der Ortsinformationen mit modernen Smartphones möglich. Zur Ortsbestimmung sind die meisten Geräte heutzutage mit Empfängern für globale Navigationssatellitensysteme (*GNSS*), wie GPS oder GLONASS, ausgestattet. Zudem kann die Position eines Endgerätes auch durch WLAN- oder netzbasierte Ortung erfolgen.

Die Probleme für die Privatsphäre, die durch die Angabe dieser Kontextdaten bei den Messungen entstehen können, sind vielfältig. Neben der allgemeine Nachverfolgung der Bewegungen von Nutzern können diese Informationen auch genutzt werden, um Wohn- und Arbeitsort des Nutzers zu bestimmen [106, 252, 253]. Ferner können Ortsdaten verwendet werden, um auf weitere persönliche Informationen zu schließen, wie die politischen Ansichten des Nutzers, seinen Gesundheitszustand oder seine persönlichen Gewohnheiten [64, 143, 157]. Gesammelte Ortsdaten können einem Angreifer verraten, wie der Nutzer sich fortbewegt [182] und wann er welche anderen Nutzer getroffen hat [173].

Neben den Bedrohungen der Privatsphäre, die durch die Informationen selbst oder durch Schlussfolgerungen aus diesen entstehen, können Ortsinformation auch weitere negative Effekte für den Nutzer haben, falls diese in falsche Hände geraten. Zum einen können die Informationen von zweifelhaften Unternehmen für unerwünschte ortsbezogene Werbung, sogenannten *Location-based Spam*, genutzt werden [64]. Zum anderen kann auch die Sicherheit und das persönliche Wohlbefinden darunter leiden, wenn es durch die bekannt gewordenen Ortsdaten zu physischen Angriffen kommt [64].

Zusätzlich zu den absichtlich erfassten Kontextdaten, die zur Einordnung der Messungen notwendig sind, können weitere Kontextdaten aus gewissen Situationen abgeleitet werden. Dies ist der Fall, wenn allein die Beantwortung einer Messanfrage zusätzliche Informationen preisgibt. Werden Messanfragen gestellt, die sehr restriktiv bei den Anforderungen sind, z. B. indem sie sehr spezielle, selten verbaute Sensoren fordert, kann eventuell auf den verwendeten Gerätetyp oder sogar die Identität des Nutzers geschlossen werden. Bei diesem Vorgehen spricht man von *Narrow Tasking* [217]. Ebenso problematisch ist es, wenn ein Angreifer den Tasking-Server so steuern kann, dass Messanfragen gezielt an einzelne Nutzer verteilt werden. Bei diesem, mit *Selective Tasking* [217] bezeichneten Vorgehen können die Ergebnisse verschiedener Antworten einfach verknüpft werden, wodurch der entsprechende Nutzer dauerhaft verfolgt und überwacht werden kann.

### 4.1.2 Sensible Informationen in Messdaten

Neben den Kontextdaten können auch die Messdaten sensible Informationen über die Nutzer bzw. deren Umgebung enthalten und stellen in den falschen Händen somit eine Bedrohung für die Privatsphäre dar. Das Ausmaß der Bedrohung hängt von der Art der Messdaten ab. In Hinblick auf den Schutz der Privatsphäre sind etwa Sound- oder Bildaufnahmen in der Regel kritischer anzusehen als die Erhebung von Umweltdaten.

Zudem entscheidet auch die Verarbeitung der Daten über das Ausmaß einer möglichen Bedrohung. Werden die Daten lokal verarbeitet, d. h. auf den Endgeräten der Nutzer, und erst anschließend an den Anwendungsserver übermittelt, ist das Risiko normalerweise geringer, als wenn die rohen Daten übermittelt werden. Ein

Beispiel hierfür ist eine Anwendung, welche die Umgebungslautstärke über Tonaufnahmen ermitteln soll. Erwartet der Anwendungsserver die Rohdaten, um die Lautstärke zentral zu bestimmen, können diese Daten auch sensible Informationen, wie z. B. Gespräche in der Umgebung des Nutzers, enthalten. Werden die Tonaufnahmen verarbeitet und nur die resultierende Lautstärke an den Server übermittelt, ist die Gefahr geringer, dass hierdurch sensible Informationen bekannt werden.

Im Folgenden soll nun ein kurzer Überblick über die verschiedenen Arten von Messdaten und die jeweils auftretenden Probleme gegeben werden. Die Auflistung ist an die Bedrohungsanalyse von Christin et al. [46] angelehnt und geht davon aus, dass mögliche Angreifer Zugriff auf die unverarbeiteten Messdaten erlangen:

- **Tonaufnahmen:** Werden, wie in dem zuvor genannten Beispiel, Tonaufnahmen als Messdaten erhoben, kann dies zu verschiedenen Bedrohungen der Privatsphäre führen. Zum einen können private Gespräche der Nutzer aufgezeichnet werden. Zum anderen können gewisse, für bestimmte Events oder Orte einzigartige Hintergrundgeräusche ungewollte Zusatzinformationen über den Nutzer liefern. Dies ist vor allem relevant für Anwendungen mit opportunistischer Datenerfassungen, bei denen automatische Tonaufnahmen ohne aktives Zutun der Nutzer aufgezeichnet werden. [46] Wichtig hierbei ist, dass nicht nur die Privatsphäre des teilnehmenden Nutzers gefährdet ist, sondern potentiell auch die der Personen in der Umgebung des Nutzers. Sind diese in „Hörreichweite“ des aufnehmenden Endgerätes, können auch Gespräche von Personen aufgezeichnet werden, die überhaupt nicht wissen, dass eine Messung in ihrer Umgebung durchgeführt wird [185].
- **Bildaufnahmen:** Werden bei einer Anwendung Bildaufnahmen, d. h. Fotos oder Videos, als Messdaten erhoben, können ähnliche Probleme wie bei Tonaufnahmen auftreten. Das Bildmaterial kann Aufnahmen von Personen und Fahrzeugen enthalten, die sich in der Umgebung des Nutzers befinden. Dadurch kann unter Umständen auf die Identität der Personen, ihre Aktivität zum Aufnahmezeitpunkt und ihre sozialen Beziehungen geschlossen werden [46]. Besonders kritisch ist hierbei, dass die Privatsphäre von Personen betroffen sein kann, die nicht am PSN teilnehmen und sich auch nicht über die Datenerfassung bewusst sind [185].
- **Beschleunigungsdaten:** In manchen Anwendungen partizipativer Sensornetze werden die Daten von Beschleunigungssensoren als Messdaten verwendet, vgl. Abschnitt 2.3. Mit ihrer Hilfe ist es möglich, auf die Aktivität der Nutzer zu schließen [28, 45, 133]. Jedoch ist dies oft auch Teil der Anwendung und den Nutzern daher bewusst. Problematischer sind Szenarien, bei denen Nutzer anhand der Beschleunigungsdaten identifiziert werden können. Dies ist beispielsweise möglich, wenn durch die Sensoren die Gangart des Nutzers ermittelt werden kann [59].
- **Umweltdaten:** Erhobene Umweltdaten, wie Luftverschmutzung oder Temperatur, sind in der Regel für die Privatsphäre der Nutzer unbedenklich. Le-

diglich in Kombination mit weiteren Daten können diese zu einer Bedrohung werden, vgl. [46]. Zudem kann ein Angreifer die Position des Nutzers bestimmen, falls etwa die gemessene Luftzusammensetzung, der Luftdruck oder auch die Temperatur Rückschlüsse auf den Ort des Nutzers ermöglichen.

Die Erhebung anderer Messdaten ist auch möglich, die hier aufgelisteten Kategorien geben nur die in bisherigen Arbeiten am häufigsten verwendeten Messdaten wieder. So wurden z. B. auf biometrische Messdaten, die nur im Rahmen von menschenzentrierten Anwendungen erhoben werden, nicht eingegangen, da der Fokus dieser Arbeit auf umweltzentrierten Anwendungen liegt.

Vom Standpunkt des Datenschutzes und der Privatsphäre betrachtet ist die Teilnahme an einem PSN folglich nicht unkritisch für die Nutzer. Sowohl die beigesteuerten Mess- als auch die Kontextdaten können sensible und persönliche Informationen enthalten. Es wurden jedoch bereits viele Ansätze vorgestellt, die versuchen, diese Bedrohungen zu beseitigen oder wenigstens die Gefahr dieser Bedrohungen zu minimieren.

## 4.2 Klassifizierung existierender Ansätze

Analog zum vorherigen Abschnitt, können existierende Ansätze in zwei Kategorien unterteilt werden: Bei der ersten Kategorie liegt der Fokus auf dem Schutz der Kontextdaten und der Vermeidung von möglichen, auf diesen Daten basierenden Bedrohungen, vgl. Abschnitt 4.1.1. Zur zweiten Kategorie gehören Ansätze, die das Bekanntwerden sensibler Informationen durch die Messdaten verhindern sollen. Dies wird durch verschiedene Formen der Verarbeitung der Messdaten erreicht.

### 4.2.1 Schutz der Kontextdaten

Zusätzlich zu den Messdaten werden, wie bereits erläutert, in der Regel auch Kontextinformationen, wie Ort und Zeit, erhoben. Letztere werden benötigt, um die Messungen in ihren örtlichen und zeitlichen Kontext einordnen zu können. Die Nutzer übermitteln folglich Tupel der Form  $\langle \text{Zeit}, \text{Ort}, \text{Messdaten} \rangle$  an den Anwendungsserver. Damit geben diese jedoch ungewollt preis, dass sie sich zu angegebener Zeit an angegebenem Ort befunden haben.

Um die Privatsphäre der Nutzer zu wahren, sollen diese Informationen aber nicht aus den übermittelten Daten ersichtlich werden. Betrachtet man die Privatsphäre nur mit diesem Bezug auf die Ortsdaten der Nutzer, spricht man auch von *Location Privacy*, also dem Schutz der Ortsdaten. Ziel ist hierbei, zu verhindern, dass Dritte den aktuellen oder einen früheren Aufenthaltsort erfahren können [27], bzw. den Nutzern die Möglichkeit zu geben, über Preisgabe und Verwendung ihrer Ortsdaten selbst zu bestimmen [64]. Neben der manuellen Kontrolle der Nutzer gibt

es verschiedene algorithmische Verfahren zum Schutz von Ortsdaten [132], die im Folgenden kurz vorgestellt werden.

#### 4.2.1.1 Anonymität

Die Verfahren dieser Kategorie versuchen die Privatsphäre der Nutzer zu schützen, indem die Nutzer anonym bleiben. Dies bedeutet, dass Ortsdaten nicht eindeutig einem Nutzer zugeordnet werden können. Eine der einfachsten Möglichkeiten dies zu realisieren, ist die Verwendung von *Pseudonymen* statt der realen Identitäten der Nutzer. Pseudonyme sind hierbei IDs, die keinerlei Bezug zu der Identität des Nutzers haben. Werden Pseudonyme allerdings sehr lange genutzt, kann ein Angreifer eventuell genügend Informationen zu einem Pseudonym erlangen, um auf dessen reale Identität schließen zu können [104, 151].

Um dies zu vermeiden, sollten Pseudonyme regelmäßig gewechselt werden. Die Bestimmung der Zeitpunkte für solche Wechsel ist jedoch nicht trivial, da bei Nutzerpfaden oftmals selbst mit einem Pseudonym-Wechsel die Zusammengehörigkeit zweier Teilpfade erkennbar bleibt. Beresford und Stajano haben hierfür das Konzept der *Mix Zones* [27] eingeführt. Die Annahme dieses Konzeptes ist, dass Nutzer ihre Positionsdaten nur in sogenannten *Application Zones*, in denen ortsbezogene Dienste angeboten werden, preisgeben. Außerhalb dieser Zonen sind die sogenannten *Mix Zones*, in denen die Nutzer ihre Pseudonyme wechseln können. Somit wird verhindert, dass ein Angreifer Pseudonyme verknüpfen kann, da ein neues Pseudonym von jedem Nutzer der *Mix Zone* stammen kann. Um den Schutz der Privatsphäre zu maximieren, wird versucht die *Mix Zones* möglichst groß zu gestalten. In PSNs ist jedoch das Ziel, möglichst viele Messdaten zu erheben, wodurch der Ansatz für dieses Einsatzfeld ungeeignet ist.

Ein anderes Konzept ist die von Gruteser und Grunwald eingeführte  $k$ -Anonymität für Ortsdaten [87]. Bei diesem Ansatz wird bei einer Anfrage statt der exakten Position eines Nutzers nur die Region des Nutzers übermittelt. Diese Region wird so bestimmt, dass diese mindestens  $k - 1$  Nutzer enthält. Ein Betreiber eines ortsbezogenen Dienstes kann somit nicht identifizieren, welcher der  $k$  Nutzer die Anfrage gesendet hat. Für die Umsetzung wird allerdings ein Anonymisierungsdienst benötigt, der die exakten Ortsinformationen der Nutzer erhält und die entsprechenden Regionen berechnet. Mokbel et al. [168] stellen einen ähnlichen Ansatz vor. Die Nutzer können neben dem  $k$ , das der Anonymisierungsdienst garantieren soll, auch eine minimale Fläche angeben, welche die an den LBS-Dienst übermittelte Region in jedem Fall haben soll. Der zweite Parameter stellt sicher, dass auch bei einer hohen Nutzerdichte die Genauigkeit der Ortsdaten ein gewisses Maß nicht überschreitet. Das von Hu und Shahabi vorgeschlagene Vorgehen [101] basiert auf einem sozialen Netz, in dem die Nutzer untereinander durch Freundschaftsbeziehungen verknüpft sind. Statt die Positionsdaten nun direkt zu übermitteln, übergibt der Nutzer die Daten an einen zufälligen Freund. Dies wird nun mehrmals wiederholt, so dass eine nicht-deterministische Nutzerkette einer zuvor festgelegten Länge

entsteht. Der letzte Nutzer in dieser Kette überträgt dann die Daten an den Server, der den ursprünglichen Nutzer nun nicht mehr bestimmen kann.

Neben den genannten Aspekten kann auch schon die Kommunikationsverbindung mit dem Anwendungsserver selbst einige Informationen verraten, wie beispielsweise die ungefähre Position eines Nutzers anhand der IP-Adresse. Um diesem Problem zu begegnen, können Verbindungen über Anonymisierungsnetze wie *TOR* [62] oder *MIX* [41] erfolgen. Bei *AnonySense* [217] wird zunächst lokal geprüft, ob die Messdaten sensible Informationen enthalten. Wenn dies der Fall ist, werden diese erst mit Hilfe eines Anonymisierungsdienstes bearbeitet. Falls dieser nicht benötigt wird, werden die Messdaten über ein MIX-Netz an den Anwendungsserver übermittelt.

#### 4.2.1.2 Unscharfe Kontextdaten

Die zweite Kategorie beinhaltet Verfahren, die versuchen durch unscharfe Kontextdaten die Privatsphäre der Nutzer zu schützen. Der Großteil der Arbeiten in diesem Bereich konzentriert sich vor allem auf die Ortsdaten. Dieses Vorgehen bietet sich bei Diensten an, die keine exakten Ortsdaten benötigen, wie ortsabhängige Nachrichtendienste, bei denen eine Genauigkeit der Position im Kilometerbereich ausreichend ist.

Das Konzept, die Genauigkeit von Ortsdaten zum Schutz der Privatsphäre herabzusetzen, wurde von Duckham und Kulik [65] formal eingeführt. Dabei wird unter anderem auch unterschieden, ob die Unschärfe durch fehlerhafte oder unpräzise Informationen erreicht wird. Bei fehlerhaften Informationen werden Ortsdaten übermittelt, die nicht der Position des Nutzers entsprechen. Eine Möglichkeit besteht darin, die Originaldaten durch Hinzufügen von zufälligem Rauschen zu verfälschen [131]. Bei unpräzisen Informationen wird lediglich die Genauigkeit herabgesetzt, so dass mehrere Orte als aktuelle Position möglich wären. Ein naiver Ansatz hierfür ist, die Koordinaten auf eine bestimmte Anzahl von Nachkommastellen zu begrenzen. Krumm [131] zeigt jedoch, dass die eingeführte Ungenauigkeit sehr groß sein muss, um effektiv die Privatsphäre der Nutzer zu schützen. Statt ungenauer Koordinaten können auch symbolische Orte oder Ortsnamen dem Dienstbetreiber übermittelt werden. So können Ortsnamen wie *Zuhause* oder *Büro* verwendet werden, oder auch beschreibende Angabe wie *In der Nähe von Platz X*. Solche Angaben werden auch als *vage* Ortsdaten bezeichnet [65]. Sind in verschiedenen Situationen unterschiedliche Genauigkeiten gewünscht, kann die Art und das Niveau der eingeführten Unschärfe auch in Form von Regeln definiert werden [139].

Eine weitere Gruppe von Ansätzen beschäftigt sich nicht nur mit dem Schutz einzelner Ortsinformationen, sondern zielt auf den Schutz der zurückgelegten Pfade. In [93] stellen Hoh und Gruteser einen Ansatz vor, bei dem die Pfade von Nutzern in örtlicher Nähe so verfälscht werden, dass diese sich kreuzen. Das Ziel hierbei ist es, dass ein möglicher Angreifer nach solch einer Kreuzung nicht mehr eindeutig bestimmen kann, auf welchem Pfad der Nutzer sich bewegt hat. You et al. [248]

verschleiern den vom Nutzer zurückgelegten Weg, indem sie zusätzlich künstliche Nutzerpfade kreieren.

Für manche Anwendungen sind jedoch exakte Ortsdaten wichtig, wohingegen eine Ungenauigkeit beim zeitlichen Kontext hinnehmbar ist. Anwendungsszenarien sind die Ermittlung des Straßenzustandes [72, 160, 166] oder die Bestimmung gefährlicher Straßenabschnitte [87] mittels partizipativer Sensornetze. Bei Ersterem werden zur Erkennung von Schlaglöchern die Beschleunigungssensoren der mobilen Endgeräte verwendet und erkannte Schäden mittels GPS lokalisiert. Dabei ist die Erfassung der exakten Position der Schlaglöcher wichtig, wohingegen der Zeitpunkt der Erfassung in aller Regel irrelevant ist. In diesem Fall können Nutzer die Übertragung der Daten beliebig verzögern oder diese z. B. monatlich in gebündelter Form übermitteln [56]. Ein anderer Ansatz adaptiert das Konzept der  $k$ -Anonymität auf die zeitliche Verschleierung [87]: Statt einem genauen Zeitpunkt der Datenerfassung wird ein Zeitintervall mit den Messdaten übertragen. Das Zeitintervall wird so groß gewählt, dass mindestens  $k$  Nutzer während dieses Intervalls an dem Ort vorbeigekommen sind.

Eine alternative Schutzmaßnahme ist die Verringerung der zeitlichen Auflösung von Ortsdaten. Die einfachste Maßnahme hierfür ist die Verringerung der Messrate [96]. Ein etwas komplexeres Verfahren stellen Hoh et al. [95] vor, bei dem die Ortsdaten nur freigegeben werden, wenn diese für die Privatsphäre der Nutzer unkritisch sind. Dafür werden einzelne Positionsdaten aus GPS-Tracks entfernt, mit dem Ziel, einen möglichen Angreifer zu verwirren.

#### 4.2.1.3 Ausblenden sensibler Informationen

Die Verfahren dieser dritten Kategorie versuchen die Privatsphäre der Nutzer durch das „Ausblenden“ sensibler Informationen zu schützen. Ausblenden bedeutet, dass keine Informationen des Nutzers an den Anwendungsserver übermittelt werden. Dieses Vorgehen kann angewendet werden, um beispielsweise Ortsdaten in bestimmten Regionen zurückzuhalten. Diese Regionen werden auch *Privatsphäre-Zonen* genannt. Nach außen hin verhalten sich die Endgeräte in diesen Zonen, als seien sie in einem Funkloch.

Wichtig bei der Erstellung von Privatsphäre-Zonen ist es, dass ein Angreifer nicht bestimmen kann, welche Informationen verschleiert werden sollen. Die Zonengrenzen dürfen folglich keine Rückschlüsse auf den zu schützenden Ort erlauben. Ein Ansatz für solche Zonen sind die von Krumm vorgestellten *Ban Zones* [131]. Dabei wird eine kreisförmige Zone erstellt, deren Mittelpunkt von der Position des sensiblen Ortes verschoben ist. Die genaue Konstruktion wird in Abschnitt 4.3.1 beschrieben. Auch die bereits genannten *Mix Zones* [27] stellen eine Form des Ausblendens dar, da in diesen auch keine Ortsdaten übertragen werden. Eine *Mix Zone* verschleiert den Ort des Pseudonymwechsels und verhindert somit, dass die Pfade zweier Pseudonyme dem gleichen Nutzer zugeordnet werden können.

Eine weitere Möglichkeit besteht darin, Ortsdaten an sensiblen Orten nicht nur auszublenden, sondern durch andere Daten zu ersetzen. Bewegt sich ein Nutzer zu einem Ort, den er nicht in seinem Bewegungsprofil enthalten haben möchte, kann dieser Pfadabschnitt durch einen anderen Pfad ersetzt werden. Mun et al. [170] haben mit ihrem *Selective Hiding* ein solches Verfahren entwickelt. Dabei werden die vom Nutzer als kritisch eingestuften Pfadabschnitte durch realistische Ersatzpfade substituiert. Der Ersetzungsalgorithmus greift hierzu auf frühere Pfade des Nutzers zurück und versucht, als Ersatz einen möglichst passenden und glaubwürdigen Pfad zu wählen.

#### 4.2.1.4 Anonyme Task-Verbreitung

Die Erhebung von Messdaten in PSNs wird in der Regel durch Messanfragen, sogenannte Tasks, angestoßen, vgl. Abschnitt 3.2. Durch das Anfordern und Bearbeiten von Tasks gibt der Nutzer jedoch schon Informationen über seinen Standort und das von ihm verwendete Gerät preis.

Besonders kritisch für die Privatsphäre ist es, wenn ein zentraler Tasking-Server alle Messanfragen verwaltet und diese für die Nutzer bereithält, da er dann durch das Herunterladen und Beantworten von Tasks die Mobilität der Nutzer verfolgen kann. Bei dezentralen Ansätzen soll dies verhindert werden. Das von Lu et al. entwickelte Bubble-Sensing [145] stellt ein hybrides Konzept dar, bei dem Tasks durch Broadcasts in der Zielregion des Tasks verteilt werden, vgl. Abschnitt 3.1.2.2. Die Verteilung erfolgt somit lokal durch Ad-hoc-Kommunikation zwischen den Endgeräten. Auf die zentrale Tasking-Instanz wird nur zurückgegriffen, falls in der Zielregion keine Endgeräte vorhanden sind, welche die sogenannte *Tasking Bubble* erhalten können. Ein ähnlicher Ansatz ist die Verteilung der Tasks durch *Tasking Beacons* [122]. Die Nutzer können die Messanfragen dabei ohne Registrierung oder Offenbarung ihrer Identität empfangen und entscheiden, ob sie die Messaufgaben erledigen möchten.

Will man die Teilnahme an einem PSN auf bestimmte Nutzer begrenzen, um glaubhafte Ergebnisse zu erhalten, aber gleichzeitig die Anonymität der Nutzer wahren, stellt die *attributbasierte Authentisierung* [122] ein mögliches Konzept dar. Bei dieser erfolgt der Zugriff auf die Informationen basierend auf bestimmten Attributen, ohne dass der Nutzer seine Identität offenbaren muss. Für eine Realisierung gibt es verschiedene kryptografische Techniken, wie beispielsweise Gruppensignaturen [36]. Bei einem weiteren Ansatz werden die Tasks nur an Orten heruntergeladen, an denen sich viele Nutzer befinden [122]. So wird eine Identifikation eines einzelnen Nutzers erschwert und gleichzeitig stellt dieses Vorgehen sicher, dass keine kritischen Ortsdaten preisgegeben werden.

Da über die verwendete Verbindung schon Informationen wie die Position des Nutzers bekannt werden können, ist auch bei der Task-Verteilung die Verwendung von Anonymisierungsnetzen wie TOR [62] sinnvoll. AnonySense [217] nutzt die-

se Technik, um beim Herunterladen der Tasks Identität und Position der Nutzer zu schützen.

Ein weiteres Problem stellen die in Abschnitt 4.1.1 bereits beschriebenen Angriffe durch Narrow Tasking oder Selective Tasking dar. Eine Gegenmaßnahme für Narrow Tasking wird von Shin et al. [217] vorgestellt: Jeder neue Task wird zunächst von einer glaubwürdigen Instanz, der *Registration Authority (RA)*, geprüft. Dabei wird untersucht, wie restriktiv der Task ist, d. h. es wird geprüft, ob mindestens  $k$  Nutzer diesen Task erfüllen könnten. Ist dies der Fall, wird der Task freigegeben und eine von der RA signierte Nachricht an den Task angehängt. Werden, wie in [68] vorgeschlagen, die Sensoren eines Gerätes um „virtuelle“ Sensoren erweitert, indem die Anfragen, die nicht direkt beantwortet werden können, an andere Endgeräte weitergeleitet werden, ist ein Angriff mittels Narrow Tasking auch nicht mehr ohne Weiteres möglich. Auch ein Angriff durch Selective Tasking wird in [217] verhindert, indem der Ersteller eines Tasks nicht bestimmen bzw. steuern kann, welche Nutzer seinen Task erhalten.

## 4.2.2 Verarbeitung der Messdaten

Messdaten können sensible Informationen enthalten und folglich Quelle für mögliche Privatsphäre-Bedrohungen sein. Daher muss vor der Übermittlung der Daten an den Server sichergestellt werden, dass diese weder sensible Informationen beinhalten noch Rückschlüsse auf solche ermöglichen.

Eine Möglichkeit dies zu realisieren besteht darin, dem Nutzer zu erlauben, die aufgenommenen Daten durchzusehen und manuell freizugeben. Bei Ansätzen wie dem *Selective Archiving* [90] oder *Buffered Capture* [216] werden aufgenommene Daten zunächst lokal zwischengespeichert und nach Ablauf eines bestimmten Zeitintervalls automatisch verworfen, wenn der Nutzer diese Daten nicht explizit freigibt. Der Schutz der Privatsphäre wird hier auf Kosten von Latenz, also der Erhöhung der Zeit von Aufnahme bis zur Verfügbarkeit der Messdaten, und einer umständlichen Handhabung für den Nutzer erreicht [3].

Da jedoch das Ziel partizipativer Sensornetze ist, möglichst viele Daten zu erfassen und gleichzeitig eine hohe Benutzerfreundlichkeit zu erreichen, werden im Folgenden verschiedene Formen der automatisierten Verarbeitung der Messdaten betrachtet. Die existierenden Ansätze basieren auf lokaler Weiterverarbeitung, Aggregation und Verfälschung der Daten.

### 4.2.2.1 Lokale Weiterverarbeitung

Wie zuvor erwähnt, entscheidet die Art der Datenverarbeitung über mögliche Bedrohungen. Werden die Messdaten lokal nicht verarbeitet und direkt in Rohform an

den Anwendungsserver übermittelt, besteht die Gefahr, dass sie sensible Informationen enthalten.

Aus Ressourcen-Sicht kann jedoch eine Übermittlung der Rohdaten sinnvoll sein, da die Auslagerung der Verarbeitung auf Backend-Server, gerade bei komplexen Verarbeitungsschritten, den Energieverbrauch der mobilen Endgeräte deutlich senken kann [51]. In diesem Fall müssen die Nutzer sich darauf verlassen, dass der Server die Daten angemessen anonymisiert und gegebenenfalls enthaltene sensible Informationen nicht veröffentlicht [46].

Eine lokale Vorverarbeitung der Daten verringert das Risiko, dass sensible Informationen preisgegeben werden. Sowohl SoundSense [146] als auch NoiseTube [155] bestimmen die Umgebungslautstärke mittels kurzer Tonaufnahmen. In beiden Fällen werden die Daten aber nur lokal verarbeitet und lediglich die aus den Aufnahmen ermittelte Lautstärke an den Server übermittelt. Hierdurch wird neben dem Schutz der Privatsphäre auch gleichzeitig die zu übertragende Datenmenge erheblich reduziert.

Ein weiterer Ansatz besteht darin, die Daten vor der Übertragung auf sensible Informationen zu untersuchen. Bei Tonaufnahmen kann beispielsweise automatisiert geprüft werden, ob sie Gespräche enthalten [163], und Bildaufnahmen können auf Gesichter [92, 100] oder bestimmten Personen eindeutig zuordenbares Material, z. B. Kennzeichen von Kraftfahrzeugen, untersucht werden. Werden sensible Informationen gefunden, können entsprechende Stellen der Aufnahme bearbeitet oder zurückgehalten werden.

#### 4.2.2.2 Aggregation

Eine andere Möglichkeit die Privatsphäre der Nutzer zu schützen besteht darin, die Daten vor der Übermittlung an den Anwendungsserver zu aggregieren. Ein solches Konzept stellt *PriSense* [214] dar. *PriSense* ist für Anwendungen konzeptioniert, die sensible Informationen der Nutzer sammeln wollen, z. B. deren Herzfrequenz, Gewicht oder Blutdruck, um daraus eine Gesamtstatistik zu bestimmen. Vor der Übermittlung tauschen die Nutzer jedoch Teile ihrer Messdaten untereinander aus und übertragen dann ein Datenverbund aus fremden und eigenen Daten. Infolgedessen ist für den Anwendungsserver nicht mehr erkenntlich, welche Daten von welchem Nutzer stammen.

Ist das Ziel eine additive Aggregation, also etwa die Summe oder der Durchschnitt der Werte, zerlegt jeder Nutzer den gemessenen Wert zufällig in  $n + 1$  Teile und sendet jeweils einen Teil an  $n$  ausgewählte Nutzer. Im nächsten Schritt übertragen alle Nutzer die erhaltenen Datenteile gemeinsam mit dem verbleibenden Teil der eigenen Messdaten. Der Anwendungsserver kann die Daten dann einfach zusammenfassen und somit die entsprechende Statistik erstellen. Nicht-additive Funktionen, wie die Bestimmung des Maximums oder des Medians, werden durch eine Reihe

von Zählabfragen, sogenannten *count queries*, realisiert, wobei die Antworten auf die einzelnen Abfragen wie bei der additiven Aggregation übertragen werden.

Bei dem von Wiesner et al. [301] entwickelten *Private Pooling* werden die Daten lokal ausgetauscht und aggregiert. Damit nicht schon bei diesem Vorgang sensible Daten preisgegeben werden, wurde ein Protokoll entwickelt, das einerseits durch die Verwendung von verschlüsselten Nachrichten eine gesicherte Kommunikation und andererseits durch das Hinzufügen von Rauschdaten auch eine geschützte Aggregation der Daten ermöglicht.

### 4.2.2.3 Datenperturbation

Eine weitere mögliche Strategie ist das Verfälschen von Daten, auch als *Datenperturbation* bezeichnet. Hierbei werden Messdaten absichtlich vor der Übermittlung verfälscht, etwa durch Hinzufügen von künstlichem Rauschen, so dass die ursprünglichen Messwerte nicht mehr eindeutig zu erkennen sind. Ziel solcher Ansätze ist es, durch Verfälschung die Privatsphäre der Nutzer zu schützen, durch Filtern des eingeführten Fehlers jedoch trotzdem relativ exakt Trends oder Verteilungen in den Gesamtdaten rekonstruieren zu können.

Ein mögliches Einsatzszenario ist die Messung des Verkehrsflusses anhand von Geschwindigkeitsdaten der Nutzer. In diesem Kontext sind vor allem die durchschnittliche Geschwindigkeit auf einer Straße sowie die Varianz der Geschwindigkeiten interessant, um beispielsweise Stau oder stockenden Verkehr zu erkennen. Durch die Perturbation soll gleichzeitig verhindert werden, dass die Geschwindigkeit eines einzelnen Nutzers rekonstruiert werden kann.

Viele allgemeine Ansätze der Datenperturbation basieren darauf, dass ein zufälliges Rauschen mit bekannter Verteilung hinzugefügt wird [4, 5, 73], wobei allerdings in späteren Arbeiten bereits gezeigt wurde, dass dies keinen zuverlässigen Schutz der Privatsphäre garantieren kann [104, 123]. Eine andere Möglichkeit besteht darin, für jeden Datentyp verschiedene Granularitätsstufen zu bestimmen und diese, basierend auf dem jeweiligen Kontext des Nutzers, auf die erhobenen Messdaten anzuwenden [46]. In unkritischen Situationen können so die Originaldaten weitergegeben werden, in anderen Fällen angepasste Messdaten in einer der Situation angemessenen Granularität.

Ein speziell für PSNs entwickelter Ansatz ist *PoolView* [79]. Zur Verfälschung der Daten wird zunächst ein Modell erzeugt, das das beobachtete Phänomen möglichst wirklichkeitsnah abbildet. Mit diesem wird dann das Rauschen für sogenannte *virtuelle Nutzer* erzeugt und ermöglicht so realistische Veränderungen der Daten. Da das Modell und dessen statistische Eigenschaften bekannt sind, können Summe, Durchschnitt und Verteilung des hinzugefügten Rauschens bestimmt und aus den Gesamtdaten heraus gefiltert werden, wodurch schließlich Trend und Verteilung der Daten ermittelt werden können.

### 4.2.3 Zusammenfassende Bewertung der Ansätze

Der vorherige Abschnitt zeigt das Spektrum existierender Ansätze zum Schutz der Privatsphäre. Die große Anzahl und die Vielfalt der Ansätze zeugen von der Wichtigkeit dieses Themas und fördern die Entwicklung partizipativer Sensornetze, da bei deren Umsetzung bereits auf viele Konzepte zurückgegriffen werden kann. Der Großteil der Ansätze lässt sich in eine der folgenden Kategorien einordnen: Bei der ersten Kategorie werden unscharfe, d. h. unpräzise oder verfälschte, Daten übermittelt. Bei der zweiten Kategorie werden die Daten vor der Übermittlung weiterverarbeitet, so dass die Übertragung dieser keine Gefahr mehr für die Privatsphäre darstellt.

Bei vielen ortsbezogenen Diensten ist eine Übertragung ungenauer Daten sinnvoll. Soll ein Dienst etwa basierend auf dem Standort des Nutzers Nachrichten oder Wetterinformationen liefern, so wird hierfür keine exakte Position des Nutzers benötigt. Im Kontext partizipativer Sensornetze sind ungenaue Daten jedoch in der Regel ungünstig. Soll die Schadstoffbelastung oder der Lärmpegel erfasst werden, so werden die exakte Position und möglichst genaue Messdaten benötigt. Ungefähre Positionsdaten sind oftmals nicht verwertbar, da schon bei geringen örtlichen Abweichungen, z. B. von einer Hauptstraße zu einer Nebenstraße, komplett andere Werte gemessen werden könnten. Ungenaue Daten sind in PSNs nur dann sinnvoll, wenn es das Anwendungsszenario wie bei PoolView lediglich erfordert, Trend und Verteilung der Daten zu ermitteln, vgl. Abschnitt 4.2.2.3.

Die Ansätze der zweiten Kategorie, bei denen die Daten lokal weiterverarbeitet werden, sind prinzipiell sinnvoll und haben mehrere Vorteile: Einerseits können durch die Verarbeitung sensible Informationen aus den Daten entfernt werden. Andererseits reduziert die Verarbeitung in der Regel auch die Datengröße und führt somit auch zu einer Reduktion der Kommunikationskosten. Allerdings sind die Ansätze sehr anwendungsspezifisch und müssen für jedes Szenario angepasst werden.

Allgemein sind zwei Problemfelder bisher nur unzureichend berücksichtigt worden: Zum einen sollte die Privatsphäre der Teilnehmer eines PSNs per se geschützt sein. Das heißt, dass Schutzmechanismen standardmäßig aktiviert sind und im Idealfall auch automatisch konfiguriert werden. Die Bereitschaft an einem PSN teilzunehmen ist typischerweise geringer, wenn zu Beginn erst umständlich komplexe Privatsphäre-Einstellungen getätigt werden müssen – vor allem, da Nutzer die Einstellungsmöglichkeiten oftmals nicht vollständig verstehen, vgl. [222]. Zum anderen müssen die Ansätze auf alle Phasen ausgedehnt werden, bei denen Interaktionen der Nutzer bzw. ihrer Endgeräte mit dem Server stattfinden. Der Fokus existierender Ansätze liegt bisher vor allem auf den Phasen der Datenerfassung, also der Task-Verteilung und dem Übertragen der Messdaten. Interaktionen in anderen Phasen, wie beispielsweise der Kalibrierung, werden bisher vernachlässigt.

Um diesen Problemen zu begegnen, wurden zwei konkrete Herausforderungen in den genannten Bereichen identifiziert, die durch die Ansätze in den folgenden Abschnitten gelöst werden sollen:

- Die Verschleierung wichtiger Orte des Nutzers ohne Eingreifen des Nutzers, um automatisierte Privatsphäre-Einstellungen zu ermöglichen.
- Die Wahrung der Privatsphäre bei Kalibrierungsvorgängen mit dem Ziel, eine umfassende Korrektur von Messwerten zu erlauben und im Zuge dessen eine verbesserte Datenqualität zu erhalten.

### 4.3 Ansatz zur automatisierten Erstellung von Privatsphäre-Zonen

Zu Beginn des Kapitels wurden die verschiedenen Bedrohungen für die Privatsphäre der Nutzer partizipativer Sensornetze skizziert. Ein Hauptproblem liegt dabei in der Übermittlung der Kontextdaten von Messungen. Durch diese gibt der Nutzer bekannt, zu welcher Zeit er sich an welchem Ort aufgehalten hat. Vor allem über die Ortsdaten können mögliche Angreifer viel über die Nutzer erfahren, weswegen die Location Privacy in PSNs ein wichtiges Thema ist.

In diesem Abschnitt soll ein Ansatz vorgestellt werden, der automatisiert Privatsphäre-Zonen erstellt. Das Ziel ist es, wichtige Orte, über die der Nutzer identifiziert werden könnte, zu verschleiern. Da PSNs auf eine große Teilnehmerzahl angewiesen sind, sollten Nutzer ohne große Einstiegshürde daran teilnehmen können. Aus diesem Grund soll die Erstellung dieser Zone automatisiert erfolgen, d. h. ohne notwendiges Eingreifen der Nutzer. Ein zweiter wichtiger Aspekt bei der Erstellung ist der effiziente Schutz der Privatsphäre durch die Zonen. Es soll mit möglichst kleiner Fläche ein möglichst großer Schutz erreicht werden, da durch die Zonen Messungen verhindert werden und folglich die Quantität der Messdaten reduziert wird.

Der Abschnitt basiert auf dem in [303] beschriebenen Konzept und ist dabei wie folgt aufgebaut: Zunächst werden in Abschnitt 4.3.1 die Problemstellung und die zu erfüllenden Anforderungen beschrieben. In Abschnitt 4.3.2 wird das Konzept der *Silent Zones*, ein Ansatz zur automatisierten Erstellung von Privatsphäre-Zonen, vorgestellt und im anschließenden Abschnitt 4.3.3 evaluiert. In Abschnitt 4.3.4 werden abschließend die Ergebnisse diskutiert und mögliche Erweiterungen des Konzeptes beschrieben.

#### 4.3.1 Problemstellung und Anforderungen

Der im Folgenden beschriebene Ansatz ist konzipiert für langfristige Messkampagnen mit opportunistischer Datenerfassung. Opportunistische Datenerfassung bedeutet, dass Messungen ohne Eingreifen des Nutzers durchgeführt werden, wenn die Situation geeignet ist, vgl. Abschnitt 2.3.1. Diese wiederholte und automatisierte Aufnahme der Daten bringt verschiedene Probleme mit sich: Durch die automatisierte Erfassung hat der Nutzer keine genaue Kenntnis, welche möglicherweise



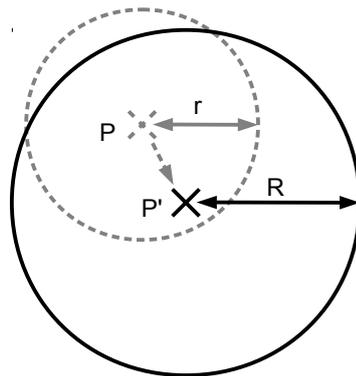


Abbildung 4.2: Konstruktion einer Ban Zone nach Krumm [131]

Der bekannteste Ansatz für Privatsphäre-Zonen sind die von Krumm vorgestellten Ban Zones [131]. Ban Zones sind Kreisflächen mit zufällig verschobenem Mittelpunkt. Die Konstruktion ist in Abbildung 4.2 dargestellt, wobei  $P$  der zu verschleiende Punkt ist. Zunächst wird um  $P$  ein „unsichtbarer“ Kreis mit Radius  $r$  gezogen, in welchem zufällig ein Punkt  $P'$  bestimmt wird. Um  $P'$  wird anschließend ein Kreis mit Radius  $R$  gezogen, welcher schließlich die Ban Zone  $z_{ban}$  darstellt. Mit  $X \in \{\text{Menge aller möglichen Punkte}\}$  gilt für die Ban Zone:  $z_{ban} = \{X | \overline{XP'} \leq R\}$ . Hierbei muss  $R > r$  gelten, damit sichergestellt ist, dass  $P$  in  $z_{ban}$  enthalten ist. Der Nachteil dieses Verfahren ist, dass die Werte für  $r$  und  $R$  für einen effektiven Schutz von jedem Nutzer selbst festgelegt werden müssen, da je nach Lage unterschiedliche Größen für die Ban Zones notwendig sein können. Eine Zone die beispielsweise in einem Stadtgebiet ausreichend groß ist, kann in einem ländlichen Umfeld völlig unzureichend sein.

