

Aus dem Institut für Telematik  
der Universität zu Lübeck

Direktor:  
Prof. Dr. rer. nat. Stefan Fischer

# **Zeitliches und räumliches Kontextbewusstsein in drahtlosen Sensornetzen**

Inauguraldissertation  
zur  
Erlangung der Doktorwürde  
der Universität zu Lübeck  
– Aus der Technisch-Naturwissenschaftlichen Fakultät –

Vorgelegt von  
Herrn Dipl.-Inform. Carsten Buschmann  
aus Gifhorn

Lübeck, im Dezember 2008

Erster Berichterstatter: Prof. Dr. rer. nat. Stefan Fischer  
Zweiter Berichterstatter: Prof. Dr.-Ing. Norbert Luttenberter  
(Christian-Albrechts-Universität zu Kiel)  
Dritter Berichterstatter: Prof. Dr. Martin Mauve  
(Heinrich-Heine-Universität Düsseldorf)  
Tag der mündlichen Prüfung: 30. Januar 2009

Zum Druck genehmigt.

Lübeck, den 30. Januar 2009

gez. Prof. Dr. Jürgen Prestin

– Dekan der Technisch-Naturwissenschaftlichen Fakultät –

# Vorwort

Forschungsarbeit ist Teamarbeit. Diese Erkenntnis wird einem selten so deutlich vor Augen geführt wie beim Verfassen der Dissertationsschrift, denn der Weg zu ihrer Fertigstellung wäre ohne die Unterstützung einer Vielzahl von Personen sehr viel steiniger gewesen.

Da sind zum Einen diejenigen, die diese Arbeit durch die von ihnen geschaffenen Rahmenbedingungen überhaupt erst ermöglicht haben. Das gilt insbesondere und ganz konkret für meinen Doktorvater: Ich danke Prof. Stefan Fischer ganz herzlich dafür, dass er mir bei meiner Arbeit alle nur erdenklichen Freiräume gelassen und mich stets mit seinem Vertrauen und hilfreichem Rat unterstützt hat. Das gilt aber auch für meine Eltern, die mich mit ihrer jederzeit liebevollen Unterstützung ein Leben lang begleitet haben.

Für die Übernahme des Koreferats bedanke ich mich bei Prof. Norbert Luttenberger ebenso wie für die gute Zusammenarbeit im Forschungsprojekt SWARMS. Die in dieser Arbeit vorgestellten Ergebnisse wären ohne die enge Zusammenarbeit und die fruchtbaren Diskussionen mit anderen Forschern nicht möglich gewesen. Die Forschungen zum Thema eines konsistenten Zeitbewusstseins und TICTAC sind gemeinsame Arbeiten mit Dr. Dennis Pfisterer und Prof. Stefan Fischer. NIDES ist in Zusammenarbeit mit Dr. Alexander Kröller und Prof. Sándor P. Fekete von der Technischen Universität Braunschweig, Prof. Horst Hellbrück von der Fachhochschule Lübeck sowie Dr. Dennis Pfisterer, Prof. Christian Werner und Prof. Stefan Fischer entstanden. Die Untersuchungen der räumlichen Approximation von Phänomenkonturen wurden gemeinsam mit Daniela Krüger und Prof. Stefan Fischer durchgeführt. Sie sind allesamt auch Koautoren der Publikationen, in denen unterschiedlichen Teilaspekte dieser Arbeit der wissenschaftlichen Öffentlichkeit vorgestellt wurden. Ich bedanke mich bei ihnen für die fruchtbare Kooperation.

Für ihre Hilfe im Kampf mit allerlei technischen Herausforderungen gilt mein Dank Dr. Dennis Pfisterer, Prof. Christian Werner, Axel Wegener, Stefan Ransom und Dirk-Frank Schmidt. Das gilt ebenso für Daniela Krüger, Birgit Schneider und meine Eltern, die sich mit unendlicher Geduld und nicht nachlassendem Eifer durch den von mir verfassten wirren Text gekämpft und mit ihren vielen Korrekturen und Kommentaren geholfen haben, ihn in eine lesbare Arbeit zu verwandeln.

Zu erwähnen bleiben meine Kollegen Martin Lipphardt, Peter Rothenpieler, Nils Glombitza, Claudia Becker, Maick Danckwardt, Alexander Carôt und Stephan Pöhlsen, die jene angenehme und inspirierende Atmosphäre am Institut für Telematik geschaffen haben, die die tägliche Arbeit zum sprichwörtlichen Vergnügen macht.



# Kurzfassung

Die Kombination von Sensorik, digitaler Datenverarbeitung und drahtloser Kommunikation führte in den letzten Jahren zu der Entwicklung drahtloser Sensornetze. Sie bestehen aus Geräten, die nicht nur Daten durch Messung erheben und dann verarbeiten und aufbereiten, sondern sie auch drahtlos untereinander austauschen können. Die Sensordaten allein sind jedoch praktisch bedeutungslos, sie erhalten ihre Aussagekraft erst durch Informationen über den Kontext ihrer Erhebung. Kontextinformationen können darüber hinaus auch zur Optimierung des Anwendungsverhaltens herangezogen werden. Aufgrund der Wichtigkeit des Kontextes in Sensornetzen werden in dieser Arbeit Verfahren vorgestellt, die Kontextinformationen bezüglich Zeit und Ort zur Verfügung stellen.

Im Hinblick auf den Kontextaspekt Zeit werden Verfahren zur Errichtung eines konsistenten Zeitbewusstseins auf ihren Kommunikationsoverhead hin untersucht, da Energie und Kanalkapazität knappe Ressourcen sind und folglich ein geringes Datenaufkommen vorteilhaft ist. Es wird ein neues Verfahren vorgestellt, das kontextadaptiv sein Verhalten so anpasst, dass der Overhead minimiert wird. Basierend auf der simulativen Untersuchung des neuen Verfahrens im Vergleich mit zwei bestehenden Ansätzen wird ein Modell zur Vorhersage des Overheads aller drei Methoden entwickelt. So können sowohl zum Zeitpunkt der Anwendungsentwicklung als auch zur Laufzeit Entscheidungen hinsichtlich der anzuwendenden Variante getroffen werden.

Eine für Sensornetze besonders geeignete Methode zur Bestimmung des räumlichen Kontextes ist die Multilateration. Weil dabei die Positionen der Sensorknoten auf Basis von Referenzpunkten und Distanzen bestimmt werden und bestehende Verfahren zur Distanzermittlung Schwächen aufweisen, wird ein neues Verfahren zur Distanzschätzung zwischen benachbarten Sensorknoten vorgestellt. Es basiert auf der Tatsache, dass nahe beieinander liegende Sensorknoten mehr gemeinsame Kommunikationspartner haben als solche, die weiter voneinander entfernt sind. Es wird gezeigt, wie die Distanzen, ausgehend von einem Modell der Funkübertragung, ermittelt werden können und wie das Verfahren zu einer Multihop-Distanzschätzung erweitert werden kann. Die Genauigkeit des Verfahrens wird in Simulationen untersucht und die Ergebnisse vorgestellt. Um aus den laterativ bestimmten Positionskoordinaten höherwertige Informationen über unterschiedliche Bereiche eines Sensornetzes ableiten zu können, wird ein Verfahren zur räumlichen Approximation von Konturen beschrieben. Es kann für alle Arten von Phänomenen, die ein Sensornetz beobachten kann, kompakte Polygonrepräsentationen ihrer Ausdehnung bestimmen. Mit Hilfe von Simulation wird gezeigt, dass die Approximationen robust gegen Fehler und trotz ihrer Kompaktheit genau sind.



# Inhaltsverzeichnis

<b>Vorwort</b>	<b>iii</b>
<b>Kurzfassung</b>	<b>v</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	2
1.2. Wissenschaftliche Beiträge und Struktur der Arbeit . . . . .	3
<b>2. Grundlagen drahtloser Sensornetze</b>	<b>7</b>
2.1. Sensorknoten . . . . .	7
2.2. Sensornetze und ihre Anwendung . . . . .	9
2.3. Anforderungen an Sensornetze . . . . .	11
2.3.1. Spontane Einsatzbereitschaft . . . . .	11
2.3.2. Skalierbarkeit . . . . .	12
2.3.3. Kosten . . . . .	13
2.3.4. Kontextbewusstsein . . . . .	13
2.3.5. Baugröße . . . . .	14
2.4. Eigenschaften von Sensornetzen . . . . .	14
2.4.1. Beschränkung des Energievorrats . . . . .	14
2.4.2. Beschränkung von Rechenleistung und Speicherkapazität . . . . .	15
2.4.3. Zuverlässigkeitsdefizite . . . . .	16
2.5. Eigenschaften und Modellierung drahtloser Kommunikation in Sensornetzen . . . . .	17
2.6. Zusammenfassung . . . . .	19
<b>3. Grundlagen des Kontextbewusstseins</b>	<b>21</b>
3.1. Zeitbewusstsein . . . . .	22
3.1.1. Verfahren zur Errichtung von Zeitbewusstsein . . . . .	24
3.1.2. Ermittlung von Zeitdifferenzen . . . . .	24
3.1.3. Anpassung der lokalen Uhrzeit . . . . .	27
3.1.4. Fortführung über mehrere Hops . . . . .	28
3.1.5. Driftkompensation . . . . .	29
3.2. Positionsbewusstsein . . . . .	30
3.2.1. Numerische und symbolische Positionen . . . . .	30
3.2.2. Verfahren zur Errichtung von numerischem Positionsbewusstsein . . . . .	31
3.3. Zusammenfassung . . . . .	42
<b>4. Kontextadaptive Errichtung konsistenten Zeitbewusstseins</b>	<b>45</b>

4.1.	Kommunikationsaufkommen bestehender Verfahren . . . . .	45
4.1.1.	Proaktive Synchronisation . . . . .	45
4.1.2.	Post-facto Synchronisation . . . . .	46
4.1.3.	Zusammenhang zwischen Kommunikation und Overhead . . . . .	46
4.2.	Ein kommunikationsabhängiges Verfahren zur Errichtung konsistenten Zeitbewusstseins . . . . .	47
4.3.	Simulative Evaluation . . . . .	51
4.3.1.	Simulationsumgebung . . . . .	51
4.3.2.	Simulationsergebnisse . . . . .	53
4.4.	Modellbildung . . . . .	55
4.4.1.	Modellbildung für Synchronisation mittels Fluten . . . . .	56
4.4.2.	Modellbildung für das Verfahren von Römer . . . . .	56
4.4.3.	Modellbildung für TICTAC . . . . .	57
4.5.	Zusammenfassung . . . . .	59
<b>5.</b>	<b>Nachbarschaftsbasierte Distanzschätzung</b>	<b>61</b>
5.1.	Verwandte Arbeiten . . . . .	61
5.1.1.	Anforderungen an Distanzschätzverfahren . . . . .	62
5.1.2.	Verfahren zur Distanzermittlung . . . . .	63
5.1.3.	Distanzschätzungen auf Basis der Funkschnittstelle . . . . .	66
5.1.4.	Multihop-Distanzschätzungen . . . . .	69
5.1.5.	Defizite bestehender Verfahren . . . . .	70
5.2.	NIDES: Nachbarschaftsbasierte Singlehop-Distanzschätzung . . . . .	71
5.2.1.	Grundlegende Funktionsweise . . . . .	71
5.2.2.	Einflussfaktoren . . . . .	73
5.2.3.	Mathematische Herleitung . . . . .	76
5.2.4.	Implementierung . . . . .	83
5.3.	Simulative Evaluation der Distanzschätzung zwischen Nachbarn . . . . .	87
5.3.1.	Simulationsumgebung . . . . .	87
5.3.2.	Leistungsfähigkeit von NIDES in Abhängigkeit von der Netzdichte . . . . .	89
5.3.3.	Einfluss des Abstandes der schätzenden Knoten . . . . .	91
5.3.4.	Einfluss schwankender Dichte . . . . .	93
5.3.5.	Einfluss des Radiomodells . . . . .	95
5.4.	Erweiterung von NIDES zur Multihop-Distanzschätzung . . . . .	97
5.4.1.	Auswirkungen von Dichteschwankungen . . . . .	98
5.4.2.	Adaption der Schätzfunktion an Multihop-Distanzen . . . . .	100
5.5.	Simulative Evaluation von Multihop-Distanzschätzungen . . . . .	101
5.5.1.	Simulationsumgebung . . . . .	101
5.5.2.	Auswertung . . . . .	102
5.6.	Eigenschaften von NIDES . . . . .	103
5.7.	Zusammenfassung . . . . .	105
<b>6.</b>	<b>Räumliche Approximation von Phänomenkonturen</b>	<b>107</b>
6.1.	Verwandte Arbeiten . . . . .	108
6.1.1.	Anforderungen an Verfahren zur Phänomenapproximation . . . . .	109
6.1.2.	Verfahren zur Kantenerkennung . . . . .	109

6.1.3.	Verfahren zur Konturapproximation . . . . .	111
6.2.	Ein Algorithmus zur Konturermittlung und -approximation . . .	112
6.2.1.	Einzelmessung . . . . .	113
6.2.2.	Voting . . . . .	114
6.2.3.	Kantendetektion . . . . .	116
6.2.4.	Netzwerkplanarisierung . . . . .	116
6.2.5.	Konturapproximation . . . . .	117
6.2.6.	Illustration anhand eines Beispiels . . . . .	120
6.3.	Simulative Evaluation . . . . .	126
6.3.1.	Simulationsumgebung . . . . .	126
6.3.2.	Robustheit gegen Messfehler . . . . .	128
6.3.3.	Robustheit gegen Paketverluste . . . . .	129
6.3.4.	Genauigkeit der Konturermittlung . . . . .	130
6.3.5.	Qualität der Approximation . . . . .	133
6.3.6.	Zusammenfassende Veranschaulichung . . . . .	134
6.4.	Eigenschaften des Verfahrens . . . . .	135
6.5.	Zusammenfassung . . . . .	136
<b>7.</b>	<b>Zusammenfassung und Ausblick</b>	<b>139</b>
<b>A.</b>	<b>Verzeichnisse</b>	<b>143</b>
	Abbildungsverzeichnis . . . . .	143
	Abkürzungsverzeichnis . . . . .	146
	Literaturverzeichnis . . . . .	147
<b>B.</b>	<b>Persönliche Informationen</b>	<b>167</b>
	Lebenslauf . . . . .	169
	Eigene Publikationen . . . . .	171



# 1. Einleitung

In den letzten Jahren fand eine rasante Entwicklung in den Bereichen der digitalen Datenverarbeitung, der drahtlosen Kommunikation und der Sensorik statt. Die verfügbaren Hardwarekomponenten wurden immer leistungsfähiger und dabei zugleich kleiner. Eine Verschmelzung dieser drei Bereiche erschloss ihnen ein neues Einsatzgebiet: die drahtlosen Sensornetze.

Diese bestehen aus sogenannten Sensorknoten, die Prozessor, Speicher, drahtlose Kommunikationstechnik, Energieversorgung sowie Sensorik bereits heute in einem Volumen von wenigen Kubikzentimetern vereinen. Solche Geräte können nicht nur praktisch überall feingranular Daten messen, verarbeiten und aufbereiten, sondern sie auch drahtlos mit anderen Geräten austauschen.

Im Gegensatz zu kabelgebundenen Messsystemen können sich Sensornetze spontan konstituieren und selbsttätig konfigurieren. Dadurch reduziert sich der Installationsaufwand erheblich, und es können nach Bedarf jederzeit drahtlose Sensorknoten hinzugefügt oder entfernt werden. Darüber hinaus profitieren Sensornetze von der Möglichkeit der Kooperation der Sensorknoten: es können von verschiedenen Geräten erhobene Daten korreliert und zu einem umfassenden Bild der Situation zusammengeführt werden.

Das langfristige Ziel ist es, der Vision der intelligenten Umgebung einen entscheidenden Schritt näher zu kommen: Sensorknoten erheben Eigenschaften von Umgebungsobjekten und machen sie direkt oder über Weitverkehrsnetze hinweg unterschiedlichsten Informationssystemen verfügbar.

Ihre Einbettung in die Umgebung lässt eine Brücke zwischen virtueller und realer Welt entstehen. Visionäre prophezeihen eine massive Verkleinerung der Sensorknoten bis hin zu „intelligenten Staubkörnern“, die eine Vielzahl neuer Anwendungsgebiete für Sensornetze erschließen würde. Millionen dieser winzigen Partikel könnten aus Flugzeugen über Katastrophen- oder Kampfgebieten abgeworfen werden und den Einsatzkräften umfassende Lagebilder liefern. Farben und Baustoffen beigemischt könnten sie intelligente Konstruktionen ermöglichen, die sich selbst auf Materialermüdung überwachen. Auch der medizinische Einsatz ist denkbar: Sensorelemente klein wie Blutkörperchen könnten in den Blutkreislauf eingebracht werden, um kontinuierlich Informationen über die Blutzirkulation oder den Blutdruck und Sauerstoffgehalt zu liefern. Prozessoptimierungen in der industriellen Fertigung oder die Überwachung von Logistikketten sind weitere Anwendungsbeispiele.

## 1.1. Motivation

Bis zur Realisierung dieser Visionen müssen allerdings noch zahlreiche Herausforderungen in verschiedenen Bereichen überwunden werden.

Während die Chiptechnologie durchaus bereits heute den Großteil der benötigten Funktionalität für Prozessor, Speicher, Sensoranbindung und Funktechnologie im Volumen weniger Kubikmillimeter unterzubringen vermag, mangelt es weiterhin an kompakten Energieversorgungen. Selbst wenn man heutige Kapazitäten in wenigen Kubikmillimetern unterbringen könnte, wird der Energieverbrauch der Geräte deutlich sinken müssen, um befriedigende Laufzeiten erzielen zu können. Nicht zuletzt die Schwierigkeit, ausreichend kleine und gleichzeitig effiziente Antennen zu konstruieren, lässt die Realisierung von Staubkörnern, die mittels elektromagnetischer Wellen kommunizieren, in weite Ferne rücken.

Neben diesen Hardwareaspekten bestehen aber auch im Bereich der Protokolle und Anwendungen zahlreiche Herausforderungen. Schon der Prozess der Softwareentwicklung muss im Vergleich zu den heute gängigen Entwicklungsmethoden neu gestaltet werden, da Sensorknoten über keinerlei Benutzerschnittstellen verfügen und sehr enge Größenbeschränkungen des verfügbaren Speichers für Programm und Daten bedingen. Die Software muss aufgrund ihres Einsatzes in unbekanntem Umgebungen extrem flexibel und fehlertolerant sein. Die große Anzahl von Geräten macht das Verhalten von Protokollen auf globaler Ebene schwer vorhersagbar, außerdem kommt dem Aspekt der Skalierbarkeit eine wesentliche Bedeutung zu.

Ebenfalls problematisch ist die Tatsache, dass für sich stehende Sensordaten zunächst nichts weiter als bedeutungslose Skalare sind. Sie erhalten ihren Wert erst durch die Verknüpfung mit dem Kontext ihrer Erhebung. Ein Temperaturmesswert beispielsweise bekommt erst eine Aussagekraft, wenn klar ist, wann und wo die Messung erfolgte. Nur so lassen sich die Temperaturwerte sinnvoll vergleichen, zeitliche Entwicklungen werden erkennbar und räumliche Temperaturunterschiede können zum Beispiel auf einen Brandherd hinweisen.