Da in PSNs Nutzer freiwillig Messdaten beitragen sollen, sollte der Aufwand für die Teilnehmer möglichst gering gehalten werden. Ein Konzept, das eine manuelle Anpassung der Zonen notwendig macht, ist daher für dieses Anwendungsgebiet ungeeignet.

Daher wurden folgende Anforderungen definiert, welche die Privatsphäre-Zonen erfüllen sollen:

- A1** Die Privatsphäre-Zonen sollen verhindern, dass ein Gebäude eindeutig als wichtiger Ort identifiziert werden kann.
- A2** Die Zonen sollen möglichst klein sein, damit dementsprechend wenige Messpunkte verloren gehen.
- A3** Die Zonen sollen automatisiert erzeugt werden. Insbesondere soll bei deren Erstellung kein Eingreifen des Nutzers nötig sein.

### 4.3.2 Konzept

Für die zuvor beschriebene Problemstellung wurde das Konzept der Silent Zones (SZs) entwickelt. Der Name leitet sich vom englischen Begriff für Funkschatten ab und signalisiert, dass sich Nutzer in diesem Bereich ruhig verhalten und keine Daten senden. Silent Zones sind folglich Privatsphäre-Zonen, in denen Nutzer weder Messdaten noch Positionsinformationen senden, und werden um wichtige Orte des Nutzers erstellt. Um die Privatsphäre der Nutzer zu schützen und eine Identifizierung der exakten Adresse der wichtigen Orte zu verhindern, müssen die Zonen groß genug sein. Hierfür wird das Konzept der  $k$ -Anonymität [223] verwendet: Jede Zone muss mindestens  $k$  Gebäude enthalten. Außerdem dürfen Ort und Form der Zone keine Schlüsse erlauben, welches dieser  $k$  Gebäude das ursprüngliche, zu dem wichtigen Ort des Nutzers gehörige Gebäude ist. Dies bedeutet, dass das ursprüngliche Gebäude höchstens mit der Wahrscheinlichkeit  $\frac{1}{k}$  bestimmt werden kann. Dieses Konzept ermöglicht eine Voreinstellung des Privatsphäre-Levels auf einen systemweiten Standardwert, so dass alle Nutzer, auch ohne manuelles Eingreifen, geschützt sind.

Formal werden Silent Zones durch Tupel der Form  $\langle u, G, b \rangle$  dargestellt.  $u$  bezeichnet hierbei das Ursprungsgebäude, also das Gebäude, um das die Silent Zone erstellt wurde.  $G$  beschreibt die Menge der Gebäude, die in der Zone enthalten sind, inklusive des Ursprungsgebäudes  $u$ , und  $b$  beschreibt die Zonengrenzen. Sei  $Z$  die Menge aller möglichen Zonen, dann müssen folgende Bedingungen erfüllt sein:

1.  $\forall z \in Z : |G_z| \geq k$
2.  $z_1, z_2 \in Z : u_{z_1} = u_{z_2} \not\Rightarrow b_{z_1} = b_{z_2}$
3.  $z_1, z_2 \in Z : b_{z_1} = b_{z_2} \not\Rightarrow u_{z_1} = u_{z_2}$

Die angestrebte  $k$ -Anonymität pro Zone wird dabei in Bedingung 1 ausgedrückt, wohingegen Bedingung 2 und 3 ausdrücken, dass weder vom Ursprungsgebäude auf die Zonengrenzen noch von den Zonengrenzen auf das Ursprungsgebäude geschlossen werden kann.

Aus Nutzersicht können die Zonen beliebig groß sein, da größere Zonen in der Regel auch einen besseren Schutz der Privatsphäre bieten. Für Dienste oder Anwendungen, die auf den Daten partizipativer Sensornetze aufbauen, ist es jedoch wichtig, dass sowohl Quantität als auch Qualität der Messdaten stimmen. Die Qualität der Daten wird unter anderem auch durch Überprüfungen mittels Ausreißertests sichergestellt, wofür jedoch eine gewisse Quantität von Nöten ist. Folglich hängt die Qualität indirekt auch von der Quantität der Messdaten ab.

Die Quantität der Messdaten ist daher sehr wichtig für den erfolgreichen Betrieb partizipativer Sensornetze und die Nutzung der gesammelten Messdaten. Je größer die Privatsphäre-Zonen sind, desto mehr potentielle Messdaten gehen verloren.

Daher ist ein Kompromiss nötig, der sowohl die Privatsphäre der Nutzer schützt, gleichzeitig aber möglichst wenig potentielle Messungen verhindert. Dies ist auch

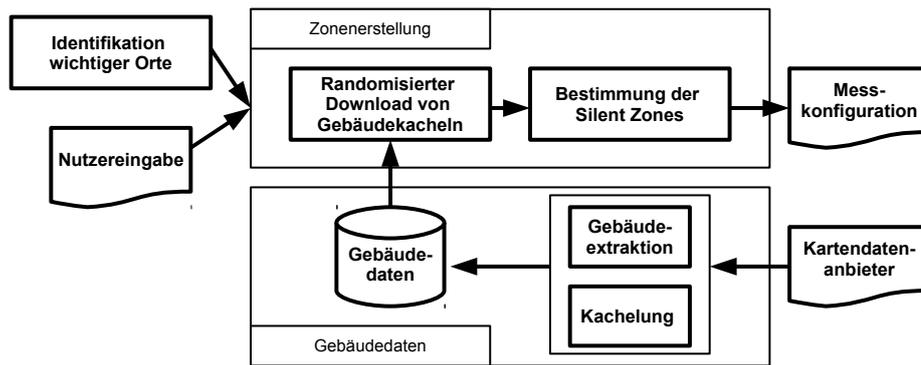


Abbildung 4.3: Architektur des Silent-Zone-Systems

in den Anforderungen A1 und A2 beschrieben, siehe Abschnitt 4.3.1. Neben den drei obigen Bedingungen sollen die Zonen möglichst klein sein. Würde immer die kleinstmögliche Zone bestimmt, widerspräche das den Bedingungen 2 und 3. Das Ziel ist daher, Zonen zu erstellen, die annähernd so klein sind wie die jeweils kleinstmögliche Zone:

$$A(s_z) \approx \min_{u_z} A(s),$$

wobei  $A(s_z)$  die Fläche der Zone  $s_z$  bezeichnet und  $\min_{u_z} A(s)$  die kleinstmögliche Zone für das Gebäude  $u_z$  beschreibt.

#### 4.3.2.1 Systemübersicht

Im Folgenden wird zunächst ein Gesamtüberblick über das für die Erstellung von Silent Zones notwendige System gegeben. In Abbildung 4.3 ist die Architektur mit den verschiedenen Komponenten dargestellt.

Das System besteht aus zwei Hauptkomponenten, der Gebäudedaten-Komponente und der Komponente zur Zonenerstellung. Damit garantiert werden kann, dass in jeder Zone mindestens  $k$  Gebäude enthalten sind, wird die Gebäudedaten-Komponente für die Zonenerstellung benötigt. Als Input werden Kartendaten benötigt, die von Kartendatenanbietern wie etwa OpenStreetMap<sup>1</sup> bezogen werden können. Die Kartendaten enthalten viele Informationen, die für den Prozess der Zonenerstellung nicht benötigt werden. Daher werden diese gefiltert und nur die Gebäudeinformationen extrahiert, siehe Abbildung 4.4. Zudem werden die Kartendaten auch in kleine Teilausschnitte, sogenannte Kacheln, zerlegt. Dies reduziert die Dateigröße und erleichtert folglich den Download durch die Nutzer. Die Gebäudedaten-Komponente kann auch lokal auf den Endgeräten der Nutzer ausgeführt werden, falls Kartendaten auf diesen verfügbar sind. Da dies aber in der Regel nicht der Fall ist, wird im

<sup>1</sup><http://www.openstreetmap.org/>

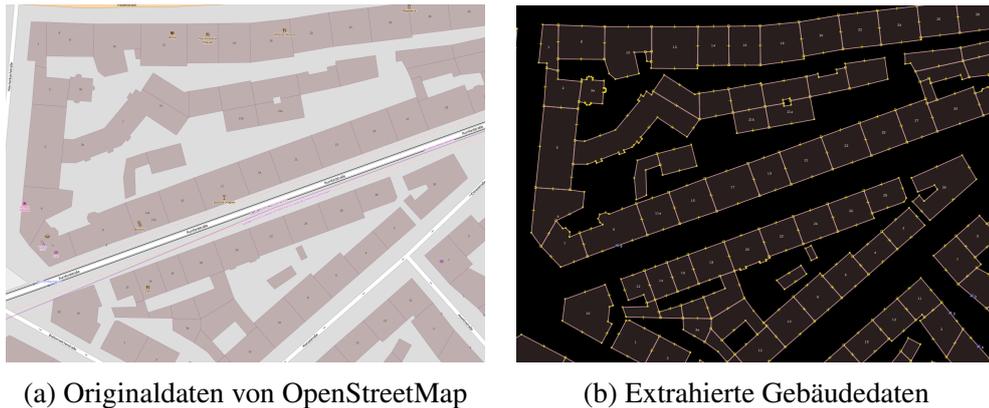
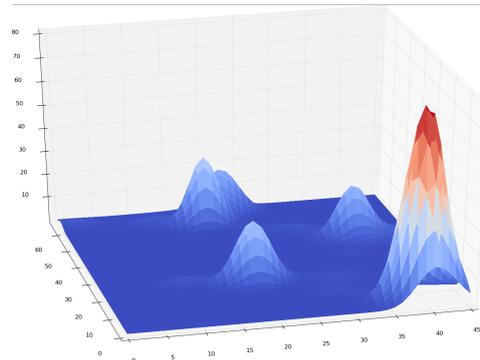


Abbildung 4.4: Gebäudeextraktion aus Kartendaten

Folgendes angenommen, dass diese Komponente durch die Betreiber des PSNs zur Verfügung gestellt wird.

Die zweite Komponente in Abbildung 4.3 ist für die Zonenerstellung zuständig. Die Zonenerstellung wird auf jeden Fall auf den Endgeräten der Nutzer durchgeführt, damit sichergestellt ist, dass auch nur diese über die erstellten Silent Zones Bescheid wissen. Neben den Gebäudedaten muss die Komponente auch Kenntnis über die wichtigen Orte des Nutzers haben. Dies kann durch eine direkte Nutzereingabe geschehen, bei welcher der Nutzer die zu verschleiern den Orte manuell spezifiziert. Alternativ können die wichtigen Orte eines Nutzers aber auch vom Endgerät gelernt werden. Hierzu wird eine Trainingsphase eingeführt, bei der das Endgerät lediglich mittels GPS die Positionen des Nutzers aufzeichnet, jedoch keinerlei Information versendet. Am Ende der Trainingsphase werden die gesammelten Positionsdaten dann verwendet, um die wichtigsten Orte automatisiert zu erkennen. Hierfür gibt es eine Vielzahl verschiedener Verfahren [16, 106, 115, 251–253], die sich durch die Komplexität und die verwendeten Eingabedaten unterscheiden. Im Rahmen des Silent-Zone-Systems wurde eine auf [127, 203] basierende Erkennung umgesetzt. Dafür wird an jedem Messpunkt eine zweidimensionale Normalverteilung angelegt und über alle Messpunkte aufaddiert. Visuell betrachtet wird an jedem aufgezeichneten Ort eine kleine Erhebung generiert, so dass häufig besuchte Orte durch große Erhebungen erkennbar sind. In Abbildung 4.5(a) ist der GPS-Trace eines Nutzers aus dem Dartmouth/Campus-Datensatz [267] dargestellt. Abbildung 4.5(b) zeigt die dazugehörige aufaddierten Normalverteilungen und die daraus resultierenden Erhebungen an häufig besuchten Orten. Das System kann nun mittels eines Schwellwertes  $t$  bestimmen, ab wann ein Ort als wichtig eingestuft werden soll. Gleichzeitig kann auch eine Mindestanzahl wichtiger Orte  $n$  definiert werden, so dass am Ende der Trainingsphase mindestens  $n$  Orte als wichtig eingestuft werden, auch wenn diese  $t$  nicht überschreiten.

Wurden die wichtigen Orte des Nutzers bestimmt, kann der Prozess der Zonenerstellung beginnen. Zunächst müssen die nötigen Gebäudedaten heruntergeladen werden. Damit durch den Download nicht darauf geschlossen werden kann, in wel-



(a) GPS-Trace eines Nutzers aus dem Dartmouth/Campus-Datensatz [267]      (b) Visualisierung der resultierenden Aufenthaltshäufigkeit des Nutzers aus Abbildung 4.5(a)

Abbildung 4.5: Identifikation wichtiger Orte

chem Gebiet sich die wichtigen Orte befinden, wird ein randomisierter Kacheldownload verwendet. Bei diesem wird neben den benötigten Kartenkacheln auch eine Menge zufälliger Kacheln ausgewählt, für die ebenfalls die Gebäudedaten angefordert werden. Sobald die Gebäudedaten lokal auf dem Endgerät verfügbar sind, kann mit der eigentlichen Bestimmung der Silent Zones begonnen werden. Für diesen Schritt werden in den folgenden Abschnitten zwei Ansätze vorgestellt: *Random Rect* und *Random Nearest Neighbor*. Diese zwei Verfahren bestimmen die Zonengrenzen, welche in einer Messkonfiguration gespeichert werden. Die Messkonfiguration verhindert, dass der Nutzer innerhalb der festgelegten Zonen Mess- und Positionsdaten beiträgt.

#### 4.3.2.2 Random Rect

Random Rect (*RR*) ist ein Ansatz, der nach der Trial-and-Error-Methode funktioniert. Wie der Name andeutet, erzeugt dieser Algorithmus zufällige Rechtecke um die zu verschleiende Position und prüft iterativ, ob genügend Gebäude in den Zonengrenzen enthalten sind. Hierfür werden folgende Parameter benötigt:

- *pos*: Die Position des wichtigen Ortes, d. h., der Ort, um den die Zone erstellt werden soll.
- *k*: Spezifiziert die minimale Menge an Gebäuden, die in der Zone enthalten sein sollen.
- *buildings*: Die Gebäudedaten für den Bereich um *pos*.

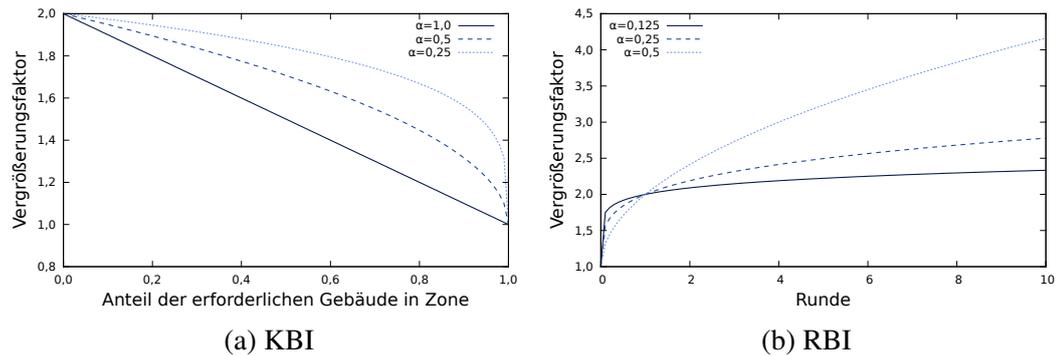


Abbildung 4.6: Vergrößerungsfaktor für Random-Rect-Varianten

- *radius*: Optionaler Parameter, der angibt, wie weit die Zonengrenzen mindestens von dem wichtigen Ort *pos* entfernt sein sollen.

Um die zuvor genannten Bedingungen, vgl. Seite 97, zu erfüllen, müssen die Zonengrenzen und das Zonenzentrum zufällig bestimmt werden. Dabei darf insbesondere zwischen der zu verschleiern Position *pos* und dem Zonenzentrum kein direkter Zusammenhang bestehen.

Im ersten Schritt bestimmt der Algorithmus Breite und Höhe der Zone. Wurde der Parameter *radius* angegeben, wird die Mindestlänge auf  $2 * radius$  festgesetzt, um sicherzustellen, dass die Zonengrenzen entsprechend weit von *pos* entfernt sind. Ist dieser Parameter nicht gesetzt, wird ein Standardwert (*initialLength*) als Mindestlänge verwendet. Die Maximallänge wird in diesem ersten Schritt auf den doppelten Wert der Mindestlänge festgesetzt. Für Breite und Höhe wird dann jeweils ein zufälliger Wert zwischen der Mindest- und der Maximallänge bestimmt. Im nächsten Schritt wird das Zonenzentrum bestimmt. Hierbei wird eine zufällige Verschiebung von der ursprünglichen Position *pos* bestimmt, die sicherstellt, dass der Abstand von *pos* zu den Zonengrenzen immer noch größer oder gleich der zuvor bestimmten Mindestlänge ist. Im letzten Schritt jeder Iteration wird geprüft, ob genügend Gebäude, d. h. mindestens *k* Gebäude, in den Zonengrenzen enthalten sind. Ist dies der Fall, endet der Algorithmus an dieser Stelle und gibt die erzeugte Zone zurück. Sind nicht genügend Gebäude enthalten, werden iterativ neue Zonen erstellt. Bei jeder Iteration wird die Fläche der Zone vergrößert. Die Vergrößerung wird dabei mit Hilfe eines *Vergrößerungsfaktors* ermittelt.

Der Vergrößerungsfaktor verändert sich bei jeder Iteration. Für diese Veränderung wurden zwei Verfahren konzipiert: *k-Based Increase* (KBI) und *Round-Based Increase* (RBI).

**KBI** Die Idee hinter k-Based Increase ist, den Vergrößerungsfaktor an die Anzahl der bereits in der Zone enthaltenen Gebäude anzupassen. Je näher die Gebäudedanzahl der aktuellen Iteration *c* an den angestrebten *k* Gebäuden ist, desto kleiner wird der Vergrößerungsfaktor. Nähert sich *c* an *k* an, fällt die Vergrößerungsfaktor.

ßerung der Zone folglich immer kleiner aus. Der Vergrößerungsfaktor  $f$  wird bei KBI wie folgt bestimmt:

$$f_{KBI} \Leftarrow 1 + \left(1 - \frac{c}{k}\right)^\alpha.$$

$\alpha$  ist hier ein Parameter zu Bestimmung des Grades der Anpassung von  $f$ . So erhält man beispielsweise mit  $\alpha = 1$  eine lineare Anpassung. In Abbildung 4.6(a) ist die Entwicklung von  $f_{KBI}$  in Abhängigkeit des Anteils der erforderlichen Gebäude  $\frac{c}{k}$  mit  $\alpha = \{0,25; 0,5; 1,0\}$  dargestellt.

**RBI** Bei Round-Based Increased wird der Vergrößerungsfaktor basierend auf Rundenanzahl  $i$  vergrößert,  $f$  wird somit bei jeder Iteration größer. Bei RBI wird der Vergrößerungsfaktor  $f$  daher wie folgt bestimmt:

$$f_{RBI} \Leftarrow 1 + i^\alpha.$$

Im Gegensatz zu KBI ist  $f_{RBI}$  nicht begrenzt, sondern wird jede Runde größer. In Abbildung 4.6(b) ist der Vergrößerungsfaktor in Abhängigkeit von der Rundenanzahl mit  $\alpha = \{0,125; 0,25; 0,5\}$  dargestellt.

Algorithmus 1 stellt die gesamte Funktionsweise von Random Rect in Pseudocode dar.

#### 4.3.2.3 Random Nearest Neighbor

Random Nearest Neighbor (*RNN*) ist ein weiterer Ansatz zur Bestimmung von Silent Zones. Im Gegensatz zu dem nach der Trial-and-Error-Methode funktionierenden RR-Algorithmus, stellt RNN ein Verfahren dar, bei dem die Zonen systematisch konstruiert werden. Eine komplett systematische Vorgehensweise hätte einen deterministischen Algorithmus zur Folge und würde damit die Bedingung 2 verletzen. Daher wird auch bei diesem Verfahren durch einen zufallsbedingten Schritt ein nichtdeterministisches Verhalten erreicht. Als Eingabeparameter erwartet RNN die gleichen Parameter wie RR sie erfordert.

Im ersten Schritt wird das Gebäude bestimmt, das am nächsten zu der angegebenen Position  $pos$  liegt. Die zugrunde liegende Annahme ist, dass der zu verschleiende wichtige Ort ein Gebäude ist. Eine weitere Annahme ist, dass das zu verschleiende Gebäude auch das geografisch nächste Gebäude zu  $pos$  ist. Hierzu wird für alle Gebäude aus dem Kartenausschnitt die Distanz zu  $pos$  bestimmt und das Gebäude mit dem kleinsten Abstand ausgewählt. Mit diesem Gebäude wird dann die Menge  $S$  initialisiert.

Im Folgenden wird solange eine Schleife durchlaufen, bis  $k$  Gebäude in  $S$  enthalten sind: Als erstes wird hierzu ein zufälliges Gebäude  $s$  aus der Gebäudemenge  $S$  ausgewählt. Im ersten Durchlauf ist dies immer das Ursprungsgebäude. Als nächstes wird der nächste Nachbar von  $s$  bestimmt und der Gebäudemenge  $S$  hinzugefügt. Für die Bestimmung des nächsten Nachbarn wird die Distanz aller Gebäude aus

---

**Algorithmus 1** Random Rect (RR)

---

**Require:**  $pos \wedge k \geq 0 \wedge B \neq \emptyset [\wedge radius > 0]$   
 $c \leftarrow 0, i \leftarrow 0$   
**if**  $isset(radius)$  **then**  
     $minLen \leftarrow 2 * radius$   
**else**  
     $minLen \leftarrow initialLength$   
**end if**  
**while**  $c < k$  **do**  
     $i \leftarrow i + 1$   
    **if**  $i = 1$  **then**  
         $w \leftarrow getRandom(minLen, 2 * minLen)$   
         $h \leftarrow getRandom(minLen, 2 * minLen)$   
    **else**  
         $f \leftarrow determineIncreaseFactor(i, c, k)$   
         $maxLen \leftarrow (area * f / minLen)$   
         $w \leftarrow getRandom(minLen, maxLen)$   
         $h \leftarrow (area * f / w)$   
    **end if**  
     $area \leftarrow w * h$   
     $dx \leftarrow getRandom(-(w - minLen) / 2, (w - minLen) / 2)$   
     $dy \leftarrow getRandom(-(h - minLen) / 2, (h - minLen) / 2)$   
     $center \leftarrow moveCenter(pos, dx, dy)$   
     $b \leftarrow calculateBounds(center, w, h)$   
     $c \leftarrow countBuildingsInArea(B, b)$   
**end while**  
**return**  $b$

---



---

**Algorithmus 2** Random Nearest Neighbor (RNN)

---

**Require:**  $pos \wedge k \geq 0 \wedge B \neq \emptyset [\wedge radius > 0]$   
 $S \leftarrow selectNearestBuilding(pos, B)$   
**while**  $|S| < k$  **do**  
     $s \leftarrow randomlySelectBuildingFromSet(S)$   
     $n \leftarrow selectNearestBuilding(s, B)$   
     $S \leftarrow S \cup n$   
**end while**  
**return**  $minimumBoundingBox(S[, radius])$

---

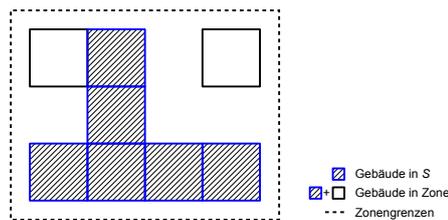


Abbildung 4.7: Schematische Darstellung der Gebäude in  $S$  und in der Zone

dem Kartenausschnitt zu  $s$  bestimmt. Dieses Vorgehen spiegelt sich in dem Namen Random Nearest Neighbor wider: Es wird in jedem Schleifendurchlauf für ein zufälliges Gebäude  $s$  der nächste Nachbar bestimmt.

Im abschließenden Schritt werden die Zonengrenzen der Silent Zone bestimmt. Hierzu wird die Gebäudemenge  $S$  verwendet und das minimal umgebende achsenparallele Rechteck bestimmt, das alle Gebäude aus  $S$  enthält. Ist ein zusätzlicher Radius angegeben, wird das Rechteck so vergrößert, dass die Distanz jedes Gebäudes aus  $S$  zum Rand des Rechteckes mindestens diesem mit *radius* angegeben Abstand entspricht.

Dieses Rechteck wird dann als Ergebnis des Algorithmus zurückgeliefert. Zu beachten ist hierbei, dass die resultierende Silent Zone, trotz der systematischen Vorgehensweise, oftmals mehr als  $k$  Gebäude enthält. Der Algorithmus stellt zunächst sicher, dass auf jeden Fall  $k$  Gebäude enthalten sind. Dann wird um diese  $k$  Gebäude jedoch ein Rechteck gezogen, welches natürlich auch weitere Gebäude enthalten kann. Dies ist auch in Abbildung 4.7 dargestellt: Die schraffierten Gebäude sind die mittels RNN bestimmten  $k$  Gebäude, in diesem Fall gilt  $k = 6$ . Die beiden anderen Gebäude liegen aber innerhalb des minimal umgebenden Rechteckes um die ausgewählten  $k$  Gebäude. Folglich übersteigt die Anzahl der enthaltenen Gebäude die geforderten  $k$  Gebäude in vielen Fällen.

### 4.3.3 Evaluation

Im Folgenden soll das vorgestellte Konzept der Silent Zones evaluiert werden. Da das Konzept möglichst universell verwendbar sein soll, werden zunächst zwei Szenarien definiert:

1. **STADT:** Das STADT-Szenario steht für ein Umfeld mit sehr hoher Bevölkerungsdichte. Als Beispiel wurde in der Evaluation Kartenmaterial der Stadt München verwendet. München ist mit ca. 1,4 Millionen Einwohnern innerhalb der Stadtgrenzen die drittgrößte Stadt Deutschlands. Die Bevölkerungsdichte liegt bei ungefähr 4 500 Einwohnern je  $km^2$ . [269]

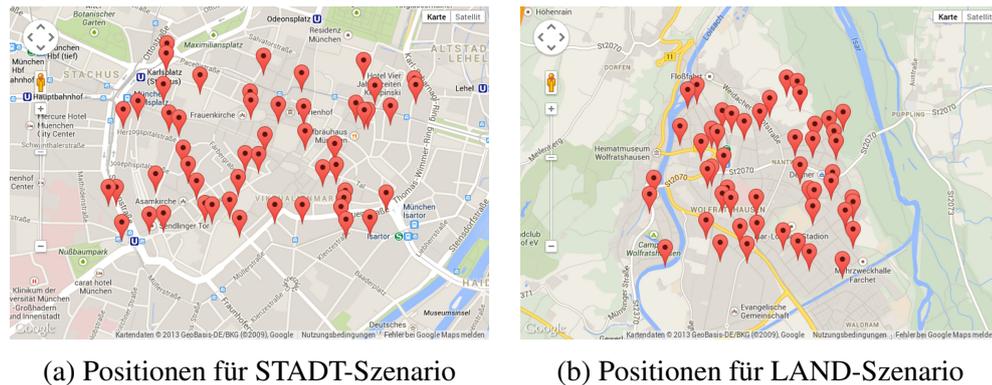


Abbildung 4.8: Zufällig bestimmte Positionen wichtiger Orte

2. **LAND:** Das zweite Szenario steht für ein ländliches dünnbesiedeltes Umfeld. Hierzu wurde Kartenmaterial der Stadt Wolfratshausen verwendet. Wolfratshausen ist eine Kleinstadt und liegt in etwa 30 km südlich von München. Wolfratshausen hat ca. 17 900 Einwohner und die Bevölkerungsdichte liegt bei rund 1 900 Einwohnern je  $km^2$ . [279]

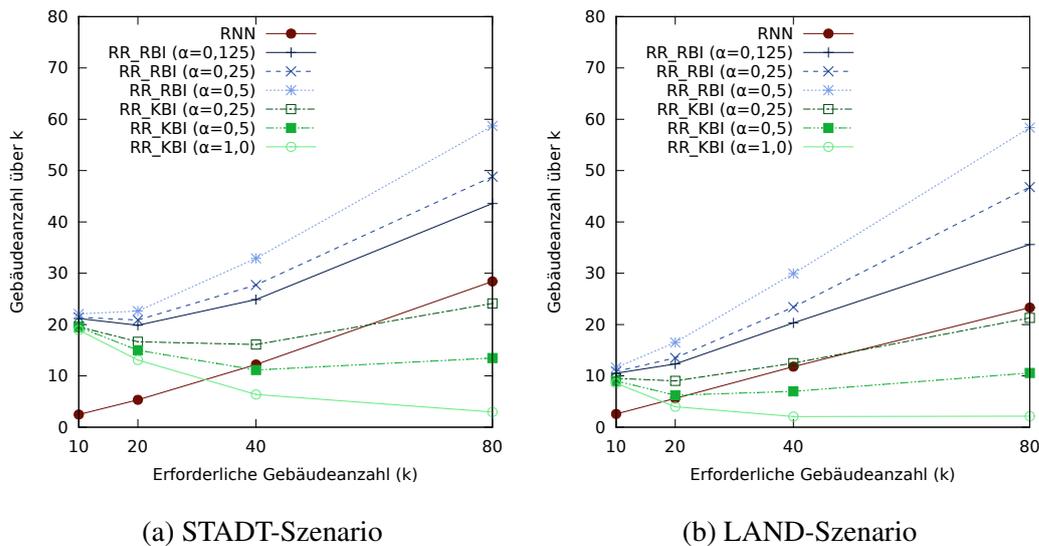
Für jedes Szenario wurden 50 Punkte zufällig bestimmt, siehe Abbildung 4.8, die bei der Analyse für Positionen wichtiger Orte stehen. Für jeden Punkt wurden dann sowohl Ban Zones als auch Silent Zones mit verschiedenen Parametern erstellt. Dabei wurden für jeden Punkt und jede Parameter-Kombination jeweils 20 Zonen pro Verfahren erstellt.

#### 4.3.3.1 Vergleich der Silent-Zone-Algorithmen

In einem ersten Schritt sollen die verschiedenen Ansätze zur Erstellung der Silent Zones, d. h. RNN, RR\_KBI und RR\_RBI, gegenübergestellt werden. Hierfür wurden für die erstellen Zonen jeweils die Anzahl der enthaltenen Gebäude und die Zonengröße erfasst. Für die Random-Rect-Varianten wurden jeweils drei verschiedene Varianten analysiert und zudem auch die benötigte Rundenanzahl, d. h. die Anzahl der benötigten Iterationen, aufgezeichnet.

Bei RR\_RBI verändert sich der Vergrößerungsfaktor  $f$  basierend auf den schon erfolgten Iterationen. Somit bestimmt  $\alpha$  die Steigung der Wachstumskurve von  $f$ , vgl. Abbildung 4.6(b). Ein größerer Wert bedeutet einen schnelleren Anstieg des Vergrößerungsfaktors. Damit die Zonen mit ansteigender Rundenanzahl nicht zu schnell wachsen, sollte ein verzögertes Wachstum von  $f$  modelliert werden. In der Evaluation wurde RR\_RBI daher mit  $\alpha = \{0,125; 0,25; 0,5\}$  verwendet.

Bei RR\_KBI verändert sich  $f$  basierend auf der Anzahl der in den aktuellen Zonengrenzen enthaltenen Gebäuden, vgl. Abbildung 4.6(a). Sind keine Gebäude enthalten, wird die Zonengröße verdoppelt. Nähert sich die Gebäudezahl an die geforderten  $k$  Gebäude an, nähert sich  $f$  dem Wert 1 an. Der Parameter  $\alpha$  bestimmt bei

Abbildung 4.9: Gebäudeanzahl über den geforderten  $k$  Gebäuden

RR\_KBI die Steigung des Abfalls von  $f$ . Mit  $\alpha = 1,0$  erreicht man eine lineare Anpassung, mit  $\alpha < 1,0$  erreicht man ein negativ beschleunigtes Wachstum. RR\_KBI wurde daher mit  $\alpha = \{0,25; 0,5; 1,0\}$  verwendet.

Abbildung 4.9 zeigt die durchschnittliche Gebäudeanzahl pro Zone in Abhängigkeit der geforderten Gebäudeanzahl  $k$ . Die dargestellten Werte zeigen dabei nicht die absolute Gebäudeanzahl  $|G|$ , sondern die Anzahl der Gebäude, die über die geforderten  $k$  Gebäude hinausgehen, d. h.  $|G| - k$ . Somit stellt beispielsweise der Abfall der Kurve von RR\_KBI ( $\alpha = 1,0$ ) keinen Rückgang der absoluten Gebäudeanzahl dar, diese steigt wie gewünscht, siehe Tabelle 4.1. Lediglich die Anzahl überschüssiger Gebäude wird geringer. Das Ziel ist es, möglichst wenige überschüssige Gebäude in den Zonen zu haben, denn diese bedeuten, dass die Zonen auch kleiner hätten sein können, ohne dass das gewünschte Privatsphäre-Level unterschritten worden wäre, vgl. Abschnitt 4.3.2.

$k$	10	20	40	80
$ G $	28,94	33,10	46,37	82,96
$ G  - k$	18,94	13,10	6,37	2,96

Tabelle 4.1: Gebäudeanzahl für RR\_KBI ( $\alpha = 1,0$ ) im STADT-Szenario

Betrachtet man zunächst die Ergebnisse der Random-Rect-Varianten, in der Abbildung 4.9 mit blauen und grünen Graphen dargestellt, ist zu erkennen, dass bei RR\_RBI generell mehr überschüssige Gebäude in den Zonen enthalten sind als bei RR\_KBI. Die besseren Ergebnisse von RR\_KBI resultieren aus dem „Herantasten“ an die richtige Zonengröße, was dadurch erreicht wird, dass der Vergrößerungsfaktor kleiner wird, sobald sich  $|G|$  an  $k$  annähert. Bei RR\_RBI wird der Vergrößerungsfaktor pro Runde erhöht. Dies führt gerade bei großem  $k$  dazu, dass durch die

höhere benötigte Rundenanzahl die Zonengröße stark vergrößert wird und folglich auch die Gebäudeanzahl entsprechend ansteigt. Bei kleineren Werten für  $\alpha$  wird dieser Effekt abgeschwächt, doch ist ein genereller Anstieg der überschüssigen Gebäude zu beobachten.

Bei RR\_KBI stabilisiert sich die Anzahl der überschüssigen Gebäude mit größer werdendem  $k$ . Bei kleinem  $k$  sind zunächst relativ viele Gebäude enthalten, was durch die initiale Zonengröße zu erklären ist. Wird die Zonengröße aber mehrmals angepasst, wie es bei größeren Werten von  $k$  in der Regel notwendig ist, kann die Zonengröße besser an die geforderte Gebäudeanzahl angepasst werden. Gerade die Ergebnisse in Abbildung 4.9(b) zeigen dies deutlich: Da die Bevölkerungsdichte kleiner ist, werden beim LAND-Szenario durchschnittlich mehr Runden benötigt, um Zonen mit bestimmter Gebäudeanzahl zu erreichen, vgl. Abbildung 4.11. Infolgedessen kommt auch das erreichte  $|G|$  sehr nahe an das geforderte  $k$  heran.

Die Random-Nearest-Neighbor-Variante liefert für kleine Werte von  $k$  ebenfalls Zonen, deren Gebäudemenge der gewünschten Anzahl sehr nahe kommt. Mit zunehmenden  $k$  ist dann aber ein relativ konstanter Anstieg der überschüssigen Gebäude zu beobachten. Wie in Abschnitt 4.3.2 bereits erwähnt, liegt dieses Verhalten darin begründet, dass das umgebende Rechteck der  $k$  ausgewählten Gebäude immer größer wird und somit auch immer mehr zusätzliche Gebäude enthält.

Vergleicht man die Ergebnisse des STADT- mit denen des LAND-Szenarios, ist kein wesentlicher Unterschied erkennbar. Die Anzahl der überschüssigen Gebäude ist im Durchschnitt etwas geringer, das grundsätzliche Verhalten der Algorithmen ist jedoch in beiden Szenarien sehr ähnlich. Die geringere Anzahl an überschüssigen Gebäuden erklärt sich durch die geringere Bevölkerungsdichte, wodurch weniger Gebäude pro Fläche vorhanden sind. Dies hat zur Folge, dass bei RNN weniger Gebäude im umgebenden Rechteck enthalten sind bzw. bei den RR-Varianten, die pro Iteration die Fläche vergrößern, die Zone folglich auch um weniger Gebäude erweitert wird.

Beim direkten Vergleich von RR und RNN muss beachtet werden, dass bei RR eine initiale Zonengröße vorgegeben wird. Daher wurden für kleine Werte von  $k$  vergleichsweise große Zonen erstellt. Dies könnte durch eine Verkleinerung der Initialgröße noch angepasst und optimiert werden. Da allerdings RNN für kleine  $k$ -Werte äußerst gut funktioniert, ist dieses Verfahren für kleine Zonen zu empfehlen. Für größere Zonen zeigten die RR\_KBI-Varianten die besten Ergebnisse.

Abbildung 4.10 zeigt die durchschnittliche Zonengröße der erzeugten Silent Zones in Abhängigkeit der geforderten Gebäudeanzahl. Hier fällt zunächst auf, dass die Ergebnisse der verschiedenen Varianten im STADT-Szenario sehr ähnlich sind, was an der hohen Bevölkerungsdichte liegt. Man kann jedoch erkennen, dass für große Werte von  $k$  die KBI-Varianten die kleinsten, d. h. die besten, Zonen erzeugen. Im LAND-Szenario verhält es sich etwas anderes. Zunächst ist ein deutlicherer Unterschied bei den Ergebnissen der verschiedenen Varianten erkennbar. Außerdem liefert hier RNN für alle untersuchten Werte von  $k$  die besten Zonengrößen.

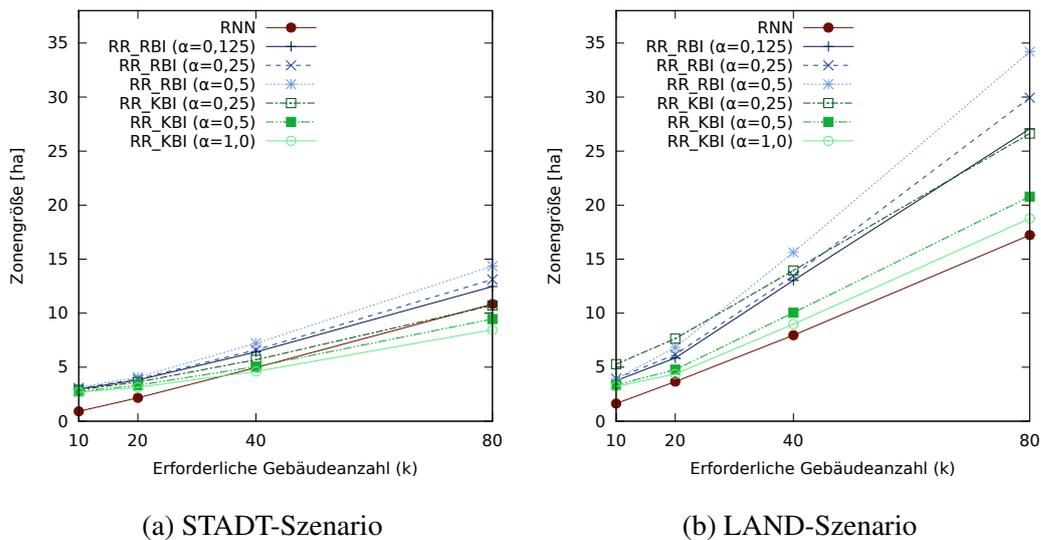


Abbildung 4.10: Durchschnittliche Zonengröße der Silent Zones

In einem letzten Schritt wurde noch für die Runden-basierten Random-Rect-Varianten die Anzahl der benötigten Runden bis zum Erreichen des  $k$ -Anonymitätslevels untersucht. Die Ergebnisse sind in Abbildung 4.11 dargestellt. Man kann erkennen, dass die KBI-Varianten im Durchschnitt mehr Runden benötigen, als dies bei RBI der Fall ist. Dies resultiert aus dem Herantasten von RR\_KBI an die richtige Größe bzw. aus dem schnellen Vergrößern der Zonen bei RR\_RBI. Da aber allgemein die Anzahl der benötigten Runden vergleichsweise gering ist, ist der Einsatz sowohl der RBI- als auch der KBI-Varianten sinnvoll.