Einerseits um Sensordaten mit Informationen über den Kontext ihrer Erhebung zu ergänzen, andererseits um der Sensornetzanwendung eine Anpassung an wechselnde Situationen zu ermöglichen, müssen Kontextinformationen in den Sensorknoten vorliegen. Allgemeiner formuliert müssen die Sensorknoten, beziehungsweise die auf ihnen laufenden Anwendungen, über Kontextbewusstsein verfügen. Dabei sind die beiden wichtigsten Kontextaspekte Zeit und Ort. Diese Informationen sind für die Interpretation von Sensordaten häufig unabdingbar, beispielsweise um zusammengehörige Sensordaten identifizieren und geeignet aggregieren zu können [115].

Aufgrund des Preises, des Energieverbrauches und der Baugröße ist es nicht möglich, jeden Sensorknoten mit einer speziellen Hardwarekomponente zur Bestimmung des Kontextes wie beispielsweise einem GPS-Empfänger, der Zeit- und Positionsinformationen liefert, auszustatten. Daher wird in der Literatur häufig vorgeschlagen, nur einige wenige Geräte mit entsprechender Hardware

auszustatten, die so erhaltenen Informationen dann mit Hilfe von speziellen Protokollen und Algorithmen auf alle anderen Geräte zu verteilen und diese so in die Lage zu versetzen, ihren eigenen Kontext zu bestimmen.

Bezüglich des zeitlichen Kontextes sind Sensornetze mit dem Problem konfrontiert, dass die Uhren der einzelnen Sensorknoten unabhängig voneinander gehen (also inkonsistent sind); sie starten zu verschiedenen Zeitpunkten und arbeiten potentiell mit unterschiedlichen Geschwindigkeiten. Das erschwert den zeitlichen Vergleich von Sensordaten unterschiedlicher Sensorknoten, da sich die entsprechenden Zeitstempel auf unterschiedliche Zeitbasen beziehen. Um diesem Problem zu begegnen, finden sich in der Literatur zwei prinzipiell verschiedene Vorgehensweisen: entweder gleicht man periodisch die Uhren aller Sensorknoten im Netz aneinander an, so dass sie synchron laufen oder man fügt jedem Datenpaket, das Zeitstempel enthält, zusätzliche Informationen hinzu, die die Transformation der Zeitstempel von der Zeitbasis des Senders in die des Empfängers ermöglichen.

Problematisch ist, dass sich beide Methoden nicht für alle Szenarien gleichermaßen eignen. Darüber hinaus können sie ihr Verhalten nicht an sich verändernde Bedingungen wie schwankendes Kommunikationsaufkommen im Sensornetz anpassen.

Bezüglich des räumlichen Kontextes gilt es, jeden Sensorknoten mit Informationen über seine Position zu versorgen. Diesen Prozess nennt man auch Lokalisierung. Eines der gängigsten Prinzipien dazu ist die Multilateration. Die Geräte mit Ortsinformationen, Anker genannt, fluten ihre Positionen durch das Sensornetz. Alle Knoten versuchen nun mit Hilfe sogenannter Distanzschätzverfahren, ihre Abstände untereinander zu bestimmen. Die Einzelabstände werden dann zu Informationen über den Abstand zu den Ankern zusammengeführt. Aus den durch das Fluten bekannten Ankerpositionen und den Distanzen zu den Ankern können die Sensorknoten dann ihre eigene Position näherungsweise berechnen.

Während die Algorithmen für die eigentliche Positionsrechnung bereits sehr ausgereift sind, weisen die Distanzschätzverfahren weiterhin Defizite auf. Sie arbeiten entweder ungenau, langsam oder benötigen spezielle zusätzliche Hardware, die die Sensorknoten teurer, größer und energiehungriger macht.

## **1.2. Wissenschaftliche Beiträge und Struktur der Arbeit**

Die vorliegende Arbeit liefert Beiträge zu den beiden zuvor thematisierten Problembereichen des Kontextes bezüglich Zeit und Raum in drahtlosen Sensornetzen.

Die beiden folgenden Kapitel machen den Leser zunächst mit den zum Verständnis der Arbeit benötigten Grundlagen vertraut. Die dann folgenden drei Kapitel beschreiben die eigenen wissenschaftlichen Beiträge des Autors. Sie bestehen aus

- einem neuen Verfahren zur kontextadaptiven lokalen Zeitsynchronisation,
- der simulativen Ermittlung des Kommunikationsaufkommens bei der Herstellung eines konsistenten Zeitbewusstseins und einem darauf aufbauenden Modell zu dessen Vorhersage,
- einem Verfahren zur nachbarschaftsbasierten Distanzschätzung sowie
- einem Protokoll zur räumlichen Approximation von Phänomenkonturen in Sensornetzen.

Kapitel 2 führt den Leser anhand der Literatur in die Funktionsweise von Sensornetzen ein, diskutiert Anforderungen an sie sowie ihre Eigenschaften. Gleichzeitig werden hieraus auch die relevanten Leitlinien für die Entwicklung von Algorithmen und Protokollen für Sensornetze abgeleitet.

Kapitel 3 stellt dem Leser die Literatur über Verfahren zur Erlangung von Kontextinformationen bezüglich Zeit und Raum vor. Die unterschiedlichen Vorschläge werden betrachtet und ihre Vor- und Nachteile diskutiert. Es wird deutlich, dass sich bezüglich des zeitlichen Kontextes im Wesentlichen zwei Verfahrensgruppen herausgebildet haben, während bei der Lokalisierung die Multilateration die Literatur klar dominiert. Da die Lateration die Sensorpositionen auf Basis von Ankerpositionen und Distanzen ermittelt, ergibt sich aus ihrer Dominanz die Relevanz von Distanzschätzungen für die Lokalisierung der Sensorknoten.

Kapitel 4 untersucht den Kommunikationsaufwand, den die beiden bereits kurz beschriebenen Verfahrensgruppen zur Herstellung eines konsistenten Zeitbewusstseins erzeugen. Es wird verdeutlicht, dass Aussagen darüber, welche der Methoden weniger Aufwand verursacht, von der Frage abhängen, wieviel sonstiger Datenaustausch im Netz stattfindet. Es wird in der Folge ein neues Verfahren namens *Traffic Induced Control of Time And Communication (TICTAC)* vorgeschlagen, das die Funktionsweise je eines Verfahrens aus den beiden Gruppen kombiniert und somit gestattet, abhängig vom Anwendungskontext die am besten geeignete Strategie zu nutzen. Es werden Simulationen vorgestellt, die einerseits Einflussfaktoren identifizieren und andererseits einen Vergleich des aus TICTAC sowie den beiden aus den anderen Verfahren resultierenden Aufwandes ermöglichen. Darüber hinaus wird ein Modell entwickelt, das Vorhersagen über den resultierenden Kommunikationsaufwand erlaubt. Es kann sowohl im Vorfeld als auch zur Laufzeit genutzt werden, um auf Basis von Applikationsparametern Entscheidungen über die einzusetzende Strategie zu fällen.

In Kapitel 5 wird ein neues Verfahren zur Distanzschätzung in Sensornetzen vorgestellt, das den Namen *Neighborhood Intersection Distance Estimation Scheme (NIDES)* trägt. Es basiert auf der erstmalig in [37] diskutierten Beobachtung, dass dicht beieinander liegende Sensorknoten mehr gemeinsame Kommunikationspartner haben als solche, die weiter voneinander entfernt sind. Es wird beschrieben, wie, ausgehend von einem Modell der Funkausbreitung der drahtlosen Kommunikationsschnittstelle, eine Distanzschätzung zwischen benachbarten Sensorknoten abgeleitet werden kann. Mit Hilfe von Simulationsergebnis-

sen wird belegt, dass die resultierenden Distanzschätzungen qualitativ mit signalstärkebasierten vergleichbar sind, obwohl der Aufwand des neuen Verfahrens geringer ist. Zusätzlich wird aufgezeigt, wie eine Verallgemeinerung zu einer Multihop-Distanzschätzung erfolgen kann, die beispielsweise für die Lokalisierung von Sensorknoten benötigt wird. In Verbindung mit den in Abschnitt 3.2 vorgestellten Multilaterationsverfahren können die Sensorknoten auf dieser Basis ihre Positionen in Form von Koordinaten bestimmen.

Kapitel 6 beschreibt ein Verfahren, das auf Basis solcher numerischen Positionsinformationen die räumliche Ausdehnung aller Phänomene, die von einem Sensornetz beobachtet werden können, und ihre Kontur approximieren kann. Anwendungsgebiete sind neben der Detektion von Ausbreitungsgebieten von Bränden, Überflutungen oder Leckagen die Erlangung sogenannter symbolischer Positionsinformationen. So können die Geräte ihr Verhalten nicht nur von ihren Koordinaten abhängig machen, sondern zum Beispiel auch davon, ob sie sich innerhalb oder außerhalb eines Phänomenbereiches befinden. In Simulationen wird belegt, dass das Verfahren genau und effizient arbeitet und dabei robust gegen Fehler ist.

Kapitel 7 beschließt die Arbeit mit einer Zusammenfassung sowie einem Ausblick auf zukünftige Arbeiten.



## 2. Grundlagen drahtloser Sensornetze

Die vorliegende Arbeit beschäftigt sich mit dem Kontextbewusstsein bezüglich Zeit und Raum speziell in Sensornetzen, deshalb sind zum Verständnis die Eigenschaften, Besonderheiten und resultierenden Einschränkungen dieses Anwendungsgebietes von großer Bedeutung. Daher macht das folgende Kapitel den Leser mit den entsprechenden Grundlagen drahtloser Sensornetze vertraut.

### 2.1. Sensorknoten

Drahtlose Sensornetze (oder kurz Sensornetze, beide Bezeichnungen werden im Folgenden synonym verwendet) bestehen aus sogenannten *Sensorknoten* (kurz: *Knoten*). Diese Geräte umfassen einen Prozessor mit Speicher, einen Transceiver (eine kombinierte Sende- und Empfangseinheit, der englische Begriff ist zusammengesetzt aus Transmitter und Receiver) zur drahtlosen Kommunikation, eine Energieversorgung sowie Sensorik. Der Speicher ist meist in zwei Bereiche unterteilt: Ein langsamer, nichtflüchtiger Teil, der meist in Flash-Technologie ausgeführt ist, wird zum dauerhaften Speichern insbesondere des Programmcodes verwendet. Ein schneller, flüchtiger Teil, der *Random Access Memory (RAM)* genannt wird, dient dem Speichern von Daten während der Programmverarbeitung. Abbildung 2.1 veranschaulicht den Aufbau eines Sensorknotens schematisch in einem Blockdiagramm.

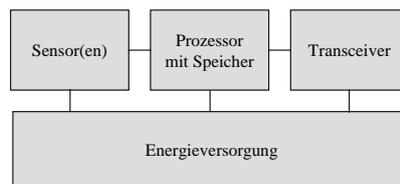
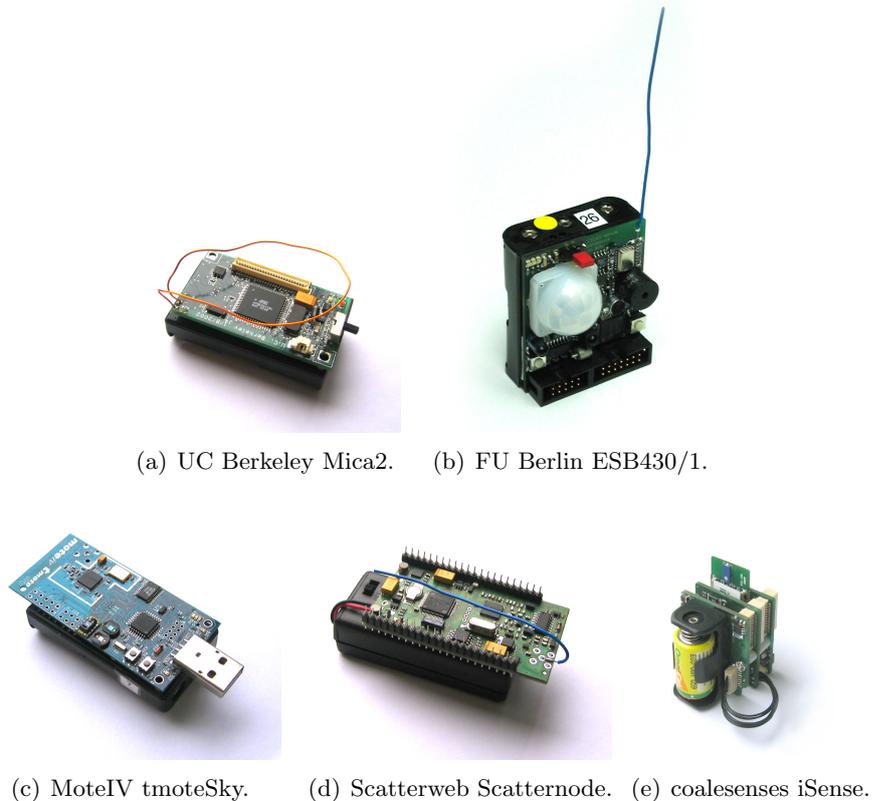


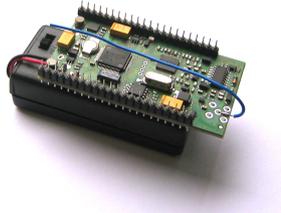
Abbildung 2.1.: Schematischer Aufbau eines Sensorknotens.

Schon heute sind eine Reihe kommerzieller Sensorknoten wie auch Forschungsprototypen verfügbar. Diese Geräte haben allerdings weiterhin noch eine Größe von einigen Kubikzentimetern.

Die von der UC Berkeley entwickelten Geräte der Mica-Familie [50, 84] (Abbildung 2.2(a)) verfügen über einen mit 4 MHz getakteten 8-Bit Prozessor mit 2 kB RAM und 128 kB Flash sowie einen bei 916 MHz arbeitenden Transceiver, der eine Datenübertragungsrate von 19,2 kBit/s ermöglicht. Etwas später entstand der ESB430/1 der FU Berlin [51, 166] (Abbildung 2.2(b)). Sein 16-Bit



(a) UC Berkeley Mica2. (b) FU Berlin ESB430/1.



(c) MoteIV tmoteSky. (d) Scatterweb Scatternode. (e) coalesenses iSense.

Abbildung 2.2.: Unterschiedliche Sensorknoten.

Prozessor mit 5 kB RAM und 60 kB Flash arbeitete ebenfalls mit 4 MHz, und auch sein Transceiver erreichte 19,2 kBit/s. Deutlich moderner präsentieren sich die ebenfalls in Berkeley als *Telos* entwickelten und von MoteIV als *tmoteSky* vertriebenen Geräte [123, 143] (Abbildung 2.2(c)). Die Datenübertragungsrate wurde auf 250 kBit/s gesteigert, der 16-Bit Prozessor arbeitet mit 8 MHz und bietet 10 kB RAM und 48 kB Flash.

Für den kommerziellen Einsatz wurden die Scatternodes [162] (Abbildung 2.2(d)) entwickelt. Sie erzielen besonders hohe Funkreichweiten bei im Gegenzug reduzierter Datenübertragungsrate. Ihr Prozessor bietet 5 kB RAM und 50 kB Flash. Ebenfalls kommerziell ausgerichtet sind die iSense Sensorknoten [28] (Abbildung 2.2(e)). Sie verfügen über einen mit 16 MHz getakteten 32-Bit RISC-Prozessor mit 96 kB RAM sowie 128 kB Flash und erreichen wie die *tmoteSky* Datenraten von 250 kBit/s.

Zukünftige Miniaturisierungsschritte könnten dazu führen, dass die entsprechenden Geräte bis auf wenige Kubikmillimeter schrumpfen [65, 190]. Als *intelligente Staubkörner* ließen sie sich sehr vielfältig einsetzen. Erste Schritte in diese Richtung sind bereits erfolgt, wie der sogenannte *Spec* Sensorknoten [86] zeigt. Dieser bündelt die meisten Funktionen des Mica Sensorknotens in einem einzigen Chip. Obwohl als Durchbruch gefeiert bleibt anzumerken, dass es sich

nicht um eine herausragende technische Entwicklung handelt: Mehrere kommerzielle Chiphersteller bieten ebenfalls in einem einzigen Chip kombinierte Prozessoren und Transceiver an.

## 2.2. Sensornetze und ihre Anwendung

Die grundlegende Idee des Einsatzes von Sensornetzen ist, dass viele Sensorknoten ein bestimmtes Phänomen gemeinsam überwachen. Dabei wird die klassische Vorgehensweise, das Objekt des Interesses in ein Labor zu holen und dort zu beobachten, genau umgedreht: Die Beobachtungsgeräte in Gestalt der Sensorknoten werden zum Phänomen gebracht und dieses in seiner natürlichen Umgebung beobachtet. Dieser Paradigmenwechsel findet besonders prägnant in dem englischen Leitsatz „Taking the lab outside“ seinen Ausdruck.

Dabei werden Objekte des Alltags instrumentiert, also möglichst unauffällig mit Sensorknoten ausgestattet. Die Knoten können die Objekte überwachen und so digitale Informationen über deren Zustand – bis hin zu einem virtuellen Abbild der realen Welt – zur Verfügung stellen. Reale und virtuelle Welt sind so über das Sensornetz miteinander verbunden, die Sensorknoten sind sozusagen in die reale Welt eingebettet.

Im Gegensatz zu klassischen Messsystemen können sich Sensornetze spontan konstituieren und selbsttätig konfigurieren. Auf diese Weise reduziert sich der Installationsaufwand, und nach Bedarf können jederzeit drahtlose Sensorknoten hinzugefügt oder entfernt werden. Zukunftsvisionen sehen sogar vor, dass man Sensorknoten beispielweise aus einem Flugzeug über dem zu beobachtenden Gebiet abwirft. In einem *Selbstorganisationsprozess* vernetzen sie sich selbständig und beginnen mit der Überwachung. Damit zeigen Sensornetze ihre Vorteile insbesondere dort, wo langwierige Installationen schwierig, teuer oder unmöglich sind.

Sensornetze sind von ihrer Natur her Adhoc-Netze. Es gibt keine prinzipielle Trennung in Endgeräte und Router, vielmehr übernimmt jeder Sensorknoten beide Rollen, speist einerseits potenziell Datenpakete in das Netz ein und leitet andererseits Pakete für andere Sensorknoten weiter.

Bei ihrem Einsatz profitieren Sensornetze von der Möglichkeit der Kooperation der Sensorknoten: so können unterschiedliche Daten korreliert, zusammengefasst und in den zeitlichen oder räumlichen Kontext gesetzt werden. Das erfordert jedoch, dass die Sensorknoten über *Kontextbewusstsein* verfügen [115]. Auf diese Weise können zum Beispiel viele Temperaturwerte zu einem Mittelwert über ein Zeitintervall aggregiert oder einzelne Objektsichtungen zu einem Bewegungsvektor zusammengeführt werden.