#### 4.3.3.2 Vergleich von Silent Zones mit Ban Zones

In einem zweiten Schritt soll nun das Verhalten von Silent Zones und Ban Zones untersucht und in Bezug auf Gebäudeanzahl und Zonengröße verglichen werden. Als Bezugsalgorithmus für die Silent Zones wurde RNN verwendet, da dieser Algorithmus im vorangegangenen Evaluationsschritt gute Ergebnisse für Gebäudeanzahl und Zonengröße lieferte. Zudem erlaubt die systematische Konstruktionsweise der Zonen durch RNN einen besseren Vergleich mit den ebenfalls systematisch konstruierten Ban Zones.

Die erstellten Ban Zones werden in den Ergebnissen mit  $\text{ban}(r, R)$  bezeichnet, wobei  $r$  die maximale Entfernung des Mittelpunkts der Ban Zone vom zu verschleiern Ursprungspunkt und  $R$  den Radius der Zone angibt. Da die Verschiebung des Mittelpunkts die Größe der Zonen nicht beeinflusst, wird  $r$  für alle erstellten Zonen auf  $R/2$  gesetzt. Die erzeugten Silent Zones werden mit  $\text{sz}(v, k)$  bezeichnet. Hierbei steht  $v$  für den verwendeten Algorithmus und  $k$  für die geforderte minimale Gebäudeanzahl innerhalb der Zone.

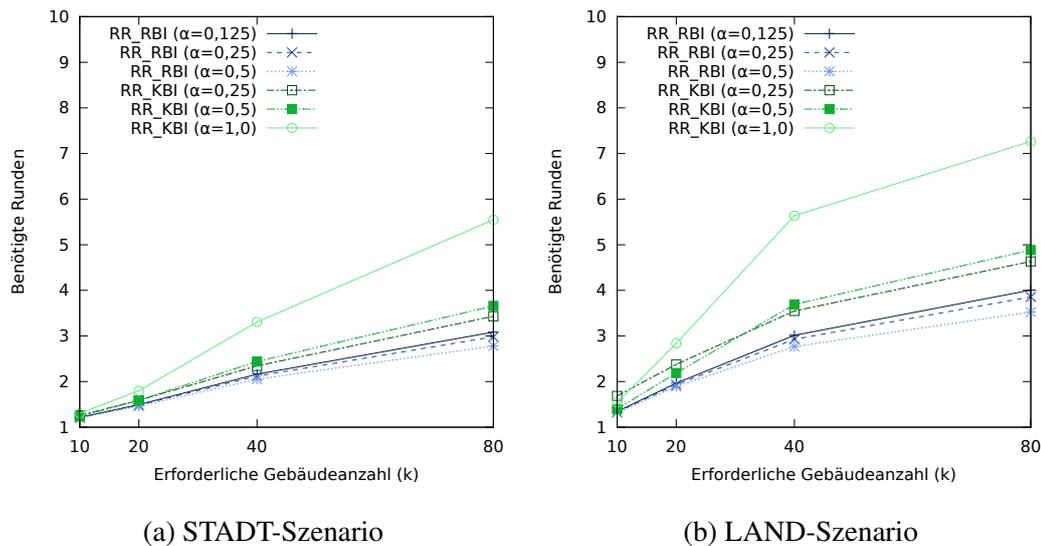
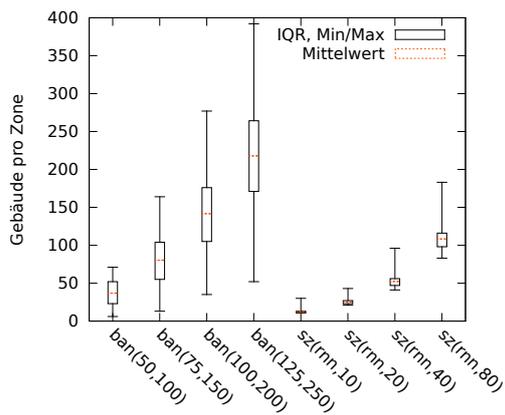
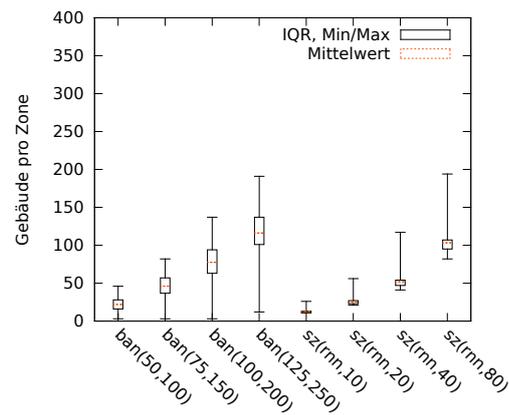


Abbildung 4.11: Benötigte Rundenanzahl der Random-Rect-Varianten

Abbildung 4.12 zeigt die Gebäudeanzahl für Ban Zones und Silent Zones verschiedener Größe. Abbildung 4.12(a) zeigt die Ergebnisse für das STADT-Szenario. Die vier Balken auf der linken Seite zeigen die Gebäudeanzahl für Ban Zones mit  $R = \{100, 150, 200, 250\}$ , die vier Balken auf der rechten Seite die für RNN mit  $k = \{10, 20, 40, 80\}$ . Bei den Silent Zones kann man erkennen, dass der Durchschnitt immer leicht über dem geforderten  $k$  liegt. Für  $k = 40$  sind in den Zonen im Durchschnitt 52,21 Gebäude enthalten, für  $k = 80$  liegt der Durchschnitt bei 108,41 Gebäuden. Viel entscheidender als die durchschnittliche ist die minimale Gebäudeanzahl aller Durchgänge, also das Ergebnis im sogenannten *Worst Case*. Soll die Privatsphäre der Nutzer geschützt werden, muss der Schutz in allen Fällen und unter allen Bedingungen garantiert werden. Hier liegt der entscheidende Vorteil der Silent Zones: Definitionsgemäß enthalten diese immer mindestens  $k$  Gebäude. Dies ist auch in den Ergebnissen zu sehen, bei denen die minimal erreichte Gebäudeanzahl immer knapp über den spezifizierten  $k$  Gebäuden liegt. Hierdurch können Silent Zones ein Mindestmaß an Privatsphäre in Form von enthaltenen Gebäuden garantieren. Dies ist bei Ban Zones nicht der Fall. Die durchschnittliche Gebäudeanzahl steigt zwar deutlich mit zunehmender Zonengröße, jedoch ist die Streuung der Ergebnisse sehr hoch. Für Ban Zones mit  $R = 250$  beträgt beispielsweise der Interquartilsabstand  $IQR = 93$  Gebäude, die Spannweite liegt in diesem Fall bei  $R_{MinMax} = 340$  Gebäuden. Die minimal erreichte Gebäudeanzahl liegt in allen untersuchten Fällen deutlich unter dem Durchschnitt. Wichtig hierbei ist vor allem, dass Ban Zones keine minimale Gebäudeanzahl garantieren können. Für  $R = 100$  gab es im STADT-Szenario Fälle, in denen nur 6 Gebäude innerhalb der Zonengrenzen enthalten waren. Deutlicher wird dies bei der Betrachtung der Ergebnisse des LAND-Szenarios in Abbildung 4.12(b): Ist bei den Silent Zones ein sehr ähnliches Verhalten wie beim STADT-Szenario zu erkennen, sind bei den Ban Zones deutliche

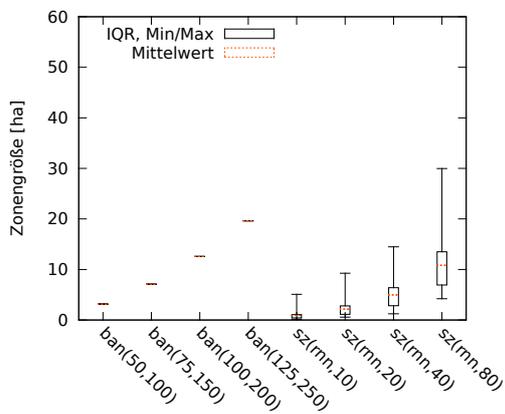


(a) STADT-Szenario

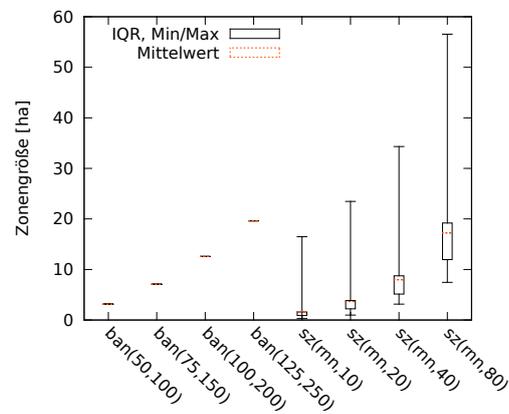


(b) LAND-Szenario

Abbildung 4.12: Gebäudeanzahl pro Zone für Ban Zones und Silent Zones



(a) STADT-Szenario



(b) LAND-Szenario

Abbildung 4.13: Zonengröße der Ban Zones und Silent Zones

Unterschiede erkennbar. Neben der geringeren durchschnittlichen Gebäudeanzahl fällt vor allem die Annäherung der minimalen Anzahl an die x-Achse auf. In diesem Szenario gibt es nun auch für größere Werte für  $R$  Zonen, die fast keine Gebäude enthalten. Bei  $\text{ban}(100,200)$  gibt es beispielsweise eine Zone, die nur drei Gebäude enthält. Dies zeigt, dass in ungünstigen Fällen auch Zonen mit großen Radien keine ausreichende Anzahl an Gebäuden enthalten und Ban Zones keine Mindestanzahl an Gebäuden garantieren können. Folglich sind diese für einen automatisierten Einsatz nicht geeignet.

Betrachtet man die Größe der erstellten Zonen, dargestellt in Abbildung 4.13, ist bezüglich der Streuung der Werte ein gegenteiliges Verhalten zu erkennen. Ban Zones stellen einfache Kreisflächen dar, wodurch die Zonengröße  $A_{\text{ban}}$  auch nur vom Kreisradius  $R$  abhängt, d. h.  $A_{\text{ban}} = R^2 * \pi$ . Die Größe der Silent Zones hingegen variiert stark, vor allem im LAND-Szenario, vgl. Abbildung 4.13(b). Für RNN mit

$k = 80$  liegt die durchschnittliche Zonengröße bei 17,22 ha, die Spannweite liegt allerdings bei  $R_{MinMax} = 49,08$  ha. Es werden also teilweise sehr große Zonen erzeugt, um die geforderten  $k$  Gebäude zu umschließen. Dieses Verhalten ist jedoch ausdrücklich gewünscht, da sich die Zonen der Bevölkerungsdichte anpassen sollen und in jedem Fall die Mindestanzahl an Gebäuden garantieren sollen.

Der direkte Vergleich von Ban Zones und Silent Zones und die Bewertung anhand der enthaltenen Gebäude ist schwierig, da bisherige Ansätze nicht das Ziel hatten, ein Mindestmaß an Gebäude zu enthalten. Trotzdem zeigt dieser Vergleich die Vorteile der Silent Zones auf und belegt, dass diese sich wie gewünscht der Bevölkerungsdichte anpassen. Die Ergebnisse der Ban Zones verdeutlichen, dass diese eine manuelle Anpassung des Radius  $R$  benötigen, um die oben genannten Fälle mit sehr wenigen Gebäuden zu vermeiden.

#### 4.3.4 Diskussion und Erweiterungen

Ein Problem des Silent-Zone-Ansatzes ist die Anfälligkeit der  $k$ -Anonymität gegenüber Angriffen, die auf der Homogenität der Daten oder auf Hintergrundwissen basieren [152]. Bei Ersteren ist das Problem, dass trotz Generalisierung der identifizierenden Attribute Gruppen mit identischen sensiblen Attributen entstehen können. Wenn z. B. in einer Datenbank mit Patienteninformationen alle zu einer  $k$ -anonymen Gruppe zusammengefassten Patienten die gleiche Krankheit haben, ist allein durch die Gruppenzugehörigkeit die Krankheit der entsprechenden Patienten bekannt. Dieses Problem ist bei den Silent Zones in der Form nicht ganz gegeben, da das sensible Attribut in diesem Fall die exakte Adresse bzw. das Gebäude ist und folglich nicht homogen sein kann. Allerdings kann die Semantik der Gebäude einer Zone homogen sein. Besteht eine von einem Nutzer regelmäßig besuchte Silent Zone beispielsweise nur aus Bürogebäuden einer Branche, kann ein Angreifer nicht nur daraus schließen, dass der Nutzer in dieser Zone arbeitet, sondern eventuell sogar welchen Job dieser hat. Angriffe, die auf diesem Problem basieren, werden jedoch in der Regel als ähnlichkeitsbasierende Attacks [141] bezeichnet.

Die zweite Gruppe von möglichen Angriffen bezieht Hintergrundwissen mit ein. Ist dem Angreifer bekannt, welchen Beruf ein Nutzer ausübt, und ist gleichzeitig in der Silent Zone des Nutzers nur ein passendes Gebäude enthalten, so kann der Angreifer entsprechende Informationen, wie Adresse und Arbeitgeber, direkt ableiten. Analog zu Konzepten wie der  $l$ -Diversität [152] und der  $t$ -Closeness [141], die entwickelt wurden, um die genannten Schwachstellen der  $k$ -Anonymität zu beheben, ist eine Erweiterung des Silent-Zone-Konzeptes sinnvoll, bei der die Semantik der Gebäude bei der Zonenerstellung mit beachtet wird. Um die zuvor beschriebenen Angriffe zu vermeiden, muss darauf geachtet werden, dass zum einen eine Mindestanzahl verschiedener Gebäude, aber gleichzeitig auch eine Mindestanzahl von Gebäuden mit gleicher Semantik enthalten ist. Solche Informationen sind bei Kartenanbietern wie OpenStreetMap teilweise bereits verfügbar. Spezielle Gebäude wie Restaurants oder Kirchen sind dort oftmals schon als `restaurant` bzw. `place_of_worship`

(dt. „Andachtsort“/„Kultstätte“) markiert. Die Qualität dieser Zusatzinformationen ist allerdings noch relativ gering, so dass die Daten aktuell entweder manuell oder mit Hilfe anderer Dienste angereichert werden müssen, bevor sie für diesen Zweck genutzt werden könnten.

Eine interessante Erweiterung bezüglich der Anwendung der Silent Zones stellt der Bereich der Online-Routenplanung dar. Im Gegensatz zur lokalen Routenbestimmung mit einem Navigationsgerät, bei dem die Route mit den auf dem Gerät vorhandenen Daten bestimmt wird, wird bei der Online-Routenplanung ein Online-Dienst genutzt. Dieser Routenplaner-Dienst benötigt als Eingabe für die Routenbestimmung sowohl die Start- als auch die Zieladresse. Bei häufiger Nutzung des Dienstes entsteht hier das in Abschnitt 4.3.1 beschriebene Problem: Der Dienst erfährt die wichtigen Orte des Nutzers – in diesem Fall sogar die wichtigsten Strecken. Im Gegenzug hat die Nutzung der Online-Routenplanung im Vergleich zur lokalen Routenbestimmung auch einige Vorzüge. Neben der Aktualität des Kartenmaterials stellen vor allem Live-Verkehrsdaten einen zentralen Vorteil dar, vgl. [258]. Über solche Daten können Informationen über gesperrte Straßen, Bauarbeiten, Staus oder Verzögerungen bei der Routenplanung mit berücksichtigt werden und schließlich zu einer besseren Route für den Nutzer führen. Eine Möglichkeit, die Vorteile der Online-Routenplanung zu nutzen, ohne dabei die Privatsphäre zu opfern, ist die Anwendung der Silent Zones auf Start- und Zieladresse [284]. Statt der wirklichen Start- und Zieladresse wird eine bestimmte Teilmenge von Punkten der jeweiligen Zone, z. B. Eintritts- bzw. Austrittspunkte der Zonen, an den Online-Dienst übermittelt, welcher dann zwischen diesen Punkten die Routen bestimmt und an den Nutzer überträgt. Durch die Ergänzung dieser online bestimmten Routen mit lokal bestimmten Routen innerhalb der Silent Zones, erhalten die Nutzer eine komplette Route für die gewünschte Strecke, ohne dass die genaue Start- und Zieladresse dem Online-Dienst bekannt wird.

## 4.4 Ansatz für eine Privatsphäre schützende Kalibrierung

Eine Kalibrierung der Sensorknoten ist ein entscheidender Aspekt für die Qualität der Messdaten in Sensornetzen. Dies trifft insbesondere auch auf PSNs zu, bei denen die mobilen Endgeräte der Nutzer als Sensorknoten verwendet werden. In Bezug auf die Qualität treten zwei Hauptprobleme auf: Zum einen ist die Hardwarelandschaft partizipativer Sensornetze durch die große Anzahl verschiedener Hersteller und Modelle sehr heterogen. Zum anderen sind die verbauten Sensoren in der Regel vergleichsweise günstige Hardwarekomponenten, da die mobilen Endgeräte nicht primär zur Verwendung als Sensorknoten konzipiert werden. Ohne eine Kalibrierung würde dies zu ungenauen und nur schwer vergleichbaren Messwerten führen.

Da eine manuelle Kalibrierung in PSNs zu aufwendig bzw. aufgrund der großen Anzahl und der örtlichen Verteilung der Geräte nicht praktikabel ist, werden die Sensorknoten typischerweise während des Betriebs mit Hilfe von Referenzstationen kalibriert. Dieses Vorgehen wird auch als *On-the-Fly*-Kalibrierung [88] bezeichnet. Die bei einer Kalibrierung ermittelten Kalibrierungsparameter können für zwei Arten von Kalibrierungsvorgängen genutzt werden: Bei der *Vorwärtskalibrierung* werden die Parameter genutzt, um zukünftige Messwerte anzupassen, wohingegen sie bei der *Rückwärtskalibrierung* auf bereits gemessene und eventuell schon übermittelte Messdaten angewendet werden.

Eine *On-the-Fly*-Kalibrierung kann allerdings nur stattfinden, wenn ein Nutzer in das direkte Umfeld einer Referenzstation kommt. Da die Mobilität der Nutzer nicht gesteuert werden kann, können folglich lange Intervalle ohne Kalibrierungen auftreten. Um dennoch möglichst genaue Messdaten zu erhalten, ist eine rückwirkende Anpassung von Messdaten, wie sie bei der Rückwärtskalibrierung geschieht, in PSNs sinnvoll.

Hierbei entsteht jedoch ein Problem für die Privatsphäre der Nutzer. Sind die Messdaten bereits an den Anwendungsserver übermittelt, muss dieser im Zuge der Rückwärtskalibrierung über die zu aktualisierenden Messdaten informiert werden, d. h. die Kalibrierungsparameter müssen dem Server kommuniziert werden. Da diese nahezu einzigartig sind, können im schlimmsten Fall die Nutzerpfade durch den Anwendungsserver komplett nachverfolgt werden. Aus diesem Grund wird in diesem Abschnitt mit sogenanntem *Privacy-Preserving Calibration for Participatory Sensing (PRICAPS)* ein neues, umfassendes Kalibrierungssystem vorgestellt. PRICAPS ermöglicht Vorwärts- und Rückwärtskalibrierung und schützt dabei die Privatsphäre der Nutzer.

Der folgende Abschnitt basiert auf dem in [298] und [299] vorgestellten Konzept und ist wie folgt aufgebaut: Zunächst wird in Abschnitt 4.4.1 das Problem beschrieben und die Notwendigkeit für ein entsprechendes Kalibrierungssystem motiviert. Darauf folgt in Abschnitt 4.4.2 eine Einführung in grundlegende Kalibrierungskonzepte und das im Folgenden verwendete Kalibrierungsmodell. In Abschnitt 4.4.3 wird das entwickelte Kalibrierungskonzept vorgestellt und im Anschluss ausführlich evaluiert, siehe Abschnitt 4.4.4. Abschließend werden die Ergebnisse in Abschnitt 4.4.5 diskutiert.

#### 4.4.1 Problemstellung und Anforderungen

Im Folgenden soll zunächst die Notwendigkeit der verschiedenen Kalibrierungsvorgänge motiviert werden. Anschließend wird die Problemstellung detailliert beschrieben.

PSNs sind ausgelegt für großflächige Datenerfassungen in dicht besiedelten Gebieten. Durch die Erstellung von solchen Kampagnen können sowohl Nutzer mit privaten als auch mit gewerblichen Interessen Messdaten erfassen, analysieren und

gewonnene Erkenntnisse der Gemeinschaft wieder zur Verfügung stellen [34]. Die Etablierung und Instandhaltung von PSNs sind jedoch aufwendig und kostenintensiv. Damit sich dieser Aufwand rentiert, muss bei der Konzeption darauf geachtet werden, dass PSNs eine möglichst große Vielfalt von Anwendungen und Diensten ermöglichen.

Ein zentraler Aspekt hierbei sind die Messdaten. Damit diese vielseitig genutzt werden können, sollen sie zum einen sehr genau, zum anderen aber auch zeitlich unmittelbar verfügbar sein. So benötigen beispielsweise Statistik-Anwendungen für ihre Auswertungen möglichst genaue Daten, wobei ein gewisser zeitlicher Versatz der Daten in vielen Fällen akzeptabel ist. Live-Anwendungen hingegen benötigen aktuelle Daten, stellen aber im Gegenzug geringere Anforderungen an die Aktualität der Daten. Ein Beispiel dafür ist der zuvor beschriebene Eco-Routing-Dienst [286], bei dem die ökologisch sinnvollste Route berechnet wird. Da die Schadstoffbelastung zeitlich durch die sich verändernde Verkehrssituation stark variiert, wird für eine solche Routenbestimmung das Wissen über die aktuelle Situation, d. h. ein möglichst aktueller Datensatz, benötigt.

Die Bereitstellung genauer und gleichzeitig aktueller Daten ist in PSNs per se nicht möglich, da die erfassten Rohdaten auf Grund der günstigen und heterogenen Sensoren zunächst unpräzise sind. Um genauere Messwerte zu erhalten, müssen die Endgeräte kalibriert werden. Dies wird in PSNs typischerweise mittels On-the-Fly-Kalibrierung gemacht, bei der die erfassten Messdaten mit denen von Referenzstationen verglichen und daraufhin angepasst werden. Eine solche Kalibrierung kann nur stattfinden, wenn ein Nutzer in das direkte Umfeld dieser Referenzstationen kommt. Dadurch können sehr lange Intervalle ohne Kalibrierungen auftreten.

Um dennoch beide Arten von Anwendungen realisieren zu können, sollte ein Kalibrierungssystem sowohl Vorwärts- als auch Rückwärtskalibrierung unterstützen. Ist dies der Fall, können Messwerte, wenngleich von noch nicht kalibrierten Geräten erfasst, unmittelbar an den Anwendungsserver übertragen werden. Im Falle einer später stattfindenden Kalibrierung können sie noch rückwirkend angepasst werden. Durch dieses Vorgehen wird erreicht, dass Daten zeitnah nutzbar und gleichzeitig langfristig so genau wie möglich sind.

Obschon die Rückwärtskalibrierung für PSNs sinnvoll ist, gefährdet dieser Prozess die Privatsphäre der Nutzer. Soll ein bereits an den Anwendungsserver übermittelter Messwert angepasst werden, muss der Nutzer den Server über den Messwert und die gewünschte Anpassung informieren. Will ein Nutzer  $u$  eine Rückwärtskalibrierung durchführen, muss er folglich die Kalibrierungsparameter  $c_u$  und eine Menge von Identifikatoren  $(m_i, \dots, m_{i+j})$  zur Bestimmung der anzupassenden Messwerte an den Server übermitteln.

Das Problem entsteht durch die nahezu einzigartigen Kalibrierungsparameter. Diese unterscheiden sich in der Regel von Gerät zu Gerät und stellen somit einen sogenannten *Quasi-Identifer* dar. Selbst bei einer Aufteilung dieser Information in  $j + 1$  separat übertragene Tupel  $\langle c_u, m_i \rangle, \dots, \langle c_u, m_{i+j} \rangle$  kann der Anwendungs-

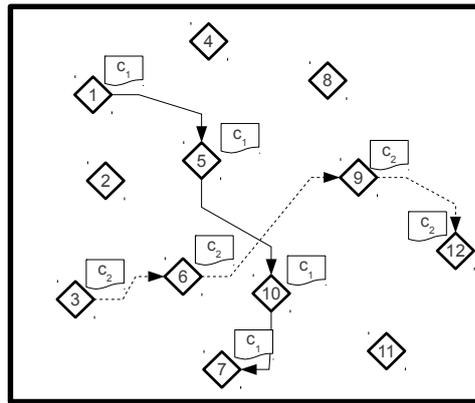


Abbildung 4.14: Die Verwendung exakter Kalibrierungsparameter (hier:  $c_1$  und  $c_2$ ) kann eine Verknüpfung von Messungen und somit eine Nachverfolgung der Nutzerpfade ermöglichen.

server auf Grund von  $c_u$  die mit  $m_i$  bis  $m_{i+j}$  identifizierten Messwerte einem Nutzer zuordnen und dadurch dessen zurückgelegten Pfad verfolgen.

Die Problematik ist in Abbildung 4.14 anhand eines schematischen Beispiels illustriert. In diesem führen zwei Nutzer  $u_1$  und  $u_2$  eine Rückwärtskalibrierung durch. Die für die Anpassung verwendeten Kalibrierungsparameter sind  $c_1$  bzw.  $c_2$ . Nutzer  $u_1$  veranlasst eine Anpassung der Messwerte 1, 5, 10 und 7. Auch wenn diese vier Anpassungen separat und anonym übertragen werden, kann der Server aufgrund der Gleichheit der Kalibrierungsparameter für diese Messwerte schließen, dass diese von einem Nutzer stammen. Hierdurch kann der Server die Pfade aller Nutzer rekonstruieren, die eine Rückwärtskalibrierung durchführen.

Aus diesem Grund wird ein Kalibrierungssystem für PSNs benötigt, das zwar eine Rückwärtskalibrierung erlaubt, aber dennoch die Privatsphäre der Nutzer schützt und keine Rekonstruktion der zurückgelegten Pfade ermöglicht.

#### 4.4.2 Kalibrierung in partizipativen Sensornetzen

Im folgenden Abschnitt sollen grundlegende Aspekte der Kalibrierung in PSNs betrachtet werden. Dafür werden im ersten Schritt die auftretenden Messfehler beschrieben, bevor anschließend das für die kommenden Abschnitte verwendete Kalibrierungsmodell dargestellt wird. Nach einer kurzen Auflistung verschiedener Kalibrierungsklassen, bildet ein Überblick über existierende Kalibrierungsansätze in PSNs den Abschluss dieses Abschnittes.

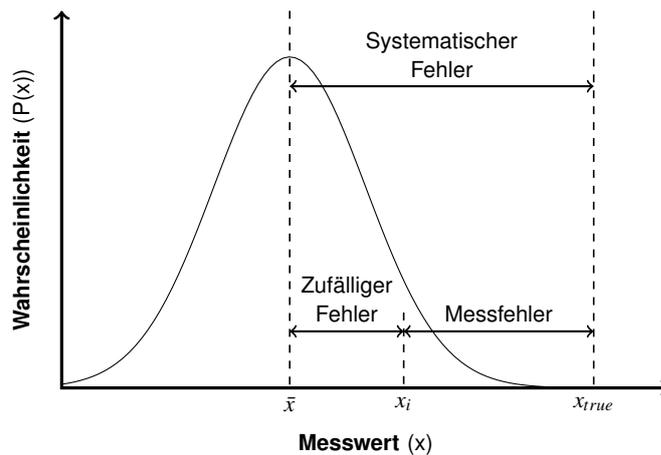


Abbildung 4.15: Systematische und zufällige Fehler bei Messungen, angelehnt an Bychkovskiy et al. [35]

#### 4.4.2.1 Messfehler

Das Ziel einer Messung ist es, den tatsächlichen Wert einer Messgröße zu ermitteln. Der gemessene Wert wird in der Regel, aufgrund verschiedenster Stör- und Fehlerquellen, eine gewisse Abweichung vom tatsächlichen Wert haben. Diese Differenz wird als *Messfehler* bezeichnet. Zwar wird in der neueren Literatur auch der Begriff der *Messabweichung* [61] verwendet, allerdings ist diese Bezeichnung in der Praxis nicht so gebräuchlich, weshalb im Folgenden der Begriff des Messfehlers bevorzugt wird.

Wie in Abbildung 4.15 dargestellt, können Messfehler in zwei Kategorien unterteilt werden:

- **Systematische Fehler:** Gränicher definiert systematische Fehler als „Fehleranteile, welche bei Wiederholungen einer Messung unter identischen Messbedingungen einen konstanten Wert besitzen, d. h. jedesmal in gleicher Größe und mit gleichem Sinn (Vorzeichen) auftreten“ [86]. Daraus folgt, dass systematische Messfehler durch Wiederholung der Einzelmessung nicht erkannt oder eliminiert werden können.
- **Zufällige Fehler:** Zufällige Fehler sind „Abweichungen (...), welche nach Betrag und Vorzeichen zufallsbedingt schwanken“ [86] und sich bei wiederholter Messung im Mittel ausgleichen. Zufällige Fehler lassen sich mit statistischen Methoden behandeln.

Der Zusammenhang zwischen systematischen und zufälligen Fehlern ist in Abbildung 4.15 dargestellt. Der tatsächliche Wert ist in der Abbildung mit  $x_{true}$  angegeben, der systematische Fehler beträgt  $\bar{x} - x_{true}$ . Die gemessenen Werte  $x_i$  schwanken zufällig um den Wert  $\bar{x}$ , wobei die Wahrscheinlichkeit mit zunehmender Distanz von  $\bar{x}$  abnimmt, hier angedeutet durch eine Gauß-Verteilung. Im Mittel gleichen sich die

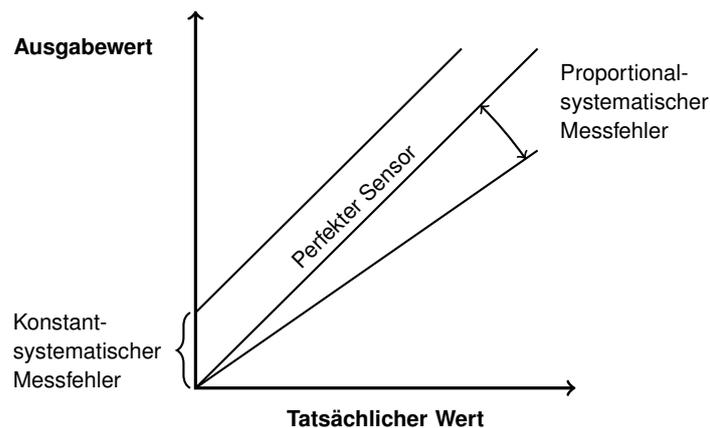


Abbildung 4.16: Systematische Messfehler, angelehnt an Honicky [99]

zufälligen Fehler aus, so dass  $\frac{1}{n} \sum_{i=1}^n x_i = \bar{x}$  gilt. Der zu beobachtende Messfehler  $x_i - x_{true}$  setzt sich aus beiden Fehlerarten zusammen und schwankt bei wiederholten Messungen auf Grund des zufälligen Fehlers.

Bei systematischen Fehlern kann man zudem noch zwischen konstant- und proportional-systematischen Messfehlern unterscheiden. Abbildung 4.16 illustriert den Unterschied. Als Referenz ist die Ausgabe eines perfekten Sensors dargestellt. Ein Sensor mit konstant-systematischen Messfehler über- oder unterschätzt den tatsächlichen Messwert immer um den gleichen Betrag, d. h. die Abweichung ist unabhängig vom Betrag des Messwertes. Bei einem proportional-systematischen Messfehler hingegen hängt die Abweichung vom Betrag des Messwertes ab, d. h. die Abweichung nimmt mit ansteigendem Messwert zu bzw. ab.

Ziel einer Kalibrierung ist es, den systematischen Fehler der Sensorknoten zu eliminieren. Systematische Fehler können durch verschiedene Quellen verursacht werden, beispielsweise durch Umwelteinflüsse, durch Unzulänglichkeiten bei der Handhabung der Messgeräte und die Unvollkommenheit der Messgeräte, vgl. [86]. Vor allem der letztgenannte Punkt ist in PSNs hervorzuheben, da die verbaute Sensorik nicht für hochpräzise Messungen ausgelegt ist.

#### 4.4.2.2 Kalibrierungsmodell

PSNs haben das Ziel, eine oder mehrere Messgrößen zu erfassen, wie z. B. die Temperatur. Um eine Messgröße  $P$  erfassen zu können, muss es ein durch  $P$  erzeugtes messbares Signal  $p : T \rightarrow D$  geben, wobei  $T \subseteq \mathbb{R}^+$  die Zeit und  $D \subseteq \mathbb{R}$  der Wertebereich der Messgröße sei. Ferner sei  $m_s(t_i)$  die Messung eines Sensors  $s$  zum Zeitpunkt  $t_i \in T$  und  $p(t_i)$  der tatsächliche Wert der Messgröße zu diesem Zeitpunkt.

Wenn Sensor  $s$  ein *perfekter Sensor* ist, vgl. Abbildung 4.16, dann gilt  $m_s(t_i) = p(t_i)$  für jeden Zeitpunkt und eine Kalibrierung ist nicht nötig. Wie im vorherigen Ab-

schnitt bereits erwähnt, sind perfekte Messungen in der Regel nicht möglich. Gerade bei kostengünstigen Gas-Sensoren, wie für das folgende Konzept angenommen, können erhebliche Genauigkeitsverluste durch das Altern der Sensoren [102] und beeinflussende kontextabhängige Faktoren [114], wie z. B. Luftfeuchtigkeit, auftreten.

Die Kalibrierung der Sensoren kann folglich als Prozess beschrieben werden, bei dem die Abweichung der gemessenen Werte  $m_s(t_i)$  von den tatsächlichen Werten  $p(t_i)$  minimiert wird. Dies wird durch das Anwenden einer *Kalibrierungskurve*  $\phi$  auf die gemessenen Werte erreicht. Als Repräsentation dieser Kurve wird im Folgenden eine Polynomfunktion des Grades  $k$  verwendet, die neben dem anzupassenden Messwert einen Vektor mit Kalibrierungsparametern  $c = (c_0, c_1, \dots, c_k) \in \mathbb{R}^{k+1}$  als Eingabeparameter erhält. Die Kalibrierungskurve ist somit eine Funktion  $\phi : \mathbb{R}^{k+1} \times D \rightarrow D$ , die einen Eingabewert  $x$  mit den Kalibrierungsparametern  $c$  anpasst:

$$\phi(c, x) = \sum_{n=0}^k c_n * x^n. \quad (4.1)$$

Da eine mehrfache Kalibrierung der Sensoren möglich ist, können sich auch die Kalibrierungsparameter über die Zeit ändern. Aus diesem Grund wird eine Funktion  $\omega : T \rightarrow \mathbb{R}^{k+1}$  eingeführt, welche die zu einem bestimmten Zeitpunkt effektiven Kalibrierungsparameter zurückliefert. Demnach ergeben sich die kalibrierten Messwerte  $\tilde{m}_s(t_i)$  eines Sensors  $s$  zum Zeitpunkt  $t_i$  wie folgt:

$$\tilde{m}_s(t_i) = \phi(\omega(t_i), m_s(t_i)) = \sum_{n=0}^k \omega(t_i)_n * m_s(t_i)^n. \quad (4.2)$$

Ein perfekter Sensor benötigt keine Kalibrierung und es gilt  $\forall t_i \in T : \omega(t_i) = (0, 1, 0, 0, \dots, 0) \in \mathbb{R}^{k+1}$  sowie  $\tilde{m}_s(t_i) = m_s(t_i) = p(t_i)$ . In der Praxis sind perfekte Sensoren allerdings kaum realisierbar. Daher wird durch die Kalibrierung versucht, einen *perfekt kalibrierten Sensor* zu erhalten. Ein perfekt kalibrierter Sensor verhält sich ab einem Zeitpunkt  $t_c$  wie ein perfekter Sensor, so dass  $\forall t_i \geq t_c : \omega(t_{i+1}) = \omega(t_i)$  und  $\tilde{m}_s(t_i) = p(t_i)$  gilt. Dieser Idealzustand wird aber in der Regel nicht dauerhaft erreicht, da die Genauigkeit der Sensoren über die Zeit hin abnimmt und die Messwerte folglich wieder von den tatsächlichen Werten abweichen. Allerdings kann durch regelmäßiges Wiederholen des Kalibrierungsprozesses eine Annäherung an den Idealzustand des perfekt kalibrierten Sensors erreicht werden.

Um die oben eingeführte Kalibrierungskurve  $\phi$  zu bestimmen, wird eine Menge  $C$  von Kalibrierungstupeln  $\langle m_s(t_i), p(t_i) \rangle$  benötigt (mit  $|C| \geq (k+1)$ ). Das bedeutet, dass für eine gewisse Menge an Messungen neben dem gemessenen Wert auch der tatsächliche Wert der Messgröße bekannt sein muss. Für diesen Zweck werden Referenzstationen verwendet. Die Annahme bei diesen ist, dass die Sensoren der Referenzstation zu jeder Zeit perfekt kalibriert sind und diese somit  $p(t_i)$

bestimmen können. Sensorknoten in der Nachbarschaft einer solchen Station können die Referenzmessungen dann abfragen und zur Kalibrierung verwenden. Dazu werden sowohl bei der Messung des mobilen Sensorknotens als auch bei der Referenzmessung der Zeitpunkt  $t$  und der Ort an diesem Zeitpunkt  $l(t)$  festgehalten. Ein Sensorknoten hat folglich eine Menge an Messungen  $M$ , die aus Tupeln der Form  $\langle t, l(t), m_s(t) \rangle$  besteht, sowie eine Menge an Referenzmessungen  $S$ , die Tupel der Form  $\langle t, l(t), p(t) \rangle$  enthält. Unter Verwendung der Punktnotation, d. h.  $m.l$  bezeichnet den Ort einer Messung  $m \in M$ , können die Kalibrierungstupel  $C$  wie folgt beschrieben werden:

$$s \in S, m \in M : C = \{(s.p, m.m_s) \mid |s.t - m.t| \leq \delta_t \wedge |s.l - m.l| \leq \delta_l\}, \quad (4.3)$$

wobei  $\delta_t$  und  $\delta_l$  die maximale zeitliche bzw. örtliche Distanz zwischen den mobilen Messungen und den Referenzmessungen angibt, bei der die Referenzdaten noch zur Kalibrierung verwendet werden dürfen. Diese Werte müssen für jede Messgröße angepasst werden.