Dieser *in-network processing* genannte Ansatz [64] hilft, das Kommunikationsaufkommen im Netz zu reduzieren, da nicht mehr viele einzelne Sensorwerte vorliegen, sondern zusammengefasste Ereignisinformationen.

In vielen Szenarien werden solche Informationen dann zu einem Gateway weitergeleitet, das einen Übergang in andere Netze zur Verfügung stellt [144].

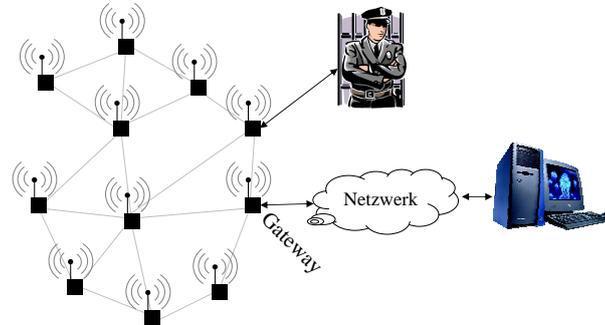


Abbildung 2.3.: Schematische Darstellung eines Sensornetzes mit Gateway.

Eine schematische Darstellung eines Sensornetzes und seiner Interaktion mit der Außenwelt ist in Abbildung 2.3 dargestellt. Einzelne Sensorknoten werden dabei in einer Region ausgebracht und erfassen durch ihre Sensoren physikalische Eigenschaften ihrer direkten Umwelt. Die im ad-hoc formierten Sensornetz erhobenen und vorverarbeiteten Daten können dann über ein Gateway und Weitverkehrsnetze hinweg zum Beispiel an eine Überwachungszentrale gesendet werden, die die relevanten Informationen in einer Datenbank sammelt oder Alarmmeldungen anzeigt.

Hier ergibt sich innerhalb des Sensornetzes hauptsächlich ein Datenstrom von den Sensorknoten hin zum Gatewayknoten. Daher wird dieser Knoten häufig auch als *Datensenke* bezeichnet.

Andererseits kann eine Interaktion natürlich außer über das Gateway auch direkt vor Ort erfolgen. Das kann beispielsweise durch ein Handgerät mit einer Funkschnittstelle, die kompatibel zu der der Sensorknoten ist, passieren.

Während das soeben beschriebene Szenario innerhalb des Sensornetzes statisch ist, ergibt sich jedoch durchaus eine gewisse *Dynamik* bezüglich der Kommunikation durch den volatilen Charakter einzelner Funkstrecken.

Darüber hinaus können einzelne oder sogar alle Sensorknoten mobil sein. Das ist zum Beispiel der Fall, wenn die Sensorknoten an Containern in einem Hafen oder an Tieren befestigt sind, deren Eigenschaften, Positionen oder Verhalten beobachtet werden sollen. Solche *mobilen Sensornetze* sind jedoch weniger das in dieser Arbeit adressierte Anwendungsgebiet, da die extreme Dynamik dieser Anwendungen eher eine Ausnahme ist.

Ein Sensornetz kann nicht nur aus einigen wenigen Sensorknoten bestehen. Insbesondere in Kombination mit einer weiteren Miniaturisierung und Verbilligung der Geräte werden Netze mit Hunderten oder Tausenden von Knoten vorhergesagt. Daher ist die *Skalierbarkeit* der verwendeten Algorithmen und Protokolle wichtig [64].

In der Literatur finden sich zahlreiche Hinweise auf Anwendungsgebiete draht-

loser Sensornetze. Beispiele sind

- die Überwachung von Gletschern [61, 114],
- die Detektion von Waldbränden [200],
- die intelligente Messung von Energieverbräuchen in Gebäuden oder Industrieanlagen [35, 120],
- Abstimmungssysteme bei Vorträgen [198],
- die Fernüberwachung von Patienten und alten Menschen in ihrer Wohnung oder im Krankenhaus [10, 38, 174],
- der Schutz von Geländen und Anlagen vor Eindringlingen [8, 196],
- die Kontrolle des baulichen Zustandes von Gebäuden [43, 95, 96, 136],
- die Überwachung militärischer Kampfgebiete [14, 18] und
- die Beobachtung von Tieren in ihrem natürlichen Lebensraum [9, 109, 179, 180, 201].

## 2.3. Anforderungen an Sensornetze

Aus den im vorangegangenen Abschnitt beschriebenen Anwendungsgebieten drahtloser Sensornetze ergeben sich eine Reihe von Anforderungen, die teilweise bereits angesprochen wurden. Der folgende Abschnitt greift die für diese Arbeit besonders relevanten noch einmal detaillierter auf.

### 2.3.1. Spontane Einsatzbereitschaft

Eine der wichtigsten Anforderungen an Sensornetze ist ihre spontane Einsatzbereitschaft.

Sie impliziert insbesondere, dass beim Ausbringen des Sensornetzes keine explizite Konfiguration der einzelnen Geräte stattfindet, wie sie heute bei lokalen Netzen üblich ist. Dafür gibt es mehrere Gründe. Zunächst ist die sehr große Anzahl von Knoten in einem Netz zu nennen. Bei potenziell mehreren Tausend Geräten ist eine individuelle manuelle Konfiguration aufgrund des immensen Aufwandes nicht realistisch.

Weiterhin ist es wünschenswert, dass Sensornetze in einem unbekanntem Umfeld zum Einsatz kommen können. Das bringt mit sich, dass eine a priori vorgegebene Konfiguration der Sensorknoten selbst in kleinen Netzen nicht sinnvoll ist, da sie die unbekanntem Gegebenheiten nicht berücksichtigen kann.

Darüber hinaus soll der Einsatz von Sensornetzen schnell möglich sein, um Phänomene von Interesse kurzfristig überwachen zu können. Daher ist es erforderlich, dass sich die Knoten in einem Sensornetz selbsttätig konfigurieren

ren [3, 64, 144]. Diese Selbstkonfiguration genannte Fähigkeit [115] ist jedoch nur eine von mehreren Selbst-X-Eigenschaften, die Sensornetze besitzen sollten. Weiterhin sind die Selbstorganisation [3, 64, 144] oder die Selbstüberwachung [85] zu nennen.

Das Streben nach diesen Eigenschaften für Computersysteme im Allgemeinen wie für Sensornetze im Speziellen unterstützte die Entwicklung eines neuen Paradigmas, des *Organic Computings (OC)* [118, 135]. Die zentrale Idee dieser Forschungsrichtung ist, sich bei dem Entwurf von Mechanismen zur Erreichung von Selbst-X-Eigenschaften durch die Natur inspirieren zu lassen [15, 57]. Die vielfältigen Vorschläge schließen beispielsweise selbstheilende Routingverfahren nach dem Vorbild von Ameisenstraßen [17, 182] oder Gruppenbildung wie in Insektenschwärmen [107] ein. Ebenfalls große Bekanntheit erlangte die speziell auf intelligenten Staub ausgerichtete dynamische Rollenverteilung mit Hilfe von virtuellen Pheromonen [47].

Der Begriff der Selbstorganisation schließt dabei die automatische, initiale Konfiguration ein, ist aber nicht auf sie beschränkt. Er umfasst vielmehr auch die dynamische Anpassung des Verhaltens an wechselnde Umgebungsbedingungen, die sich in Begriffen wie Selbstoptimierung oder Selbstheilung ausdrückt [167, 168]. Dabei kommen insbesondere solche Verfahren zum Einsatz, bei denen sich das gewünschte globale Verhalten aus lokaler Interaktion ergibt. Man spricht in diesem Zusammenhang auch von sogenanntem *emergenten Verhalten*.

Die Anforderung der spontanen Einsatzbereitschaft in unbekanntem Gelände und der damit einhergehende Verzicht auf jegliche Infrastruktur bedingt, dass die Sensorknoten im Allgemeinen eine eigene autarke Energiequelle benötigen. Es existieren zwar verschiedene Ansätze, die Sensorknoten mit Energie aus ihrer Umgebung zu versorgen, beispielsweise mit Hilfe einer Solarzelle [45, 146] oder durch Energiegewinnung aus Vibration [153], Temperaturunterschieden [176] oder elektromagnetischen Wellen [105]. Trotzdem erfolgt die Energieversorgung bisher im Allgemeinen aus einer Batterie, um die Kosten für das Energiegewinnungssystem zu sparen. Das bedingt, dass der Energievorrat der Sensorknoten begrenzt ist.

### 2.3.2. Skalierbarkeit

Da Sensornetze potenziell aus sehr vielen Sensorknoten bestehen können, ist die Skalierbarkeit der zum Einsatz kommenden Algorithmen und Protokolle eine wichtige Eigenschaft. Sie beschreibt die Tatsache, dass die Berechnungs- und Raumkomplexität sowie das verursachte Kommunikationsaufkommen pro Gerät nicht nennenswert mit der Anzahl der am Netz beteiligten Geräte steigen. Somit kann ein skalierbares Verfahren in beliebig großen Netzen angewendet werden.

Diese Anforderung bedingt einerseits den bereits angesprochenen Aspekt der Konfigurationsfreiheit, da eine manuelle Gerätekonfiguration oder ständiges Netzmanagement inhärent nicht skalierbar sind.

Andererseits ist es erforderlich, dass die Interaktion zwischen den Sensorknoten ausschließlich lokal stattfindet, da eine globale Kommunikation nicht mit der Netzgröße skaliert. Insbesondere sind zentral organisierte Verfahren, die Informationen von allen Knoten im Netz gleichzeitig verarbeiten müssen, nicht für den Einsatz in Sensornetzen geeignet. Zum Einen verursacht das Versenden dieser Daten von jedem einzelnen Gerät zu der zentralen Komponente ein extrem hohes Datenaufkommen, zum Anderen steigt häufig die Berechnungskomplexität überproportional mit der Netzgröße. Diese Aspekte sind problematisch, da Kommunikationskapazitäten und Rechenleistung der Sensorknoten stark begrenzt sind (siehe auch Abschnitte 2.4.1 und 2.4.2).

### 2.3.3. Kosten

Die Anforderung, sich selbständig administrieren, organisieren und optimieren zu können, bringt für Sensornetze auch Kostenvorteile gegenüber klassischen kabelgebundenen Messsystemen mit sich. Eine im Vergleich mit diesen Systemen insgesamt positive Kosten-Nutzen-Bilanz ergibt sich aber nur, wenn dem Vorteil des verringerten Installationsaufwands keine anderen Kostennachteile gegenüber stehen.

Da Sensornetze aus sehr vielen Geräten bestehen können, sind die Anschaffungskosten pro eingesetztem Sensorknoten ein wichtiger Einflussfaktor für die Gesamtkosten.

Die heute verfügbaren Geräte kosten meist in etwa 100 Euro, da sie noch in kleinen Stückzahlen produziert werden. Während die Entwicklungs-, Einmal- und Rüstkosten heute noch großen Einfluss auf die Stückkosten haben, werden in einer zukünftigen Massenproduktion die Kosten für die eingesetzten Teile und die Fertigung dominieren.

Als Folge befinden sich einfache und einfachste Geräte auf den Markt für Sensorknoten. Das bewirkt einerseits eine Beschränkung der Ressourcen im Hinblick beispielsweise auf die Menge des verfügbaren Speichers, andererseits aber auch den Zwang zum Verzicht auf bestimmte Komponenten wie spezielle Hardware zur Lokalisierung, Zeitsynchronisation oder Distanzmessung.

Darüber hinaus sind aus Kostengründen die Antennen der Funktransceiver häufig als Platinenantennen [50, 123, 143], einfache Drähte [51, 166] oder SMD-Bauteile [28] ausgeführt. Dadurch ergeben sich teilweise ungünstige Abstrahlcharakteristiken, die zu einer ungleichmäßigen Reichweite und häufigen Paketverlusten führen können [73, 202].

### 2.3.4. Kontextbewusstsein

Weil Informationen über Zeit und Ort der Messung von Sensordaten für ihre Interpretation unabdingbar sind, müssen die Daten dementsprechend mit Kontextinformationen ergänzt werden. Dieses ist insbesondere auch erforderlich,

um zusammengehörige Sensordaten identifizieren und geeignet aggregieren zu können [115]. Darüber hinaus gibt es eine Reihe von Protokollen wie beispielsweise geographische Routingverfahren, die für ihre Funktion Ortsinformationen benötigen [3].

Da es unter anderem aus Kostengründen im Allgemeinen nicht möglich ist, Informationen bezüglich Zeit und Raum für jeden Sensorknoten mittels spezieller Hardwareerweiterungen wie GPS [87] zu erhalten, kommen dazu in Sensornetzen häufig spezielle Protokolle und Algorithmen für deren Erlangung zum Einsatz. Diese werden detailliert in Kapitel 3 vorgestellt. Dort werden auch weitere Kontextaspekte und ihre Bedeutung für Sensornetze erläutert.

### 2.3.5. Baugröße

Vielfach ist es erwünscht, dass Sensorknoten möglichst unauffällig in die Umgebung integrierbar sind. Eine kompakte Bauweise ist daher von besonderem Vorteil, wenn die Geräte in kleine Gegenstände integriert werden sollen.

Auch die Vision vom intelligenten Staub, der aus Flugzeugen abgeworfen wird, lässt sich ebenfalls nur mit Sensorknoten realisieren, die ein sehr geringes Gewicht haben, da der Aufprall auf den Boden die Geräte sonst zerstören würde.

Die kleinsten dem Autor bekannten Sensorknoten sind neben den bereits genannten Spec Sensorknoten die Sensor Cubes [54], die etwa münzgroß sind. Momentan wird die Größe der Knoten klar durch die erforderliche Batterie dominiert. Die kompaktesten bekannten Exemplare sind Knopfzellen, die aber nur bedingt die für den vollen Betrieb der Sensorknoten erforderlichen hohen Stromimpulse liefern können.

## 2.4. Eigenschaften von Sensornetzen

Aus der Aufgabe, Verfahren, Software und Hardware zu entwickeln, die den im vorangegangenen Abschnitt dargestellten Anforderungen genügen, ergeben sich bestimmte Einschränkungen und Eigenschaften der entwickelten Komponenten. Diese Eigenschaften werden im Weiteren vorgestellt.

### 2.4.1. Beschränkung des Energievorrats

Wie bereits in Abschnitt 2.3.1 aufgezeigt, werden Sensorknoten meist aus einer Batterie mit Energie versorgt. Als Folge ist ihr Energievorrat beschränkt. Ist dieser verbraucht, fehlt dem Sensorknoten die Energie, er hört auf zu arbeiten und fällt folglich aus.

Daraus ergibt sich die Notwendigkeit, den Energieverbrauch zu minimieren, um die Lebensdauer des Sensornetzes zu maximieren. Neben dem Einsatz von Hardwarekomponenten, die wenig Energie verbrauchen, ist eine weitere Methode dazu

das sogenannte *Duty Cycling*. Dabei ist der Sensorknoten zunächst für eine gewisse Zeitspanne aktiv, danach wechselt er in einen speziellen Energiesparmodus, in dem außer einem Timer alle Hardwarekomponenten deaktiviert sind und folglich nahezu keine Energie verbraucht wird. Nach einer gewissen Zeitspanne weckt der Timer den Sensorknoten wieder auf, und ein neuer Zyklus beginnt. Über das Verhältnis der Längen von Aktivitäts- und Energiesparphasen lässt sich der mittlere Energieverbrauch der Sensorknoten variieren.

Da jedoch zum Teil im Energiesparmodus die Sensoren eines Knotens nicht aktiv sind und darüber hinaus kein Datenaustausch zwischen den Sensorknoten möglich ist, gilt es die zeitliche Lage der Phasen verschiedener Geräte zu koordinieren. Dabei sind einerseits Überlappungen erforderlich, um Kommunikation zwischen den Knoten zu ermöglichen, andererseits dürfen nicht alle Knoten gleichzeitig im Energiesparmodus sein, da sonst während dieser Phasen keine Sensorüberwachung gegeben ist.

In der Literatur gibt es eine Reihe von Vorschlägen, wie die erforderliche Koordination geleistet werden kann. Beispiele hierfür finden sich in [76, 173, 186].

Mit der Verringerung des Energieverbrauchs durch kürzere Aktivitätsphasen geht allerdings auch eine Verringerung des möglichen Datendurchsatzes im Sensornetz einher, da die Übertragung nur bei gleichzeitiger Aktivität von Sender und Empfänger erfolgreich sein kann. Als Rahmenbedingung für die Protokoll- und Algorithmenentwicklung ergibt sich hieraus insbesondere die Minimierung des erforderlichen Datenaustausches.

Neben der Minimierung des mittleren Energieverbrauches ist es auch erforderlich, dass der Verbrauch aller Sensorknoten möglichst gleich ist. Ist dieses nicht gegeben, werden einzelne Geräte ihren Energievorrat deutlich früher aufbrauchen als andere und folglich ausfallen. Daraus ergibt sich neben einer potenziell lückenhaften Sensorabdeckung die Gefahr, dass durch die Ausfälle das Sensornetz in verschiedene Partitionen zerfällt. Eine Kommunikation zwischen verschiedenen Netzteilen ist dann mangels Geräten, die die Weiterleitung der Daten übernehmen können, nicht mehr gewährleistet.

Viele Publikationen beschäftigen sich damit, bestimmte Verfahren beispielsweise zum Routing so zu optimieren, dass der Energieverbrauch möglichst gleichmäßig verteilt wird. Beispiele hierfür sind [41, 199]. Eine Verallgemeinerung der Idee der gleichmäßigen Lastverteilung sind Verfahren, die die Übernahme bestimmter Aufgaben durch einen Sensorknoten vom verbleibenden Energievorrat abhängig machen [39].

### 2.4.2. Beschränkung von Rechenleistung und Speicherkapazität

Der Zwang zu geringen Herstellungskosten einerseits und der erforderliche geringe Energieverbrauch andererseits haben zur Folge, dass die eingesetzten Hardwarekomponenten oft sehr einfach sind und nur über ein geringes Ressourcenangebot verfügen.

Die eingesetzten Prozessoren haben meist nur eine geringe Rechenleistung und verfügen nur über wenige Kilobyte an Speicher. Die Flashspeicher, die der dauerhaften Speicherung des Programmcodes sowie bestimmter Daten dienen, sind ebenfalls nur wenige Kilobyte groß.

Daher können Sensorknoten nur wenig Daten im laufenden Betrieb speichern und verwalten. Insbesondere bei der Entwicklung von Protokollen ist das zu berücksichtigen. So können beispielsweise Routingverfahren für Sensornetze keine umfangreichen Routingtabellen für alle anderen potenziellen Zielknoten im Netz speichern. Andererseits dürfen die eingesetzten Protokolle und Algorithmen nicht zu komplex sein, weil das zu einem Anstieg der Codegröße führen und den begrenzten Programmspeicher überfordern würde.