#### 4.4.2.3 Existierende Kalibrierungsansätze

Die Kalibrierung von Sensorknoten ist auch bei traditionellen Sensornetzen ein wichtiges Thema. In vielen Fällen können die Sensorknoten vor dem Einsatz manuell oder automatisiert kalibriert werden, was jedoch bei einer sehr großen Anzahl an Knoten nicht mehr praktikabel ist. Daher wurden verschiedene Konzepte entwickelt, die eine Kalibrierung während des Betriebs ermöglichen. Bychkovskiy et al. [35] stellen einen Ansatz für On-the-Fly-Kalibrierung für Sensornetze vor, bei denen die Messgröße zeitlich relativ stark variiert, wohingegen örtlich vergleichsweise wenig Veränderungen zu beobachten sind. In solchen Szenarien zeichnen benachbarte Knoten sehr ähnliche Messverläufe auf, die sich durch die schnelle Veränderung über die Zeit leicht vergleichen und aufeinander abstimmen lassen. Dabei wird in zwei Schritten vorgegangen: Zunächst werden benachbarte Knoten durch den Austausch von Messwerten gegenseitig kalibriert. Anschließend wird in Gruppen von Knoten die Konsistenz der paarweisen Kalibrierung maximiert. Da die Annahmen sehr restriktiv sind und der Ansatz für stationäre Sensorknoten entwickelt wurde, ist eine Übertragung des Konzeptes auf PSNs nur eingeschränkt möglich.

Whitehouse und Culler [238] stellen ein Framework vor, das für Szenarien entwickelt wurde, bei denen Sensoren eine von Aktoren erzeugte Messgröße aufzeichnen und sowohl die Sensoren als auch die Aktoren kalibriert werden müssen. Ein Beispiel hierfür ist die Distanzbestimmung zwischen zwei Knoten durch die Laufzeitmessung von ausgesendeten Signalen. In dem Lokalisierungssystem *Calamari* [237] senden Knoten gleichzeitig ein Akustik- und ein Funksignal aus. Knoten, die diese Signale empfangen, können durch die Laufzeitunterschiede die Entfer-

nung zu dem sendenden Knoten bestimmen. In solchen Systemen treten Fehler sowohl sender- als auch empfängerseitig auf. Folglich müssen beide Seiten bei einer Kalibrierung mit in Betracht gezogen werden. Whitehouse und Culler [238] stellen daher ein Modell auf, das die Fehlerquellen beider Seiten berücksichtigt, und führen eine Parameterschätzung mit Hilfe von aufgezeichneten Messungen durch, bei der das Gesamtsystem optimiert wird. Durch dieses Vorgehen werden alle Sensoren und Aktoren gleichzeitig kalibriert, ohne dass Referenzknoten vorhanden sind. Für einen Einsatz in PSNs ist es dennoch eher wenig geeignet, da es in der Regel keine zu kalibrierenden Aktoren gibt.

Miluzzo et al. haben in [161] ihr Kalibrierungssystem *CaliBree* vorgestellt. Dieses ist ein auf Selbstkalibrierung basierendes System für Sensornetze mit mobilen Knoten. Die Selbstkalibrierung geschieht durch eine On-the-Fly-Kalibrierung, d. h. mit der Hilfe von Referenzstationen. Diese versenden in ihrer lokalen Umgebung Beacons, um vorbeikommende Knoten über ihre Präsenz und die Möglichkeit einer Kalibrierung zu informieren. Nutzer, die in die Nähe dieser Referenzstationen kommen und deren Beacons erhalten, können auf diese antworten, falls sie den Kalibrierungsprozess mit der entsprechenden Referenzstation initiieren möchten. Erhält eine Referenzstation eine solche Antwort, beginnt diese mit der Versendung einer Folge von sogenannten *Calibration Beacons*, die aktuelle Referenzdaten enthalten. Mit den erhaltenen Daten wird ein *Consensus*-Algorithmus gestartet und die Abweichung der Messwerte des mobilen Knotens von den Referenzdaten bestimmt. Mit der Annahme, dass die minimale Abweichung auftritt, wenn der Knoten und die Referenzstation sich am nächsten sind und somit die gleiche Umgebung erfassen, wird die minimale Abweichung als Korrektur für den mobilen Sensorknoten verwendet. Ein Vorteil des Konzeptes ist, dass die mobilen Knoten ihre Position nicht kennen müssen, d. h. sie müssen nicht mit einer Positionierungstechnologie wie GPS ausgestattet sein. Allerdings muss dafür eine direkte Kommunikation mit den Referenzstationen möglich sein, wodurch bereits existierende stationäre Messstationen nicht ohne Umrüstung als Referenzstationen verwendbar sind.

Im Gegensatz zu dem dezentralen Calibree-Konzept schlägt Honicky [99] einen zentralen Ansatz vor. Mobile Sensorknoten übertragen ihre Messwerte zusammen mit der jeweiligen Position an eine zentrale Instanz. Diese kann dann mit den globalen Informationen den Kalibrierungsprozess starten und die Abweichungen mit Hilfe von Gauß-Prozessen bestimmen. Dabei wird die Annahme verwendet, dass die Messwerte von Knoten in örtlicher Nachbarschaft korrelieren. Da ein Abgleich auch mit anderen unkalibrierten Knoten möglich ist, ist dieser Ansatz nicht auf das Vorhandensein von Referenzstationen oder -knoten angewiesen.

Hasenfratz et al. [89] stellen neben der typischerweise angewandten Vorwärtskalibrierung zwei neue Kalibrierungsalgorithmen vor: Rückwärts- und Sofortkalibrierung. Bei der bereits beschriebenen Rückwärtskalibrierung werden neu bestimmte Kalibrierungsparameter genutzt, um auch zuvor gemessene Werte anzupassen. Da neue Kalibrierungsparameter nicht immer zeitnah verfügbar sind, kann es zu einem Zeitverzug zwischen Messwerterfassung und Korrektur kommen. Bei der Sofortka-

librierung wird dieses Problem durch die Verwendung eines Kalibrierungsspeichers umgangen. Dieser bestimmt mit jedem neu erfassten Kalibrierungstupel auch neue Kalibrierungsparameter. Dadurch kann eine annähernd gute Genauigkeit wie bei der Rückwärtskalibrierung erreicht werden, ohne jedoch den zeitlichen Verzug in Kauf zu nehmen.

In ihrer Arbeit stellen Hasenfratz et al. [89] ferner ein Multihop-Konzept vor. Die Idee des Konzeptes ist es, bereits kalibrierte Knoten als mobile Referenzknoten zu nutzen. Dadurch werden Kalibrierungen nicht nur auf Phasen begrenzt, in denen sich ein Knoten in der Nähe einer Referenzstation befindet, sondern erlauben diese bei jedem Aufeinandertreffen mit anderen Knoten, die bereits eine Kalibrierung mit einer solchen Station durchgeführt haben. Hierzu werden bei der Begegnung zweier Knoten sowohl die aktuellen Messwerte als auch der Zeitpunkt der letzten, mit einer Referenzstation durchgeführten Kalibrierung ausgetauscht. Mit diesen Werten wird eine Anpassung der Messwerte durchgeführt, bei der die Messwerte proportional zur Aktualität der Kalibrierung gewichtet werden.

Im folgenden Konzept werden ebenfalls Referenzstationen für die Kalibrierung verwendet, da ein solches Vorgehen zur Zeit den realistischsten Ansatz für eine breite Anwendung in PSNs darstellt.

### 4.4.3 Konzept

In diesem Abschnitt soll das Kalibrierungssystem PRICAPS beschrieben werden. PRICAPS stellt ein Kalibrierungssystem für PSNs dar, welches besonderen Wert auf die Privatsphäre der Nutzer legt. Wie schon einleitend in Abschnitt 4.4 erwähnt, ermöglicht PRICAPS die Vorwärts- und Rückwärtskalibrierung von Messwerten.

PRICAPS ist ein On-the-Fly-Kalibrierungssystem, d. h. die Kalibrierung basiert auf dem Vergleich von Messwerten mobiler Sensorknoten mit denen von benachbarten Referenzstationen. PSNs, in denen die Luftqualität gemessen werden soll, können beispielsweise stationäre Umwelt- bzw. Luftmessstationen als Referenz für die tatsächlichen Werte der zu untersuchenden Messgröße verwenden. Solche Stationen sind bereits in vielen Städten installiert. In München gibt es vier Standorte mit fest installierten Luftmessstationen [268], in Zürich sind es fünf Standorte, an denen kontinuierlich die Luftqualität aufgezeichnet wird [273]. Für PRICAPS ist es notwendig, dass die mobilen Knoten die Messwerte der Referenzstationen abfragen können, z. B. durch Web-Service-Schnittstellen.

In Abbildung 4.17 ist die „Kalibrierungs-Pipeline“ dargestellt. Bevor eine Kalibrierung initiiert wird, müssen zunächst Daten erfasst und zum Anwendungsserver hochgeladen werden. Zusammen mit entsprechenden Referenzmessungen kann anschließend sowohl eine Vorwärts- als auch eine Rückwärtskalibrierung durchgeführt werden. Die Anpassung der Messdaten im Rahmen der Vorwärtskalibrierung findet dabei lokal statt, d. h. die Korrektur geschieht nur auf dem Endgerät und der Anwendungsserver muss nicht benachrichtigt werden. Die Rückwärtskalibrierung

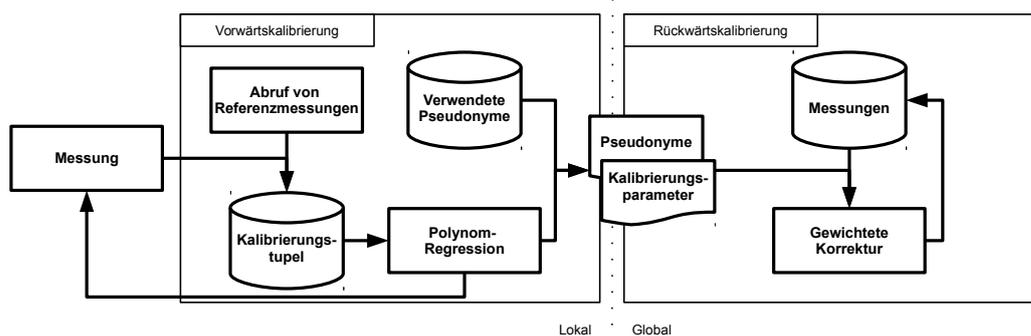


Abbildung 4.17: Die Kalibrierungs-Pipeline zeigt die zwei Kalibrierungsphasen

findet hingegen global statt, da in diesem Fall der Anwendungsserver über die rückwirkende Anpassung der Messwerte informiert werden muss. Im Folgenden werden diese drei Schritte detailliert betrachtet.

#### 4.4.3.1 Datenerfassung

Bevor eine Kalibrierung durchgeführt werden kann, müssen zunächst entsprechende Daten erfasst werden. Dies beinhaltet zwei Aspekte: Zum einen müssen Messungen durchgeführt, zum anderen müssen die erfassten Messwerte auch an den Anwendungsserver übertragen werden.

Die Messungen werden unter Verwendung der mobilen Endgeräte der Nutzer durchgeführt. Die genaue Art und Weise der Messung ist für PRICAPS nicht relevant, d. h. ob die Daten opportunistisch oder partizipativ erhoben werden, kann vernachlässigt werden. Es wird lediglich angenommen, dass nur *geeignete* Messungen beigetragen werden. Geeignete Messungen sind solche, bei denen ein Sensor  $s$  die gleiche Umgebung erfasst, wie jeder andere Sensor sie an diesem Standort erfassen könnte, und bei denen der Messwert  $m_s(t_i)$  nur aufgrund des Messfehlers des Knotens  $s$  von  $p(t_i)$  abweicht. Das schließt insbesondere Messungen aus, die in einem Kontext stattfinden, in dem andere Sensorknoten nicht messen können, wie beispielsweise in der Tasche des Nutzers. Zur Unterscheidung von geeigneten und ungeeigneten Messungen können Ansätze wie der von Miluzzo et al. [164] oder von Yang et al. [245] verwendet werden, die den Kontext des messenden Endgerätes bestimmen.

Der Upload der Messdaten von den mobilen Endgeräten zum Anwendungsserver erfolgt mit selbst generierten Pseudonymen über ein sogenanntes MIX-Netz [41]. Die Pseudonyme können vom Nutzer regelmäßig geändert werden, um eine Nachverfolgung des Bewegungspfades zu verhindern. Dies kann so weit gehen, dass Nutzer für jede Messung ein eigenes Pseudonym verwenden. Die Pseudonyme, im Folgenden mit  $ps_{id}$  bezeichnet, werden benötigt, um die Messungen bei der Rückwärtskalibrierung referenzieren zu können. Um eine Messung räumlich einordnen zu können, wird neben dem Pseudonym und dem Messwert auch der Ort der Messung  $l$  über-

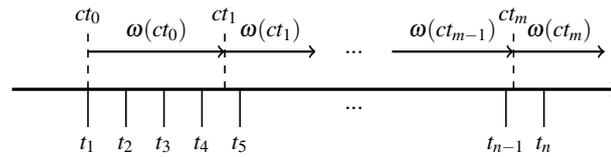


Abbildung 4.18: Die an den Kalibrierungspunkten  $ct_i$  bestimmten Parameter werden bis zur nächsten Kalibrierung verwendet, angelehnt an Hasenfratz et al. [89]

mittelt. Folglich werden Tupel der Form  $\langle ps_{id}, m_s, l \rangle$  an den Anwendungsserver übertragen.

Um zeitbasierte Angriffe auf die Privatsphäre der Nutzer zu vermeiden, wird der Zeitpunkt der Messung nicht übermittelt. Stattdessen wird die Zeit in Intervalle  $t_{int}$  eingeteilt, die der geforderten Messhäufigkeit entsprechen. Ist z. B. eine Messhäufigkeit von  $4 \times$  pro Stunde gewünscht, wird  $t_{int} = 15\text{min}$  gesetzt. Innerhalb von  $t_{int}$  wird zufällig ein Zeitpunkt gewählt, an dem die Daten an den Server übermittelt werden. Dieser zeichnet dann die Ankunftszeit  $t_{arr}$  auf und speichert somit Tupel der Form  $\langle ps_{id}, m_s, l, t_{arr} \rangle$  in seiner Datenbank.

#### 4.4.3.2 Vorwärtskalibrierung

Ziel der Vorwärtskalibrierung ist es, Messfehler von zukünftigen Messwerten zu verringern. Dafür werden die zu einem Zeitpunkt  $ct_i$  bestimmten Kalibrierungsparameter auf alle danach erhobenen Messwerte angewendet, bis eine erneute Kalibrierung zum Zeitpunkt  $ct_{i+1}$  stattfindet. Sei  $t_i$  der Zeitpunkt einer Messung und  $\omega : T \rightarrow \mathbb{R}^{k+1}$  die Funktion, welche die jeweils effektiven Kalibrierungsparameter zurückliefert, vgl. Abschnitt 4.4.2.2, dann gilt  $\omega(t_i) = \omega(ct_j)$  für alle  $t_i$ , für die gilt  $ct_j \leq t_i < ct_{j+1}$ . Das Prinzip ist Abbildung 4.18 schematisch dargestellt.

Damit eine Vorwärtskalibrierung durchgeführt werden kann, ist zunächst die Bestimmung der Kalibrierungskurve  $\phi$  und der Kalibrierungsparameter  $c$  nötig. Der dafür erforderliche Prozess kann in drei Schritte unterteilt werden, vgl. Abbildung 4.17. Der erste Schritt besteht in der Beschaffung von Referenzmessungen. Mit Hilfe dieser werden im zweiten Schritt Kalibrierungstupel bestimmt, mit denen dann im letzten Schritt die Kalibrierungskurve bestimmt wird.

Für die Beschaffung von Referenzmessungen müssen den Endgeräten die Referenzstationen bekannt sein. Hierfür stellt der Server ein Verzeichnis zur Verfügung, in dem alle Referenzstationen eines festgelegten Gebietes aufgelistet sind. Neben dem jeweiligen Standort einer Station, ist dort auch die Daten-Schnittstelle eingetragen, über welche die Endgeräte die Referenzdaten abfragen können. Sobald der Nutzer das Gebiet des zuletzt erhaltenen Verzeichnisses verlässt, fordert dieser beim Server das Verzeichnis des neuen Gebietes an. Bleibt der Nutzer über einen längeren

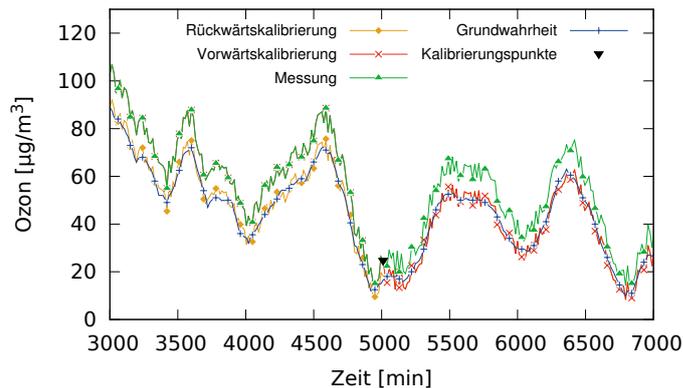


Abbildung 4.19: Beispielhafter Auszug von unkalibrierten und kalibrierten Ozonmessungen

Zeitraum in dem gleichen Gebiet, wird das Verzeichnis durch periodische Anfragen des Nutzers aktuell gehalten.

Sobald die Standorte der Referenzstationen bekannt sind, können die Endgeräte bei ihren Messungen prüfen, ob sie in der Nähe einer solchen Station sind. Falls dies der Fall ist, werden die Referenzdaten über die öffentliche Daten-Schnittstelle angefordert. Die genaue Spezifikation, was „in der Nähe“ bedeutet, d. h. ab wann die Daten der Referenzstation wirklich als Referenzwert der Messung verwendet werden kann, hängt von zu beobachtenden Messgröße ab und kann über  $\delta_i$  in Gleichung 4.3 angepasst werden.

Werden nur Referenzdaten angefordert, wenn Nutzer sich wirklich in der Nähe einer Referenzstation befinden, kann der Betreiber dieser Stationen daraus möglicherweise Rückschlüsse auf die Anzahl der Nutzer ziehen, die eine Kalibrierung durchgeführt haben. Das kann sich besonders in Szenarien als problematisch erweisen, bei denen nur sehr wenige Nutzer in der Nähe einer Station waren. Um dies zu vermeiden, täuscht ein gewisser Anteil an Nutzern  $\Gamma$  in jedem Kalibrierungsintervall vor, in der Nähe einer Station zu sein, indem Schein-Anfragen für Referenzdaten versendet werden. Jeder Nutzer  $u$  zieht hierfür eine zufällige Zahl  $\gamma_u$  aus  $[0.0; 1.0]$  und sendet eine Schein-Anfrage, wenn  $\gamma_u < \Gamma$  gilt. Die Antworten dieser Anfragen werden seitens der Nutzer einfach verworfen, da keine entsprechende lokale Messung vorhanden ist.

Werden wirklich Referenzdaten angefragt, werden diese zusammen mit den vom Endgerät aufgezeichneten Messwerten lokal gespeichert. Anschließend werden die Kalibrierungstupel durch den in Gleichung 4.3 beschriebenen raumzeitlichen Filterprozess gebildet, d. h. erhobene Messwerte werden mit Referenzdaten kombiniert, die ungefähr zur gleichen Zeit und am gleichen Ort erfasst wurden. Diese Kalibrierungstupel werden dann verwendet, um die Kalibrierungskurve  $\phi$  zu bestimmen.

Um verzerrte oder voreilige Kalibrierungsvorgänge zu vermeiden, werden bei PRICAPS vor einer Kalibrierung die folgenden Bedingungen geprüft:

- *Minimale Anzahl an Kalibrierungstupeln:* Um den Einfluss von Ausreißern zu minimieren, wird eine Vorwärtskalibrierung nur initiiert, wenn eine vordefinierte Mindestanzahl an Kalibrierungstupeln  $C_{MinCount}$  vorliegt.
- *Minimal abgedeckter Wertebereich:* Um eine Kalibrierung zu vermeiden, die auf bestimmte Wertebereiche optimiert ist, muss ein bestimmter Wertebereich  $C_{MinRange}$  durch die Messwerte der Kalibrierungstupel abgedeckt sein.
- *Maximale Kalibrierungshäufigkeit:* Um unnötige Kalibrierungsvorgänge zu vermeiden, wird ein neuer Kalibrierungsprozess nur dann initiiert, wenn eine vordefinierte Zeitspanne  $C_{Timeout}$  seit der letzten Kalibrierung überschritten wurde.

Die eigentliche Bestimmung von  $\phi$  erfolgt schließlich mittels Polynomregression. Dabei wird die Modellfunktion mit Hilfe der Methode der kleinsten Quadrate bestimmt, bei der die Summe der quadratischen Abweichung zwischen Referenzwert und mobilem Messwert minimiert wird. Die so bestimmte Kalibrierungskurve mit den Parametern  $c$  kann ab diesem Zeitpunkt verwendet werden, um zukünftige Messungen vor dem Upload zum Server anzupassen.

Abbildung 4.19 zeigt die möglichen Effekte auf die Messwerte. Der Zeitpunkt der Kalibrierung ist durch ein schwarzes Dreieck gekennzeichnet. Da es sich in diesem Fall um die erste Kalibrierung handelt, ist die rote Kurve der „Vorwärtskalibrierung“ vor diesem Zeitpunkt identisch mit denen der erhobenen Messungen. Nach der Kalibrierung zeigt sich jedoch deutlich, dass die Messwerte durch die Vorwärtskalibrierung näher an der Grundwahrheit liegen.

Die in diesem Schritt bestimmten Kalibrierungsparameter werden schließlich auch im Rahmen der Rückwärtskalibrierung genutzt, um zuvor hochgeladene Messdaten rückwirkend zu korrigieren.

#### 4.4.3.3 Rückwärtskalibrierung

Ziel der Rückwärtskalibrierung ist es, bereits gemessene und übermittelte Messdaten mittels aktuellerer Kalibrierungsparameter nachträglich zu korrigieren und dadurch den Messfehler zu verringern. Dafür sollen die im Zuge der Vorwärtskalibrierung bestimmten Kalibrierungsparameter auf die anzupassenden Daten angewandt werden.

Die Kalibrierungsparameter werden lokal, d. h. auf den mobilen Endgeräten, bestimmt, wohingegen die bereits übertragenen Messwerte auf Seiten des Anwendungsserver liegen. Der Server muss folglich über neue Kalibrierungsparameter benachrichtigt werden. Zudem hat der Server auch keine Informationen darüber, welche Werte angepasst werden sollen, da die Messungen anonym mit wechselnden Pseudonymen übermittelt werden. Die Endgeräte müssen daher die Kalibrie-

rungsparameter und die verwendeten Pseudonyme, mit denen die Messungen beim Server identifiziert werden können, an den Server kommunizieren.

Die Übermittlung der Parameter kann jedoch die Privatsphäre der Nutzer gefährden. Diesem Problem wird in PRICAPS mit dem Konzept der  $k$ -Anonymität [223] begegnet. Um den Kalibrierungsvektoren die Einzigartigkeit zu nehmen, wird die Genauigkeit der Parameter in einem Diskretisierungsschritt reduziert. Dies hat zwar einen negativen Einfluss auf die Genauigkeit der Kalibrierung, soll jedoch die Privatsphäre der Nutzer wahren. Ziel dieses Schrittes ist es, die Wahrscheinlichkeit zu erhöhen, dass mehrere Endgeräte die gleichen Parameter übermitteln. Senden  $k$  Nutzer die gleichen Parameter, kann der Server die Pfade dieser  $k$  Nutzer nicht mehr rekonstruieren, sie sind somit  $k$ -anonym. Der Genauigkeitsverlust wird in Kauf genommen, da zu erwarten ist, dass der Messfehler nach der Anpassung mit den diskretisierten Parametern im Mittel geringer ausfällt als ohne die Durchführung der Rückwärtskalibrierung.

Für diesen Schritt wird eine Diskretisierungsfunktion  $\psi: \mathbb{R}^{k+1} \times \mathbb{R}^{k+1} \rightarrow \mathbb{R}^{k+1}$  verwendet, die für einen Kalibrierungsvektor  $c$  einen diskretisierten und damit generalisierten Kalibrierungsvektor  $\tilde{c}$  zurückliefert:

$$\tilde{c} = \psi(c, d) = \begin{pmatrix} \lceil \frac{c_0}{d_0 * \theta(c)} \rceil * (d_0 * \theta(c)) \\ \vdots \\ \lceil \frac{c_k}{d_k * \theta(c)} \rceil * (d_k * \theta(c)) \end{pmatrix}. \quad (4.4)$$

Die in dieser Definition verwendeten Bezeichnungen haben die folgende Bedeutung:

- $d \in \mathbb{R}^{k+1}$  ist der Diskretisierungsvektor, der festlegt, wie stark die einzelnen Komponenten von  $c$  diskretisiert werden. Dieser wird systemweit festgelegt, das bedeutet, dass alle Knoten den gleichen Vektor  $d$  verwenden.
- $\lceil x \rceil$  beschreibt die Rundungsfunktion zur nächstgelegenen Ganzzahl. Mit  $\lfloor x \rfloor = \max\{k \in \mathbb{Z} \mid k \leq x\}$  ist diese wie folgt definiert:  $\lceil x \rceil = \lfloor x + \frac{1}{2} \rfloor$
- $\theta$  ist ein Faktor zur Anpassung der Granularität der Diskretisierung, dessen Größe auf dem Abstand  $\delta$  des Kalibrierungsvektors  $c$  vom perfekten Sensor  $s$  basiert:

$$\theta(c) = 2^{\max(\lceil \lg \delta(c) - \varphi \rceil, 0)}, \quad (4.5)$$

mit der Abstandsfunktion  $\delta(c)$

$$\delta(c) = \|c - s\|_2 = \sqrt{\sum_{n=0}^k \left(\frac{c_n - s_n}{d_n}\right)^2} \quad (4.6)$$

und der Konstanten  $\varphi$ , die bestimmt, ab welchem Abstand die Granularität der Diskretisierung vergrößert werden soll.

Zur Verdeutlichung der Vorgehensweise soll folgendes Beispiel dienen. Gegeben sei ein Kalibrierungsvektor  $c_u = (9, 2; 0, 86)^T$ . Die Kalibrierungskurve ist folglich  $\phi(c_u, x) = 0,86 * x + 9,2$  und korrigiert in diesem Fall sowohl einen konstant- als auch ein proportional-systematischen Messfehler. Des Weiteren seien der Diskretisierungsvektor  $d = (2, 0; 0, 1)^T$  und die Konstante  $\varphi = 2$  gegeben. Zunächst wird der Abstand zum perfekten Sensor bestimmt:

$$\delta(c_u) = \|c_u - s\|_2 = \sqrt{\left(\frac{9,2}{2,0}\right)^2 + \left(\frac{0,86 - 1,0}{0,1}\right)^2} \approx 4,81. \quad (4.7)$$

Damit erhält man den Faktor  $\theta(c_u) = 2^{\max(\lceil \lg(4,81) - 2,0 \rceil, 0)} = 2$  und den diskretisierten Kalibrierungsvektor:

$$\tilde{c}_u = \psi(c_u, d) = \begin{pmatrix} \lceil \frac{9,2}{2,0 * 2} \rceil * (2, 0 * 2) \\ \lceil \frac{0,86}{0,1 * 2} \rceil * (0, 1 * 2) \end{pmatrix} = \begin{pmatrix} 8, 0 \\ 0, 8 \end{pmatrix}. \quad (4.8)$$

Ein zweiter Kalibrierungsvektor  $c_v = (3, 6; 0, 86)^T$  wird aufgrund des geringeren Abstandes zum perfekten Sensor ( $\theta(c_v) \approx 2, 28$ ) weniger stark diskretisiert:

$$\tilde{c}_v = \psi(c_v, d) = \begin{pmatrix} \lceil \frac{3,6}{2,0 * 1} \rceil * (2, 0 * 1) \\ \lceil \frac{0,86}{0,1 * 1} \rceil * (0, 1 * 1) \end{pmatrix} = \begin{pmatrix} 4, 0 \\ 0, 9 \end{pmatrix}. \quad (4.9)$$

Das Prinzip, die Kalibrierungsvektoren in Abhängigkeit vom Abstand zum perfekten Sensor zu diskretisieren, liegt in der Annahme begründet, dass es mehr Knoten mit kleinen als mit großen Abweichungen gibt. Durch die mehrmalige, inkrementelle Kalibrierung nähern sich die Vektoren immer mehr dem perfekten Sensor an. Somit ist die Wahrscheinlichkeit, dass mehrere Knoten die gleichen Kalibrierungsvektoren haben, bei kleinen Abweichungen auch höher.

Die Diskretisierung führt offensichtlich zu einem Genauigkeitsverlust bei der Rückwärtskalibrierung, welcher von  $d$  und  $\phi$  abhängt. Dieser eingeführte Fehler sollte jedoch im Vergleich zu der durch die Kalibrierung eingeführten zusätzlichen Genauigkeit gering sein. Zudem wird  $\tilde{c}$  auch lediglich bei der Rückwärtskalibrierung verwendet, wodurch der eingeführte Fehler sich nicht auf zukünftige Messungen auswirkt.

Um Angriffe zu vermeiden, die auf dem Upload-Zeitpunkt der Kalibrierungsinformationen basieren, werden die Tupel aus Kalibrierungsvektoren und Pseudonymen nur an festgelegten Zeitpunkten übermittelt, d. h. es findet eine zeitliche Bündelung statt. An diesen Kalibrierungszeitpunkten übermitteln alle Knoten, die neue Kalibrierungsinformationen besitzen, diese gleichzeitig an den Server. Die Übertragung findet wieder über MIX-Netze statt, so dass der Server die erhaltenen Aktualisierungen keinem Nutzer bzw. Gerät zuordnen kann.

Der letzte Schritt im Zuge der Rückwärtskalibrierung ist die gewichtete Korrektur der Messdaten durch den Server. Das Prinzip dieser gewichteten Korrektur ist

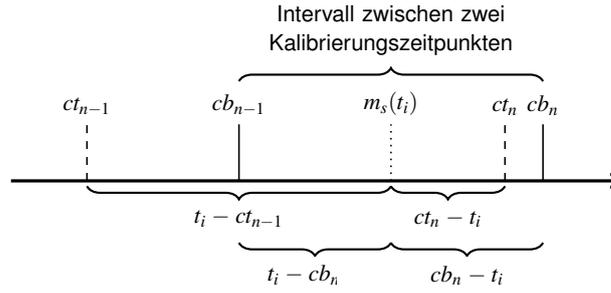


Abbildung 4.20: Die ideale Gewichtung basiert auf den wirklichen Kalibrierungszeitpunkten  $ct$ , der Server kann diese aber nur durch die Upload-Zeitpunkte  $cb$  approximieren.

es, Messwerte, die kurz vor der Kalibrierung aufgezeichnet wurden, stärker mittels der neu erhaltenen Kalibrierungsparameter anzupassen als solche, die schon lange vor dieser erfasst und übermittelt wurden. Liegt eine Messung  $m_s(t_i)$ , wie in Abbildung 4.20, näher an der zum Zeitpunkt  $ct_n$  stattfindenden Kalibrierung als an  $ct_{n-1}$ , sollte der Kalibrierungsvektor  $\omega(ct_n)$  auch stärker ins Gewicht fallen als der zuvor bestimmte  $\omega(ct_{n-1})$ . Im Idealfall sollte eine gewichtete Kombination des alten Messwertes und des durch die neue Kalibrierung angepassten Messwertes erfolgen, wobei die Gewichte auf dem Zeitpunkt innerhalb des aktuellen Kalibrierungsintervalls basieren. Liegt eine Messung  $m_s(t_i)$  zwischen den Kalibrierungszeitpunkten  $ct_{n-1}$  und  $ct_n$ , sollen die alten Kalibrierungsparameter  $\omega(ct_{n-1})$  mit  $\frac{ct_n - t_i}{ct_n - ct_{n-1}}$  und die neuen Kalibrierungsparameter  $\omega(ct_n)$  mit  $\frac{t_i - ct_{n-1}}{ct_n - ct_{n-1}}$  gewichtet werden. Der mit einer idealen Gewichtung korrigierte Messwert  $\hat{m}_s(t_i)$  wird demnach wie folgt bestimmt:

$$\hat{m}_s(t_i) = \frac{(t_i - ct_{n-1}) * \phi(\omega(ct_n), m_s(t_i))}{ct_n - ct_{n-1}} + \frac{(ct_n - t_i) * \phi(\omega(ct_{n-1}), m_s(t_i))}{ct_n - ct_{n-1}}. \quad (4.10)$$

Das Problem dabei ist, dass dem Server aufgrund der zeitlich gebündelten Übermittlung der Kalibrierungsinformationen die wirklichen Kalibrierungszeitpunkte nicht bekannt sind. Daher kann dieser nur eine Annäherung an die ideale Gewichtung bestimmen. Bei PRICAPS werden hierfür die Upload-Zeitpunkte verwendet, d. h. die Gewichtung basiert auf der zeitlichen Lage der Messung zwischen dem aktuellen und dem vorherigen Upload-Zeitpunkt. Die bei einer Rückwärtskalibrierung korrigierten Messwerte  $\tilde{m}_s(t_i)$  werden demnach wie folgt bestimmt:

$$\tilde{m}_s(t_i) = \frac{(t_i - cb_{n-1}) * \phi(\omega(cb_n), m_s(t_i))}{cb_n - cb_{n-1}} + \frac{(cb_n - t_i) * \phi(\omega(cb_{n-1}), m_s(t_i))}{cb_n - cb_{n-1}}, \quad (4.11)$$

wobei  $cb_n$  und  $cb_{n-1}$  den aktuellen bzw. den vorherigen Upload-Zeitpunkt der Kalibrierungsinformationen bezeichnen.

Da die vom Server bestimmte Annäherung deutlich von der idealen Gewichtung abweichen kann, bestimmen die Knoten vor der Übermittlung der Kalibrierungsinformationen lokal die ideale Gewichtung  $\hat{m}_s(t_i)$ . Eine Rückwärtskalibrierung wird dann nur angestoßen, wenn durch diese der Messwert näher an dem mit der idealen Gewichtung korrigierten Wert liegt. Das bedeutet, dass Knoten nur Kalibrierungsinformationen übermitteln, wenn folgende Gleichung gilt:

$$|\hat{m}_s(t_i) - m_s(t_i)| > |\hat{m}_s(t_i) - \tilde{m}_s(t_i)|. \quad (4.12)$$

In diesem Abschnitt wurde der Prozess der Rückwärtskalibrierung von PRICAPS beschrieben. Die Auswirkungen der beschriebenen Maßnahmen auf die Privatsphäre der Nutzer und auf die Qualität der Daten sollen im folgenden Abschnitt untersucht werden.

#### 4.4.4 Evaluation

Ziel von PRICAPS ist es, die Datenqualität in PSNs durch Kalibrierungsvorgänge zu erhöhen, gleichzeitig aber die Privatsphäre der Nutzer bei den dafür notwendigen Interaktionen zu wahren. In diesem Abschnitt soll nun untersucht werden, inwieweit diese Ziele durch die eingeführten Maßnahmen erreicht werden. Die Untersuchung wurde mit Hilfe einer Simulationsumgebung durchgeführt, welche im Folgenden beschrieben wird.

##### 4.4.4.1 Simulationsumgebung

Die Simulationsumgebung wurde für die Simulation eines PSNs entwickelt, in dem die Teilnehmer Ozonmessungen erfassen. Als Eingabe für die Grundwahrheit der Simulation wurden reale Ozonmessungen verwendet, welche von Luftmessstationen des *Lufthygienischen Landesüberwachungssystems Bayern (LÜB)* [268] in München über einen Zeitraum von 14 Tagen erfasst wurden.

Diese Daten wurden einerseits zeitlich interpoliert, um so die Auflösung der Daten von einem Datensatz pro Stunde auf einen pro Minute zu erhöhen. Andererseits wurde eine räumliche Interpolation durchgeführt, damit für jeden Punkt des Simulationsgebietes ein Datensatz der Grundwahrheit vorhanden ist. Für letztere wurde die *Shepard-Methode* [212] verwendet, die ein Verfahren der inversen Distanzgewichtung darstellt. Die Annahme dieser Methode ist, dass die Ähnlichkeit eines unbekanntes Messwertes zu bekannten Messwerten mit der Entfernung zu diesen abnimmt. Der unbekanntes Messwert wird daher durch eine gewichtete Verrechnung der umliegenden Messwerte geschätzt. Das jeweilige Gewicht ist dabei proportional zum Inversen des Abstandes zwischen dem Ort des unbekanntes Wertes und dem

des bekannten Messwertes. Soll ein unbekannter Messwert  $m^*$  aus den bekannten Messwerten  $m_1, \dots, m_n$  geschätzt werden, so wird dieser wie in Gleichung 4.13 bestimmt:

$$m^* = \begin{cases} \frac{\sum_{i=0}^n w_i m_i}{\sum_{i=0}^n w_i}, & \text{falls } \forall i : d(m^*.l, m_i.l) \neq 0 \\ m_i, & \text{falls } \exists i : d(m^*.l, m_i.l) = 0 \end{cases} \quad (4.13)$$

mit den Gewichten

$$w_i = \frac{1}{d(m^*.l, m_i.l)^p}, \quad (4.14)$$

wobei  $m.l$  wiederum den Ort einer Messung beschreibt und  $d$  die Abstandsfunktion bezeichnet. Der Potenz-Parameter  $p$  legt fest, wie stark die Ähnlichkeit mit zunehmenden Abstand abnimmt. Im Rahmen der Simulationsumgebung wurde dieser auf  $p = 2$  festgesetzt.

Um die Messfehler der mobilen Knoten zu simulieren, wurde das von Hasenfratz et al. [89] vorgestellte Fehlermodell für Ozonmessungen verwendet. Zur Bestimmung des Modells haben die Autoren mobile Sensorknoten mit Ozonsensoren des Typs *MiCS-OZ-47* ausgestattet, mit denen in der Umgebung von Referenzstationen Messungen durchgeführt wurden. Die Ozonkonzentration wurde dabei mit der relativen Mengenangabe *Parts per Billion* (ppb; dt. „Teile pro Milliarde“) angegeben. Ein Ozongehalt von 1 ppb entspricht einem Kubikmillimeter Ozon in einem Kubikmeter Luft. Hasenfratz et al. haben dabei beobachtet, dass die Messfehler normalverteilt sind, wenn die Sensorknoten nur initial kalibriert werden. Die Ergebnisse ergaben eine Normalverteilung  $\mathcal{N}(\mu; \sigma^2)$  mit  $\mu \sim \mathcal{U}(-9; 9)$  ppb und  $\sigma \sim \mathcal{N}(3; 1)$  ppb über die Dauer eines Tages. Um die Alterungseffekte der Sensoren zu berücksichtigen, verwenden Hasenfratz et al.  $\mu \sim \mathcal{U}(-9 - \frac{d}{5}; 9 + \frac{d}{5})$  in ihrem Modell. Dabei beschreibt  $d$  die Anzahl der Tage in Verwendung, wodurch der mögliche Fehler jeden Tag steigt. [89]

Im Gegensatz zu den Messungen von Hasenfratz et al., werden die Messdaten des LÜB in  $\mu\text{g m}^{-3}$  angegeben. Für die Umrechnung einer relativen Angabe  $c_v$  in eine massebezogene Ozonkonzentration  $c_m$  kann bei SATP-Bedingungen (*Standard Ambient Temperature and Pressure*), d. h. bei einer Temperatur von 298,15 K (= 25°C) und 100 000 Pa (= 1 bar), Formel 4.15 verwendet werden [207]:

$$c_m[\mu\text{g m}^{-3}] = c_v[\text{ppb}] * \frac{48[\text{g mol}^{-1}]}{24,789[\text{L mol}^{-1}]} \approx 1,936 * c_v[\mu\text{g m}^{-3}] \quad (4.15)$$

Für die Simulationen wurde basierend auf dem zuvor beschriebenen Modell für jeden mobilen Sensorknoten eine Fehlerkurve bestimmt. Der Grad der Fehlerkurve wurde auf 1 festgelegt, d. h. es wurden Polynome der Form  $a * x + b$  verwendet.

Zudem wurde ein im Vergleich zum obigen Modell etwas konservativer Alterungsfaktor von  $0,2 \mu\text{g m}^{-3}/\text{Tag}$  angenommen. Die simulierten Messwerte ergeben sich dann aus der Anwendung der jeweiligen Fehlerkurve auf die Werte der Grundwahrheit, worauf zusätzliche Abweichungen für die Alterung und zufälliges Rauschen addiert werden.

Mit dieser Datenbasis wurden dann die Simulationen durchgeführt. Weitere Parameter der Simulation, wie beispielsweise das verwendete Mobilitätsmodell, sind in Tabelle 4.2 aufgelistet.

Nutzer	1 000, 1 500, 2 000
Simulationszeit	14 Tage
Mobilität	Random Walk
Max. Geschwindigkeit	8,33 m/s
Messfrequenz	4x pro Stunde
Referenzstationen	5
Abgleichsdistanz $\delta_l$	250 m
$C_{MinCount}$	5
$C_{MinRange}$	$30 \mu\text{g m}^{-3}$
$C_{Timeout}$	5 Tage

Tabelle 4.2: Simulationsparameter für die Evaluation von PRICAPS

Zudem wurde das Intervall für die Rückwärtskalibrierungen auf eine Woche gesetzt. Analog zur Fehlerkurve wurde für die Kalibrierungskurve  $\phi$  ebenfalls ein Polynom ersten Grades verwendet, wodurch für die Kalibrierungs- und Diskretisierungsvektoren gilt:  $c, \tilde{c}, d \in \mathbb{R}^2$ .

Für den Diskretisierungsschritt muss der Diskretisierungsvektor  $d = (d_0, d_1)^T$  und die Konstante  $\varphi$  festgelegt werden. Für Ersteren wurde  $d_0 = \{1, 0; 1, 5; 2, 0\}$  und  $d_1 = \{0, 05; 0, 1; 0, 15; 0, 2\}$  verwendet, wohingegen für die Konstante  $\varphi = \{2; 3; 4\}$  gesetzt wurde. Durch die Permutationen der Parameter erhält man 36 verschiedene Diskretisierungsvarianten, welche im Folgenden mit Bezeichnern der Form „ $d_0; d_1/\varphi$ “ angegeben werden.

#### 4.4.4.2 Erreichtes $k$ -Anonymitätslevel

In einem ersten Schritt wurde untersucht, inwiefern die Privatsphäre der Nutzer durch PRICAPS geschützt werden kann. Wie zuvor in Abschnitt 4.4.1 beschrieben, ist bei der Übertragung der exakten Kalibrierungsparameter eine Rückverfolgung der Nutzerpfade möglich. Das Ziel in PRICAPS ist somit, diese Möglichkeit der eindeutigen Rückverfolgung durch das Konzept der  $k$ -Anonymität zu verhindern. Diese Kalibrierungsparameter werden also mit der Absicht manipuliert, dass zu jedem Kalibrierungszeitpunkt jeder übertragene Kalibrierungsvektor mindestens  $k$ -mal vorkommt.

In den Simulationen wurden die im vorigen Abschnitt genannten 36 Diskretisierungsvarianten verwendet und untersucht, wie oft die gewünschte  $k$ -Anonymität erreicht wurde. Das Level der Anonymität wurde dabei mit  $k = \{2, 3, \dots, 10\}$  variiert.

Die Abbildungen 4.21(a) bis (c) zeigen die Ergebnisse für die Simulationen mit 1 000 mobilen Sensorknoten. Auf der x-Achse sind die verschiedenen  $k$ -Anonymitätslevel aufgetragen, die y-Achse zeigt den Anteil der Kalibrierungsvorgänge, bei denen das entsprechende  $k$ -Anonymitätslevel erreicht wurde. Die verschiedenen Graphen zeigen die Ergebnisse für die Diskretisierungsvarianten. Wie zuvor erwähnt, sind diese in der Form „ $d_0; d_1/\varphi$ “ angegeben, d. h. der erste in der Legende von Abbildung 4.21(a) aufgelistete Graph zeigt die Ergebnisse der Diskretisierung mit einem Diskretisierungsvektor  $d = (1, 0; 0, 05)^T$  und  $\varphi = 2, 0$ .

Die Ergebnisse zeigen sehr deutliche Unterschiede zwischen den verschiedenen Varianten. Da bei größeren Diskretisierungsvarianten, wie z. B.  $2, 0; 0, 2/2, 0$ , die Kalibrierungsvektoren sehr stark manipuliert wurden, siehe Abbildung 4.21(c), wurde bei diesen das  $k$ -Anonymitätslevel deutlich öfter erreicht als bei sehr feinen Varianten, wie beispielsweise  $1, 0; 0, 05/4, 0$ , siehe Abbildung 4.21(a). Zudem lässt sich erkennen, dass der Diskretisierungsparameter  $d_1$  einen starken Einfluss auf die Ergebnisse hatte. Mit  $d_1 = 0.15$  oder  $d_1 = 0.2$  wurde das gewünschte  $k$ -Anonymitätslevel signifikant häufiger erreicht als bei der Verwendung kleinerer Werte. Auch die Wahl von  $\varphi$  hatte erheblichen Einfluss. Der Parameter  $\varphi$  bestimmt, ab welcher Abweichung vom perfekten Sensor der Diskretisierungsvektor angepasst, d. h. ein größerer Diskretisierungsvektor verwendet werden soll. Kleine Werte für  $\varphi$  führten somit öfter zu größeren Diskretisierungen, was sich positiv auf das Erreichen des  $k$ -Anonymitätslevels auswirkte.

Insgesamt zeigen die Ergebnisse, dass mit PRICAPS in vielen Fällen die gewünschte Anonymität erreicht werden konnte. Mit  $k = 10$  wurde bei 23 der 36 Diskretisierungsvarianten immerhin in 60 % der Fälle das  $k$ -Anonymitätslevel erreicht. Mit  $k = 5$  wurde bei 28 Varianten in mehr als 80 % der Fälle das gewünschte Level erreicht. Dies ist als sehr positiv zu bewerten, da schon ab  $k = 2$  eine eindeutige und belastbare Zuordnung nicht mehr möglich ist, vgl. [56].

In einem weiteren Schritt wurden die Auswirkungen der Anzahl der Messknoten auf das  $k$ -Anonymitätslevel untersucht. Daher wurden die beste und schlechteste Diskretisierungsvariante bezüglich des erreichten  $k$ -Anonymitätslevels ausgewählt und in Simulationsläufen mit 1 000, 1 500 und 2 000 Knoten untersucht. Die Ergebnisse sind in Abbildung 4.22 dargestellt. Dabei war vor allem bei der sehr feinen Diskretisierungsvariante ein signifikanter Anstieg bei der Erreichung des  $k$ -Anonymitätslevels zu erkennen. Da das Anwendungsszenario von PRICAPS partizipative Sensornetze sind, bei denen von sehr vielen Nutzern großflächig Messungen durchgeführt werden, sind die untersuchten Knotenzahlen und damit auch die Ergebnisse für das Erreichen des  $k$ -Anonymitätslevels eher als untere Grenze zu sehen.

Das Ziel von PRICAPS ist der Schutz der Privatsphäre der Nutzer bei gleichzeitig möglichst guter Datenqualität. Daher muss neben dem erreichten  $k$ -Anonymitäts-

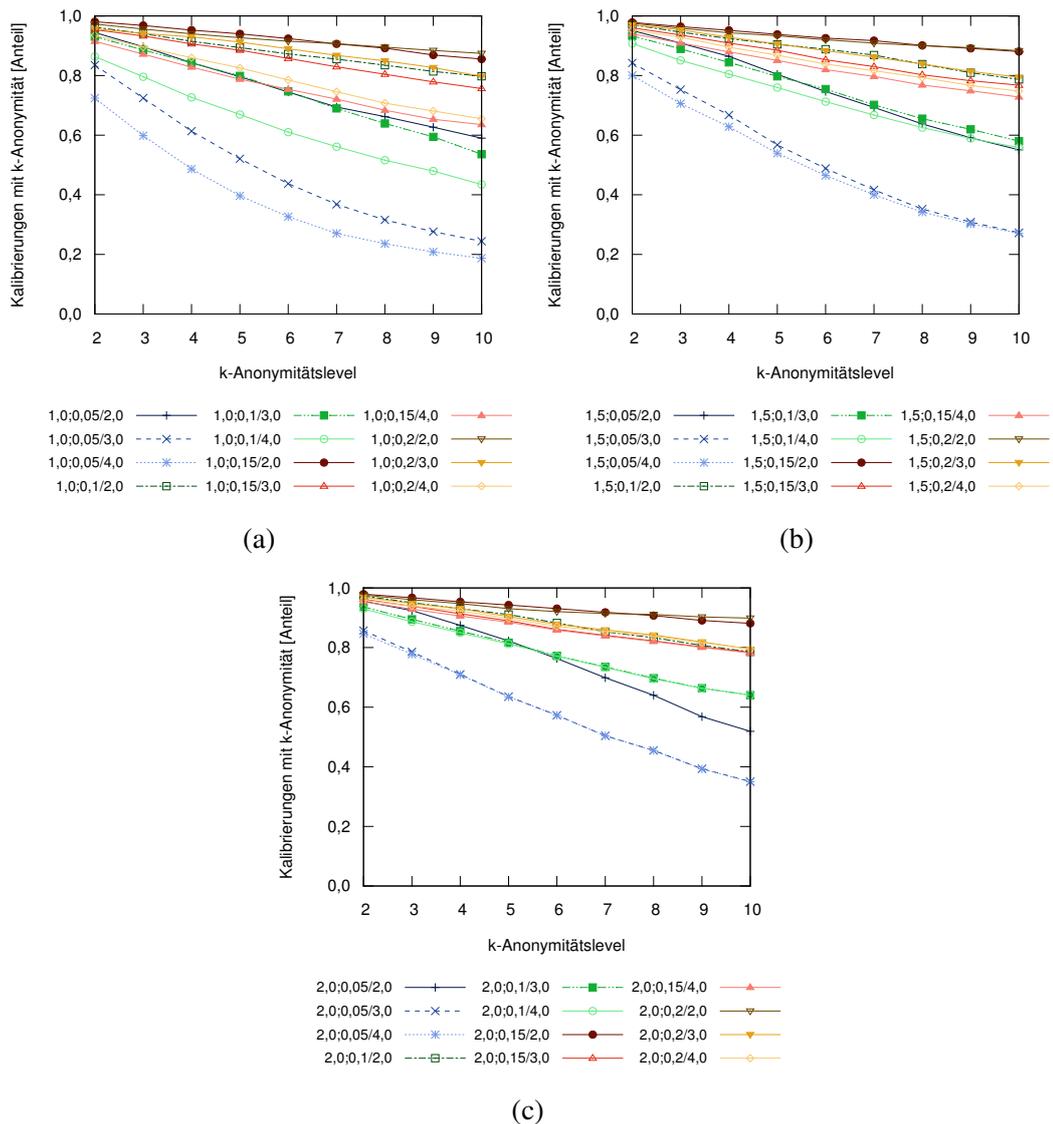


Abbildung 4.21: Erreichtes  $k$ -Anonymitätslevel bei 1 000 Knoten für verschiedene Diskretisierungsvarianten

level auch der Genauigkeitsverlust untersucht werden, der durch den Diskretisierungsvorgang entsteht.

#### 4.4.4.3 Genauigkeitsverlust bei Messdaten

Durch die in Abschnitt 4.4.3.3 vorgestellten Maßnahmen entsteht ein Genauigkeitsverlust. Dieser wird in Kauf genommen, um die Privatsphäre der Nutzer zu wahren. Es muss allerdings gewährleistet werden, dass die Datenqualität dennoch hoch bleibt. Vor allem muss sichergestellt werden, dass durch die Schutzmaßnahmen im Rahmen der Rückwärtskalibrierung die Daten nicht ungenauer werden als ohne eine

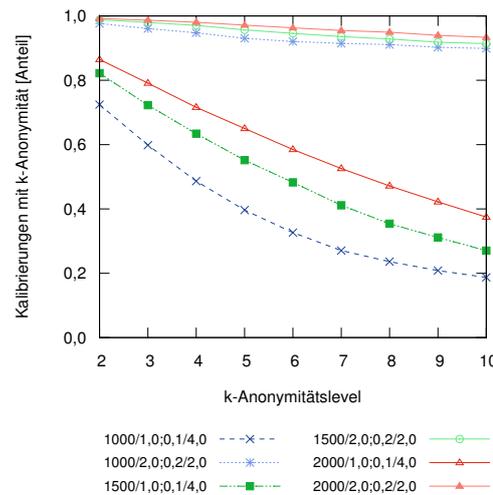


Abbildung 4.22: Erreichtes  $k$ -Anonymitätslevel bei variierender Knotenzahl für beste bzw. schlechteste Diskretisierungsvariante

Kalibrierung. Im Folgenden soll daher untersucht werden, wie hoch der Genauigkeitsgewinn bzw. -verlust für die unterschiedlichen Diskretisierungsvarianten ausfällt.

In diesem Schritt wurden nun lediglich die Diskretisierungsvarianten untersucht, die bei der vorherigen Untersuchung eine  $k$ -Anonymität mit  $k = 10$  bei mindestens 60 % der Kalibrierungsvorgänge erreicht haben, d. h. es wurden die 23 besten Varianten bezüglich des Schutzes der Privatsphäre ausgewählt und untersucht. Die Ergebnisse sind in Abbildung 4.23(a) und 4.23(b) dargestellt. In diesen werden der durchschnittliche Diskretisierungsfehler  $\tilde{c}_{err}$  und der durch die Rückwärtskalibrierung entstehende durchschnittliche Genauigkeitsgewinn  $\tilde{c}_{gain}$  für die genannten Diskretisierungsvarianten gegenüber gestellt.

Für den Diskretisierungsfehler wurde analysiert, welcher Fehler durch die Diskretisierung eingeführt wurde, d. h. es wurde die Abweichung der exakten Rückwärtskalibrierung  $\phi(c, m_s(t_i))$  von der Grundwahrheit  $p(t_i)$  mit der Abweichung der diskretisierten Rückwärtskalibrierung  $\phi(\tilde{c}, m_s(t_i))$  von dieser verglichen:

$$\tilde{c}_{err}(t_i) = |\phi(\tilde{c}, m_s(t_i)) - p(t_i)| - |\phi(c, m_s(t_i)) - p(t_i)|. \quad (4.16)$$

Die Diskretisierungsfehler wurden dann für jede Messung eines Knotens aufaddiert. Um dabei größere Abweichungen stärker zu bestrafen als kleinere Abweichungen, wurde die Wurzel aus der mittleren quadratischen Abweichung (engl. *root mean square deviation (RMSD)*) bestimmt:

$$\text{RMSD}(\tilde{c}_{err}) = \sqrt{\frac{\sum_{i=1}^n (\tilde{c}_{err}(t_i))^2}{n}}. \quad (4.17)$$

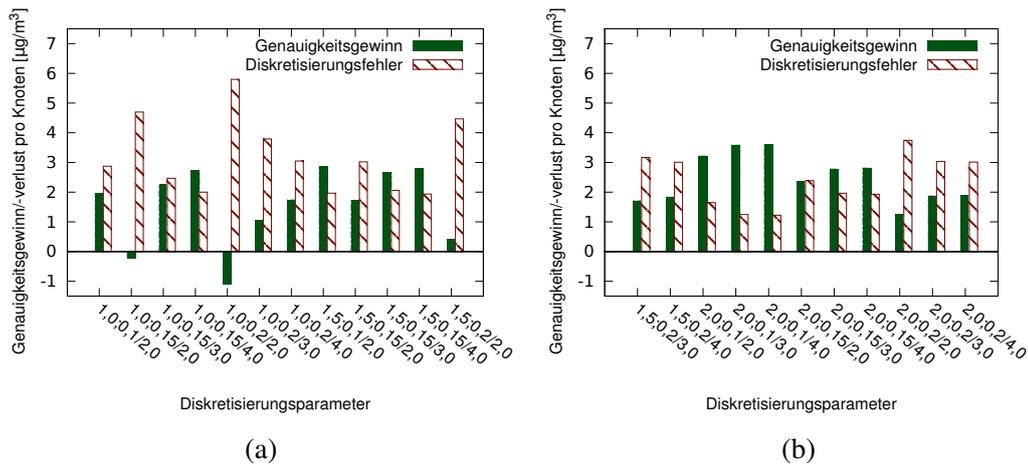


Abbildung 4.23: Durchschnittlicher Genauigkeitsgewinn/-verlust für verschiedene Diskretisierungsvarianten

Der Durchschnitt von  $\text{RMSD}(\tilde{c}_{err})$  über alle Knoten ist in Abbildung 4.23 dargestellt.

Für  $\tilde{c}_{gain}$  wurde untersucht, welcher Genauigkeitsgewinn durch die Rückwärtskalibrierung erreicht werden konnte. Dafür wurde die Abweichung des Messwertes von der Grundwahrheit ohne Rückwärtskalibrierung bestimmt und mit der Abweichung des Messwertes mit angewandter Rückwärtskalibrierung  $\phi(\tilde{c}, m_s(t_i))$  verglichen. Da der Messwert  $m_s(t_i)$  durch die Vorwärtskalibrierung bereits angepasst worden sein kann, soll  $m'_s(t_i)$  im Folgenden den Messwert ohne erfolgte Rückwärtskalibrierung bezeichnen. Der Genauigkeitsgewinn wurde somit wie folgt berechnet:

$$\tilde{c}_{gain}(t_i) = |m'_s(t_i) - p(t_i)| - |\phi(\tilde{c}, m_s(t_i)) - p(t_i)|. \quad (4.18)$$

Analog zum Diskretisierungsfehler wurde auch hier für jeden Knoten  $\text{RMSD}(\tilde{c}_{gain})$  bestimmt und anschließend der Durchschnitt über alle Knoten ermittelt.

Die Ergebnisse sind, wie erwartet, orthogonal zu den Ergebnissen zum Schutz der Privatsphäre: Je feiner die Diskretisierung gewählt wurde, desto kleiner war der Diskretisierungsfehler bzw. desto größer der Genauigkeitsgewinn. Man kann außerdem erkennen, dass auch hier die Wahl von  $d_1$  und  $\varphi$  entscheidend für die Qualität der Ergebnisse war. Der Genauigkeitsgewinn war größtenteils im positiven Bereich, mit nur wenigen Ausnahmen, d. h. trotz der Diskretisierung wurde durch die Rückwärtskalibrierung ein genauere Messwert bestimmt als ohne diese. Ein negativer Genauigkeitsgewinn, wie z. B. bei  $1,0;0,2/2,0$ , zeigt, dass durch die Diskretisierung in diesem Fall ein zu großer Fehler erzeugt wurde. Der durch die Rückwärtskalibrierung angepasste Messwert war in diesem Fall ungenauer als der Messwert ohne diese Anpassung. Insgesamt zeigt sich, dass Varianten mit kleinem  $d_1$  oder mit großem  $\varphi$  kleinere Fehler und folglich auch einen größeren Genauigkeitsgewinn zur Folge hatten.

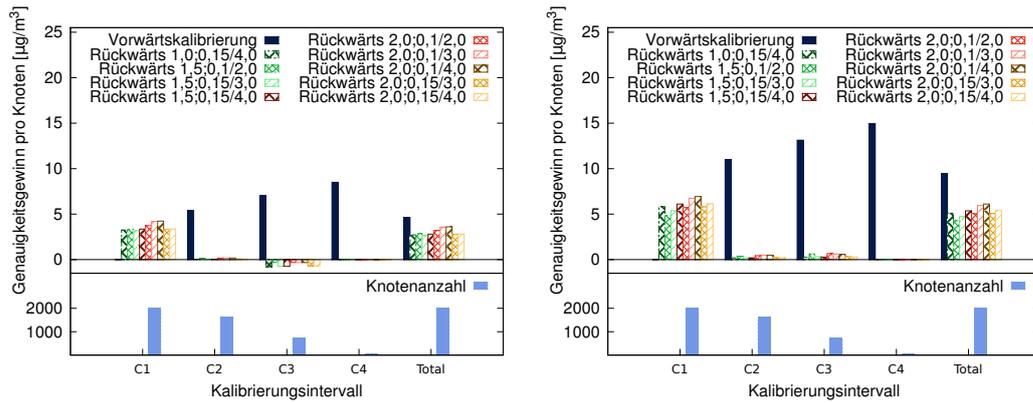
(a) Mit Alterungsfaktor  $0,2 \mu\text{g m}^{-3}/\text{Tag}$ (b) Mit Alterungsfaktor  $1,2 \mu\text{g m}^{-3}/\text{Tag}$ 

Abbildung 4.24: Genauigkeitsgewinn für Vorwärts- und Rückwärtskalibrierung in den einzelnen Kalibrierungsintervallen mit verschiedenen Alterungsfaktoren

Zudem wurde der Genauigkeitsgewinn in den einzelnen Kalibrierungsintervallen untersucht. Als Kalibrierungsintervall wird die Zeitspanne zwischen zwei Kalibrierungszeitpunkten bezeichnet. Formal sind diese wie folgt definiert:  $\{i \in 1, \dots, n+1 : C_i = [ct_{i-1}; ct_i]\}$ , wobei  $\{ct_1, ct_2, \dots, ct_n\}$  die Kalibrierungszeitpunkte bezeichnen und  $ct_0$  und  $ct_{n+1}$  den Start- bzw. den Endzeitpunkt der Simulation angeben. Das erste Intervall  $C_1$  bezeichnet somit die Zeit zwischen dem Simulationsstart und dem ersten Kalibrierungszeitpunkt. Die maximale Kalibrierungshäufigkeit wurde mit  $C_{Timeout} = 5$  Tage begrenzt, vgl. Tabelle 4.2. Deshalb waren in den Simulationsläufen maximal drei Kalibrierungszeitpunkte pro Knoten und folglich maximal vier Kalibrierungsintervalle ( $C_1$  bis  $C_4$ ) möglich.

Die Abbildungen 4.24(a) und (b) zeigen die Ergebnisse. Im oberen Teil der Abbildungen ist der durchschnittliche Genauigkeitsgewinn pro Knoten für die einzelnen Kalibrierungsintervalle dargestellt. Dabei wurden für diese Analyse nur die neun Varianten ausgewählt, deren Genauigkeitsgewinn im vorherigen Schritt den Diskretisierungsfehler überstiegen hat. Zum Vergleich wurde auch der Genauigkeitsgewinn aufgetragen, der durch die Vorwärtskalibrierung entstand, wenngleich man beachten muss, dass bei dieser keine Diskretisierung stattfand und diese zudem vor der Rückwärtskalibrierung stattfand, wodurch immer deutlich höhere Genauigkeitsgewinne erreicht wurden. Der untere Teil der Abbildungen zeigt die Anzahl der Knoten, die in diesem Intervall eine Kalibrierung durchgeführt haben. Für diese Analyse wurden Simulationen mit 2 000 Knoten verwendet. Die Knotenanzahl mit mehreren Kalibrierungsintervallen war geringer, da die Anzahl der Kalibrierungsvorgänge eines Knotens von dessen Mobilität abhängt, d. h. ob und wie oft dieser in die Umgebung einer Referenzstation kommt.

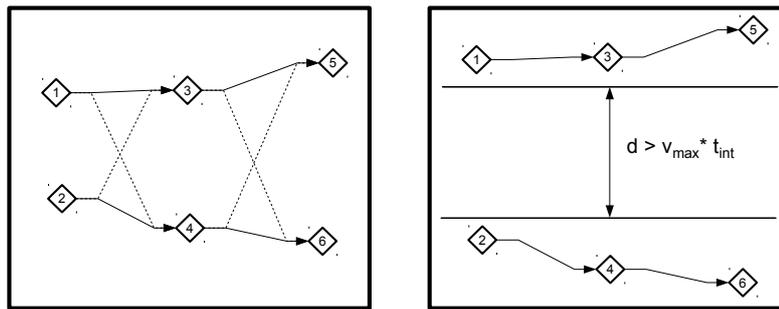
In Abbildung 4.24(a) sind die Ergebnisse bei Verwendung eines Alterungsfaktors von  $0,2 \mu\text{g m}^{-3}/\text{Tag}$  dargestellt. Im Intervall  $C_1$  ist noch kein Genauigkeitsgewinn

durch die Vorwärtskalibrierung zu erkennen, da zu diesem Zeitpunkt noch keine Kalibrierung stattgefunden hatte und somit die Messwerte nicht angepasst wurden. In den darauf folgenden Intervallen war ein deutlicher Genauigkeitsgewinn zu beobachten, welcher mit zunehmender Intervallanzahl leicht anstieg. Dies kann zum einen darauf zurückgeführt werden, dass die Kalibrierung zunehmend exakter wird. Zum anderen sind bei häufigeren Kalibrierungen auch die Intervalle kürzer, weshalb die Sensoren innerhalb dieser Intervalle weniger von der bestimmten Kalibrierung abweichen können.

Bei der Rückwärtskalibrierung war der größte Genauigkeitsgewinn im Intervall  $C_1$  zu beobachten. Zu diesem Zeitpunkt waren die Knoten noch komplett unkalibriert, wodurch die rückwirkende Anpassung der Werte natürlich den größten Effekt hatte. In den darauf folgenden Intervallen war die Verbesserung durch die Rückwärtskalibrierung minimal bzw. sogar negativ. Dies resultierte aus den vergleichsweise kurzen Intervallen zwischen zwei Kalibrierungsintervallen. In  $C_3$  waren die Sensoren bereits zweimal kalibriert worden und die Effekte durch die Alterung sehr gering, so dass die diskretisierte Kalibrierung die Genauigkeit nicht verbesserte. In  $C_4$  traten keine Effekte der Rückwärtskalibrierung in Erscheinung, da am Ende dieses Intervalls keine Kalibrierung stattfand und eine Rückwärtskalibrierung somit nicht möglich war. Der durchschnittliche Genauigkeitsgewinn über alle Knoten und alle Intervalle ist in der Abbildung mit *Total* angegeben. Da hier auch die Knoten mit einbezogen wurden, die während des kompletten Durchlaufes keine Kalibrierung durchgeführt und somit einen durchschnittlichen Genauigkeitsgewinn von 0 hatten, fiel dieser geringer aus als bei den Ergebnissen in  $C_1$  bis  $C_4$ , bei denen nur die Knoten betrachtet wurden, die eine Kalibrierung durchgeführt hatten.

In Abbildung 4.24 wurde der Alterungsfaktor auf  $1,2 \mu\text{g m}^{-3}/\text{Tag}$  erhöht, wodurch ein schnelleres Altern der Sensoren simuliert wird. Dies kann auch verwendet werden, um längere Intervalle zwischen der Kalibrierungszeitpunkten mit gleichbleibendem Alterungsfaktor zu simulieren, d. h. in diesem Fall sechsmal längere Zeitintervalle mit einem Alterungsfaktor von  $0,2 \mu\text{g m}^{-3}/\text{Tag}$ . Im Vergleich zum vorherigen Szenario war eine deutliche Steigerung des Genauigkeitsgewinns sowohl bei der Vorwärts- als auch bei der Rückwärtskalibrierung zu erkennen. Vor allem bei Letzterer fällt auf, dass der Genauigkeitsgewinn nun immer positiv war. Eine Rückwärtskalibrierung ist folglich vor allem dann sinnvoll, wenn die Intervalle zwischen zwei Kalibrierungszeitpunkten so groß sind, dass die Sensoren durch die Alterung schon während der Intervalle stark von der vorherigen Kalibrierung abweichen.

Die Ergebnisse zeigen, dass der Einsatz von PRICAPS sowohl in Hinblick auf die Datenqualität als auch aus Sicht der Privatsphäre sinnvoll ist. Allerdings kann keine pauschale Empfehlung für eine Diskretisierungsvariante gegeben werden, da der Einsatz dieser immer vom Anwendungsszenario abhängt. Es gibt zwar Varianten, wie beispielsweise  $2, 0; 0, 1/2, 0$ , die insgesamt gute Ergebnisse zeigen. In manchen Szenarien, etwa mit strikteren Anforderungen an das  $k$ -Anonymitätslevel oder die Datenqualität, können diese aber eventuell nicht ausreichen. Sollen die Ergebnisse in Bezug auf erreichte  $k$ -Anonymitätslevel verbessert werden, kann eine etwas grö-



(a) Nutzerpfade mit unterschiedlicher Wahrscheinlichkeit  
 (b) Entfernte Messungen können nicht zu einem Pfad gehören

Abbildung 4.25: Beispiele möglicher Einschränkungen des Konzeptes

bere Diskretisierungsvariante gewählt, wohingegen für genauere Daten eine feinere Variante gewählt werden sollte.

#### 4.4.5 Diskussion und Erweiterungen

In diesem Abschnitt werden die vorherigen Ergebnisse diskutiert und Einschränkungen des PRICAPS-Ansatzes aufgezeigt. Die Ergebnisse haben gezeigt, dass PRICAPS kein bestimmtes Privatsphäre-Level garantieren kann. Dieses ist abhängig von der Anzahl der teilnehmenden Nutzer sowie deren Mobilität. Dadurch kann PRICAPS als ein *Best-Effort*-Ansatz gesehen werden, dessen Ziel der Schutz der Privatsphäre ist.

Zudem fehlt in PRICAPS die Betrachtung der Wahrscheinlichkeiten verschiedener Nutzerpfade. Teilweise können Nutzerpfade allein dadurch rekonstruiert werden, dass gewisse Pfade wahrscheinlicher sind als andere. Ein stark vereinfachtes Beispiel ist in Abbildung 4.25(a) dargestellt. In diesem sei angenommen, dass die durchgezogenen Verbindungen auf Grund des Straßennetzes wahrscheinlicher sind als die gepunkteten Verbindungen. In diesem Fall kann ein Angreifer relativ sicher die Pfade der zwei Nutzer bestimmen. Allerdings wird bei PRICAPS angenommen, dass die Messfrequenz relativ niedrig ist. Dadurch kann jede Messung von einer Vielzahl an Nutzern durchgeführt worden sein, wodurch eine verlässliche Rekonstruktion der Pfade nicht mehr möglich ist.

Ein weiterer Aspekt, der das effektive Privatsphäre-Level reduzieren könnte, ist die Möglichkeit, dass die Entfernung zwischen zwei Messungen zu groß ist, als dass diese auf einem Nutzerpfad liegen könnten. In Abbildung 4.25(b) ist dies illustriert: Die Entfernung zwischen Messung 3 und 4 ist hier größer als die Strecke, die ein Nutzer in dem Intervall zwischen zwei Messungen  $t_{int}$  mit maximaler Geschwindigkeit  $v_{max}$  zurücklegen kann. Folglich scheint es so, als ob die Messungen von zwei verschiedenen Nutzern beigetragen worden sind. Allerdings ist weder dem Server

noch einem potentiellen Angreifer bekannt, wie viele Nutzer zu einem Zeitpunkt Messungen beitragen. In dem skizzierten Beispiel könnten folglich die Messungen von zwei bis sechs Nutzern stammen, wodurch auch in diesem Fall eine verlässliche Rekonstruktion der Nutzerpfade nicht möglich ist.

Zudem sei angemerkt, dass bei der Bestimmung des  $k$ -Anonymitätslevels jeweils der *Worst Case* angenommen wurde. Das bedeutet, es wurde angenommen, dass zu jedem Zeitpunkt bekannt ist, wie viele Nutzer eine Kalibrierung durchführen. Wenn zu einem Zeitpunkt  $n$  Nutzer eine Rückwärtskalibrierung mit dem gleichen Kalibrierungsvektor  $\tilde{c}$  durchführen und jeder Nutzer  $m$  Messungen rückwirkend anpassen möchte, so werden bei diesem Vorgang  $n * m$  Messungen mit  $\tilde{c}$  korrigiert. In diesem Fall wurde bei den Ergebnissen ein  $k$ -Anonymitätslevel von  $n$  gewertet. Allerdings ist dem Server oder einem Angreifer nicht bekannt, wie viele Nutzer wirklich beteiligt sind. Die Updates könnten dabei von einem einzelnen Nutzer bis hin zu  $n * m$  Nutzern stammen, so dass das effektiv erreichte Privatsphäre-Level sogar über den in den Ergebnissen angegebenen Werten liegen sollte.

Bei möglichen Erweiterungen sollte ein Hauptziel sein, die Häufigkeit von Kalibrierungsvorgängen zu erhöhen, da sich dies sowohl auf die Datenqualität als auch auf die Privatsphäre positiv auswirken würde. Eine Möglichkeit hierfür kann die Integration eines Multihop-Kalibrierungskonzeptes in PRICAPS sein. Wie von Hasenfratz et al. [89] vorgeschlagen, können kalibrierte Geräte als Referenzknoten für andere Knoten dienen, wodurch mehr Geräte die für eine Kalibrierung notwendigen Referenzmesswerte erhalten.

Eine zusätzliche Erweiterungsmöglichkeit ist die Einführung von spieltypischen Elementen, sogenannter *Gamification*-Konzepte, als Anreizsystem, um die Mobilität der Nutzer zu beeinflussen. Wie bereits in verschiedenen Konzepten vorgeschlagen [32, 159, 201], können Nutzer belohnt werden, wenn sie ihre Mobilität auf eine bestimmte Weise anpassen. So können Nutzer dazu motiviert werden, ihre Pfade so zu adaptieren, dass sie sich öfters, bzw. sobald eine Kalibrierung wieder sinnvoll ist, in die Umgebung von Referenzstationen bewegen.

## 4.5 Zusammenfassung

Diese Kapitel befasste sich mit der zweiten großen Herausforderung – dem Schutz der Privatsphäre – und erörterte Bedrohungen und existierende Schutzmaßnahmen, bevor zwei neue, die Privatsphäre in PSNs betreffende Konzepte vorgestellt wurden.

Zunächst wurden Risiken und Probleme, welche die Privatsphäre der Teilnehmer von PSNs gefährden können, betrachtet und analysiert. Dabei wurden sowohl Bedrohungen betrachtet, die durch die Kontextdaten der durchgeführten Messungen entstehen, als auch solche, die aus den Messdaten selbst resultieren. Anschließend wurden existierende Schutzmaßnahmen und -konzepte für beide Bedrohungsfor-

men vorgestellt und offene Herausforderungen identifiziert. Für letztere wurden im Anschluss eigene Ansätze präsentiert.

Der erste vorgestellte Ansatz beschäftigte sich mit der automatisierten Erstellung von Privatsphäre-Zonen basierend auf Gebäudeinformationen. Diese helfen dabei, wichtige Orte des Nutzers, die durch eine langfristige Teilnahme an einem PSN identifiziert werden und zu einer Deanonymisierung des Nutzers führen können, zu verschleiern. Mit den Silent Zones wurde ein Konzept vorgestellt, bei dem sich die Zonen automatisch an das jeweilige Umfeld anpassen und somit ein Mindestlevel an Privatsphäre garantieren können. Die automatisierte Erstellung ist wichtig, da so die Privatsphäre der Nutzer auch ohne deren aktives Eingreifen geschützt wird, d. h. Nutzer können ohne Bedenken an einem PSN teilnehmen und müssen zu Beginn keine komplexen Privatsphäre-Einstellungen durcharbeiten. Gleichzeitig ist das Ziel der Silent Zones, die Privatsphäre möglichst effizient zu schützen, d. h. dem PSN sollen möglichst wenige Messdaten vorenthalten werden. Es wurden mit RNN und RR verschiedene Varianten zur Erstellung dieser Zonen beschrieben und anschließend wurde das Konzept mit existierenden Lösungen verglichen und evaluiert. Die Ergebnisse zeigen, dass die Silent Zones ein sinnvolles Konzept für PSNs darstellen und die spezifizierten Anforderungen erfüllt werden. In der abschließenden Diskussion wurden Verbesserungs- und Erweiterungsmöglichkeiten vorgestellt.

Das zweite präsentierte Konzept stellt einen Ansatz für die Kalibrierung in PSNs dar. Ziel von PSNs ist es, die erfassten Daten möglichst vielseitig verwenden zu können, wozu diese einerseits umgehend verfügbar andererseits möglichst genau sein sollten. Daher werden Messdaten unverzüglich an den Anwendungsserver übermittelt und, im Falle einer späteren Kalibrierung, rückwirkend korrigiert. Bei einer rückwirkenden Kalibrierung entsteht jedoch das Problem, dass durch die Übermittlung der Kalibrierungsparameter eventuell eine Nachverfolgung der Nutzerpfade ermöglicht wird. Daher wurde mit PRICAPS ein umfassendes Kalibrierungssystem für PSNs vorgestellt. Dieses ermöglicht eine rückwirkende Kalibrierung bereits übermittelter Daten, schützt aber gleichzeitig, durch eine intelligente Diskretisierung der Kalibrierungsparameter, die Privatsphäre der Nutzer. Die Anwendbarkeit des Ansatzes und dessen Auswirkung auf die Privatsphäre und die Datenqualität wurden mit Hilfe von Simulationen analysiert und bewertet.

## 5 Zusammenfassung und Ausblick

Infolge der Integration zahlreicher Sensoren entwickeln sich moderne Smartphones zu allgegenwärtigen und universell einsetzbaren Messgeräten. Dieser Trend, in Kombination mit der steigenden Verbreitung der Geräte, bietet großes Potenzial für die Erfassung von Messdaten. Durch den Zusammenschluss vieler solcher, freiwillig zur Verfügung gestellter Mobiltelefone entsteht eine neue Art von Sensornetzen: die partizipativen Sensornetze. PSNs nutzen die Mobilität der Teilnehmer zur Datenerfassung und ermöglichen so großflächige Beobachtungen im Umfeld der Nutzer.

Mit dem Konzept der PSNs liegt die „perfekte Plattform zur Erfassung der Welt“ bereits in unseren Händen und muss nur entsprechend genutzt werden, vgl. [271]. Bevor dies jedoch geschehen kann, müssen noch einige Herausforderungen überwunden werden, sowohl technischer als auch nicht-technischer Natur. Bei den nicht-technischen Problemen sind die Motivation und Aufklärung der Nutzer zentrale Aspekte. Bei den technischen Problemen sind vor allem zwei Themenfelder essenziell: Zum einen müssen die Energie- und Übertragungskosten auf Seiten der Nutzer möglichst gering ausfallen. Zum anderen muss auch die Privatsphäre der Teilnehmer gewahrt werden. Diese Arbeit begegnet den beschriebenen Herausforderungen, indem sie sowohl Ansätze zur ressourcenschonenden Datenerfassung als auch zur Wahrung der Privatsphäre vorstellt.

Der erste Teil der Arbeit beschäftigte sich mit der Datenerfassung in PSNs. Je nach Anwendungsszenario muss bei der Datenerfassung geklärt werden, welche Knoten Messungen durchführen sollen, welches Phänomen es zu beobachten gilt und wann bzw. wo gemessen werden soll. Man unterscheidet dabei zwei grundsätzliche Anwendungsszenarien: Auf der einen Seite gibt es Anwendungen, bei denen Messungen nur an spezifizierten Orten und zu spezifizierten Zeiten erfasst werden. Ziel bei diesen ist es, punktuell Informationen über das Untersuchungsgebiet zu erhalten. Auf der anderen Seite stehen Anwendungen, bei denen Messungen flächendeckend und langfristig durchgeführt werden. Das Ziel ist hier, möglichst im gesamten Untersuchungsgebiet kontinuierlich Messdaten zu erfassen. Für beide Szenarien wurde jeweils ein Ansatz entwickelt, der die Datenerfassung im Hinblick auf die benötigten Ressourcen verbessert.

Zunächst wurde ein Ansatz für lokal und zeitlich beschränkte Messungen vorgestellt. Das Hauptproblem bei diesem Szenario ist die Kommunikation der Messaufgaben an die teilnehmenden Knoten, wobei vorrangig Ort und Zeitpunkt der gewünschten Messungen von Interesse sind. Zur Lösung dieses Problems wurde

ein Ansatz zur energieeffizienten Verteilung von Messaufgaben präsentiert. Dieser besteht aus drei Komponenten: Erstens wird mit Hilfe der zellbasierten Task-Verteilung die Menge der für einen Nutzer relevanten Aufgaben verkleinert. Zweitens wird die Anzahl der Positionsbestimmungen durch ein adaptives Monitoring mit Bewegungserkennung reduziert. Drittens wird durch eine Bündelung der noch ausstehenden Messaufgaben eine Senkung der Anzahl übertragener Nachrichten erreicht. Durch die Einsparungen bei Positionsbestimmungen und Kommunikation erreicht der vorgestellte Ansatz eine Reduzierung des Energiebedarfs. Gleichzeitig wird jedoch die Effektivität des Systems nicht beeinträchtigt, da der Ansatz sicherstellt, dass Nutzer alle Tasks ihres Umfeldes rechtzeitig erhalten.