Die geringe Rechenleistung der eingesetzten Prozessoren führt darüber hinaus dazu, dass komplexe Berechnungen insbesondere zur Signalverarbeitung von Sensordaten oder beispielsweise zur Datenverschlüsselung leicht eine Laufzeit von vielen Sekunden haben können.

Die Beschränkung auf eine geringe Berechnungskomplexität und minimalen Speicherbedarf ist daher eine wichtige Leitlinie bei der Entwicklung von Protokollen und Algorithmen für drahtlose Sensornetze.

### **2.4.3. Zuverlässigkeitsdefizite**

Die für Sensornetze häufig eingesetzten preiswerten Hardwarekomponenten, die unzuverlässige drahtlose Kommunikation oder auch der begrenzte Energievorrat der Sensorknoten führen dazu, dass die Geräte teilweise Zuverlässigkeitsdefizite aufweisen.

So kann beispielsweise eine Ausbringung durch Abwurf aus einem Fahrzeug oder Flugzeug bereits zu Beginn zu Verlusten unter den Sensorknoten führen. Während des Betriebs können entleerte Batterien, bewegliche Hindernisse oder mechanische Zerstörung zum zeitweiligen oder dauerhaften Ausfallen beziehungsweise zu der Abtrennung vom restlichen Netz führen.

Diesen Tatsachen kann einerseits mit einer gewissen Redundanz im Netz begegnet werden, beispielsweise indem mehr Sensorknoten auf einer Fläche ausgebracht werden als zum Funktionieren der betreffenden Anwendung unbedingt erforderlich sind. Andererseits müssen die eingesetzten Algorithmen und Protokolle mit dem Ausfall eines gewissen Anteils von Knoten umgehen können.

Ein weiterer wichtiger Aspekt ist, dass die drahtlose Kommunikation zwischen den Sensorknoten ebenfalls fehleranfällig ist. So schwankt die Kommunikationsreichweite zum Teil erheblich: Experimente mit realen Sensorknoten haben ergeben, dass aufgrund der relativ einfachen Transceiver und Antennen die Funkausbreitung keineswegs kreisförmig ist [73,202]. Vielmehr variiert die Reichweite richtungsabhängig. In [184] und [185] wird anhand von Auswertungen eines Experimentes einerseits gezeigt, dass Kommunikationsstrecken in Sensornetzen zu einem nicht unerheblichen Anteil unidirektional sind. Andererseits wird deut-

lich, dass die Paketverlustrate drahtloser Kommunikation zum Teil erheblich variiert. In [73] wurde darüber hinaus gezeigt, dass die Paketankunftswahrscheinlichkeit mit der Distanz zwischen Sender und Empfänger abnimmt.

Beide Aspekte sind bei der Protokollentwicklung zu beachten. Viele klassische Verfahren gehen von stabilen und bidirektionalen Kommunikationsverbindungen aus, so dass sie beispielsweise einmalig eine Route aufbauen, die dann dauerhaft zum Datenaustausch in beide Richtungen genutzt wird. Solche Verfahren versagen in Sensornetzen, da sie die bestehenden Kommunikationscharakteristika nicht berücksichtigen.

## 2.5. Eigenschaften und Modellierung drahtloser Kommunikation in Sensornetzen

Die Charakteristik der Funkkommunikation beeinflusst das Verhalten von Protokollen nachhaltig. Sie bestimmt nicht nur über Paketverlustraten, sondern auch über die Anzahl der Nachbarn, mit denen ein Gerät direkt kommunizieren kann, und über die Frage, ob eine direkte Kommunikationsverbindung uni- oder bidirektional ist. Die Tatsache, dass zwei Sensorknoten (zumindest gelegentlich und in einer Richtung) direkt Daten austauschen können, wird im Weiteren durch die Formulierung beschrieben, dass zwischen ihnen ein *Link* besteht.

Um die Eigenschaften der Kommunikation in Sensornetzen beschreiben zu können, entwickelt man Modelle, die das Verhalten (oft vereinfacht) widerspiegeln. Diese Modelle können dann einerseits eingesetzt werden, um zum Beispiel Protokolle analytisch zu untersuchen, andererseits können sie in Simulatoren zur Vorhersage des Protokollverhaltens dienen.

Geht man zunächst davon aus, dass sich Funkwellen kreisförmig ausbreiten und mit zunehmender Distanz abschwächen, so ergibt sich, dass um jedes Gerät ein kreisförmiger Bereich (beziehungsweise kugelförmig bei drei betrachteten Dimensionen) mit dem Radius  $r$  existiert, in dem Funksignale des Gerätes noch von anderen Geräten empfangen werden. Der Radius  $r$  wird im Weiteren *Kommunikationsradius* genannt und der durch ihn beschriebene Bereich *Kommunikationsbereich*. Sind Sendeleistung und Signalabschwächung für alle Geräte gleich, ergibt sich für alle Sensorknoten ein einheitlicher Kommunikationsradius  $r$ . Das führt dazu, dass alle Kommunikationsverbindungen bidirektional sind.

Diese Annahmen sind durchaus nicht unüblich bei der Simulation von Protokollen für Sensornetze. Sie werden durch das *Unit Disk Graph (UDG) Modell* in Simulatoren nachempfunden. Das Modell besagt, dass ein versendetes Paket bis zu einer gewissen Distanz  $r$  (vom Absender aus gesehen) empfangen werden kann und jenseits von  $r$  nicht mehr.

Gibt man für dieses Modell die Empfangswahrscheinlichkeit in Abhängigkeit von der Distanz zwischen Sender und Empfänger an, ergibt sich der Graph

in Abbildung 2.4(a). Zu beachten ist, dass in der Abbildung der Kommunikationsradius auf 1 normalisiert wurde. Die Rotationssymmetrie dieses *Funkmodells* ermöglicht es dabei, das Modell als Funktion der Distanz zu beschreiben.

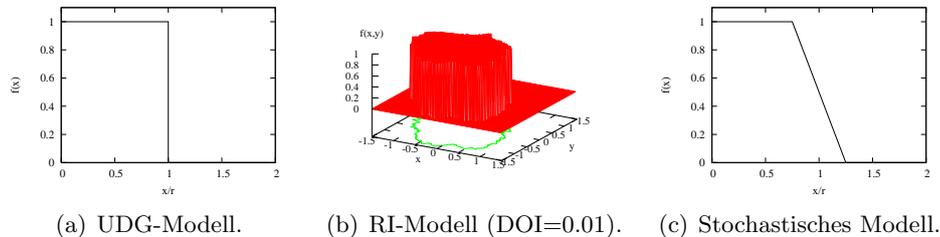


Abbildung 2.4.: Funkmodelle.

Wie in Abschnitt 2.4.3 angesprochen, ist jedoch die Funkausbreitung in Sensornetzen keineswegs kreisförmig [73, 202], sondern die Reichweite variiert richtungsabhängig. Um diese Ungleichmäßigkeiten beschreiben zu können, wurden neue Modelle vorgeschlagen.

Das *Radio Irregularity Model (RIM, auch RI-Modell)* beschreibt die Kommunikationsreichweite als Funktion des Winkels zwischen Sender und Empfänger [202]. Somit hängt die Empfangswahrscheinlichkeit nicht nur von der Distanz, sondern auch von der relativen Lage ab. Von oben betrachtet ergibt sich somit kein kreisförmiger Kommunikationsbereich, sondern eine mehr oder minder ungleichmäßige Form. Wie stark die Unförmigkeit ausgeprägt ist, bestimmt ein *Degree of Irregularity (DOI)* genannter Parameter, der die maximale prozentuale Reichweitenveränderung pro Winkelgrad angibt. Mit seiner Hilfe kann man die Form des Kommunikationsbereiches zwischen Kreis (DOI=0) und Stern (beispielsweise DOI=0,2) variieren. Abbildung 2.4(b) zeigt das RI-Modell für DOI=0,01. Dieser Wert kann als realistische Größe für Sensornetze angenommen gelten [202] und wird im Weiteren durchgehend verwendet, wenn das RI-Modell zum Einsatz kommt. Es ist zu erkennen, dass das Funkmodell nicht rotationsymmetrisch ist und folglich als dreidimensionale Funktion angegeben werden muss.

Obwohl das RI-Modell anders als das Kreismodell auch unidirektionale Links beschreiben kann, stellt es noch immer eine deutliche Vereinfachung der in Sensornetzen vorgefundenen Situation dar, da es sich um ein binäres Modell handelt. Es kennt solche Links, auf denen eine Kommunikation zeitinvariant immer oder niemals möglich ist, nicht aber solche, in denen zeitweise Kommunikation möglich ist oder auch nicht.

Um diesen zeitlichen Schwankungen und der Distanzabhängigkeit der Paketverlustrate [73] Rechnung zu tragen, haben Zuniga und Krishnamachari [204] ein *stochastisches Funkmodell* entwickelt. Es kennt zwei Distanzschwellen: liegt die Distanz zwischen Sender und Empfänger unterhalb der kleineren Schwelle, ist der Paketempfang stets möglich. Bei Distanzen zwischen den beiden Schwel-

len nimmt die Empfangswahrscheinlichkeit mit der Distanz ab, oberhalb der zweiten Schwelle ist kein Empfang möglich. Es gibt folglich keinen eindeutigen Kommunikationsradius mehr, sondern es ergibt sich ein Mittelwert. Daher spricht man auch vom mittleren Kommunikationsradius  $r$ . Das entsprechende Funkmodell ist in Abbildung 2.4(c) dargestellt. Da das stochastische Modell rotationssymmetrisch ist, genügt wie beim Kreismodell eine zweidimensionale Darstellung. Die beiden Schwellenwerte bestimmen das Verhalten des Modells maßgeblich. Sie erlauben eine weitgehende Anpassung an die Eigenschaften des eingesetzten Transceivers und der entsprechenden Antenne, im Beispiel liegen sie bei  $0.75r$  und  $1.25r$ .

Die in den Experimenten festgestellten Unregelmäßigkeiten und Schwankungen der Funkeigenschaften von Sensorknoten sind natürlich nicht ausschließlich auf die Sensorknoten selbst zurückzuführen. Sie resultieren auch aus Veränderungen der Umgebung wie beispielsweise Schwankungen von Temperatur und Luftfeuchte, beweglichen Hindernissen und resultierenden Reflektionen, Mehrwegeausbreitung und Abschattung. Somit repräsentieren Funkmodelle wie das RI-Modell und das stochastische Modell, die diese Effekte berücksichtigen, neben Eigenschaften der Sensorknoten auch Umgebungsvariationen.

Vor dem Hintergrund einer stochastischen Ankunfts-wahrscheinlichkeit für Pakete und oszillierenden Links in Sensornetzen muss der Begriff des verbundenen, also unpartitionierten Netzes überdacht werden. Es stellt sich die Frage, wie viele Nachbarn, mit denen eine direkte Kommunikation möglich ist, die Sensorknoten im Mittel haben müssen, damit über mehrere Hops hinweg mit jedem anderen Knoten ein Datenaustausch möglich ist.

Es konnte gezeigt werden, dass mobile Adhoc-Netze mit großer Wahrscheinlichkeit zur Partitionierung neigen, wenn die Netzdichte (das heißt die mittlere Anzahl der Nachbarn) unterhalb von 10 liegt [83]. Andere Publikationen belegen, dass auch in statischen Szenarien die für ein verbundenes Netz erforderliche Dichte von der Anzahl der Geräte abhängt. Je größer ein Netz ist, desto höher ist die erforderliche Dichte [197]. Für ein Netz mit 300 Knoten ist etwa eine Dichte von 13 notwendig. Aus diesem Grund wird im Weiteren davon ausgegangen, dass für größere Sensornetze eine mittlere Anzahl von etwa 15 Nachbarn einen sinnvollen Wert darstellt.

## 2.6. Zusammenfassung

In diesem Kapitel wurden der grundlegende Aufbau von Sensorknoten, die Funktionsweise von Sensornetzen und ihre Anwendungen beschrieben.

Dabei wurde auf unterschiedliche Anforderungen eingegangen. Drahtlose Sensornetze müssen spontan einsatzbereit sein, und die verwendeten Protokolle und Verfahren müssen auch mit großen Gerätezahlen skalieren. Die Geräte müssen klein und preiswert sein und Kontextbewusstsein verfügen.

Auch verschiedene aus diesen Anforderungen resultierende Eigenschaften von

Sensornetzen und Sensorknoten wurden diskutiert. Der begrenzte Energievorrat, die Ressourcenbeschränkungen sowie die Zuverlässigkeitsdefizite müssen beim Entwurf von Protokollen und Verfahren für Sensornetze berücksichtigt werden.

Speziell die Besonderheiten der drahtlosen Kommunikation in Sensornetzen sowie Ansätze zu deren Modellierung wurden vorgestellt, da sie in Kapitel 5 aufgegriffen werden und zur Anwendung kommen.

Das folgende Kapitel vertieft nun die Betrachtung des Kontextes in Sensornetzen mit dem Schwerpunkt der Kontextaspekte Zeit und Ort.

### 3. Grundlagen des Kontextbewusstseins

Der Begriff des Kontextbewusstseins fand schwerpunktmäßig mit dem Aufkommen mobiler Anwendungen Einzug in die Informatik. Er beschreibt den Umstand, dass Anwendungen Informationen über ihr Umfeld kennen und ihr Verhalten an Veränderungen der Bedingungen in ihrer Umwelt anpassen können. Es gibt eine Vielzahl von Vorschlägen, welche Eigenschaften der Umwelt sowie der Anwendung selbst zum Kontext gehören.

Erstmals wurde der Begriff des Kontextbewusstseins im Bereich mobiler Anwendungen von Schilit und Theimer [165] vorgestellt. Die Autoren zählen Lokation, nahe Personen und Objekte sowie Veränderungen an diesen Parametern als zum Kontext gehörig.

Während diese Autoren Kontext vornehmlich durch Beispiele definieren, verwenden andere Synonyme wie *Umgebung* oder *Situation* zur Definition. Als Bezugsgrößen finden sich hier sowohl der Nutzer als auch die Anwendung selbst.

Die Autoren in [152, 189] beziehen sich ausschließlich auf die Applikation und sehen Kontext als den Zustand ihrer Umgebung. Franklin und Flaschbart hingegen [69] bezeichnen Kontext als die Situation des Nutzers, Brown [19] definiert ihn als die Summe der Elemente der Umgebung des Nutzers, die der Anwendung bekannt sind. Dey [53] zählt darüber hinaus den emotionalen Zustand des Nutzers sowie den Fokus seiner Aufmerksamkeit dazu.

Da Sensornetze im Allgemeinen keine direkte Interaktion der einzelnen Geräte mit dem Nutzer aufweisen, sondern sich wesentlich durch die Zusammenarbeit verschiedener Geräte auszeichnen, eignet sich in diesem Bereich als Bezugsgröße die Anwendung eher als der Nutzer.

Ähnlich zu der Definition in [165] werden in [20, 154] Lokation, die Identität naher Personen, die Tages- und Jahreszeit, Temperatur und ähnliches als zum Kontext gehörig angesehen. Zusätzlich wird in [164] der Aspekt der die Anwendung umgebenden Ressourcen einbezogen, darunter andere Computer, Ein- und Ausgabegeräte, Konnektivität oder die Netzkapazität.

In der konkret auf drahtlose Sensornetze bezogenen Literatur werden als relevante Kontextaspekte zunächst Lokation und Zeit genannt [64, 115], in [2, 44] treten mit Hilfe der integrierten Sensoren ermittelte Umgebungseigenschaften hinzu. Letztere haben jedoch die Eigenschaft, dass die relevanten Parameter im Allgemeinen anwendungsabhängig sind und sich somit nur schwer verallgemeinern lassen. Ort und Zeit hingegen sind für praktisch alle Anwendungen von

Interesse, beispielsweise um Messwerte räumlich oder in zeitliche Reihenfolgen einordnen zu können. Darüber hinaus sind sie auch geeignet, miteinander in Zusammenhang stehende Messwerte zu erkennen und entsprechend zu aggregieren.

Neben diesen anwendungsbezogenen Kontextaspekten ergibt sich aus der für Sensornetze typischen Kooperation von mehreren Geräten untereinander, dass die technische Umgebung, das heißt Eigenschaften und Zustand nahegelegener Sensorknoten, von Interesse ist. So können beispielsweise Informationen über die Sensorausstattung benachbarter Knoten oder die Qualität der Funkverbindung zu diesen genutzt werden, um die Funktion der auf allen diesen Geräten verteilt arbeitenden Anwendung zu optimieren.

In diesem Zusammenhang sind Informationen über den Energievorrat der Sensorknoten ebenfalls von Interesse. So gibt es in der Literatur zahlreiche Beiträge über die Berücksichtigung der verbleibenden Energie beispielsweise beim Medienzugriff [5], bei der Weiterleitung von Daten [79, 89, 108, 110, 203, 205], bei der Gruppierung von Geräten in Clustern [82] oder zur Steuerung der Schlaf- und Wachzyklen der Sensorknoten [175].

Die vorliegende Arbeit beschäftigt sich hauptsächlich mit den beiden Kontextaspekten Zeit und Ort, so dass auf diese beiden im Weiteren besonders eingegangen wird.

### 3.1. Zeitbewusstsein

Die Zeit ist einer der wichtigsten Kontextaspekte in drahtlosen Sensornetzen. Ein einheitliches Zeitbewusstsein der an einem Netz beteiligten Sensorknoten ist aus mehreren Gründen wichtig:

- Ein wesentliches Ziel bei der Verarbeitung von Messdaten ist das Zusammenführen von Daten verschiedener Sensorknoten innerhalb des Netzes. Um zum gleichen Ereignis gehörende Daten zu erkennen und miteinander korrelieren zu können, ist ein einheitliches Zeitbewusstsein über alle Knoten des Netzes hinweg unerlässlich.
- Das Zusammenführen von mehreren, nacheinander erfassten Ereignissen zu einem Ereignisverlauf verlangt ebenfalls eine konsistentes Zeitbewusstsein, beispielsweise um bei der Objektverfolgung mehrere Objekterkennungen zu einer Objektspur mit Geschwindigkeit und Richtung zu aggregieren.
- Viele Protokolle erfordern darüber hinaus ein gleichzeitiges Handeln mehrerer Sensorknoten, zum Beispiel das synchrone Aufwachen von Sender und Empfänger einer Nachricht aus dem Schlafzustand. Auch dieses ist nur mittels eines einheitlichen Zeitbewusstseins möglich.

Das Vorhandensein eines solchen konsistenten Zeitbewusstseins bedeutet nicht notwendigerweise, dass die Uhren aller Sensorknoten möglichst exakt gleich gehen. Es kann auch bedeuten, dass die einzelnen Geräte sich der Unterschiede zwischen ihren Uhren bewusst sind und diese im Ablauf ihrer Protokolle und Algorithmen berücksichtigen.

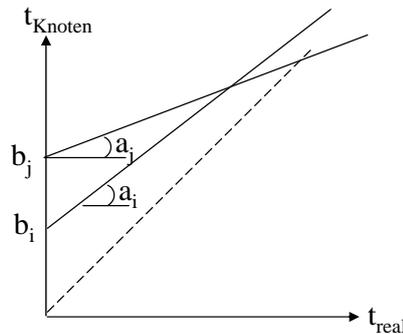


Abbildung 3.1.: Inkonsistentes Zeitbewusstsein.