Als Zweites wurde mit dem Distributed Multi-Head Clustering ein Ansatz für Szenarien mit flächendeckender und langfristiger Datenerfassung vorgestellt. Ziel des Ansatzes ist es, sowohl die Energie- als auch die Übertragungskosten der Nutzer zu reduzieren und durch die Einsparung von Messungen insgesamt eine weniger redundante Datenerfassung zu erreichen. Hierzu gruppiert DMHC die Messknoten in lokaler Nachbarschaft zu Clustern und verteilt innerhalb dieser verschiedene Aufgaben. Ein Knoten erhält dabei die Rolle des Clusterheads, welcher die Messdaten des Clusters einsammelt und diese gebündelt an den Anwendungsserver überträgt. Eine Teilmenge der verbleibenden Knoten des Clusters wird zu Sensingheads. Letztere sind für die Durchführung der gewünschten Messungen verantwortlich. Die Aufteilung der anfallenden Aufgaben und die intelligente Auswahl der entsprechenden Knoten basieren auf den verfügbaren Energiereserven und den anfallenden Übertragungskosten. Dies führt zu der anvisierten ressourcenschonenden und weniger redundanten Datenerfassung.

Der zweite Teil der Arbeit beschäftigte sich mit dem Schutz der Privatsphäre. Hier wurden zwei Problemfelder identifiziert, die in bisherigen Arbeiten nur unzureichend Berücksichtigung fanden. Zum einen automatisierte Schutzmechanismen, bei denen die Privatsphäre der Teilnehmer gewahrt wird, ohne dass die Nutzer eingreifen und Einstellungen vornehmen müssen. Zum anderen Privatsphäre-Konzepte für Interaktionen der Knoten mit dem Anwendungsserver, die nicht direkt mit der Datenerfassung im Zusammenhang stehen, wie etwa bei Kalibrierungsvorgängen.

Für das erste Problemfeld wurde das Konzept der Silent Zones vorgestellt. Bei diesem werden automatisierte Privatsphäre-Zonen an häufig besuchten Orten erstellt, in denen weder Mess- noch Kontextdaten an den Anwendungsserver übermittelt werden. Die Zonen können dabei ohne Eingreifen des Nutzers erzeugt werden. Basierend auf existierendem Kartenmaterial werden Zonen erstellt, die an die jeweilige Umgebung angepasst sind und garantiert  $k$  Gebäude enthalten. Durch diese Zonen wird eine Häufung von Messungen an wichtigen Orten der Nutzer verhindert. Dies ist wichtig, da eine Identifikation dieser Orte zu einer Deanonymisierung der Nutzer führen kann.

Bei dem zweiten Problemfeld wurde mit PRICAPS der Schutz der Privatsphäre bei den angesprochenen Kalibrierungsvorgängen adressiert. PRICAPS ermöglicht sowohl die Vorwärts- als auch die Rückwärtskalibrierung von Messdaten. Letztere

erlaubt es, dass Messdaten unmittelbar an den Anwendungsserver übertragen und später noch rückwirkend korrigiert werden können. Hierdurch können sowohl Anwendungen realisiert werden, die Echtzeitdaten benötigen, als auch solche, die sehr exakte Daten erfordern. Damit durch die Rückwärtskalibrierung keine Nachverfolgung der Bewegungspfade möglich wird, setzt PRICAPS auf eine Diskretisierung der Kalibrierungsparameter sowie eine zeitlich gebündelte Übertragung der Informationen an den Server mit dem Ziel, durch  $k$ -Anonymität die exakten Pfade der Nutzer zu verschleiern. Der Server verwendet die empfangenen Kalibrierungsparameter, um mit einer gewichteten Korrektur die zuvor erhaltenen Messwerte anzupassen.

Die im Rahmen dieser Arbeit bearbeiteten Fragestellungen bieten Potenzial für weitere Arbeiten. Bei den vorgestellten Ansätzen wurden bereits mögliche Erweiterungen aufgezeigt, wie etwa eine verbesserte Bewegungserkennung oder die Integration von Wegenetzen bei der Task-Verteilung, vgl. Abschnitt 3.2.5, die adaptive Auswahl der Messknoten bei DMHC, vgl. Abschnitt 3.3.4, die Nutzung semantischer Information bei der Erzeugung der Silent Zones, vgl. Abschnitt 4.3.4, oder die zusätzliche Multihop-Kalibrierung bei PRICAPS, vgl. Abschnitt 4.4.5.

Neben diesen Erweiterungsmöglichkeiten für die speziellen Ansätze gibt es zahlreiche allgemeine Herausforderungen. Eine sehr wichtige Thematik ist beispielsweise die Schaffung realistischer und großflächiger Testumgebungen für PSNs. Bisherige Arbeiten prüfen ihre Konzepte größtenteils mit Hilfe von Simulationen, wie in dieser Arbeit geschehen, oder anhand kleiner, prototypischer PSNs. Bei Ersteren kann die Realität nie wirklich vollständig abgebildet werden. Bei Zweiteren treten viele Probleme größerer Netze noch gar nicht auf, z. B. Skalierbarkeit und Heterogenität der Hardware, wodurch eine wirklichkeitsnahe Untersuchung nicht möglich ist. Es gilt also, großflächige Testumgebungen zu schaffen und diese der Forschungsgemeinde zur Verfügung zu stellen, damit neue Konzepte und Lösungen in einem realistischen Umfeld evaluiert werden können. Dies ist vor allem für Ansätze der Datenerfassung wichtig. Auch die in dieser Arbeit vorgestellten Konzepte zur Task-Verteilung und der Gruppierung von Messknoten würden von solchen Umgebungen profitieren. Aber auch für Konzepte zum Schutz der Privatsphäre sind realistische Testumgebungen unerlässlich, um die Probleme und Wünsche der Nutzer diesbezüglich wirklich verstehen und begegnen zu können.

Durch solche, von verschiedenen Forschern gemeinsam genutzte Testumgebungen könnte das Zusammenspiel mehrerer, gleichzeitig ausgeführter Anwendungen für unterschiedliche PSNs analysiert bzw. entsprechende Optimierungen für diese Szenarien entwickelt werden. Will ein Nutzer z. B. gleichzeitig an einem PSN zur Überwachung der Luftqualität, einem weiteren zur Bestimmung der Wettersituation, sowie einem dritten zur Erfassung der Verkehrssituation teilnehmen, so sollte die Datenerfassung dieser drei Anwendungen koordiniert und eine gemeinsame Strategie festgelegt werden.

Neben diesen technischen Herausforderungen müssen auch Konzepte zur Überwindung der nicht-technischen Herausforderungen entworfen werden, vgl. Abschnitt

2.5.1. Zentraler Aspekt ist hier die Aufklärung der Gesellschaft bezüglich partizipativer Sensornetze. Nach den Abhöraktionen im Rahmen der NSA-Affäre [204] stehen viele Bürger dem Erfassen von Messdaten mit Hilfe privater Mobiltelefone kritisch gegenüber. Daher muss die Gesellschaft zunächst über die Möglichkeiten unterrichtet und zur Teilnahme motiviert werden. Aber auch eine Aufklärung bezüglich der Gefahren muss stattfinden. Hierfür sollten geeignete Hilfsmittel entwickelt werden, welche die Nutzer bei der Einschätzung der Gefahren unterstützen.

Die Arbeit hat gezeigt, dass PSNs zahlreiche neue Möglichkeiten bieten können. Vor einem großflächigen Einsatz dieser in der realen Welt müssen allerdings noch einige Herausforderungen überwunden werden. Diese Arbeit und die darin vorgestellten Ansätze liefern hierzu einen Beitrag und sollen dabei helfen, der Etablierung und Nutzung von PSNs in der Praxis ein Stück näher zu kommen.

# Abbildungsverzeichnis

2.1	Klassifizierung von Anwendungen partizipativer Sensornetze . . . . .	14
2.2	Screenshots der Plattform für Umweltbeobachtungen . . . . .	21
2.3	Screenshots der mobilen WRAP-Anwendung . . . . .	22
2.4	Architektur von PSNs . . . . .	23
3.1	Klassifizierung ressourcenschonender Ansätze für WSNs . . . . .	33
3.2	Klassifizierung ressourcenschonender Ansätze für PSNs . . . . .	40
3.3	Akteure und deren Interaktionen innerhalb eines PSNs . . . . .	49
3.4	Zellbasierte Caching-Strategien . . . . .	51
3.5	Beschleunigungsdaten eines Nutzers . . . . .	52
3.6	Schematische Darstellung der Task-Bündelung . . . . .	54
3.7	Anzahl der Positionsbestimmungen pro Knoten . . . . .	57
3.8	Anzahl der Kommunikationsverbindungen . . . . .	59
3.9	Anteil der gebündelt versendeten Tasks . . . . .	60
3.10	Schematische Darstellung eines Clusters . . . . .	63
3.11	Visualisierung der Simulationsumgebung mit vergrößertem Cluster . . . . .	68
3.12	Analyse der Mobilfunk-Übertragungen und Kosten . . . . .	71
3.13	Energiekosten für zwei Szenarien . . . . .	72
3.14	Anzahl der Messungen pro Knoten . . . . .	73
3.15	Ad-hoc-Overhead zur Bestimmung der Messknoten . . . . .	74
3.16	Wertebasierte Anpassung der Anzahl an Messknoten . . . . .	77
3.17	Drei Messknoten mit überlappender Abdeckung . . . . .	78
4.1	Anhäufung von Messpunkten um den Arbeitsort des Nutzers . . . . .	95
4.2	Konstruktion einer Ban Zone . . . . .	96
4.3	Architektur des Silent-Zone-Systems . . . . .	98
4.4	Gebäudeextraktion aus Kartendaten . . . . .	99
4.5	Identifikation wichtiger Orte . . . . .	100
4.6	Vergrößerungsfaktor für Random-Rect-Varianten . . . . .	101
4.7	Schematische Darstellung der Gebäude in $S$ und in der Zone . . . . .	104
4.8	Zufällig bestimmte Positionen wichtiger Orte . . . . .	105
4.9	Gebäudeanzahl über den geforderten $k$ Gebäuden . . . . .	106
4.10	Durchschnittliche Zonengröße der Silent Zones . . . . .	108
4.11	Benötigte Rundenzahl der Random-Rect-Varianten . . . . .	109
4.12	Gebäudeanzahl pro Zone für Ban Zones und Silent Zones . . . . .	110
4.13	Zonengröße der Ban Zones und Silent Zones . . . . .	110
4.14	Exakte Parameter ermöglichen eine Nachverfolgung der Nutzerpfade	115

---

4.15	Systematische und zufällige Fehler bei Messungen . . . . .	116
4.16	Systematische Messfehler . . . . .	117
4.17	Die Kalibrierungs-Pipeline zeigt die zwei Kalibrierungsphasen . . .	122
4.18	Parameter werden bis zur nächsten Kalibrierung verwendet . . . . .	123
4.19	Auszug von unkalibrierten und kalibrierten Ozonmessungen . . . . .	124
4.20	Die Gewichtung wird anhand der Upload-Zeitpunkte bestimmt . . . . .	128
4.21	Erreichtes $k$ -Anonymitätslevel bei 1 000 Knoten . . . . .	133
4.22	Erreichtes $k$ -Anonymitätslevel bei variierender Knotenzahl . . . . .	134
4.23	Durchschnittlicher Genauigkeitsgewinn/-verlust . . . . .	135
4.24	Genauigkeitsgewinn für Vorwärts- und Rückwärtskalibrierung . . .	136
4.25	Beispiele möglicher Einschränkungen des Konzeptes . . . . .	138

# Literaturverzeichnis

- [1] AANENSEN, David M.; HUNTLEY, Derek M.; FEIL, Edward J.; SPRATT, Brian G.: EpiCollect: Linking Smartphones to Web Applications for Epidemiology, Ecology and Community Data Collection. In: *PloS ONE* 4 (2009), September, Nr. 9. – DOI 10.1371/journal.pone.0006968. – ISSN 1932–6203, S. 1–7
- [2] ABBASI, Ameer A.; YOUNIS, Mohamed: A survey on clustering algorithms for wireless sensor networks. In: *Computer Communications* 30 (2007), Oktober, Nr. 14–15. – DOI 10.1016/j.comcom.2007.05.024. – ISSN 0140–3664, S. 2826–2841
- [3] ABDELZAHER, Tarek; ANOKWA, Yaw; BODA, Peter; BURKE, Jeff; ESTRIN, Deborah; GUIBAS, Leonidas; KANSAL, Aman; MADDEN, Samuel; REICH, Jim: Mobiscopes for Human Spaces. In: *IEEE Pervasive Computing* 6 (2007), April–Juni, Nr. 2. – DOI 10.1109/MPRV.2007.38. – ISSN 1536–1268, S. 20–29
- [4] AGRAWAL, Dakshi; AGGARWAL, Charu C.: On the Design and Quantification of Privacy Preserving Data Mining Algorithms. In: *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '01)*. New York, NY, USA : ACM, 2001. – DOI 10.1145/375551.375602. – ISBN 1–58113–361–8, S. 247–255
- [5] AGRAWAL, Rakesh; SRIKANT, Ramakrishnan: Privacy-preserving Data Mining. In: *ACM SIGMOD Record* 29 (2000), Mai, Nr. 2. – DOI 10.1145/335191.335438. – ISSN 0163–5808, S. 439–450
- [6] AHN, Gahng-Seop; MUSOLESI, Mirco; LU, Hong; OLFATI-SABER, Reza; CAMPBELL, Andrew T.: MetroTrack: Predictive Tracking of Mobile Events Using Mobile Phones. In: RAJARAMAN, Rajmohan (Hrsg.); MOSCIBRODA, Thomas (Hrsg.); DUNKELS, Adam (Hrsg.); SCAGLIONE, Anna (Hrsg.): *Distributed Computing in Sensor Systems*. Bd. 6131. Berlin / Heidelberg : Springer, 2010. – DOI 10.1007/978–3–642–13651–1\_17. – ISBN 978–3–642–13650–4, S. 230–243
- [7] AKKAYA, Kemal; YOUNIS, Mohamed: A survey on routing protocols for wireless sensor networks. In: *Ad Hoc Networks* 3 (2005), Mai, Nr. 3. – DOI 10.1016/j.adhoc.2003.09.010. – ISSN 1570–8705, S. 325–349

- [8] AKYILDIZ, Ian F.; MELODIA, Tommaso; CHOWDHURY, Kaushik R.: A survey on wireless multimedia sensor networks. In: *Computer Networks* 51 (2007), März, Nr. 4. – DOI 10.1016/j.comnet.2006.10.002. – ISSN 1389–1286, S. 921–960
- [9] AKYILDIZ, Ian F.; SU, Weilian; SANKARASUBRAMANIAM, Yogesh; CA-YIRCI, Erdal: Wireless sensor networks: a survey . In: *Computer Networks* 38 (2002), März, Nr. 4. – DOI 10.1016/S1389–1286(01)00302–4. – ISSN 1389–1286, S. 393–422
- [10] AKYILDIZ, Ian F.; VURAN, Mehmet C.: *Wireless Sensor Networks*. 1. Auflage. Chichester, England : John Wiley & Sons, 2010 (Ian F. Akyildiz Series in Communications and Networking). – 516 S. – ISBN 978–0–470–03601–3
- [11] ALIPPI, Cesare; ANASTASI, Giuseppe; GALPERTI, Cristian; MANCINI, Francesca; ROVERI, Manuel: Adaptive Sampling for Energy Conservation in Wireless Sensor Networks for Snow Monitoring Applications. In: *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS 2007)*. Pisa, Italy, Oktober 2007. – DOI 10.1109/MOB-HOC.2007.4428700, S. 1–6
- [12] ALTMAN, Irwin: *The environment and social behavior - Privacy, personal space, territory, crowding*. 4. Auflage. Monterey, CA, USA : Brooks/Cole, 1975. – 256 S. – ISBN 0–8185–0168–5
- [13] ANASTASI, Giuseppe; CONTI, Marco; FRANCESCO, Mario D.; PASSARELLA, Andrea: Energy conservation in wireless sensor networks: A survey . In: *Ad Hoc Networks* 7 (2009), Mai, Nr. 3. – DOI 10.1016/j.adhoc.2008.06.003. – ISSN 1570–8705, S. 537–568
- [14] ANNAVARAM, Murali; MEDVIDOVIC, Nenad; MITRA, Urbashi; NARAYANAN, Shrikanth; SUKHATME, Gaurav; MENG, Zhaoshi; QIU, Shi; KUMAR, Rakesh; THATTE, Gautam; SPRUIJT-METZ, Donna: Multimodal Sensing for Pediatric Obesity Applications. In: *Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense '08)*. Raleigh, North Carolina, USA, November 2008, S. 21–25
- [15] ARAMPATZIS, Th; LYGEROS, John; MANESIS, Stamatis: A Survey of Applications of Wireless Sensors and Wireless Sensor Networks. In: *Proceedings of the 13th Mediterranean Conference on Control and Automation (MED '05)*. New York, NY, USA : IEEE, Juni 2005. – DOI 10.1109/.2005.1467103. – ISSN 2158–9860, S. 719–724
- [16] ASHBROOK, Daniel; STARNER, Thad: Using GPS to learn significant locations and predict movement across multiple users. In: *Personal and Ubiquitous Computing* 7 (2003), Oktober, Nr. 5. – DOI 10.1007/s00779–003–0240–0. – ISSN 1617–4909, S. 275–286

- [17] BAIER, Patrick; DÜRR, Frank; ROTHERMEL, Kurt: Efficient Distribution of Sensing Queries in Public Sensing Systems. In: *Proceedings of the IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2013)*, IEEE, Oktober 2013. – DOI 10.1109/MASS.2013.11, S. 272–280
- [18] BAIER, Patrick; DÜRR, Frank; ROTHERMEL, Kurt: PSense: Reducing Energy Consumption in Public Sensing Systems. In: *Proceedings of the IEEE 26th International Conference on Advanced Information Networking and Applications (AINA 2012)*, 2012. – DOI 10.1109/AINA.2012.33. – ISSN 1550–445X, S. 136–143
- [19] BALASUBRAMANIAN, Niranjan; BALASUBRAMANIAN, Aruna; VENKATARAMANI, Arun: Energy consumption in mobile phones: a measurement study and implications for network applications. In: *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC '09)*. New York, NY, USA : ACM, 2009. – ISBN 978–1–60558–771–4, S. 280–293
- [20] BALES, Elizabeth; NIKZAD, Nima; QUICK, Nichole; ZIFTCI, Celal; PATRICK, Kevin; GRISWOLD, William: Citisense: Mobile Air Quality Sensing for Individuals and Communities. Design and deployment of the Citisense mobile air-quality system. In: *Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth 2012)*. New York, NY, USA : IEEE, Mai 2012. – ISBN 978–1–4673–1483–1, S. 155–158
- [21] BARETH, Ulrich; KÜPPER, Axel: Energy-Efficient Position Tracking in Proactive Location-Based Services for Smartphone Environments. In: *IEEE 35th Annual Computer Software and Applications Conference (COMPSAC 2011)*, 2011. – DOI 10.1109/COMPSAC.2011.72. – ISSN 0730–3157, S. 516–521
- [22] BARR, Kenneth C.; ASANOVIĆ, Krste: Energy-aware Lossless Data Compression. In: *ACM Transactions on Computer Systems (TOCS)* 24 (2006), August, Nr. 3. – DOI 10.1145/1151690.1151692. – ISSN 0734–2071, S. 250–291
- [23] BASAGNI, Stefano: Distributed Clustering for Ad Hoc Networks. In: *Proceedings of the Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN '99)*. Perth/Fremantle, WA, Australia, Juni 1999. – DOI 10.1109/ISPAN.1999.778957. – ISSN 1087–4089, S. 310–315
- [24] BASAGNI, Stefano; CAROSI, Alessio; MELACHRINOUDIS, Emanuel; PETRIOLI, Chiara; WANG, Z. M.: Controlled Sink Mobility for Prolonging Wireless Sensor Networks Lifetime. In: *Wireless Networks* 14 (2008), Dezember, Nr. 6. – DOI 10.1007/s11276–007–0017–x. – ISSN 1022–0038, S. 831–858

- [25] BASSEVILLE, Michèle; NIKIFOROV, Igor V.: *Detection of Abrupt Changes: Theory and Application*. 1. Auflage. Upper Saddle River, NJ, USA : Prentice Hall, 1993 (Prentice Hall Information and System Sciences Series). – 528 S. – ISBN 0–13–126780–9
- [26] BAYERISCHES STAATSMINISTERIUM FÜR LANDESENTWICKLUNG UND UMWELTFRAGEN: *Grundinformation zur neuen Mobilfunkgeneration UMTS*. 1. Auflage. München : StMLU, 2003 (Fachinformation Umwelt & Entwicklung Bayern). – 28 S.
- [27] BERESFORD, Alastair R.; STAJANO, Frank: Location Privacy in Pervasive Computing. In: *IEEE Pervasive Computing* 2 (2003), Januar, Nr. 1. – DOI 10.1109/MPRV.2003.1186725. – ISSN 1536–1268, S. 46–55
- [28] BIEBER, Gerald; VOSKAMP, Jörg; URBAN, Bodo: Activity Recognition for Everyday Life on Mobile Phones. In: STEPHANIDIS, Constantine (Hrsg.): *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*. Bd. 5615. Springer Berlin Heidelberg, 2009. – DOI 10.1007/978–3–642–02710–9\_32. – ISBN 978–3–642–02709–3, S. 289–296
- [29] BILANDZIC, Mark; BANHOLZER, Michael; PEEV, Deyan; GEORGIEV, Vesko; BALAGTAS-FERNANDEZ, Florence; DE LUCA, Alexander: Laermometer: A Mobile Noise Mapping Application. In: *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges (NordiCHI '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1463160.1463208. – ISBN 978–1–59593–704–9, S. 415–418
- [30] BOUCSEIN, Wolfram: *Elektrodermale Aktivität: Grundlagen, Methoden und Anwendungen*. 1. Auflage. Berlin u.a. : Springer, 1988. – 546 S. – ISBN 3–540–18586–0
- [31] BRÉMAUD, Pierre: *Texts in Applied Mathematics*. Bd. 31: *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. 1. Auflage. New York, NY, USA : Springer, 1999. – 445 S. – ISBN 978–1–4419–3131–3
- [32] BROLL, Gregor; CAO, Hu; EBBEN, Peter; HOLLEIS, Paul; JACOBS, Koen; KOOLWAAIJ, Johan; LUTHER, Marko; SOUVILLE, Bertrand: Tripzoom: An App to Improve Your Mobility Behavior. In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. New York, NY, USA : ACM, 2012 (MUM '12). – DOI 10.1145/2406367.2406436. – ISBN 978–1–4503–1815–0, S. 57:1–57:4
- [33] BURGOON, Michael: Privacy and Communication. In: BURGOON, Michael (Hrsg.): *Communication Yearbook*. Bd. 6. 1. Auflage. Beverly Hills, CA, USA : SAGE Publications, 1982. – ISBN 978–0–8039–1862–7, S. 206–249

- [34] BURKE, Jeffrey A.; ESTRIN, Deborah; HANSEN, Mark; PARKER, Andrew; RAMANATHAN, Nithya; REDDY, Sasank; SRIVASTAVA, Mani B.: Participatory Sensing. In: *Proceedings of the First Workshop on World-Sensor-Web (WSW'06) at the 4th ACM Conference on Embedded Networked Sensor System (SenSys '06)*. Boulder, CO, USA, Oktober 2006, S. 1–5
- [35] BYCHKOVSKIY, Vladimir; MEGERIAN, Seapahn; ESTRIN, Deborah; POTKONJAK, Miodrag: A Collaborative Approach to In-Place Sensor Calibration. In: ZHAO, Feng (Hrsg.); GUIBAS, Leonidas (Hrsg.): *Information Processing in Sensor Networks*. Bd. 2634. Berlin / Heidelberg : Springer, 2003. – DOI 10.1007/3-540-36978-3\_20. – ISBN 978-3-540-02111-7, S. 301–316
- [36] CAMENISCH, Jan; STADLER, Markus: Efficient Group Signature Schemes for Large Groups. In: KALISKI, Jr. Burton S. (Hrsg.): *Advances in Cryptology – CRYPTO '97*. Bd. 1294. Berlin / Heidelberg : Springer, 1997. – DOI 10.1007/BFb0052252. – ISBN 978-3-540-63384-6, S. 410–424
- [37] CAMPBELL, Andrew T.; EISENMAN, Shane B.; LANE, Nicholas D.; MILUZZO, Emiliano; PETERSON, Ronald A.: People-centric Urban Sensing. In: *Proceedings of the 2nd Annual International Workshop on Wireless Internet (WICON '06)*. New York, NY, USA : ACM, August 2006. – DOI 10.1145/1234161.1234179. – ISBN 1-59593-510-X, S. 1–14
- [38] CAMPBELL, Andrew T.; EISENMAN, Shane B.; LANE, Nicholas D.; MILUZZO, Emiliano; PETERSON, Ronald A.; LU, Hong; ZHENG, Xiao; MUSOLESI, Mirco; FODOR, Kristóf; AHN, Gahng-Seop: The Rise of People-Centric Sensing. In: *IEEE Internet Computing* 12 (2008), Juli/August, Nr. 4. – DOI 10.1109/MIC.2008.90. – ISSN 1089-7801, S. 12–21
- [39] CARDONE, Giuseppe; FOSCHINI, Luca; BELLAVISTA, Paolo; CORRADI, Antonio; BORCEA, Cristian; TALASILA, Manoop; CURTMOLA, Reza: Fostering ParticipAction in Smart Cities: A Geo-Social Crowdsensing Platform. In: *IEEE Communications Magazine* 51 (2013), Juni, Nr. 6. – DOI 10.1109/MCOM.2013.6525603. – ISSN 0163-6804, S. 112–119
- [40] CASALE, Pierluigi; PUJOL, Oriol; RADEVA, Petia: Personalization and User Verification in Wearable Systems Using Biometric Walking Patterns. In: *Personal and Ubiquitous Computing* 16 (2012), Juni, Nr. 5. – DOI 10.1007/s00779-011-0415-z. – ISSN 1617-4909, S. 563–580
- [41] CHAUM, David L.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In: *Communications of the ACM* 24 (1981), Februar, Nr. 2. – DOI 10.1145/358549.358563, S. 84–90

- [42] CHEN, Benjie; JAMIESON, Kyle; BALAKRISHNAN, Hari; MORRIS, Robert: Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In: *Wireless Networks* 8 (2002), September, Nr. 5. – DOI 10.1023/A:1016542229220. – ISSN 1022–0038, S. 481–494
- [43] CHEN, Xiang; HO, Chee T.; LIM, Eng T.; KYAW, T.Z.: Cellular Phone Based Online ECG Processing for Ambulatory and Continuous Detection. In: *Proceedings of the Computers in Cardiology Conference (CinC 2007)*. Bd. 34. New York, NY, USA : IEEE, September 2007. – DOI 10.1109/CIC.2007.4745570. – ISSN 0276–6547, S. 653–656
- [44] CHEN, Yuequan; CHAN, Edward; HAN, Song: Energy Efficient Multipath Routing in Large Scale Sensor Networks with Multiple Sink Nodes. In: CAO, Jiannong (Hrsg.); NEJDL, Wolfgang (Hrsg.); XU, Ming (Hrsg.): *Advanced Parallel Processing Technologies*. Bd. 3756. Berlin / Heidelberg : Springer, 2005. – DOI 10.1007/11573937\_42. – ISBN 978–3–540–29639–3, S. 390–399
- [45] CHOUDHURY, Tanzeem; BORRIELLO, Gaetano; CONSOLVO, Sunny; HAEHNEL, Dirk; HARRISON, Beverly; HEMINGWAY, Bruce; HIGHTOWER, Jeffrey; KLASNJA, Predrag; KOSCHER, Karl; LAMARCA, Anthony; LANDAY, James A.; LEGRAND, Louis; LESTER, Jonathan; RAHIMI, Ali; REA, Adam; WYATT, Danny: The Mobile Sensing Platform: An Embedded Activity Recognition System. In: *IEEE Pervasive Computing* 7 (2008), April–Juni, Nr. 2. – DOI 10.1109/MPRV.2008.39. – ISSN 1536–1268, S. 32–41
- [46] CHRISTIN, Delphine; REINHARDT, Andreas; KANHERE, Salil S.; HOLLICK, Matthias: A Survey on Privacy in Mobile Participatory Sensing Applications. In: *Journal of Systems and Software* 84 (2011), November, Nr. 11. – DOI 10.1016/j.jss.2011.06.073. – ISSN 0164–1212, S. 1928–1946
- [47] CHU, David; DESHPANDE, Amol; HELLERSTEIN, Joseph M.; HONG, Wei: Approximate Data Collection in Sensor Networks using Probabilistic Models. In: *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, 2006. – DOI 10.1109/ICDE.2006.21, S. 48–48
- [48] CLAUS, Volker; SCHWILL, Andreas: *Duden Informatik A - Z: Fachlexikon für Studium, Ausbildung und Beruf*. 4., überarbeitete und aktualisierte Auflage. Berlin : Bibliographisches Institut, 2006. – 768 S. – ISBN 3–41–105234–1

- [49] CONSOLVO, Sunny; McDONALD, David W.; TOSCOS, Tammy; CHEN, Mike Y.; FROEHLICH, Jon; HARRISON, Beverly; KLASNJA, Predrag; LAMARCA, Anthony; LEGRAND, Louis; LIBBY, Ryan; SMITH, Ian; LANDAY, James A.: Activity Sensing in the Wild: A Field Trial of Ubifit Garden. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1357054.1357335. – ISBN 978-1-60558-011-1, S. 1797–1806
- [50] CONSTANDACHE, Ionut; GAONKAR, Shravan; SAYLER, Matt; CHOUDHURY, Romit R.; COX, Landon: EnLoc: Energy-Efficient Localization for Mobile Phones. In: *Proceedings of the IEEE INFOCOM 2009*, 2009. – DOI 10.1109/INFCOM.2009.5062218. – ISSN 0743-166X, S. 2716–2720
- [51] CUERVO, Eduardo; BALASUBRAMANIAN, Aruna; CHO, Dae-ki; WOLMAN, Alec; SAROIU, Stefan; CHANDRA, Ranveer; BAHL, Paramvir: MAUI: Making Smartphones Last Longer with Code Offload. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1814433.1814441. – ISBN 978-1-60558-985-5, S. 49–62
- [52] D'ROZA, Thomas; BILCHEV, George: An Overview of Location-Based Services. In: *BT Technology Journal* 21 (2003), Januar, Nr. 1. – DOI 10.1023/A:1022491825047. – ISSN 1358-3948, S. 20–27
- [53] DARGIE, Walteneagus; POELLABAUER, Christian: *Fundamentals of Wireless Sensor Networks: Theory and Practice*. 1. Auflage. Chichester, England : John Wiley & Sons, 2010 (Wiley Series on Wireless Communications and Mobile Computing). – 336 S. – ISBN 978-0-470-99765-9
- [54] DAS, Tathagata; MOHAN, Prashanth; PADMANABHAN, Venkata N.; RAMJEE, Ramachandran; SHARMA, Asankhaya: PRISM: platform for remote sensing using smartphones. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1814433.1814442. – ISBN 978-1-60558-985-5, S. 63–76
- [55] DE CRISTOFARO, Emiliano; SORIENTE, Claudio: Participatory privacy: Enabling privacy in participatory sensing. In: *IEEE Network* 27 (2013), Januar, Nr. 1. – DOI 10.1109/MNET.2013.6423189. – ISSN 0890-8044, S. 32–36
- [56] DECKER, Michael: Location Privacy-An Overview. In: *Proceedings of the 7th International Conference on Mobile Business (ICMB '08)*, 2008. – DOI 10.1109/ICMB.2008.14, S. 221–230

- [57] DENG, Linda; COX, Landon P.: LiveCompare: Grocery Bargain Hunting Through Participatory Sensing. In: *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications (HotMobile '09)*. New York, NY, USA : ACM, Februar 2009. – DOI 10.1145/1514411.1514415. – ISBN 978-1-60558-283-2, S. 4:1-4:6
- [58] DENNING, Tamara; ANDREW, Adrienne; CHAUDHRI, Rohit; HARTUNG, Carl; LESTER, Jonathan; BORRIELLO, Gaetano; DUNCAN, Glen: BALANCE: Towards a Usable Pervasive Wellness Application with Accurate Activity Inference. In: *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications (HotMobile '09)*. New York, NY, USA : ACM, Februar 2009. – DOI 10.1145/1514411.1514416. – ISBN 978-1-60558-283-2, S. 5:1-5:6
- [59] DERAWI, Mohammad O.; NICKEL, Claudia; BOURS, Partick; BUSCH, Christoph: Unobtrusive User-Authentication on Mobile Phones Using Biometric Gait Recognition. In: *Proceedings of the 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*. Darmstadt, Germany, Oct 2010. – DOI 10.1109/IIHMSP.2010.83, S. 306-311
- [60] DESHPANDE, Amol; GUESTRIN, Carlos; MADDEN, Samuel R.; HELLERSTEIN, Joseph M.; HONG, Wei: Model-driven Data Acquisition in Sensor Networks. In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB '04) - Volume 30*, VLDB Endowment, 2004. – ISBN 0-12-088469-0, S. 588-599
- [61] DEUTSCHES INSTITUT FÜR NORMUNG E. V.: DIN 1319-1:1995-01. Grundlagen der Meßtechnik - Teil 1: Grundbegriffe. Berlin : Beuth, Januar 1995 (<http://www.beuth.de/de/norm/din-1319-1/2440447>). – DIN-Norm
- [62] DINGLEDINE, Roger; MATHEWSON, Nick; SYVERSON, Paul: Tor: The Second-Generation Onion Router. In: *Proceedings of the 13th USENIX Security Symposium (Security '04)*. San Diego, CA, USA, August 2004, S. 303-320
- [63] DONG, Yi F.; KANHERE, Salil; CHOU, Chun T.; BULUSU, Nirupama: Automatic Collection of Fuel Prices from a Network of Mobile Cameras. In: NIKOLETSEAS, Sotiris E. (Hrsg.); CHLEBUS, Bogdan S. (Hrsg.); JOHNSON, David B. (Hrsg.); KRISHNAMACHARI, Bhaskar (Hrsg.): *Distributed Computing in Sensor Systems*. Bd. 5067. Berlin / Heidelberg : Springer, 2008. – DOI 10.1007/978-3-540-69170-9\_10. – ISBN 978-3-540-69169-3, S. 140-156

- [64] DUCKHAM, Matt; KULIK, Lars: Location privacy and location-aware computing. In: DRUMMOND, Jane (Hrsg.); BILLEN, Roland (Hrsg.); JOÃO, Elsa (Hrsg.); FORREST, David (Hrsg.): *Dynamic & Mobile GIS: Investigating Change in Space and Time*. Boca Raton, FL : CRC Press, 2006. – ISBN 0-8493-9092-3, S. 34–51
- [65] DUCKHAM, Matt; KULIK, Lars: A Formal Model of Obfuscation and Negotiation for Location Privacy. In: GELLERSEN, Hans-W. (Hrsg.); WANT, Roy (Hrsg.); SCHMIDT, Albrecht (Hrsg.): *Pervasive Computing*. Bd. 3468. Berlin / Heidelberg : Springer, 2005. – DOI 10.1007/11428572\_10. – ISBN 978-3-540-26008-0, S. 152–170
- [66] DÜRR, Michael: *Sicherheit und Privatsphäre in Online Sozialen Netzwerken*. München, Ludwig-Maximilians-Universität München, Fakultät für Mathematik, Informatik und Statistik, Dissertation, November 2013
- [67] EISENMAN, Shane B.; CAMPBELL, Andrew T.: SkiScape Sensing. In: *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*. New York, NY, USA : ACM, 2006. – DOI 10.1145/1182807.1182876. – ISBN 1-59593-343-3, S. 401–402
- [68] EISENMAN, Shane B.; LANE, Nicholas D.; CAMPBELL, Andrew T.: Techniques for Improving Opportunistic Sensor Networking Performance. In: NIKOLETSEAS, Sotiris E. (Hrsg.); CHLEBUS, Bogdan S. (Hrsg.); JOHNSON, David B. (Hrsg.); KRISHNAMACHARI, Bhaskar (Hrsg.): *Distributed Computing in Sensor Systems*. Bd. 5067. Berlin / Heidelberg : Springer, 2008. – DOI 10.1007/978-3-540-69170-9\_11. – ISBN 978-3-540-69169-3, S. 157–175
- [69] EISENMAN, Shane B.; LANE, Nicholas D.; MILUZZO, Emiliano; PETERSON, Ronald A.; AHN, Gahng-Seop; CAMPBELL, Andrew T.: MetroSense Project: People-Centric Sensing at Scale. In: *Proceedings of the First Workshop on World-Sensor-Web (WSW'06) at the 4th ACM Conference on Embedded Networked Sensor System (SenSys '06)*. Boulder, CO, USA, 2006, S. 1–6
- [70] EISENMAN, Shane B.; MILUZZO, Emiliano; LANE, Nicholas D.; PETERSON, Ronald A.; AHN, Gahng-Seop; CAMPBELL, Andrew T.: BikeNet: A Mobile Sensing System for Cyclist Experience Mapping. In: *ACM Transactions on Sensor Networks* 6 (2010), Januar, Nr. 1. – DOI 10.1145/1653760.1653766. – ISSN 1550-4859, S. 6:1–6:39
- [71] ELSON, Jeremy E.; ESTRIN, Deborah: Time synchronization for wireless sensor networks. In: *Proceedings of the 15th International Parallel and Distributed Processing Symposium*. San Francisco, CA, USA, April 2000. – DOI 10.1109/IPDPS.2001.925191. – ISSN 1530-2075, S. 1965–1970

- [72] ERIKSSON, Jakob; GIROD, Lewis; HULL, Bret; NEWTON, Ryan; MADDEN, Samuel; BALAKRISHNAN, Hari: The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. In: *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1378600.1378605. – ISBN 978-1-60558-139-2, S. 29–39
- [73] EVFIMIEVSKI, Alexandre: Randomization in Privacy Preserving Data Mining. In: *ACM SIGKDD Explorations Newsletter* 4 (2002), Dezember, Nr. 2. – DOI 10.1145/772862.772869. – ISSN 1931-0145, S. 43–48
- [74] FASOLO, Elena; ROSSI, Michele; WIDMER, Jorg; ZORZI, Michele: In-network aggregation techniques for wireless sensor networks: a survey. In: *IEEE Wireless Communications* 14 (2007), April, Nr. 2. – DOI 10.1109/MWC.2007.358967. – ISSN 1536-1284, S. 70–87
- [75] FERREIRA, Heitor; DUARTE, Sérgio; PREGUIÇA, Nuno: 4Sensing – Decentralized Processing for Participatory Sensing Data. In: *IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS 2010)*. New York, NY : IEEE, Dezember 2010. – DOI 10.1109/ICPADS.2010.20. – ISSN 1521-9097, S. 306–313
- [76] FERREIRA, Michel; FERNANDES, Ricardo; CONCEIÇÃO, Hugo; GOMES, Pedro; D'OREY, Pedro; MOREIRA-MATIAS, Luís; GAMA, João; LIMA, Fernanda; DAMAS, Luís: Vehicular Sensing: Emergence of a Massive Urban Scanner. In: MARTINS, Francisco (Hrsg.); LOPES, Luís (Hrsg.); PAULINO, Hervé (Hrsg.): *Sensor Systems and Software*. Bd. 102. Berlin / Heidelberg : Springer, 2012. – DOI 10.1007/978-3-642-32778-0\_1. – ISBN 978-3-642-32777-3, S. 1–14
- [77] GANDHAM, Shashidhar R.; DAWANDE, Milind; PRAKASH, Ravi; VENKATESAN, Subbarayan: Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In: *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*. Bd. 1, IEEE, Dezember 2003. – DOI 10.1109/GLOCOM.2003.1258265, S. 377–381
- [78] GANTI, Raghu K.; PHAM, Nam; AHMADI, Hossein; NANGIA, Saurabh; ABDELZAHER, Tarek F.: GreenGPS: A Participatory Sensing Fuel-efficient Maps Application. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1814433.1814450. – ISBN 978-1-60558-985-5, S. 151–164
- [79] GANTI, Raghu K.; PHAM, Nam; TSAI, Yu-En; ABDELZAHER, Tarek F.: PoolView: Stream Privacy for Grassroots Participatory Sensing. In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1460412.1460440. – ISBN 978-1-59593-990-6, S. 281–294