Die Ursachen für ein inkonsistentes Zeitbewusstsein sind in Abbildung 3.1 dargestellt, sie zeigt die lokale Uhrzeit  $t_{Knoten}$  zweier Sensorknoten  $i$  und  $j$  als durchgezogene Geraden relativ zur tatsächlichen Zeit  $t_{real}$ . Einerseits beginnen die Uhren auf verschiedenen Geräten mit unterschiedlichen Startwerten und eventuell zu unterschiedlichen Zeitpunkten zu laufen. Dieser initiale *Versatz* drückt sich durch die unterschiedlichen Ordinatenabschnitte  $b_i$  und  $b_j$  in Abbildung 3.1 aus. Doch auch wenn diese anfänglichen Differenzen ausgeglichen wurden, führen Hardwaretoleranzen der für die Sensorknoten verwendeten Bauteile, schwankende Umgebungstemperaturen und Batteriespannungen dazu, dass ihre internen Uhren unterschiedlich schnell voranschreiten. Neben dem Versatz der Uhrzeiten ergibt sich daher eine *Drift* der Uhren, die Steigungen  $a_i$  und  $a_j$  differieren. Deshalb ist es nicht ausreichend, die Konsistenz einmalig herzustellen, vielmehr ist ein periodisches Eingreifen erforderlich, um diese dauerhaft zu erhalten.

Der Prozess des Herstellens einer konsistenten Zeitbasis auf allen Knoten wird im Weiteren *Synchronisation* genannt. Dabei kann die *interne* von der *externen* Synchronisation unterschieden werden. Während erstere nur auf ein einheitliches Zeitverständnis innerhalb des Sensornetzes abzielt, strebt letztere zusätzlich nach Konsistenz mit einer externen, tatsächlichen Zeit. Bezüglich Abbildung 3.1 heißt dieses, dass eine interne Synchronisation lediglich die Geraden der Knoten  $i$  und  $j$  zur Deckung bringt, während die externe Synchronisation zusätzlich die Deckung mit der realen Zeit (gestrichelte Gerade) anstrebt.

Eine Möglichkeit zur Synchronisation stellt die Ausrüstung eines jeden Sensorknotens mit einem speziellen Empfänger für eine externe Referenzzeit wie etwa DCF77 [142] oder die im GPS-Signal kodierten Zeitinformationen [111] dar. Der Einsatz solcher Empfänger wirkt sich aber negativ auf Baugröße, Preis und Energieverbrauch der Geräte aus, so dass ein flächendeckender Einsatz nicht in Frage kommt. Einzelne Sensorknoten können aber eine solche Ausstattung auf-

weisen und dann Ausgangspunkte für die Synchronisation im Netz darstellen.

Logische Uhren, wie Lamport sie vorschlägt [101], sowie die entsprechenden Verfahren zur Definition von Ereignisreihenfolgen sind für Sensornetze im Allgemeinen nicht geeignet, weil beispielsweise für die Aggregation von Sensordaten nicht nur die Reihenfolge der Sensorereignisse, sondern auch ihr zeitlicher Abstand von Relevanz ist.

Die Anforderungen an die Genauigkeit der Synchronisation sind letztendlich anwendungsabhängig. Um beispielsweise zusammengehörige Temperaturwerte zu identifizieren, ist auch ein Fehler von mehreren Hundert Millisekunden akzeptabel, während die Fusion von Sensordaten zur akustischen Lagebestimmung eines Objektes zum Beispiel nur Abweichungen im Mikrosekundenbereich toleriert [63].

### 3.1.1. Verfahren zur Errichtung von Zeitbewusstsein

Die Synchronisation lässt sich in mehrere Teilschritte unterteilen, wobei einige davon in allen im Weiteren vorgestellten Verfahren (wenn auch in unterschiedlichen Ausprägungen) vorkommen, während andere nur in einzelnen Verfahren eingesetzt werden. Diese Teilschritte sind:

- Ermittlung der Zeitdifferenz zwischen verschiedenen Sensorknoten,
- Anpassung der lokalen Uhrzeiten,
- Fortführung dieser Schritte über mehrere Hops hinweg,
- Kompensation der Drift.

Um die unterschiedlichen Systematiken der in der Literatur zu findenden Verfahren besser zu verdeutlichen, werden die Verfahren im Weiteren in diese Teilschritte zerlegt besprochen.

### 3.1.2. Ermittlung von Zeitdifferenzen

Die Ermittlung der Zeitdifferenzen funktioniert prinzipiell per Übertragung der lokalen Zeit des Senders. Nach Empfang wird dann die Differenz zur Empfängerzeit gebildet. Die einfachste Methode beschreiben der IEEE 802.11 Standard [46] und das *Time Transmission Protocol (TTP)* [6], siehe auch Abbildung 3.2(a). Der Sender liest seine Uhrzeit zum Zeitpunkt  $t_1$  aus, überträgt sie ab  $t_2$ , der Empfang ist bei  $t_3$  abgeschlossen und der Empfänger berechnet die Differenz zum Zeitpunkt  $t_4$ . Nachteilig bei dieser Methode ist der lange *kritische Pfad* (grau hinterlegt). Er beschreibt Prozesse, deren Dauer nicht genau bekannt ist, und die somit zu Ungenauigkeiten beitragen. Er umfasst hier die Zeit zwischen dem Ablesen der Uhr und dem Beginn der Übertragung  $t_1 - t_2$ , die eigentliche Übertragungszeit  $t_2 - t_3$  sowie die Zeit zwischen Empfang und dem Ablesen der Uhr des Empfängers  $t_3 - t_4$ . Die größten Unsicherheiten ergeben sich in aller

Regel vor Übertragungsbeginn, da die Medienzugriffszeit schwanken kann.

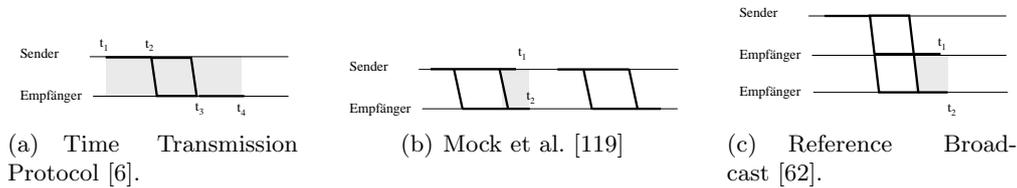


Abbildung 3.2.: Ermittlung von Zeitdifferenzen mit einzelnen Nachrichten.

Diesen Nachteil versuchen die Autoren von [119] auszugleichen, indem sie zunächst nicht den Zeitstempel selbst übertragen, sondern lediglich eine sogenannte *Indication Message* (Abbildung 3.2(b)). Bei Abschluss der Übertragung lesen Sender und Empfänger die Uhrzeit ab. Die Zeit des Senders wird dann in einer weiteren *Synchronization Message* übertragen. Als Folge werden der verkürzte kritische Pfad und die resultierende höhere Genauigkeit durch zusätzlichen Kommunikationsaufwand erkauft. Einen ähnlichen Ansatz verfolgt ein *Reference Broadcast* [62] genanntes Verfahren, das jedoch nicht eine Differenz zwischen Sender und Empfänger, sondern die Differenz zwischen mehreren Empfängern ermittelt (Abbildung 3.2(c)). Auch hier ist eine zweite Nachricht erforderlich, mit der die Empfangszeitpunkte zwischen den Empfängern ausgetauscht werden, um die Zeitdifferenzen ermitteln zu können.

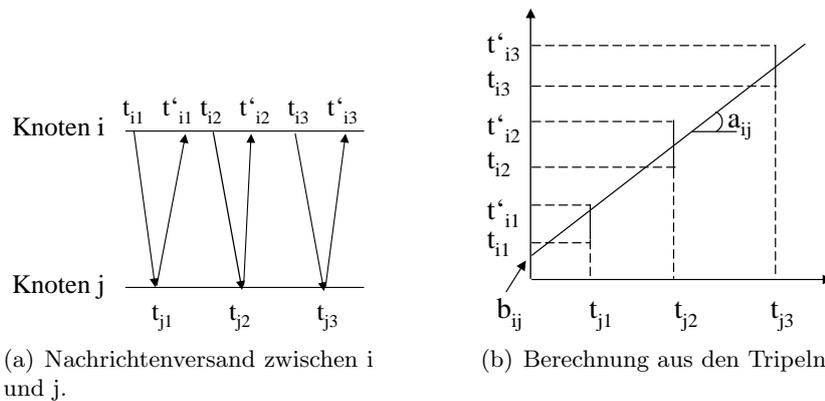


Abbildung 3.3.: Ermittlung von Zeitdifferenzen mit mehreren Nachrichten [104].

In anderen Verfahren wird versucht, die Dauer des kritischen Pfades durch wiederholte Übertragungen zuverlässig zu ermitteln. Dem liegt die Annahme zugrunde, dass sich zufällige Schwankungen durch die Wiederholungen ausgleichen. Die so ermittelte Übertragungsdauer der Zeitstempel kann dann bei der Bestimmung der Zeitdifferenz berücksichtigt werden. So beschreibt die *Set-Valued Estimation* [104] das Verhältnis zwischen den Uhren der Knoten i und j

als lineare Korrelation der Form

$$t_i = a_{ij}t_j + b_{ij} \quad (3.1)$$

mit  $a_{ij}$  als relativer Drift und  $b_{ij}$  als Offset. Um diese Werte zu ermitteln, versendet Knoten  $i$   $n$  Nachrichten, auf die Knoten  $j$  umgehend antwortet (Abbildung 3.3(a)). Somit entstehen eine Reihe von Tripeln  $(t_{ik}, t_{jk}, t'_{ik}), k = 1, \dots, n$ , die dann in ein Diagramm eingetragen werden (Abbildung 3.3(b)). Anhand einer durch die vertikalen Unsicherheitsintervalle gelegten Geraden können dann  $a_{ij}$  und  $b_{ij}$  ermittelt werden. Die Autoren von [170] schlagen mit ihren *Tiny-Sync* und *Mini-Sync* genannten Verfahren Verbesserungen vor, die keine sofortige Antwort von Knoten  $j$  verlangen und darüber hinaus nur die beiden optimalen Tripel speichern müssen.

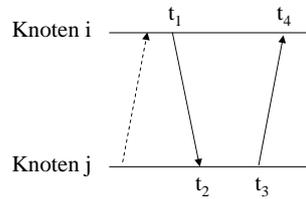


Abbildung 3.4.: Austausch von Nachrichten mit Zeitstempeln.

Ähnlich arbeitet das in [71,72] vorgeschlagene Verfahren mit dem Namen *Timing-sync Protocol for Sensor Networks (TPSN)* (siehe Abbildung 3.4), das sich am bekannten Algorithmus von Christian [49] orientiert. Dabei liest Knoten  $i$  zum Zeitpunkt  $t_1$  seine Uhr aus und sendet das Ergebnis an seinen Nachbarn  $j$ . Dieser liest bei Empfang zum Zeitpunkt  $t_2$  die eigene Uhr aus. Zum Zeitpunkt  $t_3$  liest  $j$  nun wiederum seine Uhr aus, und sendet  $t_2$  und  $t_3$  zurück zu  $i$ , der wiederum zum Empfangszeitpunkt  $t_4$  auf die Uhr schaut. Er kann nun die Zeitdifferenz  $d$  wie auch die mittlere Nachrichtenverzögerung  $\delta$  der beiden Nachrichten bestimmen. Es gilt

$$d = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}, \quad (3.2)$$

$$\delta = \frac{(t_2 - t_1) - (t_4 - t_3)}{2}. \quad (3.3)$$

Das Verfahren von Römer [149] arbeitet ähnlich, verwendet jedoch eine weitere Nachricht (in Abbildung 3.4 gestrichelt eingezeichnet). Während hier also Knoten  $j$  den Austauschprozess anstößt, fragt bei TPSN Knoten  $i$  nach der Zeit. Römer betont jedoch, dass die Informationen der dritten Nachricht prinzipiell mittels Piggy-Backing einer ohnehin zu sendenden Nachricht mitgegeben werden können.

In [72] wird darüber hinaus beschrieben, wie die Genauigkeit von TPSN (wie auch anderer Verfahren) weiter gesteigert werden kann. Die Autoren schlagen

dazu das Einfügen von Zeitstempeln innerhalb der Medienzugriffs- und Sicherungsschicht (MAC-Layer/Link Layer) vor. Die Grundidee ist, dass der Sender seine Uhr direkt vor oder während der Übertragung des Pakets ausliest und den Zeitstempel in das Paket einfügt. Auf diese Weise können die Unsicherheiten bezüglich der Zeitdauer der Entgegennahme des Sendeauftrages von einer höheren Protokollschicht, der Zusammenstellung des Paketes und des Medienzugriffes aus dem kritischen Pfad entfernt werden. Analog kann auf Empfängerseite die Differenz zwischen lokaler Uhr und empfangenem Zeitstempel ebenfalls in der Sicherungsschicht erfolgen. Dieses Vorgehen ist allerdings nur dann möglich, wenn diese Schichten in Software und nur die Bitübertragungsschicht in Hardware realisiert ist, oder aber die betreffende Hardware diese Möglichkeit vorsieht.

### 3.1.3. Anpassung der lokalen Uhrzeit

Um zu einer einheitlichen Zeit zu gelangen, muss nach der Bestimmung des Zeitunterschiedes zweier lokaler Uhren diese Information genutzt werden, um für ein konsistentes Zeitverständnis zu sorgen.

Die *proaktive* Vorgehensweise sieht dabei vor, dass eine der beiden Uhren an die andere angepasst wird und folglich alle weiteren auf diesen beiden Sensorknoten erhobenen Uhrzeiten zueinander konsistent sind.

Im einfachsten Fall wird die neue Uhrzeit aus der lokalen alten Zeit und der bestimmten Zeitdifferenz ermittelt, und diese dann sofort durch Setzen angepasst. Dieses Verfahren nennt man *unmittelbare Uhrenanpassung*. Die resultierende sprunghafte Veränderung der lokalen Zeit führt jedoch notwendigerweise zu einer zeitlichen Diskontinuität, die entsprechend auf Anwendungs- und Betriebssystemebene gehandhabt werden muss.

Eine Alternative wird in [119] vorgeschlagen. Die *kontinuierliche Uhrenanpassung* gleicht den Unterschied zwischen neuer und alter Zeit über einen längeren Zeitraum hinweg aus, indem man die lokale Uhr schneller (wenn die Uhr vorge stellt werden muss) oder langsamer (wenn die Uhr zurückgestellt werden muss) laufen lässt. Dieses Verfahren hat jedoch den Nachteil eines nicht unerheblichen Laufzeitaufwandes.

Das *Asynchronous Diffusion Protocol (ADP)* [106] geht anders vor. Es teilt mit dem bekannten Berkeley-Algorithmus [77] die auf Mittelwertbildung basierende Zeitbestimmung. Anders als beim Berkeley- Algorithmus gibt es aber keine zentralen Zeitmaster, der regelmäßig alle anderen Knoten im Netz abfragt. Statt dessen ermittelt jeder Sensorknoten die lokalen Uhrzeiten seiner Nachbarn und setzt seine eigene Zeit auf deren Mittelwert. Dabei werden mehrere Verfahren zur Mittelwertbildung, wie auch zur Initiierung des Austausches vorgestellt. Dieser Vorgang wird mehrfach wiederholt, so dass die Zeit auf allen Knoten über mehrere Iterationen hinweg gegen eine mittlere Netzzeit konvergiert. Das Verfahren hat den Vorteil, dass es nicht erforderlich ist, eine globale externe Zeit vorzugeben.

Anders als die bisher besprochenen gehen die *post-facto* arbeitenden Verfahren vor.

Mit dem vollständigen Verzicht auf die Uhrenanpassung stellt Römer eine solche Möglichkeit vor [149]. Sein Verfahren transformiert statt dessen Zeitstempel bei der Übertragung von einem Gerät zum anderen in das lokale Zeitsystem des Empfängers. Darüber hinaus schlägt Römer vor, wie aus dem Drift resultierende Ungewissheiten bezüglich der Zeit durch die Betrachtung von Intervallen anstelle von Zeitpunkten repräsentiert werden können.

Eine ähnliche Methode verwenden Elson et al. [62] bei der Reference Broadcast Synchronization. Jeder Sensorknoten verwaltet eine Tabelle mit Parametern bezüglich Uhrenversatz und Drift für die anderen Kommunikationspartner im Netz, die eine Umrechnung der Zeitstempel ins eigene Zeitsystem ermöglicht.

### 3.1.4. Fortführung über mehrere Hops

Mit den bisher beschriebenen Funktionalitäten kann ein konsistentes Zeitverständnis lediglich zwischen direkt benachbarten Sensorknoten hergestellt werden.

Während einige Publikationen sich auf die Betrachtung solcher benachbarter Knoten beschränken, machen andere auch Vorschläge, wie sich ein netzweites Zeitverständnis herstellen lässt. Bestehende Verfahren zur Zeitsynchronisation im Internet wie das *Network Time Protocol (NTP)* [116, 117] sind für den Einsatz in Sensornetzen nicht geeignet, da sie Sensornetzanforderungen wie die Kommunikationsminimierung nicht erfüllen. Da NTP für Synchronisation zwischen Servern entwickelt wurde, bedarf es darüber hinaus der manuellen Konfiguration eines Synchronisationsbaumes, die in Sensornetzen nicht möglich ist.

Um dennoch eine Synchronisation entlang eines Baumes zu ermöglichen, schlagen die Autoren von TSPN [71, 72] vor, dass ein dedizierter Sensorknoten zunächst eine solche Baumstruktur aufbaut. Jeder Knoten synchronisiert sich dann sukzessive mit seinem direkten Vater im Baum. Dieses Vorgehen hat zwei wesentliche Nachteile: zum Einen erfordert der vorgeschaltete Baumaufbau den Versand zusätzlicher Nachrichten, zum Anderen ist die statische Baumstruktur wenig robust gegen wechselnde Konnektivität im Netz. Eine Weiterentwicklung dieses Vorgehens wird in [187] vorgestellt. Das *Lightweight Tree-Based Synchronization (LTS)* genannte Verfahren ermöglicht in seiner dezentralisierten Variante, dass jeder beliebige Sensorknoten eine Synchronisation erbitten kann. Dazu baut er mit einem beliebigen Routingverfahren einen Pfad zur Wurzel des Baumes auf, entlang dessen dann die Synchronisation wie im Baum stattfindet.

Die oben genannten Nachteile der Synchronisation mittels Baumaufbau versuchen die Autoren von [112] auszugleichen. Bei dem *Flooding Time Synchronization Protocol (FTSP)* genannten Verfahren führt jeder Sensorknoten, der synchronisiert wurde, die Synchronisation mit seinen Nachbarn fort. Auf diese Weise entsteht eine Flutwelle, die nach und nach alle Knoten im gesamten Netz

synchronisiert.