- [80] GANTI, Raghu K.; YE, Fan; LEI, Hui: Mobile Crowdsensing: Current State and Future Challenges. In: *IEEE Communications Magazine* 49 (2011), November, Nr. 11. – DOI 10.1109/MCOM.2011.6069707. – ISSN 0163–6804, S. 32–39
- [81] GAO, Chunming; KONG, Fanyu; TAN, Jindong: HealthAware: Tackling Obesity with Health Aware Smart Phone Systems. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO 2009)*. New York, NY : IEEE, Dezember 2009. – DOI 10.1109/ROBIO.2009.5420399. – ISBN 978–1–4244–4774–9, S. 1549–1554
- [82] GAONKAR, Shravan; LI, Jack; CHOUDHURY, Romit R.; COX, Landon; SCHMIDT, Al: Micro-Blog: Sharing and Querying Content Through Mobile Phones and Social Participation. In: *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1378600.1378620. – ISBN 978–1–60558–139–2, S. 174–186
- [83] GHOSH, Amitabha; DAS, Sajal K.: Coverage and connectivity issues in wireless sensor networks: A survey. In: *Pervasive and Mobile Computing* 4 (2008), Juni, Nr. 3. – DOI 10.1016/j.pmcj.2008.02.001. – ISSN 1574–1192, S. 303–334
- [84] GOEL, Samir; IMIELINSKI, Tomasz: Prediction-based Monitoring in Sensor Networks: Taking Lessons from MPEG. In: *ACM SIGCOMM Computer Communication Review* 31 (2001), Oktober, Nr. 5. – DOI 10.1145/1037107.1037117. – ISSN 0146–4833, S. 82–98
- [85] GOLLE, Philippe; PARTRIDGE, Kurt: On the Anonymity of Home/Work Location Pairs. In: TOKUDA, Hideyuki (Hrsg.); BEIGL, Michael (Hrsg.); FRIDAY, Adrian (Hrsg.); BRUSH, A.J.Bernheim (Hrsg.); TOBE, Yoshito (Hrsg.): *Pervasive Computing*. Bd. 5538. Berlin / Heidelberg : Springer, 2009. – DOI 10.1007/978–3–642–01516–8\_26. – ISBN 978–3–642–01515–1, S. 390–397
- [86] GRÄNICHER, W.H. H.: *Messung beendet-was nun?: Einführung und Nachschlagewerk für die Planung und Auswertung von Messungen*. 2., überarbeitete Auflage. Zürich, Schweiz : vdf Hochschulverlag AG an der ETH Zürich, 1996. – 156 S. – ISBN 3–7281–2258–0
- [87] GRUTESER, Marco; GRUNWALD, Dirk: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In: *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys '03)*. New York, NY, USA : ACM, 2003. – DOI 10.1145/1066116.1189037, S. 31–42

- [88] HASENFRATZ, David; SAUKH, Olga; STURZENEGGER, Silvan; THIELE, Lothar: Participatory Air Pollution Monitoring Using Smartphones. In: *Proceedings of the 2nd International Workshop on Mobile Sensing ("Mobile Sensing: From Smartphones and Wearables to Big Data") at the 11th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN 2012)*. Peking, China, April 2012, S. 1–5
- [89] HASENFRATZ, David; SAUKH, Olga; THIELE, Lothar: On-the-Fly Calibration of Low-Cost Gas Sensors. In: PICCO, Gian (Hrsg.); HEINZELMAN, Wendi (Hrsg.): *Wireless Sensor Networks*. Bd. 7158. Berlin / Heidelberg : Springer, 2012. – DOI 10.1007/978-3-642-28169-3\_15. – ISBN 978-3-642-28168-6, S. 228–244
- [90] HAYES, G.R.; SHEHAN POOLE, E.; IACHELLO, G.; PATEL, S.N.; GRIMES, A.; ABOWD, G.D.; TRUONG, K.N.: Physical, Social, and Experiential Knowledge in Pervasive Computing Environments. In: *IEEE Pervasive Computing* 6 (2007), Oktober, Nr. 4. – DOI 10.1109/MPRV.2007.82. – ISSN 1536–1268, S. 56–63
- [91] HERCOG, Darko; GERGIČ, Bojan: A Flexible Microcontroller-Based Data Acquisition Device. In: *Sensors* 14 (2014), Juni, Nr. 6. – DOI 10.3390/s140609755. – ISSN 1424–8220, S. 9755–9775
- [92] HJELMÅS, Erik; LOW, Boon K.: Face Detection: A Survey. In: *Computer Vision and Image Understanding* 83 (2001), September, Nr. 3. – DOI 10.1006/cviu.2001.0921. – ISSN 1077–3142, S. 236–274
- [93] HOH, Baik; GRUTESER, Marco: Protecting Location Privacy Through Path Confusion. In: *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm 2005)*, 2005. – DOI 10.1109/SECURECOMM.2005.33, S. 194–205
- [94] HOH, Baik; GRUTESER, Marco; HERRING, Ryan; BAN, Jeff; WORK, Daniel; HERRERA, Juan-Carlos; BAYEN, Alexandre M.; ANNAVARAM, Murali; JACOBSON, Quinn: Virtual Trip Lines for Distributed Privacy-preserving Traffic Monitoring. In: *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1378600.1378604. – ISBN 978-1-60558-139-2, S. 15–28
- [95] HOH, Baik; GRUTESER, Marco; XIONG, Hui; ALRABADY, Ansaf: Preserving Privacy in Gps Traces via Uncertainty-aware Path Cloaking. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*. New York, NY, USA : ACM, 2007. – DOI 10.1145/1315245.1315266. – ISBN 978-1-59593-703-2, S. 161–171

- [96] HOH, Baik; GRUTESER, Marco; XIONG, Hui; ALRABADY, Ansaf: Enhancing Security and Privacy in Traffic-Monitoring Systems. In: *IEEE Pervasive Computing* 5 (2006), Oktober, Nr. 4. – DOI 10.1109/MPRV.2006.69. – ISSN 1536–1268, S. 38–46
- [97] HOH, Baik; IWUCHUKWU, Toch; JACOBSON, Quinn; WORK, Daniel; BAYEN, Alexandre M.; HERRING, Ryan; HERRERA, Juan C.; GRUTESER, Marco; ANNAVARAM, Murali; BAN, Jeff: Enhancing Privacy and Accuracy in Probe Vehicle-Based Traffic Monitoring via Virtual Trip Lines. In: *IEEE Transactions on Mobile Computing* 11 (2012), Mai, Nr. 5. – DOI 10.1109/TMC.2011.116. – ISSN 1536–1233, S. 849–864
- [98] HOLLICK, Matthias; CHRISTIN, Delphine: Fünf Milliarden vernetzte Sensoren: Chancen und Gefahren. In: *Proceedings of the Essener Workshop zur Netzsicherheit (EWNS11)*. Essen, Deutschland, April 2011, S. 1–2
- [99] HONICKY, Richard E.: Automatic calibration of sensor-phones using gaussian processes / Electrical Engineering and Computer Sciences, UC Berkeley. Berkeley, CA, USA, März 2007 (UCB/EECS-2007-34, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-34.pdf>). – Forschungsbericht. – 11 S.
- [100] HSU, Rein-Lien; ABDEL-MOTTALEB, M.; JAIN, A.K.: Face Detection in Color Images. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), Mai, Nr. 5. – DOI 10.1109/34.1000242. – ISSN 0162–8828, S. 696–706
- [101] HU, Ling; SHAHABI, Cyrus: Privacy Assurance in Mobile Sensing Networks: Go Beyond Trusted Servers. In: *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2010)*. Mannheim, Germany, March 2010. – DOI 10.1109/PERCOMW.2010.5470509, S. 613–619
- [102] HUAN, Chen; ZHIYU, Liu; GANG, Fu: Analysis of the aging characteristics of SnO<sub>2</sub> gas sensors. In: *Sensors and Actuators B: Chemical* 156 (2011), August, Nr. 2. – DOI 10.1016/j.snb.2011.03.005. – ISSN 0925–4005, S. 912–917
- [103] HUANG, Kuan L.; KANHERE, Salil S.; HU, Wen: Are You Contributing Trustworthy Data?: The Case for a Reputation System in Participatory Sensing. In: *Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWIM '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1868521.1868526. – ISBN 978–1–4503–0274–6, S. 14–22

- [104] HUANG, Zhengli; DU, Wenliang; CHEN, Biao: Deriving Private Information from Randomized Data. In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD '05)*. New York, NY, USA : ACM, 2005. – DOI 10.1145/1066157.1066163. – ISBN 1-59593-060-4, S. 37-48
- [105] HULL, Bret; BYCHKOVSKY, Vladimir; ZHANG, Yang; CHEN, Kevin; GORACZKO, Michel; MIU, Allen; SHIH, Eugene; BALAKRISHNAN, Hari; MADDEN, Samuel: CarTel: A Distributed Mobile Sensor Computing System. In: *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*. New York, NY, USA : ACM, 2006. – DOI 10.1145/1182807.1182821. – ISBN 1-59593-343-3, S. 125-138
- [106] ISAACMAN, Sibren; BECKER, Richard; CÁCERES, Ramón; KOBOUROV, Stephen; MARTONOSI, Margaret; ROWLAND, James; VARSHAVSKY, Alexander: Identifying Important Places in People's Lives from Cellular Network Data. In: LYONS, Kent (Hrsg.); HIGHTOWER, Jeffrey (Hrsg.); HUANG, Elaine M. (Hrsg.): *Pervasive Computing*. Bd. 6696. Berlin / Heidelberg : Springer, 2011. – DOI 10.1007/978-3-642-21726-5\_9. – ISBN 978-3-642-21725-8, S. 133-151
- [107] ISHIGAKI, Yang; MATSUMOTO, Yoshinori; ICHIMIYA, Ryo; TANAKA, Kenji: Development of Mobile Radiation Monitoring System Utilizing Smartphone and Its Field Tests in Fukushima. In: *IEEE Sensors Journal* 13 (2013), Oktober, Nr. 10. – DOI 10.1109/JSEN.2013.2272734. – ISSN 1530-437X, S. 3520-3526
- [108] JAIN, Ankur; CHANG, Edward Y.: Adaptive Sampling for Sensor Networks. In: *Proceedings of the 1st International Workshop on Data Management for Sensor Networks (DMSN '04)*. New York, NY, USA : ACM, 2004. – DOI 10.1145/1052199.1052202, S. 10-16
- [109] JAIN, Sushant; SHAH, Rahul C.; BRUNETTE, Waylon; BORRIELLO, Gaetano; ROY, Sumit: Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks. In: *Mobile Networks and Applications* 11 (2006), Juni, Nr. 3. – DOI 10.1007/s11036-006-5186-9. – ISSN 1383-469X, S. 327-339
- [110] JOHNSON, David B.; MALTZ, David A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: IMIELINSKI, Tomasz (Hrsg.); KORTH, Henry F. (Hrsg.): *Mobile Computing*. Bd. 353. New York / Philadelphia, US : Springer US, 1996. – DOI 10.1007/978-0-585-29603-6\_5. – ISBN 978-0-7923-9697-0, S. 153-181

- [111] JUANG, Philo; OKI, Hidekazu; WANG, Yong; MARTONOSI, Margaret; PEH, Li S.; RUBENSTEIN, Daniel: Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In: *ACM SIGOPS Operating Systems Review* 36 (2002), Oktober, Nr. 5. – DOI 10.1145/635508.605408. – ISSN 0163–5980, S. 96–107
- [112] KAENEL, Michael von; SOMMER, Philipp; WATTENHOFER, Roger: Ikarus: Large-scale Participatory Sensing at High Altitudes. In: *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications (HotMobile '11)*. New York, NY, USA : ACM, 2011. – DOI 10.1145/2184489.2184503. – ISBN 978–1–4503–0649–2, S. 63–68
- [113] KALMAN, Rudolph E.: A New Approach to Linear Filtering and Prediction Problems. In: *Journal of Fluids Engineering* 82 (1960), März, Nr. 1. – DOI 10.1115/1.3662552. – ISSN 0098–2202, S. 35–45
- [114] KAMIONKA, Marc; BREUIL, Philippe; PIJOLAT, Christophe: Calibration of a multivariate gas sensing device for atmospheric pollution measurement. In: *Sensors and Actuators B: Chemical* 118 (2006), Oktober, Nr. 1–2. – DOI 10.1016/j.snb.2006.04.058. – ISSN 0925–4005, S. 323–327
- [115] KANG, Jong H.; WELBOURNE, William; STEWART, Benjamin; BORRIELLO, Gaetano: Extracting Places from Traces of Locations. In: *Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH '04)*. New York, NY, USA : ACM, 2004. – DOI 10.1145/1024733.1024748. – ISBN 1–58113–877–6, S. 110–118
- [116] KANG, Seungwoo; LEE, Jinwon; JANG, Hyukjae; LEE, Hyonik; LEE, Youngki; PARK, Souneil; PARK, Taiwoo; SONG, Junehwa: SeeMon: Scalable and Energy-efficient Context Monitoring Framework for Sensor-rich Mobile Environments. In: *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1378600.1378630. – ISBN 978–1–60558–139–2, S. 267–280
- [117] KANHERE, Salil S.: Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces. In: *Proceedings of the 12th IEEE International Conference on Mobile Data Management (MDM 2011)*. Bd. 2. New York, NY : IEEE, Juni 2011. – DOI 10.1109/MDM.2011.16, S. 3–6
- [118] KANJO, Eiman: NoiseSPY: A Real-Time Mobile Phone Platform for Urban Noise Monitoring and Mapping. In: *Mobile Networks and Applications* 15 (2010), August, Nr. 4. – DOI 10.1007/s11036–009–0217–y. – ISSN 1383–469X, S. 562–574

- [119] KANJO, Eiman; BACON, Jean; ROBERTS, David; LANDSHOFF, Peter: MobSens: Making Smart Phones Smarter. In: *IEEE Pervasive Computing* 8 (2009), Oktober–Dezember, Nr. 4. – DOI 10.1109/MPRV.2009.79. – ISSN 1536–1268, S. 50–57
- [120] KANJO, Eiman; BENFORD, Steve; PAXTON, Mark; CHAMBERLAIN, Alan; FRASER, Danae S.; WOODGATE, Dawn; CRELLIN, David; WOOLARD, Adrain: MobGeoSen: Facilitating Personal Geosensor Data Collection and Visualization Using Mobile Phones. In: *Personal and Ubiquitous Computing* 12 (2008), November, Nr. 8. – DOI 10.1007/s00779–007–0180–1. – ISSN 1617–4909, S. 599–607
- [121] KANSAL, Aman; SOMASUNDARA, Arun A.; JEA, David D.; SRIVASTAVA, Mani B.; ESTRIN, Deborah: Intelligent Fluid Infrastructure for Embedded Networks. In: *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys '04)*. New York, NY, USA : ACM, 2004. – DOI 10.1145/990064.990080. – ISBN 1–58113–793–1, S. 111–124
- [122] KAPADIA, A.; KOTZ, D.; TRIANDOPOULOS, N.: Opportunistic sensing: Security challenges for the new paradigm. In: *First International Communication Systems and Networks and Workshops (COMSNETS 2009)*, 2009. – DOI 10.1109/COMSNETS.2009.4808850, S. 1–10
- [123] KARGUPTA, H.; DATTA, S.; WANG, Q.; SIVAKUMAR, K.: On the Privacy Preserving properties of Random Data Perturbation Techniques. In: *Proceedings of the Third IEEE International Conference on Data Mining (ICDM 2003)*, 2003. – DOI 10.1109/ICDM.2003.1250908, S. 99–106
- [124] KESHAVARZIAN, Abtin; LEE, Huang; VENKATRAMAN, Lakshmi: Wakeup Scheduling in Wireless Sensor Networks. In: *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*. New York, NY, USA : ACM, 2006. – DOI 10.1145/1132905.1132941. – ISBN 1–59593–368–9, S. 322–333
- [125] KHAN, Wazir Z.; XIANG, Yang; AALSALEM, Mohammed Y.; ARSHAD, Quratulain: Mobile Phone Sensing Systems: A Survey. In: *IEEE Communications Surveys & Tutorials* 15 (2013), Januar–März, Nr. 1. – DOI 10.1109/SURV.2012.031412.00077. – ISSN 1553–877X, S. 402–427
- [126] KIJEWski-CORREA, Tracy; HAENGGI, Martin; ANTSAKLIS, Panos: Wireless Sensor Networks for Structural Health Monitoring: A Multi-Scale Approach. In: *Proceedings of the ASCE Structures Congress ASCE*, 2006, S. 1–16
- [127] KIM, Minkyong; KOTZ, David; KIM, Songkuk: Extracting a Mobility Model from Real User Traces. In: *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, 2006. – DOI 10.1109/INFOCOM.2006.173. – ISSN 0743–166X, S. 1–13

- [128] KIM, Sukun; PAKZAD, Shamim; CULLER, David; DEMMEL, James; FENVES, Gregory; GLASER, Steven; TURON, Martin: Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks. In: *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN 2007)*. New York, NY, USA : IEEE, April 2007. – DOI 10.1109/IPSN.2007.4379685. – ISBN 978-1-59593-638-7, S. 254–263
- [129] KIMURA, Naoto; LATIFI, Shahram: A survey on data compression in wireless sensor networks. In: *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2005)*. Bd. 2, 2005. – DOI 10.1109/ITCC.2005.43, S. 8–13
- [130] KJÆRGAARD, Mikkel B.; LANGDAL, Jakob; GODSK, Torben; TOFTKJÆR, Thomas: EnTracked: Energy-efficient Robust Position Tracking for Mobile Devices. In: *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*. New York, NY, USA : ACM, 2009. – DOI 10.1145/1555816.1555839. – ISBN 978-1-60558-566-6, S. 221–234
- [131] KRUMM, John: Inference Attacks on Location Tracks. In: LAMARCA, Anthony (Hrsg.); LANGHEINRICH, Marc (Hrsg.); TRUONG, KhaiN. (Hrsg.): *Pervasive Computing*. Bd. 4480. Berlin / Heidelberg : Springer, 2007. – DOI 10.1007/978-3-540-72037-9\_8. – ISBN 978-3-540-72036-2, S. 127–143
- [132] KRUMM, John: A Survey of Computational Location Privacy. In: *Personal and Ubiquitous Computing* 13 (2009), August, Nr. 6. – DOI 10.1007/s00779-008-0212-5. – ISSN 1617-4909, S. 391–399
- [133] KWAPISZ, Jennifer R.; WEISS, Gary M.; MOORE, Samuel A.: Activity Recognition Using Cell Phone Accelerometers. In: *ACM SIGKDD Explorations Newsletter* 12 (2011), März, Nr. 2. – DOI 10.1145/1964897.1964918. – ISSN 1931-0145, S. 74–82
- [134] LANE, Nicholas D.; EISENMAN, Shane B.; MUSOLESI, Mirco; MILUZZO, Emiliano; CAMPBELL, Andrew T.: Urban Sensing Systems: Opportunistic or Participatory? In: *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications (HotMobile '08)*. New York, NY, USA : ACM, Februar 2008. – DOI 10.1145/1411759.1411763. – ISBN 978-1-60558-118-7, S. 11–16
- [135] LANE, Nicholas D.; MILUZZO, Emiliano; LU, Hong; PEEBLES, Daniel; CHOUDHURY, Tanzeem; CAMPBELL, Andrew T.: A survey of mobile phone sensing. In: *IEEE Communications Magazine* 48 (2010), September, Nr. 9. – DOI 10.1109/MCOM.2010.5560598. – ISSN 0163-6804, S. 140–150
- [136] LANGENDOEN, Koen; REIJERS, Niels: Distributed localization in wireless sensor networks: a quantitative comparison. In: *Computer Networks* 43 (2003), November, Nr. 4. – DOI 10.1016/S1389-1286(03)00356-6. – ISSN 1389-1286, S. 499 – 518. – Wireless Sensor Networks

- [137] LE BORGNE, Yann-Aël; SANTINI, Silvia; BONTEMPI, Gianluca: Adaptive model selection for time series prediction in wireless sensor networks. In: *Signal Processing* 87 (2007), Dezember, Nr. 12. – DOI 10.1016/j.sigpro.2007.05.015. – ISSN 0165–1684, S. 3010–3020. – Special Section: Information Processing and Data Management in Wireless Sensor Networks
- [138] LE GALL, Didier: MPEG: A Video Compression Standard for Multimedia Applications. In: *Communications of the ACM - Special issue on digital multimedia systems* 34 (1991), April, Nr. 4. – DOI 10.1145/103085.103090. – ISSN 0001–0782, S. 46–58
- [139] LEONHARDT, Ulf; MAGEE, Jeff: Security Considerations for a Distributed Location Service. In: *Journal of Network and Systems Management* 6 (1998), März, Nr. 1. – DOI 10.1023/A:1018777802208. – ISSN 1064–7570, S. 51–70
- [140] LI, Jason J.; FALTINGS, Boi; SAUKH, Olga; HASENFRATZ, David; BEUTEL, Jan: Sensing the Air We Breathe – The OpenSense Zurich Dataset. In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*. Bd. 1. Palo Alto, California : AAAI, 2012, S. 323–325
- [141] LI, Ninghui; LI, Tiancheng; VENKATASUBRAMANIAN, Suresh: t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In: *Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE 2007)*. New York, NY, USA : IEEE, April 2007. – DOI 10.1109/ICDE.2007.367856. – ISBN 1–4244–0802–4, S. 106–115
- [142] LIN, Kaisen; KANSAL, Aman; LYMBEROPOULOS, Dimitrios; ZHAO, Feng: Energy-Accuracy Trade-off for Continuous Mobile Device Location. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1814433.1814462. – ISBN 978–1–60558–985–5, S. 285–298
- [143] LIU, Ling: From Data Privacy to Location Privacy: Models and Algorithms. In: *Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB Endowment, 2007 (VLDB '07)*. – ISBN 978–1–59593–649–3, S. 1429–1430
- [144] LU, Hong; BERNHEIM BRUSH, A.J.; PRIYANTHA, Bodhi; KARLSON, AmyK.; LIU, Jie: SpeakerSense: Energy Efficient Unobtrusive Speaker Identification on Mobile Phones. In: LYONS, Kent (Hrsg.); HIGHTOWER, Jeffrey (Hrsg.); HUANG, ElaineM. (Hrsg.): *Pervasive Computing*. Bd. 6696. Springer Berlin Heidelberg, 2011. – DOI 10.1007/978–3–642–21726–5\_12. – ISBN 978–3–642–21725–8, S. 188–205

- [145] LU, Hong; LANE, Nicholas D.; EISENMAN, Shane B.; CAMPBELL, Andrew T.: Bubble-sensing: Binding sensing tasks to the physical world. In: *Pervasive and Mobile Computing* 6 (2010), Februar, Nr. 1. – DOI 10.1016/j.pmcj.2009.10.005. – ISSN 1574–1192, S. 58–71
- [146] LU, Hong; PAN, Wei; LANE, Nicholas D.; CHOUDHURY, Tanzeem; CAMPBELL, Andrew T.: SoundSense: Scalable Sound Sensing for People-Centric Applications on Mobile Phones. In: *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*. New York, NY, USA : ACM, 2009. – DOI <http://doi.acm.org/10.1145/1555816.1555834>. – ISBN 978–1–60558–566–6, S. 165–178
- [147] LU, Hong; YANG, Jun; LIU, Zhigang; LANE, Nicholas D.; CHOUDHURY, Tanzeem; CAMPBELL, Andrew T.: The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1869983.1869992. – ISBN 978–1–4503–0344–6, S. 71–84
- [148] LUO, Jun; HUBAUX, Jean-Pierre: Joint mobility and routing for lifetime elongation in wireless sensor networks. In: *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*. Bd. 3, IEEE, März 2005. – DOI 10.1109/INFCOM.2005.1498454. – ISSN 0743–166X, S. 1735–1746
- [149] LUO, Jun; PANCHARD, Jacques; PIÓRKOWSKI, Michał; GROSSGLAUSER, Matthias; HUBAUX, Jean-Pierre: MobiRoute: Routing Towards a Mobile Sink for Improving Lifetime in Sensor Networks. In: GIBBONS, Phillip B. (Hrsg.); ABDELZAHER, Tarek (Hrsg.); ASPNES, James (Hrsg.); RAO, Ramesh (Hrsg.): *Distributed Computing in Sensor Systems*. Bd. 4026. Berlin / Heidelberg : Springer, 2006. – DOI 10.1007/11776178\_29. – ISBN 978–3–540–35227–3, S. 480–497
- [150] LUO, Tie; KANHERE, Salil S.; TAN, Hwee-Pink: SEW-ing a Simple Endorsement Web to Incentivize Trustworthy Participatory Sensing. In: *Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON 2014)*. Singapur, Singapur, Juli 2014, S. 1–9
- [151] MA, Chris Y.; YAU, David K.; YIP, Nung K.; RAO, Nageswara S.: Privacy Vulnerability of Published Anonymous Mobility Traces. In: *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking (MobiCom '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1859995.1860017. – ISBN 978–1–4503–0181–7, S. 185–196

- [152] MACHANAVAJHALA, Ashwin; KIFER, Daniel; GEHRKE, Johannes; VENKITASUBRAMANIAM, Muthuramakrishnan: L-diversity: Privacy Beyond K-anonymity. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1 (2007), März, Nr. 1. – DOI 10.1145/1217299.1217302. – ISSN 1556–4681, S. 3:1–3:52
- [153] MADDEN, Samuel; FRANKLIN, Michael J.; HELLERSTEIN, Joseph M.; HONG, Wei: TAG: A Tiny AGgregation Service for Ad-hoc Sensor Networks. In: *ACM SIGOPS Operating Systems Review: Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)* 36 (2002), Dezember, Nr. SI. – DOI 10.1145/844128.844142. – ISSN 0163–5980, S. 131–146
- [154] MAINWARING, Alan; CULLER, David; POLASTRE, Joseph; SZEWCZYK, Robert; ANDERSON, John: Wireless Sensor Networks for Habitat Monitoring. In: *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*. New York, NY, USA : ACM, September 2002. – DOI 10.1145/570738.570751. – ISBN 1–58113–589–0, S. 88–97
- [155] MAISONNEUVE, Nicolas; STEVENS, Matthias; NIESSEN, Maria E.; STEELS, Luc: NoiseTube: Measuring and mapping noise pollution with mobile phones. In: ATHANASIADIS, IoannisN. (Hrsg.); RIZZOLI, Andrea E. (Hrsg.); MITKAS, Pericles A. (Hrsg.); GÓMEZ, JorgeMarx (Hrsg.): *Information Technologies in Environmental Engineering*. Berlin / Heidelberg : Springer, 2009 (Environmental Science and Engineering). – DOI 10.1007/978–3–540–88351–7\_16. – ISBN 978–3–540–88350–0, S. 215–228
- [156] MASS, Clifford: Nowcasting: The Promise of New Technologies of Communication, Modeling, and Observation. In: *Bulletin of the American Meteorological Society* 93 (2012), Juni, Nr. 6. – DOI 10.1175/BAMS–D–11–00153.1. – ISSN 0003–0007, S. 797–809
- [157] MATSUO, Yutaka; OKAZAKI, Naoaki; IZUMI, Kiyoshi; NAKAMURA, Yoshiyuki; NISHIMURA, Takuichi; HASIDA, Kôiti; NAKASHIMA, Hideyuki: Inferring Long-term User Properties Based on Users' Location History. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2007 (IJCAI'07), S. 2159–2165
- [158] MATTERN, Friedemann: Allgegenwärtige Datenverarbeitung – Trends, Visionen, Auswirkungen. In: ROSSNAGEL, Alexander (Hrsg.); SOMMERLATTE, Tom (Hrsg.); WINAND, Udo (Hrsg.): *Digitale Visionen*. Berlin / Heidelberg : Springer, 2008. – DOI 10.1007/978–3–540–77022–0\_1. – ISBN 978–3–540–77021–3, S. 3–29

- [159] MCCALL, Roderick; KRACHEEL, Martin; KOENIG, Vincent: Reducing Congestion through Persuasive Gaming. In: *Proceedings of "The Car as an Arena for Gaming" Workshop at MobileHCI 2012*. San Francisco, CA., September 2012
- [160] MEDNIS, Artis; STRAZDINS, Girts; ZVIEDRIS, Reinholds; KANONIRS, Georgijs; SELAVO, Leo: Real Time Pothole Detection using Android Smartphones with Accelerometers. In: *Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS 2011)*. New York, NY, USA : IEEE, Juni 2011. – DOI 10.1109/DCOSS.2011.5982206. – ISBN 978-1-4577-0512-0, S. 1-6
- [161] MILUZZO, Emiliano; LANE, Nicholas; CAMPBELL, Andrew; OLFATI-SABER, Reza: CaliBree: A Self-calibration System for Mobile Sensor Networks. In: NIKOLETSEAS, Sotiris (Hrsg.); CHLEBUS, Bogdan (Hrsg.); JOHNSON, David (Hrsg.); KRISHNAMACHARI, Bhaskar (Hrsg.): *Distributed Computing in Sensor Systems*. Bd. 5067. Berlin / Heidelberg : Springer, 2008. – DOI 10.1007/978-3-540-69170-9\_21. – ISBN 978-3-540-69169-3, S. 314-331
- [162] MILUZZO, Emiliano; LANE, Nicholas D.; EISENMAN, Shane B.; CAMPBELL, Andrew T.: CenceMe – Injecting Sensing Presence into Social Networking Applications. Berlin / Heidelberg : Springer, 2007. – DOI 10.1007/978-3-540-75696-5\_1. – ISBN 978-3-540-75695-8, S. 1-28
- [163] MILUZZO, Emiliano; LANE, Nicholas D.; FODOR, Kristóf; PETERSON, Ronald; LU, Hong; MUSOLESI, Mirco; EISENMAN, Shane B.; ZHENG, Xiao; CAMPBELL, Andrew T.: Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1460412.1460445. – ISBN 978-1-59593-990-6, S. 337-350
- [164] MILUZZO, Emiliano; PAPANDREA, Michela; LANE, Nicholas D.; LU, Hong; CAMPBELL, Andrew T.: Pocket, Bag, Hand, etc.-Automatically Detecting Phone Context through Discovery. In: *Proceedings of the International Workshop on Sensing for App Phones (PhoneSense 2010)*. Zurich, Switzerland, November 2010, S. 21-25
- [165] MOHAN, Prashanth; PADMANABHAN, Venkata N.; RAMJEE, Ramachandran: Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones. In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1460412.1460444. – ISBN 978-1-59593-990-6, S. 323-336

- [166] MOHAN, Prashanth; PADMANABHAN, Venkata N.; RAMJEE, Ramachandran: Nericell: Using Mobile Smartphones for Rich Monitoring of Road and Traffic Conditions. In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1460412.1460450. – ISBN 978-1-59593-990-6, S. 357–358
- [167] MOHAN, Prashanth; PADMANABHAN, Venkata N.; RAMJEE, Ramachandran: TrafficSense: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones / Microsoft Research India. Bangalore, Indien, April 2008 (MSR-TR-2008-59, <http://research.microsoft.com/apps/pubs/default.aspx?id=70573>). – Forschungsbericht. – 29 S.
- [168] MOKBEL, Mohamed F.; CHOW, Chi-Yin; AREF, Walid G.: The New Casper: A Privacy-Aware Location-Based Database Server. In: *Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE 2007)*. New York, NY, USA : IEEE, April 2007. – DOI 10.1109/ICDE.2007.369052. – ISBN 1-4244-0802-4, S. 1499–1500
- [169] MOTEGI, Shinji; NISHIMURA, Yasutaka; YOSHIHARA, Kiyohito: Communication Algorithm for Statistic Monitoring in People-Centric Sensing Networks. In: *Proceedings of the 2010 Ninth International Conference on Networks (ICN '10)*. Washington, DC, USA : IEEE Computer Society, 2010. – ISBN 978-0-7695-3979-9, S. 133–138
- [170] MUN, Min; REDDY, Sasank; SHILTON, Katie; YAU, Nathan; BURKE, Jeff; ESTRIN, Deborah; HANSEN, Mark; HOWARD, Eric; WEST, Ruth; BODA, Péter: PEIR, the Personal Environmental Impact Report, As a Platform for Participatory Sensing Systems Research. In: *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*. New York, NY, USA : ACM, 2009. – DOI 10.1145/1555816.1555823. – ISBN 978-1-60558-566-6, S. 55–68
- [171] MUSOLESI, Mirco; PIRACCINI, Mattia; FODOR, Kristof; CORRADI, Antonio; CAMPBELL, Andrew T.: Supporting Energy-Efficient Uploading Strategies for Continuous Sensing Applications on Mobile Phones. In: FLORÉEN, Patrik (Hrsg.); KRÜGER, Antonio (Hrsg.); SPASOJEVIC, Mirjana (Hrsg.): *Pervasive Computing*. Bd. 6030. Berlin / Heidelberg : Springer, 2010. – DOI 10.1007/978-3-642-12654-3\_21. – ISBN 978-3-642-12653-6, S. 355–372

- [172] NACHMAN, Lama; BAXI, Amit; BHATTACHARYA, Sangeeta; DARERA, Vivek; DESHPANDE, Piyush; KODALAPURA, Nagaraju; MAGESHKUMAR, Vincent; RATH, Satish; SHAHABDEEN, Junaith; ACHARYA, Raviraja: Jog Falls: A Pervasive Healthcare Platform for Diabetes Management. In: FLORÉEN, Patrik (Hrsg.); KRÜGER, Antonio (Hrsg.); SPASOJEVIC, Mirjana (Hrsg.): *Pervasive Computing*. Bd. 6030. Berlin / Heidelberg : Springer, 2010. – DOI 10.1007/978-3-642-12654-3\_6. – ISBN 978-3-642-12653-6, S. 94–111
- [173] NEWMAN, William M.; ELDRIDGE, Margery A.; LAMMING, Michael G.: PEPYS: Generating Autobiographies by Automatic Tracking. In: BANNON, Liam (Hrsg.); ROBINSON, Mike (Hrsg.); SCHMIDT, Kjeld (Hrsg.): *Proceedings of the Second European Conference on Computer-Supported Cooperative Work ECSCW '91*. Dordrecht, NL : Springer Netherlands, 1991. – DOI 10.1007/978-94-011-3506-1\_13. – ISBN 978-0-7923-1439-4, S. 175–188
- [174] NIFORATOS, Evangelos; CAMPOS, Pedro; VOURVOPOULOS, Athanasios; DORIA, Andre; LANGHEINRICH, Marc: Atmos: A Hybrid Crowdsourcing Approach to Weather Estimation. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. New York, NY, USA : ACM, 2014. – DOI 10.1145/2638728.2638780. – ISBN 978-1-4503-3047-3, S. 135–138
- [175] NIKZAD, Nima; VERMA, Nakul; ZIFTCI, Celal; BALES, Elizabeth; QUICK, Nichole; ZAPPI, Piero; PATRICK, Kevin; DASGUPTA, Sanjoy; KRUEGER, Ingolf; ROSING, Tajana ; GRISWOLD, William G.: CitiSense: Improving Geospatial Environmental Assessment of Air Quality Using a Wireless Personal Exposure Monitoring System. In: *Proceedings of the Conference on Wireless Health (WH '12)*. New York, NY, USA : ACM, Oktober 2012. – DOI 10.1145/2448096.2448107. – ISBN 978-1-4503-1760-3, S. 11:1–11:8
- [176] OSCAR WANG, Xinlei; CHENG, Wei; MOHAPATRA, Prasant; ABDELZAHER, Tarek: ARTSense: Anonymous reputation and trust in participatory sensing. In: *Proceedings of IEEE INFOCOM 2013*, IEEE, April 2013. – DOI 10.1109/INFCOM.2013.6567058. – ISSN 0743-166X, S. 2517–2525
- [177] OUTRAM, Christine; RATTI, Carlo; BIDERMAN, Assaf: The Copenhagen Wheel: An innovative electric bicycle system that harnesses the power of real-time information and crowd sourcing. In: *Proceedings of International Conference and Exhibition on Ecological Vehicles and Renewable Energies (EVER 2010)*. Monaco : Environmental association of Monaco MC2D, März 2010, S. 1–8

- [178] OVEREEM, Aart; ROBINSON, James C. R.; LEIJNSE, Hidde; STEENEVELD, Gert-Jan; HORN, Berthold K. P.; UIJLENHOET, Remko: Crowdsourcing Urban Air Temperatures from Smartphone Battery Temperatures. In: *Geophysical Research Letters* 40 (2013), August, Nr. 15. – DOI 10.1002/grl.50786. – ISSN 1944–8007, S. 4081–4085
- [179] PADHY, Paritosh; DASH, Rajdeep K.; MARTINEZ, Kirk; JENNINGS, Nicholas R.: A Utility-based Sensing and Communication Model for a Glacial Sensor Network. In: *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '06)*. New York, NY, USA : ACM, 2006. – DOI 10.1145/1160633.1160885. – ISBN 1–59593–303–4, S. 1353–1360
- [180] PAEK, Jeongyeup; KIM, Joongheon; GOVINDAN, Ramesh: Energy-efficient Rate-adaptive GPS-based Positioning for Smartphones. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1814433.1814463. – ISBN 978–1–60558–985–5, S. 299–314
- [181] PARUCHURI, Vamsi; BASAVARAJU, Shivakumar; DURRESI, Arjan; KANNAN, Rajgopal; IYENGAR, Sundaraja S.: Random asynchronous wakeup protocol for sensor networks. In: *Proceedings of the First International Conference on Broadband Networks (BroadNets 2004)*, 2004. – DOI 10.1109/BROADNETS.2004.71, S. 710–717
- [182] PATTERSON, Donald J.; LIAO, Lin; FOX, Dieter; KAUTZ, Henry: Inferring High-Level Behavior from Low-Level Sensors. In: DEY, Anind K. (Hrsg.); SCHMIDT, Albrecht (Hrsg.); MCCARTHY, Joseph F. (Hrsg.): *UbiComp 2003: Ubiquitous Computing*. Bd. 2864. Berlin / Heidelberg : Springer, 2003. – DOI 10.1007/978–3–540–39653–6\_6. – ISBN 978–3–540–20301–8, S. 73–89
- [183] PAULOS, Eric; HONICKY, Richard J.; GOODMAN, Elizabeth: Sensing atmosphere. In: *Proceedings of the "Sensing on Everyday Mobile Phones in support of participatory research" workshop at the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys 2007)*. Sydney, Australia, November 2007, S. 1–3
- [184] PHILIPP, Damian; STACHOWIAK, Jaroslaw; ALT, Patrick; DÜRR, Frank; ROTHERMEL, Kurt: DrOPS: Model-driven optimization for Public Sensing systems. In: *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom 2013)*, 2013. – DOI 10.1109/PerCom.2013.6526731, S. 185–192

- [185] PIDCOCK, Sarah; SMITS, Rob; HENGARTNER, Urs; GOLDBERG, Ian: NotiSense: An Urban Sensing Notification System To Improve Bystander Privacy. In: *Proceedings of Second International Workshop on Sensing Applications on Mobile Phones (PhoneSense 2011) at ACM SenSys 2011*. Seattle, USA, November 2011, S. 1–5
- [186] POH, Ming-Zher; KIM, Kyunghye; GOESSLING, Andrew; SWENSON, Nicholas; PICARD, Rosalind: Cardiovascular Monitoring Using Earphones and a Mobile Device. In: *IEEE Pervasive Computing* 11 (2012), Oktober, Nr. 4. – DOI 10.1109/MPRV.2010.91. – ISSN 1536–1268, S. 18–26
- [187] POLASTRE, Joseph; HILL, Jason; CULLER, David: Versatile Low Power Media Access for Wireless Sensor Networks. In: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*. New York, NY, USA : ACM, 2004. – DOI 10.1145/1031495.1031508. – ISBN 1–58113–879–2, S. 95–107
- [188] PRIYANTHA, Bodhi; LYMBEROPOULOS, Dimitrios; LIU, Jie: LittleRock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones. In: *IEEE Pervasive Computing* 10 (2011), April, Nr. 2. – DOI 10.1109/MPRV.2011.28. – ISSN 1536–1268, S. 12–15
- [189] RACHURI, Kiran K.; MASCOLO, Cecilia; MUSOLESI, Mirco; RENTFROW, Peter J.: SociableSense: Exploring the Trade-offs of Adaptive Sampling and Computation Offloading for Social Sensing. In: *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom '11)*. New York, NY, USA : ACM, 2011. – DOI 10.1145/2030613.2030623. – ISBN 978–1–4503–0492–4, S. 73–84
- [190] RAGHUNATHAN, Vijay; GANERIWAL, Saurabh; SRIVASTAVA, Mani: Emerging Techniques for Long Lived Wireless Sensor Networks. In: *IEEE Communications Magazine* 44 (2006), April, Nr. 4. – DOI 10.1109/MCOM.2006.1632657. – ISSN 0163–6804, S. 108–114
- [191] RAHIMI, Mohammad; HANSEN, Mark; KAISER, William J.; SUKHATME, Gaurav; ESTRIN, Deborah: Adaptive sampling for environmental field estimation using robotic sensors. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005. – DOI 10.1109/IROS.2005.1545070, S. 3692–3698
- [192] RAJENDRAN, Venkatesh; OBRACZKA, Katia; GARCIA-LUNA-ACEVES, Jose J.: Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Networks. In: *Wireless Networks* 12 (2006), Februar, Nr. 1. – DOI 10.1007/s11276–006–6151–z. – ISSN 1022–0038, S. 63–78