Eine auf strikt lokaler Interaktion beruhende Methode wird in [106] vorgeschlagen. Hier tauscht jeder Sensorknoten spontan zu einem zufälligen Zeitpunkt Zeitinformationen mit seinen Nachbarn aus und stellt seine Uhr auf deren Mittelwert. So ergibt sich eine mittlere Netzzeit, indem die Mittelwerte langsam über mehrere Iterationsschritte hinweg durch das Netz diffundieren.

Das in [177] beschriebene *Time-Diffusion Synchronization Protocol (TDP)* arbeitet rundenbasiert und periodisch, um Uhrendrift auszugleichen. In jeder Runde werden unter Berücksichtigung verschiedener Eingangsgrößen wie zum Beispiel dem Drift der verschiedenen Uhren unterschiedliche Cluster gebildet, innerhalb derer die Synchronisation stattfindet. Innerhalb der Cluster werden verschiedene Rollen unterschieden. Durch die unterschiedlichen resultierenden Cluster und die somit zwischen den Sensorknoten wechselnden Rollen wird mittelfristig der Arbeitsaufwand für die Synchronisation gleichmäßig über die Knoten verteilt. Auch bei diesem Ansatz resultiert eine mittlere Netzzeit. Das Verfahren hat den Nachteil, dass für die Clusterbildung ein erheblicher Nachrichtenaufwand erforderlich ist.

### 3.1.5. Driftkompensation

Bauteiltoleranzen, Temperatur- und Spannungsunterschiede führen dazu, dass die Hardwarebauteile, die die Uhren auf den Sensorknoten takten, unterschiedlich schnell arbeiten. Das führt dazu, dass selbst zu einem bestimmten Zeitpunkt perfekt synchronisierte Uhren auf verschiedenen Knoten mit der Zeit auseinanderdriften.

Dieser Effekt kann entweder durch regelmäßige Resynchronisation ausgeglichen werden oder bei der Umrechnung von Zeitstempeln zwischen unterschiedlichen Zeitbasen berücksichtigt werden, sofern die Driftrate der Uhren bekannt ist. Letzteres erfordert jedoch die Abschätzung der betreffenden Raten. Dazu kommen mehrere aufeinander folgende Synchronisationspunkte zum Einsatz: aus der Veränderung des Uhrenversatzes über die Zeit kann die unterschiedliche Laufgeschwindigkeit der Uhren beispielsweise mit der Methode der kleinsten Quadrate [62] bestimmt werden.

Ist die Driftrate der Uhren bekannt, kann diese Information auch dazu genutzt werden, die Laufgeschwindigkeit der lokalen Uhr an die eines anderen Sensorknotens anzupassen, so dass spätere Synchronisationen entfallen können. Das erfordert allerdings, dass die Hardware eine entsprechende Kompensationsmöglichkeit mit der notwendigen Granularität anbietet.

Nahezu alle Synchronisationsverfahren setzen sich aus den behandelten Verfahrensschritten zusammen. Für eine weitere Diskussion von Eigenschaften, Vor- und Nachteilen sei auf Kapitel 4 verwiesen. Nach der Vorstellung der verschiedenen Verfahren zur Errichtung eines einheitlichen Zeitbewusstseins wird dieses Kapitel nun mit der Behandlung der Position als Kontextaspekt sowie mit den

Verfahren zur Erlangung von Positionsinformationen fortgesetzt.

## 3.2. Positionsbewusstsein

Zahlreiche Arbeiten betonen die Relevanz von Positionsinformationen für die Funktion von Sensornetzanwendungen. Gerade in Sensornetzen sind ihre Verwendungsmöglichkeiten vielfältig. Zum Einen ist für die Bewertung von Messdaten der Ort ihres Entstehens inhärent relevant, zum Anderen können mit Hilfe von Ortsinformationen räumlich benachbarte Daten zusammengefasst werden [115]. Spezielle Aufgaben wie das Verfolgen von Objekten benötigen ebenfalls Positionsdaten, um die Bewegungsrichtung ermitteln zu können [183].

Darüber hinaus können Positionsinformationen zum sogenannten *geographischen Routing* genutzt werden, das heißt um Daten auf Basis der Position der Zielknoten weiterzuleiten [94, 100].

In den folgenden Abschnitten werden unterschiedliche Verfahren zur Erlangung von Positionsbewusstsein vorgestellt.

### 3.2.1. Numerische und symbolische Positionen

Im Hinblick auf die Art der Positionsangaben lassen sich numerische von symbolischen Angaben unterscheiden.

Erstere drücken die Positionen in Form von Koordinaten aus. Das hat den Vorteil, dass die entsprechenden Angaben sehr abstrakt sind und sich somit für nahezu jede Anwendung eignen. Darüber hinaus lassen sich aus den Koordinaten weitere Angaben wie beispielsweise Distanzen direkt ableiten. Diese Eigenschaften haben dazu geführt, dass die überwältigende Mehrheit der Verfahren zur Positionsbestimmung numerische Angaben liefert. Abschnitt 3.2.2 stellt deren unterschiedliche Funktionsweisen vor.

Symbolische Positionsangaben hingegen schließen bereits eine gewisse Interpretation ein, sie lauten etwa „am Rand des Sensornetzes“ oder „Raum X in Gebäude Y“. Beispiele für solche Verfahren sind [67], das die Randlage von Sensorknoten ermittelt, oder [99], das als Ergebnis die Lage der Sensorknoten in Straßen- und Kreuzungsbereichen eines Straßennetzes ermittelt und benennt. Die entsprechenden Ansätze haben den Vorteil, dass sie auf den Einsatz von Heuristiken und meist fehlerbehafteten Distanzschätzungen verzichten können, die bei numerischen Verfahren speziell in schwierigen Szenarien im Extremfall zum kompletten Versagen führen können [24]. Andererseits sind die resultierenden symbolischen Positionsangaben nicht für alle Anwendungsbereiche geeignet und konnten sich deshalb nicht auf breiter Front durchsetzen.

### 3.2.2. Verfahren zur Errichtung von numerischem Positionsbewusstsein

Nahezu alle Verfahren, die dazu dienen, die Knoten eines Sensornetzes mit Informationen über ihre Position in Form von Koordinaten zu versorgen, gehen davon aus, dass einige Sensorknoten ihre Position bereits kennen. Diese a priori positionsbewussten Sensorknoten werden im Weiteren *Ankerknoten* genannt.

Die Positionsinformationen der Anker können aus unterschiedlichsten Quellen stammen. Die Anker können beispielsweise gezielt an einer bekannten Position ausgebracht werden und die entsprechenden Daten manuell mitgeteilt bekommen. Dieses Vorgehen steigert jedoch den Aufwand bei der Ausbringung des Sensornetzes erheblich.

Alternativ können die Anker mit einer separaten Technik zur Positionsbestimmung wie beispielsweise dem Global Positioning System (GPS) [87] ausgestattet sein. Nachteile dieser Variante sind, dass die entsprechende zusätzliche Hardware sich negativ auf Baugröße, Energieverbrauch und Preis auswirkt. Somit ist es vorteilhaft, den Anteil der Ankerknoten relativ zur Gesamtanzahl der Sensorknoten möglichst gering zu halten.

Auf den Anker aufbauend ist es Inhalt der meisten in diesem Abschnitt vorgestellten Verfahren, die Informationen über die Positionen der Anker gemeinsam mit zusätzlichen, meist relativen räumlichen Information so an alle anderen Sensorknoten zu verteilen, dass diese ihre eigene Position mindestens näherungsweise bestimmen können.

Dieses kann innerhalb des Sensornetzes, also verteilt auf den einzelnen Sensorknoten, geschehen oder aber auf einer zentralen Berechnungskomponente. Dem Vorteil der zentralisierten Variante, dass global auf alle verfügbaren Informationen zurückgegriffen werden kann, stehen die Nachteile eines erhöhten Kommunikationsbedarfs und schlechter Skalierbarkeit gegenüber.

Im Folgenden werden unterschiedliche Gruppen von Verfahren zur Errichtung von Positionsbewusstsein vorgestellt und in ihrer Funktionsweise beschrieben.

#### **Mustervergleich**

Verfahren zur Positionsbestimmung auf Basis von Mustervergleichen, in der Literatur vielfach auch *Fingerprinting* genannt, basieren auf der Annahme, dass für den Bereich, innerhalb dessen Positionen bestimmt werden sollen, eine Datenbasis existiert. Sie verzeichnet die messbaren Eigenschaften jeder Position innerhalb dieses Bereiches.

Ein Gerät, das seine Position bestimmen will, misst zunächst die entsprechenden Eigenschaften am momentanen Aufenthaltsort. Dann sucht es innerhalb der Datenbasis diejenigen Koordinaten, deren gespeicherte Eigenschaften am besten zu den gemessenen passen und nimmt diese Koordinaten als aktuellen Aufenthaltsort an.

In der Literatur finden sich hauptsächlich Beiträge, die Signalcharakteristika von WLAN-Zugangspunkten zur Positionsbestimmung nutzen (zum Beispiel [181]). Andere Publikationen beschreiben Verfahren, die das Bild einer auf einem mobilen Roboter montierten Kamera zur Positionsbestimmung mit bekannten Bildern vergleichen [68].

Der zentrale Nachteil dieser Verfahren liegt im immensen initialen Aufwand, der für die Erstellung der verwendeten Datenbasis erforderlich ist. So muss der gesamte Bereich im Vorfeld vermessen und die Messergebnisse in der Datenbasis aufgezeichnet werden. Daher lässt sich die für Sensornetze gewünschte spontane Einsatzbereitschaft mit diesem Verfahren nicht erzielen. Darüber hinaus nimmt die Datenbasis große Mengen Speicherplatz in Anspruch, so dass davon auszugehen ist, dass sie nicht auf jedem Gerät zur Verfügung stehen kann, sondern nur auf einer zentralen Komponente vorgehalten werden kann. Die Sensorknoten müssten somit die Messdaten an diese Instanz weiterleiten, und die zentral berechneten Positionsdaten müssten an die Sensorknoten zurückgesandt werden.

Nicht nur der hohe initiale Aufwand, auch das hohe Kommunikationsaufkommen und der resultierende Energieverbrauch sowie die mangelnde Skalierbarkeit einer solchen zentralisierten Lösung führen dazu, dass dieses Verfahren für Sensornetze nicht geeignet ist.

### Flächenschnitte und Positionsmittlung

Eines der einfachsten Verfahren, um Sensorknoten mit Positionsinformationen zu versorgen, ist die Positionsmittlung. Dabei wird davon ausgegangen, dass jeder Sensorknoten mehrere Anker zu seinen direkten Nachbarn zählt.

Jeder Anker sendet seine Position mittels eines Broadcasts an seine Nachbarn, die ihrerseits alle empfangenen Positionen speichern. Aus diesen Positionsinformationen bestimmen sie dann ihre eigene Position.

Dies geschieht im einfachsten Fall ausschließlich durch eine Mittelwertbildung, das heißt die Position  $P_i$  eines Sensorknotens  $i$  ergibt sich aus den empfangenen Positionen  $B_j = (x_j, y_j, z_j)$  durch

$$P_i = (x_i, y_i, z_i) = \frac{1}{n} \sum_{j=1}^n B_j. \quad (3.4)$$

Dieses *Centroid Location* genannte Verfahren wird in [22] beschrieben und kommt ebenfalls in [21, 97] zum Einsatz. Diese Art der Positionsberechnung ist mathematisch besonders anspruchslos und lässt sich auch auf leistungsschwachen Mikrokontrollern leicht umsetzen. Allerdings hat sie den Nachteil, dass alle Sensorknoten, die die gleichen Anker zu ihren Nachbarn zählen, die exakt selbe Position berechnen, auch wenn sie sich in Wirklichkeit an unterschiedlichen Orten befinden.

Um diesen Nachteil zu kompensieren, wurden Erweiterungen vorgeschlagen, die die Positionen der Anker unterschiedlich gewichten [13]. Bei diesem *Weighted Centroid* oder *gewichtete Mittelung* genannten Ansatz ergibt sich die Position  $P$  des Sensorknotens  $i$  durch multiplikative Gewichte  $w_{ij}$  bezüglich der Ankerpositionen  $B_j$  gemäß der Formel

$$P_i = (x_i, y_i, z_i) = \frac{\sum_{j=1}^n (w_{ij} \cdot B_j)}{\sum_{j=1}^n w_{ij}}. \quad (3.5)$$

Das Gewicht  $w_{ij}$  ergibt sich aus der Distanz  $d_{ij}$  zwischen Ankern  $j$  und Sensorknoten  $i$ , dabei kann der Einfluss der Distanz mit Hilfe eines Exponenten  $e$  angepasst werden:

$$w_{ij} = \frac{1}{(d_{ij})^e}. \quad (3.6)$$

Konkret ließe sich beispielsweise der Signalstärkewert (auch *Link Quality Indicator (LQI)*) der Anker als Gewicht einsetzen, er ergibt sich  $w_{ij} = LQI_{ij}$ . In [11] wird dieses Verfahren verfeinert, um allgemeine Schwankungen in der Stärke der Signale von den Ankern auszugleichen. Die Autoren schlagen vor, einen gewissen Bruchteil  $q$  der niedrigsten Signalstärke von den Signalstärken aller Anker abzuziehen; die Position ergibt sich damit zu

$$P_i = (x_i, y_i, z_i) = \frac{\sum_{j=1}^n ((LQI_{ij} - q \cdot \min LQI_{i,1..n}) \cdot B_j)}{\sum_{j=1}^n (LQI_{ij} - q \cdot \min LQI_{i,1..n})}. \quad (3.7)$$

Der in [172] vorgestellte Ansatz arbeitet ähnlich, verwendet jedoch anstelle der gewichteten Mittelung die sogenannte *Monte-Carlo-Integration*.

Dem Vorteil der geringen Berechnungskomplexität steht bei all diesen Verfahren der Nachteil gegenüber, dass jeder Sensorknoten mehrere Anker in der direkten Nachbarschaft benötigt. Darüber hinaus können naturgemäß nur solche Positionen Ergebnis der Mittelung sein, die innerhalb der konvexen Hülle der Ankerpositionen liegen, obwohl dieses nicht zwangsläufig der tatsächlichen Lage entsprechen muss.

### Constraints und Optimierungen

Eine weitere Gruppe von Verfahren basiert auf der Auswertung von Beschränkungen der räumlichen Lage von Sensorknoten untereinander sowie zwischen Sensorknoten und Ankern. Im Rahmen eines globalen Optimierungsprozesses können dann die Positionen der Sensorknoten näherungsweise bestimmt werden.

Das in [88] vorgestellte, zentralisierte Verfahren geht davon aus, dass die Anker ihre Positionen broadcasten. Empfangen mehrere Sensorknoten diese Informationen, lässt sich zwischen ihnen ein *Constraint* aufstellen, das heißt, ihre möglichen relativen Positionen zueinander werden eingeschränkt. Aus allen diesen

Constraints lassen sich im Rahmen einer globalen Optimierung die Positionen der Sensorknoten berechnen.

Ähnlich arbeitet der Ansatz aus [56]. Hier werden die Nachbarschaftsbeziehungen aller Sensorknoten in einem sogenannten *Konnektivitätsgraphen* dargestellt. Mit Hilfe eines zentralen Optimierungsschrittes, der auf dem Verfahren der *linearen Programmierung* basiert, werden dann die resultierenden Knotenpositionen derart bestimmt, dass sie die Bedingungen des Konnektivitätsgraphen erfüllen.

Eine Erweiterung dieser beiden Ansätze wird in [178] beschrieben. Zusätzlich zu den Konnektivitätsbeziehungen wird hier berücksichtigt, dass Sensorknoten, die nicht direkt miteinander kommunizieren können, einen gewissen Mindestabstand haben müssen. Nach Aussagen der Autoren reduziert die Hinzunahme dieser Informationen den Fehler der berechneten Positionen um etwa ein Drittel.

Während die obigen Ansätze auf der binären Information der Konnektivität beruhen, werten die im Folgenden beschriebenen Verfahren Distanzen zwischen den Sensorknoten aus.

In [169] wird ein *MDS-MAP* genanntes, zentralisiertes Verfahren vorgeschlagen, das auf dem Prinzip der *Multidimensionalen Skalierung (MDS)* [16] beruht. Es berechnet die relativen Positionen der Sensorknoten untereinander mit einer Komplexität von  $O(n^3)$  unter der Annahme, dass alle relativen Distanzen zwischen den Sensorknoten bekannt sind. In einem weiteren Berechnungsschritt der Komplexität  $O(n)$  können die relativen Knotenpositionen mit Hilfe der Ankerpositionen dann in absolute Positionen überführt werden.

Mehrere Autoren schlagen vor, Positionsbestimmung auf Basis von Distanzinformationen mit Hilfe der *Maximum-Likelihood-Methode* durchzuführen. Während in [139] ein Verfahren zur Positionsbestimmung in mittels WLAN verbundenen Peer-to-Peer-Netzen vorgestellt wird, beschreibt [137] die Adaption für Sensornetze. Ausgehend von einigen wenigen bekannten Ankerpositionen sowie Informationen über die Distanzen zwischen benachbarten Sensorknoten und Ankern werden die Positionen zentral berechnet. Ähnliche Ansätze werden in [147, 148] beschrieben.

In [138] wird untersucht, inwieweit global optimierende Verfahren, die aus empfangenen Signalstärken erlangte Distanzinformationen verwenden, bessere Ergebnisse erzielen als solche Verfahren, die lediglich auf Konnektivitätsinformationen beruhen. Die Autoren kommen zu dem Ergebnis, dass erstere um bis zu 50% geringere Fehler erzielen können als letztere, was den Wert von Distanzinformationen für die Positionsbestimmung unterstreicht.

Allen bisher besprochenen Verfahren ist gemein, dass sie auf zentralen Berechnungen beruhen. Dieses ist meist schon aufgrund der immensen Komplexität der eingesetzten Verfahren erforderlich, die die Rechen- und Speicherkapazitäten eines Sensorknoten weit überschreiten. Das bringt jedoch mit sich, dass alle im Netz ermittelten Informationen über Konnektivität oder Distanzen zur Berech-

nungsinstanz geleitet werden müssen. Die berechneten Positionen müssen im Anschluss wiederum den einzelnen Sensorknoten mitgeteilt werden, so dass ein erheblicher Kommunikationsaufwand entsteht.

Dieser Nachteil führt dazu, dass die meisten Verfahren nur bedingt für Sensornetzwerke geeignet sind. Dieses schlägt sich auch darin nieder, dass praktisch keine entsprechenden Implementierungen existieren.

Es gibt jedoch auch verteilt arbeitende Verfahren, die auf Einschränkungen möglicher Positionen oder Optimierungen basieren.

Das in [191] beschriebene *Calamari-System* verwendet neben anderen Verfahrensbestandteilen, die sich in den Bereich der Multilateration einordnen lassen, einen *Resolution of Forces* genannten Ansatz, um bereits berechnete, grobe Positionsinformationen iterativ zu verfeinern. Hierbei wird das Spannungsverhältnis zwischen Distanzen zu Anker einerseits und der im Vorfeld bestimmten Position andererseits als eine Kraft mit Richtung und Betrag aufgefasst, die den Sensorknoten bildlich betrachtet zu seiner korrekten Position hinzieht. Durch iterative Anpassung der berechneten Positionen wird nun versucht, alle Kräfte im Sensornetz möglichst auf Null zu reduzieren.

Obwohl das in [81] beschriebene Verfahren völlig anders arbeitet als die bisher besprochenen, lässt es sich doch am ehesten in diesen Abschnitt einordnen, da es die mögliche Position eines Sensorknotens sukzessive einschränkt. Die Autoren gehen davon aus, dass die Anker im Netz über eine besonders hohe Kommunikationsreichweite verfügen, so dass jeder Sensorknoten seine Distanz zu vielen Anker sowie deren Positionen kennt. Die Sensorknoten bilden nun aus den Positionen von jeweils 3 Anker Dreiecke und prüfen, ob sie innerhalb dieser liegen. Dieses geschieht mit Hilfe des sogenannten *Approximate Point-In-Triangulation Test* (APIT). Der besagt, dass ein Sensorknoten innerhalb des Dreiecks aus Anker liegt, wenn keiner seiner Nachbarn näher an allen drei Anker liegt als er selber.

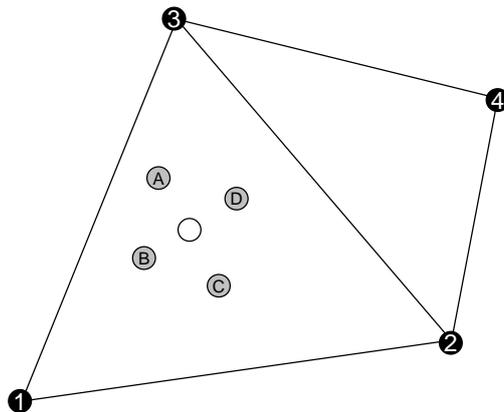


Abbildung 3.5.: Der Approximate Point-In-Triangulation Test.

Abbildung 3.5 verdeutlicht dieses an einem Beispiel. Bezüglich des Dreiecks aus den Anker 1, 2 und 3 liegt keiner der grauen Nachbarn näher an allen drei

Ankern als der weiße Sensorknoten, bezüglich der Anker 2, 3 und 4 jedoch liegt D näher an allen drei Ankern. Somit kann der weiße Sensorknoten schlussfolgern, dass er innerhalb des ersten, aber außerhalb des zweiten Dreiecks liegt. Alle Sensorknoten bilden nun den Schnitt all jener Dreiecke, innerhalb derer sie liegen, und schränken so ihre Position sukzessive ein. Schließlich platzieren sie sich in die Mitte die verbleibenden Fläche.

### Multilateration

Das meistverbreitete Verfahren zur Positionsbestimmung ist die Multilateration. Hierbei wird die gesuchte Position der nicht positionsbewussten Sensorknoten  $p = (x, y, z)$  aus den bekannten Positionen  $p_i = (x_i, y_i, z_i)$  der Anker sowie den Distanzen zu den Ankern  $d_i$  berechnet.

Lokationsverfahren, die nach dem Prinzip der Multilateration arbeiten, verlaufen im Allgemeinen in drei Phasen [102]:

1. Zuerst werden die Distanzen zwischen den Ankerknoten und den nicht positionsbewussten Sensorknoten ermittelt. Dieses geschieht meist, indem zunächst die Distanzen zwischen benachbarten Sensorknoten bestimmt werden, um diese im Anschluss zu den Distanzen zu den Ankern zusammenzuführen. Entsprechende Verfahren werden in den Abschnitten 5.1.2 bis 5.1.4 vorgestellt.
2. Aus diesen Distanzen und den Ankerpositionen wird dann die Position der Sensorknoten bestimmt.
3. Bei einigen Verfahren schließt sich eine Verfeinerung an. Hier werden große Unterschiede der im vorangegangenen Schritt erlangten Positionsinformationen bei benachbarten Sensorknoten ausgeglichen. Dieses kann zum Beispiel durch eine erneute, lokale Lateration oder Positionsmittelung geschehen.

Typisch für die Positionsbestimmung in Schritt zwei ist die Lateration. Dazu kommen die folgenden Gleichungen zum Einsatz.

$$\begin{aligned}
 (x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 &= d_1^2 \\
 &\vdots \\
 (x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2 &= d_n^2
 \end{aligned}
 \tag{3.8}$$

Dieses Gleichungssystem kann linearisiert werden, indem man die n-te Gleichung von den ersten n-1 Gleichungen subtrahiert.

$$\begin{aligned}
 x_1^2 - x_n^2 - 2(x_1 - x_n)x + y_1^2 - y_n^2 - 2(y_1 - y_n)y + \\
 z_1^2 - z_n^2 - 2(z_1 - z_n)z &= d_1^2 - d_n^2 \\
 &\vdots
 \end{aligned}
 \tag{3.9}$$

$$x_{n-1}^2 - x_n^2 - 2(x_{n-1} - x_n)x + y_{n-1}^2 - y_n^2 - 2(y_{n-1} - y_n)y + z_{n-1}^2 - z_n^2 - 2(z_{n-1} - z_n)z = d_{n-1}^2 - d_n^2$$

Eine Neuordnung der Gleichungen führt zu einem linearen Gleichungssystem der Form  $Ap = b$  mit

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) & 2(z_1 - z_n) \\ \vdots & \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) & 2(z_{n-1} - z_n) \end{bmatrix}, \quad (3.10)$$

$$b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + z_1^2 - z_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + z_{n-1}^2 - z_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix}. \quad (3.11)$$

Dieses Gleichungssystem kann nun mittels des Verfahrens der kleinsten Quadrate gelöst werden:

$$p = (A^T A)^{-1} A^T b. \quad (3.12)$$

In Einzelfällen kann es vorkommen, dass die inverse Matrix aufgrund von fehlerbehafteten Distanzen nicht berechnet werden kann, und die Lateration scheitert. In der Mehrheit der Fälle jedoch ist die Lateration erfolgreich und liefert eine Position  $p$ . Einen weiteren Plausibilitätstest ermöglicht die Berechnung des sogenannten Residuums

$$r = \frac{\sum_{i=1}^n \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 - d_i^2}}{n}. \quad (3.13)$$

Ein großes Residuum signalisiert ein inkonsistentes Gleichungssystem. Daher ist es sinnvoll, die auf diese Weise berechnete Position  $p$  zu verwerfen, wenn der Wert des Residuums die mittlere Kommunikationsreichweite überschreitet [102]. Stehen mehr Gleichungen zur Verfügung als zur Positionsberechnung notwendig sind, kann diejenige Menge von Gleichungen zur Berechnung ausgewählt werden, die zum kleinsten Residuum führt.

Die große Anzahl von Publikationen, die Lokationsverfahren nach dem Prinzip der Lateration beschreiben, belegt die hohe Akzeptanz dieses Verfahrens. Beispiele für die Funktion nach dem oben beschriebenen Phasenschema finden sich in [124, 125, 127, 156, 157, 161], wobei sich die entsprechenden Verfahren lediglich in Details wie der Art der Distanzschätzung, der Bewertung der Residuen oder der Art der Verfeinerung unterscheiden.

Andere Beiträge verzichten auf das Zusammenführen von Distanzen über mehrere Hops hinweg. Hier ist eine Positionsbestimmung für solche Sensorknoten möglich, die über mindestens drei Anker als direkte Nachbarn verfügen [12].

Um diesen Nachteil zumindest abzuschwächen, wurden verschiedene iterative Ansätze vorgeschlagen [1, 4, 36, 151, 159]. Hier übernehmen Sensorknoten, die

bereits eine Position berechnet haben, im Weiteren die Rollen von Anker, so dass weitere Knoten ihre Position bestimmen können.

Neben der Tatsache, dass die Laterationsberechnung scheitern kann, ist der hohe mathematische Aufwand gerade für leistungsschwache Sensorknoten ein Nachteil. Als Ausweg wurde das Min-Max-Verfahren vorgeschlagen [160, 171].

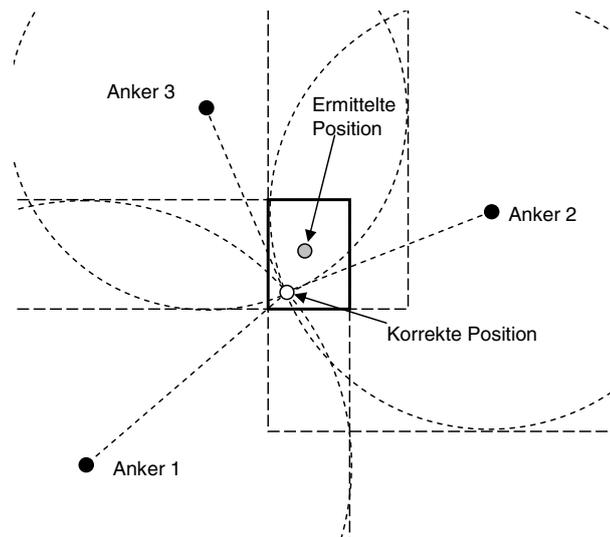


Abbildung 3.6.: Positionsbestimmung mit dem Min-Max-Verfahren.

Auch hier wird die Position der Sensorknoten aus Distanzen zu Anker sowie deren Positionen berechnet (vergleiche Abbildung 3.6). Dazu wird um die Ankerpositionen (schwarz ausgefüllte Kreise) jeweils ein Quadrat mit der doppelten Kantenlänge des Abstandes zu diesem Anker gelegt. Die Schnitte dieser Quadrate um alle bekannten Anker ergeben ein Rechteck, in dessen Mitte sich der Sensorknoten positioniert. Gegenüber der klassischen Lateration ist der Berechnungsaufwand erheblich geringer, allerdings wird in aller Regel nicht die exakt richtige Position (weißer Kreis) berechnet, sondern lediglich eine Näherungslösung (grau ausgefüllter Kreis).

Während [171] nur dann eine Position berechnen kann, wenn mindestens drei Anker zu den direkten Nachbarn eines Sensorknoten gehören, ermittelt das *N-hop Multilateration* genannte Verfahren aus [160] auch dann Positionen, wenn die Anker mehrere Hops weit entfernt sind. Dazu verwendet es das *Sum-Dist-Schema* (siehe auch Abschnitt 5.1.4), das Distanzen über mehrere Hops hinweg addieren kann.

Neben den bisher beschriebenen Verfahren, die einerseits verteilt und andererseits netzweit arbeiten, gibt es auch Vorschläge, das Sensornetz in Cluster zu unterteilen. Innerhalb dieser Cluster werden im ersten Schritt Positionen berechnet, die meist relativ zu einem Clusterhead bestimmt werden. Hierzu kommen prinzipiell die bisher beschriebenen Verfahren zu Einsatz, zum Teil jedoch in abgewandelter Form und ergänzt durch trigonometrische Erwägungen. In einem zweiten Schritt werden dann die bezüglich lokaler Koordinatensysteme

me angegebenen Positionen in ein globales System überführt. Beispiele für diese Vorgehensweise finden sich etwa in [91] und [157].

Ein problematischer Punkt an der Multilateration ist die Schwierigkeit, die Distanzen zwischen den Knoten und den Anker zu bestimmen. Dazu kommen *Distanzschätzverfahren* zum Einsatz, deren Genauigkeit die Präzision der ermittelten Positionen maßgeblich beeinflusst. Diese Verfahren arbeiten jedoch häufig ungenau oder langsam, benötigen zusätzliche Hardware wie spezialisierte Sende- und Empfangseinrichtungen oder weisen eine hohe Berechnungskomplexität auf. Die einzelnen Verfahren sowie ihre Vor- und Nachteile werden in den Abschnitten 5.1.2 bis 5.1.5 vorgestellt.

### Multiangulation

Eine weitere Gruppe bilden die Verfahren nach dem Prinzip der Multiangulation. Hier werden neben der bekannten Lage der Anker Winkelinformationen zur Positionsbestimmung eingesetzt. Die entsprechenden Verfahren wurden meist ursprünglich zur Positionsbestimmung im Bereich der Luft- und Seefahrt entwickelt.

Einen guten Überblick über die möglichen Varianten der Multiangulation gibt [128]. Grundsätzlich lassen sich zwei Methoden unterscheiden:

- *VOR* oder auch *Radial*: die Kurzform VOR steht für *VHF Omnidirectional Radio Range*, VHF wiederum heißt *Very High Frequency* - zu Deutsch Ultrakurzwellen (UKW). VOR bedeutet also übersetzt „UKW-Drehfunkfeuer“.
- *Angle of Arrival (AoA)* oder auch *Bearing*

Die VOR-Methode verlangt, dass die Anker Signale in einem schmalen Winkel aussenden können, der über die Zeit variiert. Auf diese Weise wird ähnlich einem Leuchtturm ein rotierender Signalstrahl versendet, in dem die Position des Senders sowie der aktuelle Winkel kodiert sind. So können die Empfänger ihre Position auf eine Gerade in der Ebene (zweidimensionaler Fall) beziehungsweise auf eine Ebene im Raum (dreidimensionaler Fall) einschränken.

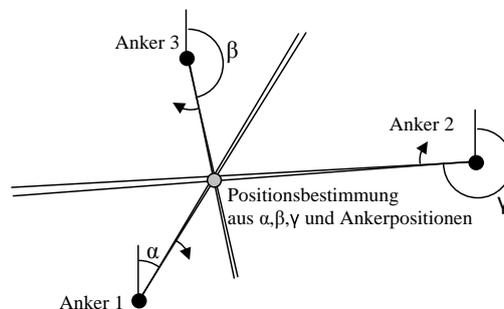


Abbildung 3.7.: Positionsbestimmung mit der VOR-Methode.

Empfängt ein Sensorknoten entsprechende Informationen mehrerer Anker, so

kann er die jeweiligen Geraden- oder Ebenengleichungen gleichsetzen und auf diese Weise seine Position berechnen. Dieses Vorgehen wird in Abbildung 3.7 veranschaulicht, die entsprechenden Berechnungsvorschriften können [130] entnommen werden. Diese Methode verlangt, dass die Anker ihre absolute Orientierung kennen. Vorteilhafterweise ist es nicht erforderlich, dass die übrigen Sensorknoten die Richtung, aus der die Signale kommen, detektieren können.

Die Autoren von [126] schlagen Übertragung der VOR-Methode auf Sensornetze vor. Vereinfachend gehen sie aber davon aus, dass vier Anker in einem Rechteck am Rand des Sensornetzes angeordnet sind. Diese senden rotierende, in einem engen Winkel begrenzte Signale aus, die im gesamten Sensornetz empfangen werden können. Dabei können die Sensorknoten den Winkel zu den Ankern ermitteln, indem sie den bekannten Versatz der Winkel der Ankersignale nutzen, um den Empfangszeitpunkt der Signale aller Anker in einen Winkel zu überführen. Die Berechnung der Positionen der Sensorknoten kann dann lokal im Prinzip nach der oben beschriebenen Methode erfolgen, durch die Rechteckanordnung der Anker ergeben sich zusätzliche Vereinfachungen.

Die AoA-Methode hingegen erfordert, dass die Sensorknoten feststellen können, aus welcher Richtung sie die Signale der Anker empfangen, die ihrerseits omnidirektionale Sender sind. Zur Verdeutlichung des weiteren Vorgehens (hier für den zweidimensionalen Fall) sei auf Abbildung 3.8 verwiesen.

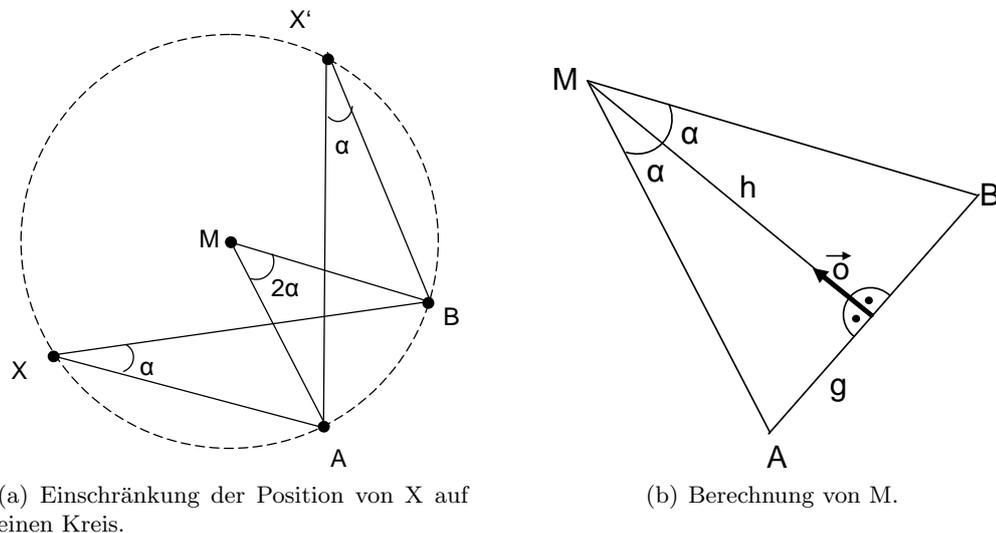


Abbildung 3.8.: Rückführung der Multiangulation auf die Multilateration.

Empfängt ein Sensorknoten X entsprechende Signale von zwei Ankern A und B mit deren Positionen, so kann er den Winkel  $\alpha$  zwischen den beiden Ankern durch Subtraktion der Empfangswinkel berechnen. Mit Hilfe von  $\alpha$  kann X nun seine Position auf einen Kreis um den Punkt M einschränken, auf dem auch A und B liegen (Abbildung 3.8(a)). Dabei können der Punkt M wie auch der Radius des Kreises  $r$  aus den Positionen von A und B sowie dem Winkel  $\alpha$

berechnet werden.

Es gilt

$$\vec{M} = \vec{A} + \frac{\vec{B} - \vec{A}}{2} + h\vec{o}, \quad (3.14)$$

$$h = \frac{g}{\sin \alpha}, \quad (3.15)$$

$$r = \sqrt{g^2 + h^2}, \quad (3.16)$$

wobei  $\vec{o}$  den auf  $\overline{AB}$  senkrecht stehenden Vektor der Länge 1 bezeichnet. Somit erhält X die Information, dass X sich in der Distanz  $r$  von Punkt  $M$  befindet. Mit Hilfe von weiteren Anker kann auf gleiche Weise der Abstand zu weiteren Punkten ermittelt werden. Die Positionsberechnung von X erfolgt dann mittels Multilateration.

Ein besonderer Vorteil von Angle-of-Arrival ist, dass es neben der Positionsbestimmung auch die Ermittlung der Ausrichtung der Sensorknoten ermöglicht. Dann kann beispielsweise die Position eines Objektes weiter eingeschränkt werden, das mittels eines gerichteten Sensors detektiert wurde.

Das in [128] vorgestellte Verfahren zur Positionsermittlung nutzt die soeben beschriebene Vorgehensweise. Um darüber hinaus auch dann Positionen ermitteln zu können, wenn sich nicht ausreichend viele Anker in direkter Nachbarschaft befinden, wird ein *DV-Bearing* genannter Ansatz beschrieben, der die Bestimmung von Winkeln über mehrere Hops hinweg ermöglicht.