- [193] RANA, Rajib K.; CHOU, Chun T.; KANHERE, Salil S.; BULUSU, Nirupama; HU, Wen: Ear-phone: An End-to-end Participatory Urban Noise Mapping System. In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*. New York, NY, USA : ACM, April 2010. – DOI 10.1145/1791212.1791226. – ISBN 978-1-60558-988-6, S. 105–116
- [194] REDDY, Sasank; ESTRIN, Deborah; HANSEN, Mark; SRIVASTAVA, Mani: Examining Micro-payments for Participatory Sensing Data Collections. In: *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (Ubicomp '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1864349.1864355. – ISBN 978-1-60558-843-8, S. 33–36
- [195] REDDY, Sasank; ESTRIN, Deborah; SRIVASTAVA, Mani: Recruitment Framework for Participatory Sensing Data Collections. In: FLORÉEN, Patrik (Hrsg.); KRÜGER, Antonio (Hrsg.); SPASOJEVIC, Mirjana (Hrsg.): *Pervasive Computing*. Bd. 6030. Berlin / Heidelberg : Springer, 2010. – DOI 10.1007/978-3-642-12654-3\_9. – ISBN 978-3-642-12653-6, S. 138–155
- [196] REDDY, Sasank; PARKER, Andrew; HYMAN, Josh; BURKE, Jeff; ESTRIN, Deborah; HANSEN, Mark: Image Browsing, Processing, and Clustering for Participatory Sensing: Lessons from a DietSense Prototype. In: *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets '07)*. New York, NY, USA : ACM, 2007. – DOI 10.1145/1278972.1278975. – ISBN 978-1-59593-694-3, S. 13–17
- [197] REDDY, Sasank; SHILTON, Katie; DENISOV, Gleb; CENIZAL, Christian; ESTRIN, Deborah; SRIVASTAVA, Mani: Biketastic: Sensing and Mapping for Better Biking. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1753326.1753598. – ISBN 978-1-60558-929-9, S. 1817–1820
- [198] RHEE, Injong; WARRIER, Ajit; AIA, Mahesh; MIN, Jeongki; SICHITIU, Mihail L.: Z-MAC: A Hybrid MAC for Wireless Sensor Networks. In: *IEEE/ACM Transactions on Networking (TON)* 16 (2008), Juni, Nr. 3. – DOI 10.1109/TNET.2007.900704. – ISSN 1063-6692, S. 511–524
- [199] RÖMER, Kay; MATTERN, Friedemann: The Design Space of Wireless Sensor Networks. In: *IEEE Wireless Communications* 11 (2004), Dezember, Nr. 6. – DOI 10.1109/MWC.2004.1368897. – ISSN 1536-1284, S. 54–61
- [200] ROTH, Jörg: *Mobile Computing: Grundlagen, Technik, Konzepte*. 2., aktualisierte Auflage. Heidelberg : dpunkt.verlag, 2005. – 494 S. – ISBN 3-89864-366-2
- [201] RULA, John; BUSTAMANTE, Fabián E.: Crowd (Soft) Control. In: *Proceedings of The Thirteenth Workshop on Mobile Computing Systems and Applications (HotMobile 2012)*. San Diego, CA, USA, Februar 2012, S. 1–6

- [202] SADLER, Christopher M.; MARTONOSI, Margaret: Data Compression Algorithms for Energy-constrained Devices in Delay Tolerant Networks. In: *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*. New York, NY, USA : ACM, 2006. – DOI 10.1145/1182807.1182834. – ISBN 1-59593-343-3, S. 265–278
- [203] SCCELLATO, Salvatore; MUSOLESI, Mirco; MASCOLO, Cecilia; LATORA, Vito; CAMPBELL, Andrew T.: NextPlace: A Spatio-temporal Prediction Framework for Pervasive Systems. In: LYONS, Kent (Hrsg.); HIGHTOWER, Jeffrey (Hrsg.); HUANG, ElaineM. (Hrsg.): *Pervasive Computing*. Bd. 6696. Berlin / Heidelberg : Springer, 2011. – DOI 10.1007/978-3-642-21726-5\_10. – ISBN 978-3-642-21725-8, S. 152–169
- [204] SCHMID, Gerhard: Abhören in der Premiumklasse. In: *Zeitschrift für Außen- und Sicherheitspolitik* 7 (2014), Januar, Nr. 1. – DOI 10.1007/s12399-013-0386-4. – ISSN 1866-2188, S. 11–21
- [205] SCHURGERS, Curt; TSIATSI, Vlasios; SRIVASTAVA, Mani B.: STEM: Topology management for energy efficient sensor networks. In: *Proceedings of the IEEE Aerospace Conference '02*. Bd. 3, 2002. – DOI 10.1109/AERO.2002.1035239, S. 3–1099-3-1108
- [206] SCHWEIZER, Immanuel; BÄRTL, Roman; SCHULZ, Axel; PROBST, Florian; MÜHLHÄUSER, Max: NoiseMap - Real-time participatory noise maps. In: *Proceedings of the Second International Workshop on Sensing Applications on Mobile Phones (PhoneSense '11)*. Seattle, WA, USA, 2011, S. 1–5
- [207] SCHWISTER, Karl: *Kleine Formelsammlung Chemie*. 4., aktualisierte Auflage. München : Carl Hanser Verlag, 2014. – 176 S. – ISBN 978-3-446-44213-9
- [208] SEHGAL, Shitiz; KANHERE, Salil S.; CHOU, Chun T.: MobiShop: Using Mobile Phones for Sharing Consumer Pricing Information. In: *Demo Session of the 4th International Conference on Distributed Computing in Sensor Systems (DCOSS 2008)*. Santorini Island, Greece, Juni 2008, S. 1–2
- [209] SELAVO, Leo; WOOD, Anthony; CAO, Qing; SOOKOOR, Tamim; LIU, Hengchang; SRINIVASAN, Aravind; WU, Yafeng; KANG, Woochul; STANKOVIC, John; YOUNG, Don; PORTER, John: LUSTER: Wireless Sensor Network for Environmental Research. In: *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*. New York, NY, USA : ACM, November 2007. – DOI 10.1145/1322263.1322274. – ISBN 978-1-59593-763-6, S. 103–116

- [210] SHA, Kewei; ZHAN, Guoxing; SHI, Weisong; LUMLEY, Mark; WIHOLM, Clairy; ARNETZ, Bengt: SPA: A Smart Phone Assisted Chronic Illness Self-management System with Participatory Sensing. In: *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments (HealthNet '08)*. New York, NY, USA : ACM, 2008. – DOI 10.1145/1515747.1515755. – ISBN 978-1-60558-199-6, S. 5:1–5:3
- [211] SHAH, Rahul C.; ROY, Sumit; JAIN, Sushant; BRUNETTE, Waylon: Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. In: *Ad Hoc Networks* 1 (2003), September, Nr. 2–3. – DOI 10.1016/S1570-8705(03)00003-9. – ISSN 1570-8705, S. 215 – 233
- [212] SHEPARD, Donald: A Two-dimensional Interpolation Function for Irregularly-spaced Data. In: *Proceedings of the 1968 23rd ACM National Conference*. New York, NY, USA : ACM, 1968 (ACM '68). – DOI 10.1145/800186.810616, S. 517–524
- [213] SHETH, Amit: Citizen Sensing, Social Signals, and Enriching Human Experience. In: *IEEE Internet Computing* 13 (2009), Juli/August, Nr. 4. – DOI 10.1109/MIC.2009.77. – ISSN 1089-7801, S. 87–92
- [214] SHI, Jing; ZHANG, Yanchao; LIU, Yunzhong; ZHANG, Yanchao: PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems. In: *Proceedings IEEE INFOCOM 2010*, 2010. – DOI 10.1109/INF-COM.2010.5462147. – ISSN 0743-166X, S. 1–9
- [215] SHILTON, Katie: Four Billion Little Brothers?: Privacy, Mobile Phones, and Ubiquitous Data Collection. In: *Communications of the ACM* 52 (2009), November, Nr. 11. – DOI 10.1145/1592761.1592778. – ISSN 0001-0782, S. 48–53
- [216] SHILTON, Katie; BURKE, Jeff; ESTRIN, Deborah; HANSEN, Mark; SRIVASTAVA, Mani B.: Participatory Privacy in Urban Sensing. In: *Proceedings of the International Workshop on Mobile Device and Urban Sensing (MODUS 2008)*, 2008, S. 1–7
- [217] SHIN, Minh; CORNELIUS, Cory; PEEBLES, Dan; KAPADIA, Apu; KOTZ, David; TRIANDOPOULOS, Nikos: AnonySense: A system for anonymous opportunistic sensing. In: *Pervasive and Mobile Computing* 7 (2011), Februar, Nr. 1. – DOI 10.1016/j.pmcj.2010.04.001. – ISSN 1574-1192, S. 16–30
- [218] SINGH, Amarjeet; BUDZIK, Diane; CHEN, Willie; BATALIN, Maxim A.; STEALEY, Michael; BORGSTROM, Henrik; KAISER, William J.: Multiscale Sensing: A new paradigm for actuated sensing of high frequency dynamic phenomena. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China, Oktober 2006. – DOI 10.1109/IROS.2006.282517, S. 328–335

- [219] SIVARAMAN, Vijay; CARRAPETTA, James; HU, Ke; LUXAN, Blanca G.: HazeWatch: A Participatory Sensor System for Monitoring Air Pollution in Sydney. In: *Proceedings of the IEEE 38th Conference on Local Computer Networks Workshops (LCN Workshops 2013)*. New York, NY, USA : IEEE, Oktober 2013. – DOI 10.1109/LCNW.2013.6758498, S. 56–64
- [220] SOMASUNDARA, Arun A.; KANSAL, Aman; JEA, David D.; ESTRIN, Deborah; SRIVASTAVA, Mani B.: Controllably mobile infrastructure for low energy embedded networks. In: *IEEE Transactions on Mobile Computing* 5 (2006), August, Nr. 8. – DOI 10.1109/TMC.2006.109. – ISSN 1536–1233, S. 958–973
- [221] STANKOVIC, John A.: Wireless Sensor Networks. In: *IEEE Computer* 41 (2008), Oktober, Nr. 10. – DOI 10.1109/MC.2008.441. – ISSN 0018–9162, S. 92–95
- [222] STRATER, Katherine; LIPFORD, Heather R.: Strategies and Struggles with Privacy in an Online Social Networking Community. In: *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers (BCS-HCI '08)*. Swinton, UK, UK : British Computer Society, 2008. – ISBN 978–1–906124–04–5, S. 111–119
- [223] SWEENEY, Latanya: k-anonymity: A model for protecting privacy. In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10 (2002), Oktober, Nr. 5. – DOI 10.1142/S0218488502001648. – ISSN 0218–4885, S. 557–570
- [224] TESKE, Holger; FURTHMÜLLER, Jochen; WALDHORST, Oliver P.: A Resilient and Energy-saving Incentive System for Resource Sharing in MANETs. In: LUTTENBERGER, Norbert (Hrsg.); PETERS, Hagen (Hrsg.): *17th GI/ITG Conference on Communication in Distributed Systems (KiVS 2011)*. Bd. 17. Dagstuhl, Germany : Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011 (OpenAccess Series in Informatics (OASICs)). – URN urn:nbn:de:0030–drops–29621. – ISBN 978–3–939897–27–9, S. 109–120
- [225] THIAGARAJAN, Arvind; BIAGIONI, James; GERLICH, Tomas; ERIKSSON, Jakob: Cooperative Transit Tracking Using Smart-phones. In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1869983.1869993. – ISBN 978–1–4503–0344–6, S. 85–98
- [226] THIAGARAJAN, Arvind; RAVINDRANATH, Lenin; LACURTS, Katrina; MADDEN, Samuel; BALAKRISHNAN, Hari; TOLEDO, Sivan; ERIKSSON, Jakob: VTrack: Accurate, Energy-aware Road Traffic Delay Estimation Using Mobile Phones. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*. New York, NY, USA : ACM, 2009. – DOI 10.1145/1644038.1644048. – ISBN 978–1–60558–519–2, S. 85–98

- [227] TREPTE, Sabine; DIENLIN, Tobias: Privatsphäre im Internet. In: PORSCHE, Torsten (Hrsg.); PIESCHL, Stephanie (Hrsg.): *Neue Medien und deren Schattent: Mediennutzung, Medienwirkung und Medienkompetenz*. 1. Auflage. Göttingen : Hogrefe Verlag, 2014. – ISBN 978–3–8409–2479–8, S. 53–80
- [228] TSENG, Yu-Chee; HSU, Chih-Shun; HSIEH, Ten-Yueng: Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks . In: *Computer Networks* 43 (2003), Oktober, Nr. 3. – DOI 10.1016/S1389–1286(03)00284–6. – ISSN 1389–1286, S. 317–337
- [229] TSENG, Yu-Chee; WANG, You-Chiun; CHENG, Kai-Yang; HSIEH, Yao-Yu: iMouse: An Integrated Mobile Surveillance and Wireless Sensor System. In: *IEEE Computer* 40 (2007), Juni, Nr. 6. – DOI 10.1109/MC.2007.211. – ISSN 0018–9162, S. 60–66
- [230] TSUJIMORI, Tomoya; THEPVILOJANAPONG, Niwat; OHTA, Yoshikatsu; ZHAO, Yunlong; TOBE, Yoshito: History-based Incentive for Crowd Sensing. In: *Proceedings of the 2014 International Workshop on Web Intelligence and Smart Sensing (IWWISS '14)*. New York, NY, USA : ACM, September 2014. – DOI 10.1145/2637064.2637089. – ISBN 978–1–4503–2747–3, S. 2:1–2:6
- [231] WANG, Guiling; CAO, Guohong; LA PORTA, Tom; ZHANG, Wensheng: Sensor Relocation in Mobile Sensor Networks. In: *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*. Bd. 4, 2005. – DOI 10.1109/INFCOM.2005.1498517. – ISSN 0743–166X, S. 2302–2312
- [232] WANG, Yi; LIN, Jialiu; ANNAVARAM, Murali; JACOBSON, Quinn A.; HONG, Jason; KRISHNAMACHARI, Bhaskar; SADEH, Norman: A Framework of Energy Efficient Mobile sensing for Automatic User State Recognition. In: *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*. New York, NY, USA : ACM, 2009. – DOI <http://doi.acm.org/10.1145/1555816.1555835>. – ISBN 978–1–60558–566–6, S. 179–192
- [233] WANG, Z. M.; BASAGNI, Stefano; MELACHRINOUDIS, Emanuel; PETRIOLI, Chiara: Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime. In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05)*, IEEE, Januar 2005. – DOI 10.1109/HICSS.2005.259. – ISSN 1530–1605, S. 287–295
- [234] WELCH, Terry A.: A Technique for High-Performance Data Compression. In: *IEEE Computer* 17 (1984), Juni, Nr. 6. – DOI 10.1109/MC.1984.1659158. – ISSN 0018–9162, S. 8–19
- [235] WERNER-ALLEN, Geoffrey; SWIESKOWSKI, Patrick; WELSH, Matt: Mote-Lab: A Wireless Sensor Network Testbed. In: *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*. Piscataway, NJ, USA : IEEE Press, 2005. – ISBN 0–7803–9202–7

- [236] WESTIN, Alan F.: *Privacy and Freedom*. 4. Auflage. New York, NY, USA : Atheneum, 1967. – 487 S.
- [237] WHITEHOUSE, Kamin: *The Design of Calamari: an Ad-hoc Localization System for Sensor Networks*. Berkeley, USA, Department of Electrical Engineering and Computer Sciences, University of California Berkeley, Masterarbeit, 2002
- [238] WHITEHOUSE, Kamin; CULLER, David: Calibration As Parameter Estimation in Sensor Networks. In: *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*. New York, NY, USA : ACM, 2002. – DOI 10.1145/570738.570747. – ISBN 1–58113–589–0, S. 59–67
- [239] WILLETT, Rebecca; MARTIN, Aline; NOWAK, Robert: Backcasting: Adaptive Sampling for Sensor Networks. In: *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*. New York, NY, USA : ACM, 2004. – DOI 10.1145/984622.984641. – ISBN 1–58113–846–6, S. 124–133
- [240] WISNIEWSKI, Mariusz; DEMARTINI, Gianluca; MALATRAS, Apostolos; CUDRÉ-MAUROUX, Philippe: NoizCrowd: A Crowd-Based Data Gathering and Management System for Noise Level Data. In: DANIEL, Florian (Hrsg.); PAPADOPOULOS, George A. (Hrsg.); THIRAN, Philippe (Hrsg.): *Mobile Web Information Systems*. Bd. 8093. Berlin / Heidelberg : Springer, 2013. – DOI 10.1007/978–3–642–40276–0\_14. – ISBN 978–3–642–40275–3, S. 172–186
- [241] WORK, Daniel B.; HERRING, Ryan; HOFLEITNER, Aude; BAYEN, Alexandre M.: Mobile Millennium Demonstration - Participatory Traffic Estimation Using Mobile Phones. In: *Proceedings of the CPS Forum at the Cyber-Physical Systems Week*. San Francisco, CA, USA, April 2009, S. 1–2
- [242] XIONG, Zixiang; LIVERIS, Angelos D.; CHENG, Samuel: Distributed Source Coding for Sensor Networks. In: *IEEE Signal Processing Magazine* 21 (2004), September, Nr. 5. – DOI 10.1109/MSP.2004.1328091. – ISSN 1053–5888, S. 80–94
- [243] XU, Ya; HEIDEMANN, John; ESTRIN, Deborah: Geography-informed Energy Conservation for Ad Hoc Routing. In: *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*. New York, NY, USA : ACM, 2001. – DOI 10.1145/381677.381685. – ISBN 1–58113–422–3, S. 70–84
- [244] YAN, Zhixian; SUBBARAJU, Vigneshwaran; CHAKRABORTY, Dipanjan; MISRA, Archan; ABERER, Karl: Energy-Efficient Continuous Activity Recognition on Mobile Phones: An Activity-Adaptive Approach. In: *Proceedings of the 16th International Symposium on Wearable Computers (ISWC 2012)*, 2012. – DOI 10.1109/ISWC.2012.23. – ISSN 1550–4816, S. 17–24

- [245] YANG, Jun; MUNGUIA-TAPIA, Emmanuel; GIBBS, Simon: Efficient In-pocket Detection with Mobile Phones. In: *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp '13 Adjunct)*. New York, NY, USA : ACM, 2013. – DOI 10.1145/2494091.2494099. – ISBN 978-1-4503-2215-7, S. 31-34
- [246] YE, Fan; LUO, Haiyun; CHENG, Jerry; LU, Songwu; ZHANG, Lixia: A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks. In: *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA : ACM, 2002 (MobiCom '02). – DOI 10.1145/570645.570664. – ISBN 1-58113-486-X, S. 148-159
- [247] YE, Wei; HEIDEMANN, John; ESTRIN, Deborah: An energy-efficient MAC protocol for wireless sensor networks. In: *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*. Bd. 3. New York, NY, USA : IEEE, 2002. – DOI 10.1109/INFCOM.2002.1019408. – ISSN 0743-166X, S. 1567-1576
- [248] YOU, Tun-Hao; PENG, Wen-Chih; LEE, Wang-Chien: Protecting Moving Trajectories with Dummies. In: *Proceedings of the International Conference on Mobile Data Management (MDM 2007)*, 2007. – DOI 10.1109/MDM.2007.58, S. 278-282
- [249] YU, Jane Y.; CHONG, Peter Han J.: A survey of clustering schemes for mobile ad hoc networks. In: *IEEE Communications Surveys & Tutorials 7* (2005), Januar-März, Nr. 1. – DOI 10.1109/COMST.2005.1423333. – ISSN 1553-877X, S. 32-48
- [250] ZHENG, Rong; HOU, Jennifer C.; SHA, Lui: Asynchronous Wakeup for Ad Hoc Networks. In: *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '03)*. New York, NY, USA : ACM, 2003. – DOI 10.1145/778415.778420. – ISBN 1-58113-684-6, S. 35-45
- [251] ZHENG, Yu; ZHANG, Lizhu; XIE, Xing; MA, Wei-Ying: Mining Interesting Locations and Travel Sequences from GPS Trajectories. In: *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. New York, NY, USA : ACM, 2009. – DOI 10.1145/1526709.1526816. – ISBN 978-1-60558-487-4, S. 791-800
- [252] ZHOU, Changqing; BHATNAGAR, Nupur; SHEKHAR, Shashi; TERVEEN, Loren: Mining Personally Important Places from GPS Tracks. In: *Proceedings of the IEEE 29th International Conference on Data Engineering Workshops (ICDEW 2007)*. Los Alamitos, CA, USA : IEEE Computer Society, April 2007. – DOI 10.1109/ICDEW.2007.4401037. – ISBN 978-1-4244-0831-3, S. 517-526

- [253] ZHOU, Changqing; FRANKOWSKI, Dan; LUDFORD, Pamela; SHEKHAR, Shashi; TERVEEN, Loren: Discovering Personally Meaningful Places: An Interactive Clustering Approach. In: *ACM Transactions on Information Systems (TOIS)* 25 (2007), Juli, Nr. 3. – DOI 10.1145/1247715.1247718. – ISSN 1046–8188
- [254] ZHUANG, Zhenyun; KIM, Kyu-Han; SINGH, Jatinder P.: Improving Energy Efficiency of Location Sensing on Smartphones. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. New York, NY, USA : ACM, 2010. – DOI 10.1145/1814433.1814464. – ISBN 978–1–60558–985–5, S. 315–330
- [255] SJURTS, Insa (Hrsg.): *Gabler Lexikon Medienwirtschaft*. 2., aktualisierte und erweiterte Auflage. Wiesbaden : Springer Gabler, 2011. – 702 S. – ISBN 978–3–8349–6487–8. – Stichwort: Smartphone, S. 564–565

## Internetquellen

- [256] ANDERSEN, Andreas B.; KRØGHOLT, Pelle; BIERRE, Stine; TABARD, Aurélien: *NoxDroid – A Bicycle Sensor System for Air Pollution Monitoring*. <http://noxdroid.org/images/2011-12/Noxdroid.pdf>, Dezember 2011. – Letzter Zugriff: 13. Oktober 2014
- [257] AQUALAB RESEARCH GROUP, EECS DEPARTMENT, NORTHWESTERN UNIVERSITY: *SWANS++ (alpha) – Extensions to the Scalable Wireless Ad-hoc Network Simulator*. <http://www.aqualab.cs.northwestern.edu/projects/143-swans-extensions-to-the-scalable-wireless-ad-hoc-network-simulator>, April 2007. – Letzter Zugriff: 15.06.2012
- [258] BARTH, Dave: *The bright side of sitting in traffic: Crowdsourcing road congestion data*. <http://googleblog.blogspot.de/2009/08/bright-side-of-sitting-in-traffic.html>, August 2009. – Letzter Zugriff: 16.05.2014
- [259] BITKOM: *Smartphones stärker verbreitet als normale Handys*. [http://www.bitkom.org/files/documents/BITKOM-Presseinfo\\_Smartphone-Nutzung\\_in\\_Deutschland\\_11\\_06\\_2014\(1\).pdf](http://www.bitkom.org/files/documents/BITKOM-Presseinfo_Smartphone-Nutzung_in_Deutschland_11_06_2014(1).pdf), Juni 2014. – Letzter Zugriff: 04.11.2014
- [260] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *Sicherheit in drahtlosen Sensornetzwerken*. [https://www.bsi.bund.de/DE/Themen/weitereThemen/DrahtloseSensornetzwerke/DrahtloseSensornetzwerke\\_node.html](https://www.bsi.bund.de/DE/Themen/weitereThemen/DrahtloseSensornetzwerke/DrahtloseSensornetzwerke_node.html), März 2013. – Letzter Zugriff: 03.11.2014

- [261] BUNDESNETZAGENTUR FÜR ELEKTRIZITÄT, GAS, TELEKOMMUNIKATION, POST UND EISENBAHNEN: *Teilnehmerentwicklung im Mobilfunk*. [http://www.bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen\\_Institutionen/Marktbeobachtung/Deutschland/Mobilfunkteilnehmer/Mobilfunkteilnehmer\\_node.html](http://www.bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Marktbeobachtung/Deutschland/Mobilfunkteilnehmer/Mobilfunkteilnehmer_node.html), August 2014. – Letzter Zugriff: 04.11.2014
- [262] CARRAPETTA, James; YOUDALE, Nik; CHOW, Amanda; SIVARAMAN, Vijay: *Haze Watch Project*. <http://www.pollution.ee.unsw.edu.au/>, 2010. – Letzter Zugriff: 17.06.2013
- [263] GOOGLE: *Google Play - Supported devices*. <https://storage.googleapis.com/support-kms-prod/CBD5B3E453BC19229BFB46447F033B35E316>, September 2014. – Letzter Zugriff: 30.09.2014
- [264] INFORMATIONSZENTRUM MOBILFUNK E.V. (IZMF): *Wie sind Funkzellen und Mobilfunknetze beschaffen?* <http://www.izmf.de/de/content/wie-sind-funkzellen-und-mobilfunknetze-beschaffen>, September 2011. – Letzter Zugriff: 22.09.2014
- [265] INITIATIVE D21 E.V.: *Mobile Internetnutzung*. <http://www.initiatived21.de/portfolio/mobile-internetnutzung-2014/>, Dezember 2014. – Letzter Zugriff: 02.12.2014
- [266] INTERNATIONAL TELECOMMUNICATION UNION: *Mobile-cellular subscriptions*. [http://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2014/Mobile\\_cellular\\_2000-2013.xls](http://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2014/Mobile_cellular_2000-2013.xls), 2014. – Letzter Zugriff: 04.11.2014
- [267] KOTZ, David; HENDERSON, Tristan; ABYZOV, Ilya; YEO, Jihwang: *CRAWDAD data set dartmouth/campus (v. 2007-02-08)*. <http://crawdad.org/dartmouth/campus/>, Februar 2007. – Letzter Zugriff: 08.05.2014
- [268] LANDESHAUPTSTADT MÜNCHEN: REFERAT FÜR GESUNDHEIT UND UMWELT: *Luftmessstationen in München (LÜB-Stationen)*. <http://maps.muenchen.de/rgu/luftmessstationen>, 2014. – Letzter Zugriff: 01.07.2014
- [269] LANDESHAUPTSTADT MÜNCHEN: *Statistik*. <http://www.muenchen.de/rathaus/Stadtinfos/Statistik/Bev-lkerung.html>, September 2014. – Letzter Zugriff: 31.10.2014
- [270] MICROSOFT: *Microsoft Band*. <http://www.microsoft.com/microsoft-band/>, Oktober 2014. – Letzter Zugriff: 05.11.2014
- [271] NOKIA RESEARCH CENTER (NRC): *Sensing the World with Mobile Devices*. [http://research.nokia.com/sites/default/files/NTI\\_Sensing\\_-\\_Dec\\_2008.pdf](http://research.nokia.com/sites/default/files/NTI_Sensing_-_Dec_2008.pdf), Dezember 2008. – Letzter Zugriff: 03.11.2014

- [272] OPENSIGNAL: *Android Fragmentation Report 2014*. [http://opensignal.com/assets/pdf/reports/2014\\_08\\_fragmentation\\_report.pdf](http://opensignal.com/assets/pdf/reports/2014_08_fragmentation_report.pdf), August 2014. – Letzter Zugriff: 30.09.2014
- [273] OSTLUFT: *Luftqualität in der Ostschweiz und in Liechtenstein*. <http://www.ostluft.ch/>, 2014. – Letzter Zugriff: 01.07.2014
- [274] SAMSUNG ELECTRONICS: *Samsung Galaxy S5*. <http://www.samsung.com/de/microsite/galaxys5/features.html>, April 2014. – Letzter Zugriff: 04.11.2014
- [275] SAMSUNG TOMORROW: *What You May Not Know About GALAXY S4 Innovative Technology*. <http://global.samsungtomorrow.com/?p=23610>, April 2013. – Letzter Zugriff: 14.10.2014
- [276] SCHWAN, Ben: *Mini-Spektrometer fürs Handy*. <http://www.heise.de/tr/artikel/Mini-Spektrometer-fuers-Handy-2059100.html>, Januar 2014. – Letzter Zugriff: 05.11.2014
- [277] SCHWAN, Ben: *Smartphone hart in der Brise*. <http://www.heise.de/tr/artikel/Smartphone-hart-in-der-Brise-2298560.html>, September 2014. – Letzter Zugriff: 05.11.2014
- [278] SOFTBANK MOBILE: *PANTONE 5 107SH*. <http://mb.softbank.jp/en/products/sharp/107sh.html>, Mai 2012. – Letzter Zugriff: 05.11.2014
- [279] STADT WOLFRATSHAUSEN: *Stadtinfo: Zahlen - Daten - Fakten*. <https://www.wolfratshausen.de/rathaus/stadtinfo/>, Januar 2013. – Letzter Zugriff: 31.10.2014
- [280] WAZE.COM: *Participate in Waze's Holiday Treasure Hunt!* <http://blog.waze.com/2009/12/participate-in-wazes-holiday-treasure.html>, Dezember 2009. – Letzter Zugriff: 20.10.2014
- [281] WORLD METEOROLOGICAL ORGANIZATION: *Nowcasting*. <http://www.wmo.int/pages/prog/amp/pwsp/Nowcasting.htm>, Februar 2007. – Letzter Zugriff: 30.10.2014
- [282] JiST / SWANS: *Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator - v1.0.6*. <http://jist.ece.cornell.edu/>, März 2005. – Letzter Zugriff 15.06.2012



# Eigene Publikationen

- [283] BROLL, Gregor; HAMARD, John; PAOLUCCI, Massimo; HAARLÄNDER, Markus; WAGNER, Matthias; STORPAES, Sven; RUKZIO, Enrico; SCHMIDT, Albrecht; WIESNER, Kevin: Mobile Interaction with Web Services Through Associated Real World Objects. In: *Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '07)*. New York, NY, USA : ACM, September 2007. – DOI 10.1145/1377999.1378025. – ISBN 978-1-59593-862-6, S. 319–321
- [284] DORFMEISTER, Florian; WIESNER, Kevin; SCHUSTER, Michael; MAIER, Marco: Preventing Restricted Space Inference in Online Route Planning Services. – Eingereicht: *The 13th International Conference on Mobile Systems, Applications, and Services (MobiSys 2015)*, November 2014.
- [285] DUCHON, Markus; DÜRR, Michael; WIESNER, Kevin: Kollaboratives Parkplatzmanagement: Ein Community basierter Ansatz. In: WERNER, Martin (Hrsg.); ROTH, Jörg (Hrsg.): 8. *GI/KuVS-Fachgespräch „Ortsbezogene Anwendungen und Dienste“*. Berlin : Logos, Oktober 2011. – ISBN 978-3-8325-3041-9, S. 181–191
- [286] DUCHON, Markus; WIESNER, Kevin; MÜLLER, Alexander; LINNHOFF-POPIEN, Claudia: Collaborative Sensing Platform for Eco Routing and Environmental Monitoring. In: MARTINS, Francisco (Hrsg.); LOPES, Luís (Hrsg.); PAULINO, Hervé (Hrsg.): *Sensor Systems and Software*. Bd. 102. Berlin / Heidelberg : Springer, 2012. – DOI 10.1007/978-3-642-32778-0\_8. – ISBN 978-3-642-32777-3, S. 89–104
- [287] DÜRR, Michael; DUCHON, Markus; WIESNER, Kevin; SEDLMEIER, Andreas: Distributed Group and Rights Management for Mobile Ad Hoc Networks. In: *Proceedings of the 4th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011)*. New York, NY, USA : IEEE, Oktober 2011. – DOI 10.1109/WMNC.2011.6097258. – ISBN 978-1-4577-1192-3, S. 1–8
- [288] DÜRR, Michael; MAIER, Marco; WIESNER, Kevin: An Analysis of Query Forwarding Strategies for Secure and Privacy-Preserving Social Networks. In: *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM '12)*. Washington, DC, USA : IEEE Computer Society, 2012. – DOI 10.1109/ASONAM.2012.92. – ISBN 978-0-7695-4799-2, S. 535–542

- [289] DÜRR, Michael; MARCUS, Philipp; WIESNER, Kevin: Secure, Privacy-Preserving, and Context-Restricted Information Sharing for Location-based Social Networks. In: *Proceedings of the Seventh International Conference on Wireless and Mobile Communications (ICWMC 2011)*. Luxemburg, Luxemburg : IARIA, Juni 2011. – ISBN 978–1–61208–140–3, S. 243–248
- [290] DÜRR, Michael; WIESNER, Kevin: A Privacy-Preserving Social P2P Infrastructure for People-Centric Sensing. In: LUTTENBERGER, Norbert (Hrsg.); PETERS, Hagen (Hrsg.): *17th GI/ITG Conference on Communication in Distributed Systems (KiVS 2011)*. Bd. 17. Dagstuhl, Germany : Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011 (OpenAccess Series in Informatics (OASIS)). – DOI 10.4230/OASIS.KiVS.2011.176. – ISBN 978–3–939897–27–9, S. 176–181
- [291] FRANK, Korbinian; ROBERTSON, Partick; GROSS, Michael; WIESNER, Kevin: Sensor-based Identification of Human Stress Levels. In: *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2013)*. New York, NY, USA : IEEE, März 2013. – DOI 10.1109/PerComW.2013.6529469. – ISBN 978–1–4673–5075–4, S. 127–132
- [292] HAAS, Stefan; WIESNER, Kevin; STONE, Thomas C.: Car Ride Classification for Drive Context Recognition. In: *Proceedings of the Fourth International Conference on Mobile Services, Resources, and Users (MOBILITY 2014)*. Paris, Frankreich : IARIA, Juli 2014. – ISBN 978–1–61208–366–7, S. 61–66
- [293] KEIS, Felix; WIESNER, Kevin: Participatory Sensing Utilized by an Advanced Meteorological Nowcasting System. In: *Proceedings of the IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2014)*. New York, NY, USA : IEEE, April 2014. – DOI 10.1109/ISSNIP.2014.6827629. – ISBN 978–1–4799–2842–2, S. 1–6
- [294] PROTSCHKY, Valentin; WIESNER, Kevin; FEIT, Stefan: Adaptive Traffic Light Prediction via Kalman Filtering. In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IV 2014)*. New York, NY, USA : IEEE, Juni 2014. – DOI 10.1109/IVS.2014.6856394. – ISBN 978–1–4799–3637–3, S. 151–157
- [295] VACULÍN, Roman; WIESNER, Kevin; SYCARA, Katia: Exception Handling and Recovery of Semantic Web Services. In: *Proceedings of the Fourth International Conference on Networking and Services (ICNS 2008)*. New York, NY, USA : IEEE, März 2008. – DOI 10.1109/ICNS.2008.35. – ISBN 978–0–7695–3094–9, S. 217–222

- [296] WERNER, Martin; KESSEL, Moritz; GSCHWANDTNER, Florian; DÜRR, Michael; WIESNER, Kevin; MAIR, Thomas: Technologische Herausforderungen für kontextsensitive Geschäftsanwendungen. In: VERCLAS, Stephan (Hrsg.); LINNHOFF-POPIEN, Claudia (Hrsg.): *Smart Mobile Apps: Mit Business-Apps ins Zeitalter mobiler Geschäftsprozesse*. Berlin / Heidelberg : Springer, 2012. – DOI 10.1007/978-3-642-22259-7\_19. – ISBN 978-3-642-22258-0, S. 285–300
- [297] WIESNER, Kevin: Towards Semantic-enabled Integration of Virtual Sensor Services. In: *Proceedings of the "Mobile Context-Awareness: Capabilities, Challenges and Applications" Workshop at the ACM UbiComp '10*. Kopenhagen, Dänemark, September 2010, S. 1–4
- [298] WIESNER, Kevin; DORFMEISTER, Florian; LINNHOFF-POPIEN, Claudia: Privacy-Preserving Calibration for Participatory Sensing. In: STOJMIENOVIC, Ivan (Hrsg.); CHENG, Zixue (Hrsg.); GUO, Song (Hrsg.): *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Bd. 131. Cham, Schweiz : Springer International Publishing, 2014. – DOI 10.1007/978-3-319-11569-6\_22. – ISBN 978-3-319-11568-9, S. 276–288
- [299] WIESNER, Kevin; DORFMEISTER, Florian; LINNHOFF-POPIEN, Claudia: PRICAPS: A System for Privacy-Preserving Calibration in Participatory Sensing Networks. In: *EAI Endorsed Transactions on Ubiquitous Environments* 14 (2014), November, Nr. 3. – DOI 10.4108/ue.1.3.e5. – ISSN 2032-9377, S. 5:1–5:11
- [300] WIESNER, Kevin; DUCHON, Marcus; DÜRR, Michael: Distributed Multi-Head Clustering for People-Centric Sensor Networks. In: *Proceedings of the 6th International Conference on Sensor Technologies and Applications (SENSORCOMM 2012)*. Rome, Italy : IARIA, August 2012. – ISBN 978-1-61208-207-3, S. 53–58
- [301] WIESNER, Kevin; DÜRR, Michael; DUCHON, Markus: Private Pooling: A Privacy-Preserving Approach for Mobile Collaborative Sensing. In: PRASAD, Ramjee (Hrsg.); FARKAS, Károly (Hrsg.); SCHMIDT, Andreas U. (Hrsg.); LIOY, Antonio (Hrsg.); RUSSELLO, Giovanni (Hrsg.); LUCCIO, Flaminia L. (Hrsg.): *Security and Privacy in Mobile Information and Communication Systems*. Bd. 94. Berlin / Heidelberg : Springer, 2012. – DOI 10.1007/978-3-642-30244-2\_5. – ISBN 978-3-642-30244-2, S. 52–63
- [302] WIESNER, Kevin; FELD, Sebastian: Energy-efficient Tasking in Participatory Sensing Systems. In: SCHAU, Volkmar (Hrsg.); EICHLER, Gerald (Hrsg.); ROTH, Jörg (Hrsg.): *Ortsbezogene Anwendungen und Dienste (LBAS 2013)*. Berlin : Logos Verlag, November 2014. – ISBN 978-3-8325-3813-2, S. 127–136

- [303] WIESNER, Kevin; FELD, Sebastian; DORFMEISTER, Florian; LINNHOFF-POPIEN, Claudia: Right to Silence: Establishing Map-based Silent Zones for Participatory Sensing. In: *Proceedings of the IEEE ISSNIP 2014 - Symposium on Participatory Sensing and Crowdsourcing*. New York, NY, USA : IEEE, April 2014. – DOI 10.1109/ISSNIP.2014.6827657. – ISBN 978-1-4799-2842-2, S. 1-6
- [304] WIESNER, Kevin; FOTH, Marcus; BILANDZIC, Mark: Unleashing Creative Writers: Situated Engagement with Mobile Narratives. In: *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7 (OZCHI '09)*. New York, NY, USA : ACM, November 2009. – DOI 10.1145/1738826.1738901. – ISBN 978-1-60558-854-4, S. 373-376
- [305] WIESNER, Kevin; FOTH, Marcus; BILANDZIC, Mark; KRCCMAR, Helmut: Restrictions and Constraints in Mobile Narratives for Place-based Community Engagement. In: *Proceedings of the Community Practices and Locative Media Workshop at MobileHCI '09*. Bonn, September 2009, S. 1-4
- [306] WIESNER, Kevin; VACULÍN, Roman; KOLLINGBAUM, Martin; SYCARA, Katia: Recovery Mechanisms for Semantic Web Services. In: MEIER, René (Hrsg.); TERZIS, Sotirios (Hrsg.): *Distributed Applications and Interoperable Systems*. Bd. 5053. Berlin / Heidelberg : Springer, 2008. – DOI 10.1007/978-3-540-68642-2\_8. – ISBN 978-3-540-68639-2, S. 100-105