Ein zentralisierter Ansatz wird in [7] vorgestellt. Hier kommt ebenfalls Angle-of-Arrival zum Einsatz, allerdings werden alle Winkelmessungen zu einem großen Gleichungssystem zusammengefasst und zentral die resultierenden Positionen berechnet. Die Betrachtung eines Systems von Winkeln findet ihren Ausdruck auch in der Verfahrensbezeichnung *Robust System Multiangulation*.

Insgesamt konnten sich Ansätze zur Positionsbestimmung auf Basis der Multiangulation nur sehr bedingt durchsetzen, was sich neben der relativ geringen Anzahl der Publikationen auch an der minimalen Anzahl von vorgestellten praktischen Realisierungen ablesen lässt. Ursächlich ist neben dem höheren mathematischen Aufwand durch den zusätzlichen Berechnungsschritt bei der Rückführung auf die Multilateration die hohe technische Komplexität geeigneter Sender und Empfänger. Obwohl die für den VOR-Ansatz notwendigen rotierenden schmalwinkligen Sender prinzipiell auch ohne mechanisch bewegliche Teile umsetzbar wären, sind keine entsprechenden Realisierungen für drahtlose Sensornetze bekannt. Für den Angle-of-Arrival-Ansatz gibt es zwar prototypische Umsetzungen, diese benötigen jedoch mehrere Empfangseinheiten pro Gerät. Das wirkt sich negativ auf Energieverbrauch, Preis und Baugröße der Geräte aus, so dass die Multiangulation speziell im Vergleich mit der Multilateration gravierende Nachteile aufweist.

### Multimodale Verfahren

Aufbauend auf den beschriebenen Verfahren der Multilateration und Multiangulation gibt es in der Literatur eine Reihe von Vorschlägen, wie beide Ansätze kombiniert werden können. Die Autoren nennen neben besonderer Flexibilität gute Genauigkeiten schon bei einer geringen Anzahl von Ankern als Vorteile ihrer Ansätze.

So wird in [121] gezeigt, dass bereits mit nur zwei Ankern und drei Sensorknoten die relativen Positionen aller beteiligten Geräte abgeleitet werden können. Dabei ist es nicht zwingend erforderlich, dass die Anker ihre absoluten Positionen kennen. Fehlen die entsprechenden Informationen, werden alle Positionen in einem relativen Koordinatensystem bestimmt. Dazu kommen hier Distanzschätzungen und Angle-of-Arrival-Messungen zum Einsatz. Während dieses Verfahren bereits verteilt arbeitet, präsentierten die Autoren in [122] noch einen ähnlichen, zentralen Ansatz.

Auch in [59] wird betont, dass im Vergleich zu [131] eine ähnliche Genauigkeit bei weniger Ankern erreicht werden kann. Hier wird der VOR-Ansatz mit Signalstärkemessungen kombiniert. Zur Positionsbestimmung aus diesen unterschiedlichen Eingangsgrößen kommen sogenannte *Bayessche Netze* zum Einsatz, die wiederum auf bekannte Lokationsverfahren zurückgreifen.

Die Autoren von [129] betonen primär die besondere Flexibilität ihres Ansatzes, so dass das *Local Positioning System (LPS)* genannte Verfahren auf Geräten mit unterschiedlichen technischen Voraussetzungen im Hinblick auf die Positionsbestimmung verwendet werden kann: LPS kann Anker mit bekannten Positionen, Kompassdaten, Distanzschätzungen und Angle-of-Arrival-Messungen flexibel für die Positionsbestimmung nutzen.

Insgesamt haften den kombinierten Verfahren inhärent die Nachteile der Multiangulation an. Erschwerend kommt die durch die Kombination höhere Komplexität der Verfahren hinzu.

### 3.3. Zusammenfassung

Dieses Kapitel beschäftigte sich mit den beiden wichtigsten Kontextdimensionen in drahtlosen Sensornetzwerken: Positionsbewusstsein und Zeitbewusstsein.

In ersten Teil wurde auf Verfahren zur Errichtung eines konsistenten Zeitbewusstseins eingegangen. Es wurde verdeutlicht, dass es für die Phasen der Zeitdifferenzermittlung, der Uhrenanpassung und der Multihopweiterleitung recht unterschiedliche Vorgehensweisen gibt. Während bei der Post-facto-Synchronisation wie beispielsweise von Römer vorgeschlagen [149] immer dann zusätzliche Informationen lokal ausgetauscht werden, wenn ein konsistentes Zeitverständnis benötigt wird, stellen die proaktiven Verfahren bereits im Vorfeld eine netzweit einheitliche Zeit her. Ein typisches Beispiel hierfür ist das Flooding Time Synchronization Protocol [112].

Kapitel 4 geht auf diesen Gegensatz näher ein, untersucht das resultierende Datenaufkommen und schlägt schließlich eine Möglichkeit vor, die Vorteile beider Ansätze zu kombinieren.

Im zweiten Teil des Kapitels wurde eine Vielzahl von Verfahren vorgestellt, die sich zum Ziel gesetzt haben, jeden Knoten in einem Sensornetz mit Informationen über seine Position zu versorgen. Dabei wurde deutlich, dass die meisten Verfahren auf Basis von Informationen über die Distanz zu anderen Sensorknoten arbeiten, die somit die wichtigste Eingangsgröße zur Positionsermittlung darstellt.

Während die Algorithmen für die eigentliche Positionsbestimmung bereits sehr ausgereift sind, zeigen die Verfahren zur Bestimmung der Distanzen weiterhin zum Teil erhebliche Schwächen. Aus diesem Grund untersucht Kapitel 5 bekannte Verfahren zur Distanzschätzung und zeigt ihre Defizite auf. Es stellt darüber hinaus ein neues Verfahren zur Distanzermittlung vor und ordnet dessen Leistungen gegenüber bestehenden Verfahren ein.



## 4. Kontextadaptive Errichtung konsistenten Zeitbewusstseins

Nachdem im ersten Teil des vorigen Kapitels die Funktionsweise von Verfahren zur Errichtung eines konsistenten Zeitverständnisses erläutert wurde, beschäftigt sich dieses Kapitel mit dem aus den konzeptionellen Unterschieden resultierenden Kommunikationsaufwand. Da dieser Aufwand nicht direkt der eigentlichen Aufgabe des Sensornetzes zuzurechnen ist, sondern aus dem Verfahren zur Erlangung eines konsistenten Zeitverständnisses resultiert, wird er im Folgenden *Overhead* genannt. Es wird eine Methode vorgeschlagen, die Vorteile verschiedener bestehender Verfahrenansätze zu kombinieren. Der resultierende Overhead wird mit dem anderer Verfahren simulativ verglichen. Aus den Simulationsergebnissen wird dann ein Modell entwickelt, mit dessen Hilfe der Overhead verschiedener Verfahren prognostiziert werden kann. Die hier vorgestellten Ergebnisse wurden bereits veröffentlicht [25, 30].

### 4.1. Kommunikationsaufkommen bestehender Verfahren

Wie in Kapitel 3.3 angesprochen, lassen sich die bestehenden Verfahren zur Errichtung eines konsistenten Zeitverständnisses in *proaktiv* und *post-facto* arbeitende Protokolle einteilen.

#### 4.1.1. Proaktive Synchronisation

Die proaktiven Verfahren überwiegen in der Literatur zahlenmäßig. Bei diesen Verfahren wird von einem Sensorknoten, dem sogenannten *Synchronisationsmaster*, ausgehend dessen Zeit an alle anderen Geräte verteilt. Diese stellen dann ihre interne Uhr auf die Uhrzeit des Ausgangsgerätes (des Zeitgebers) um, so dass im Anschluss alle Uhren (bis auf eine gewisse Ungenauigkeit) synchronisiert sind. Somit geben nach diesem Synchronisationsvorgang die Uhren aller Geräte eine nahezu gleiche Zeit an. Werden nun von verschiedenen Knoten Sensorereignisse aufgezeichnet und mit einem Zeitstempel versehen, beziehen sich alle Zeitstempel auf ein konsistentes Zeitsystem. Wird ihr Drift nicht ausgeglichen, verschieben sich jedoch die Zeitsysteme langsam wieder gegeneinander. Abhängig vom Ausmaß des Drifts und der Toleranz der Anwendung gegenüber Ungenauigkeiten muss der Synchronisationsvorgang dann nach einer bestimmten Zeit wiederholt werden.

Bei der Zeitsynchronisation entsteht bei jedem Synchronisationsvorgang, bedingt durch das Verteilen der Uhrzeit, ein bestimmter Overhead, während danach bei der normalen Kommunikation im Sensornetz keinerlei Overhead anfällt.

#### 4.1.2. Post-facto Synchronisation

Eine grundlegend andere Herangehensweise stellt das Umrechnen der Zeiten in das lokale Zeitsystem des Empfängergerätes bei jedem Kommunikationsvorgang, die post-facto Synchronisation, dar, die Römer in [149] vorgeschlägt. Dabei werden bewusst die unterschiedlichen Zeitbasen der verschiedenen Geräte beibehalten, der Overhead für das Synchronisieren der Uhren entfällt. Statt dessen wird beim Versenden eines Datenpaketes, das Zeitinformationen enthält, der Zeitpunkt des Versandes vom Sender an das Paket angefügt. Der Empfänger kann durch Vergleich dieses Zeitstempels mit seiner lokalen Uhr (unter Einberechnung einer gewissen Kommunikationszeit) den Versatz zwischen seiner und der Uhr des Senders errechnen und alle Zeitinformationen im Datenpaket entsprechend anpassen.

Das Verfahren von Römer lässt sich verbessern, indem wie in [72] vorgeschlagen der Zeitstempel direkt vor dem Versand in der MAC-Schicht des Protokollstapels eingefügt wird (vergleiche Abschnitt 3.1.1). Somit reicht es aus, in einem Paket, das Informationen zum Beispiel zum Zeitpunkt eines Sensorereignisses enthält, einen weiteren Zeitstempel einzufügen, der auf den Zeitpunkt des Paketversandes verweist. Anhand dieses Zeitstempels kann dann die Umrechnung der Zeitinformationen des Ereignisses in das Zeitsystem des Empfängers erfolgen. Als Folge wird lediglich das gesendete Datenpaket um wenige Bytes verlängert, der in [149] beschriebene Austausch weiterer Nachrichten entfällt.

Verfahrensinhärent entsteht bei jedem Datenpaket, das zwischen Sensorknoten ausgetauscht wird und das Zeitinformationen enthält, durch den zusätzlichen Zeitstempel ein gewisser Overhead.

An dieser Stelle ist anzumerken, dass nicht alle in einem Sensornetz versendeten Pakete Zeitinformationen enthalten. Dennoch ist die Einordnung von ausgetauschten Daten in einen zeitlichen Kontext kennzeichnend für Sensornetze [115]. Daher ist davon auszugehen, dass ein nicht zu vernachlässigender Anteil der Pakete Zeitinformationen enthält, und dementsprechend der Overhead für den zusätzlichen Zeitstempel häufig erforderlich ist.

Im weiteren Verlauf dieses Kapitels sind bei der Betrachtung von Paketaustauschraten oder des Verkehrsaufkommens im Netz stets nur solche Pakete gemeint, die tatsächlich Zeitinformationen enthalten.

#### 4.1.3. Zusammenhang zwischen Kommunikation und Overhead

Im direkten Vergleich der beiden Verfahren ergibt sich also die Bilanz, dass die proaktiven Verfahren durch das Verteilen der Zeit im Netz einen Startoverhead

mit sich bringen, der von der Größe des Sensornetzes, dem Synchronisationsintervall sowie der Größe der benötigten Pakete abhängig ist. Pro Synchronisationsvorgang und Sensorknoten ist der Aufwand praktisch konstant. Erfolgt die Synchronisation beispielsweise durch Fluten, muss jeder Knoten genau ein Synchronisationspaket versenden.

Bei der Post-facto-Umrechnung der Zeiten stellt sich die Situation grundlegend anders dar. Hier gibt es keinen prinzipbedingten Startoverhead, dafür fällt bei jedem Kommunikationsvorgang durch den zusätzlich übertragenen Zeitstempel Overhead an. Der Gesamtoverhead hängt also netzweit wie auch pro Knoten sowohl von der Anzahl der versendeten Datenpakete als auch von der Länge des Zeitstempels ab.

Dieses legt nahe, dass die proaktive Zeitsynchronisation bei hohem Kommunikationsaufkommen im Netz besonders geeignet ist, weil sich der Startoverhead durch die Ersparnis pro Paket schnell amortisiert. Dagegen ist die Umrechnung von Zeiten bei geringem Datenverkehr im Netz von Vorteil, weil hier kein Startoverhead anfällt. Mit fortschreitender Zeit wird aber der Overhead langsam steigen, bis zu einem bestimmten Zeitpunkt genau die Menge an Overhead erreicht ist, die auch durch die Synchronisation angefallen wäre. Es bleibt jedoch zu bedenken, dass bei der proaktiven Vorgehensweise die Synchronisation früher oder später wiederholt werden muss, um den Drift der lokalen Uhren gegeneinander auszugleichen. Zu diesem Zeitpunkt endet folglich der Betrachtungszeitraum, und die Analyse reduziert sich auf die Fragestellung, welche der beiden Varianten bis dahin weniger Overhead verursacht hat.

## 4.2. Ein kommunikationsabhängiges Verfahren zur Errichtung konsistenten Zeitbewusstseins

Neben diesem recht offensichtlichen Ergebnis stellt sich die Frage, wie sich die Situation darstellt, wenn das Verkehrsaufkommen im Sensornetz regional und zeitlich schwankt.

Um dieser Situation besser gerecht werden zu können, wird im Weiteren jetzt ein neues Verfahren vorgestellt, das die Vorteile beider Ansätze kombiniert. Die Grundidee dabei ist, dass es in Sensornetzen Regionen gibt, in denen das Kommunikationsaufkommen temporär ansteigt, um danach wieder auf das normale Niveau zurückzufallen. Dieses könnte der Fall sein, wenn an einer bestimmten Stelle das zu beobachtende Phänomen auftritt. Daraufhin steigt der Kommunikationsbedarf beispielsweise durch Koordination zwischen den Knoten oder durch Datenfusion zeitweise an und führt zu einer sogenannten *temporären Hochverkehrsregion*.

Das neue, kombinierte Verfahren mit dem Namen *Traffic Induced Control of Time And Communication (TICTAC)* geht davon aus, dass ein zu erwartender Anstieg des Kommunikationsaufkommens in Grenzen bezüglich Zeitpunkt, Dauer, Intensität und räumlicher Ausdehnung vorhersagbar ist. Das könnte bei-

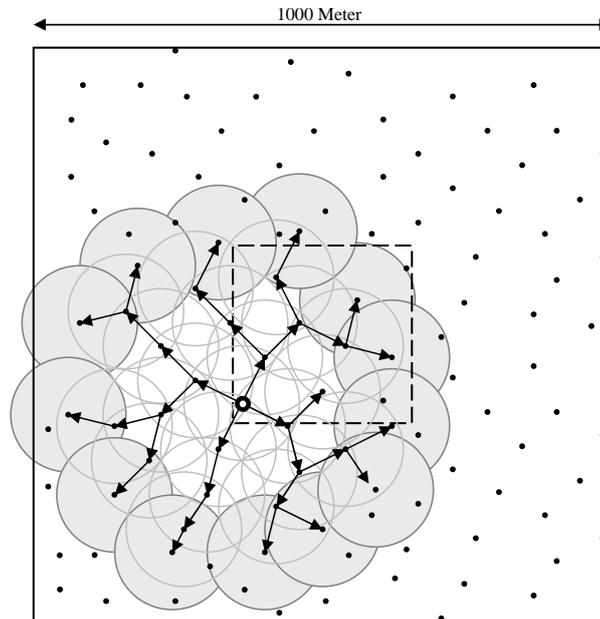


Abbildung 4.1.: Sensornetzbereich (durchgezogenes Quadrat), Hochverkehrsbeereich (gestricheltes Quadrat), sowie die Entwicklung des vom dick eingezeichneten Knoten angestoßenen Flutens bei TIC-TAC (qualitative Darstellung).

spielsweise der Fall sein, wenn ein Sensorknoten als Erster das zu beobachtende Phänomen wahrnimmt.

Abbildung 4.1 zeigt qualitativ ein Sensornetz innerhalb einer quadratischen Fläche mit 1000 Metern Seitenlänge. Die temporäre Hochverkehrsregion, in der zeitlich begrenzt verstärkte Kommunikation auftritt, ist durch ein gestricheltes Quadrat angedeutet.

Nachfolgend wird die Arbeitsweise von TICTAC beschrieben, die auch noch einmal in Algorithmus 1 bis 4 zusammenfassend dargestellt wird. Bei normalem, das heißt geringem Verkehrsaufkommen, wird das Verfahren von Römer [149] angewandt. Ist zu erwarten, dass die Kommunikation an einer Stelle des Netzes zunehmen wird, leitet einer der Knoten in der betroffenen Region eine lokale Synchronisation der Uhren ein. Der Einfachheit halber wird zunächst davon ausgegangen, dass der erste Knoten, der vermehrte Kommunikation vermutet, mit der Synchronisation beginnt. Dazu flutet er seine momentane Uhrzeit an die Nachbarn innerhalb einer gewissen (in Hops gemessenen) Entfernung.

Die Steuerung dieses Flutungsvorganges erfolgt mit Hilfe eines speziellen Feldes im Synchronisationspaket, das bei jedem Weiterleiten dekrementiert wird und die Anzahl der noch verbleibenden Weiterleitungsschritte angibt (das *Time-to-live-* oder *TTL-Feld*). Erhält ein Sensorknoten ein solches Synchronisationspaket mit einer Null in diesem Feld, wird er seine Uhr zwar der des Zeitgebers anpassen, sich aber anders verhalten als diejenigen Knoten, die ein solches Paket mit einem höheren Wert bekommen haben. Einerseits wird er das Paket

nicht weiterleiten, darüber hinaus wird er aber auch weiterhin Datenpakete versenden, die (wie beim Verfahren von Römer) noch einen Zeitstempel mit der Versanduhrzeit enthalten. Diese Knoten bilden somit einen Ring (siehe auch die dunkel gefärbten Kommunikationsradien in Abbildung 4.1) um die Flutungszone, der ermöglicht, dass Knoten außerhalb der Zone die aus der Flutungszone kommenden Daten korrekt in ihren lokalen zeitlichen Kontext einordnen können.

Der initiale Wert im TTL-Feld, der die Ausdehnung der lokalen Flutungszone steuert, muss in Abhängigkeit von der erwarteten Ausdehnung der Hochverkehrsregion gewählt werden. Geht man davon aus, dass der Initiator im ungünstigsten Fall am Rand der Region liegt, ergibt sich der passende Wert aus dem maximalen Durchmesser der Region geteilt durch die mittlere pro Weiterleitungsvorgang zurückgelegte Distanz.

Empfangen Knoten in dem Ring Daten von außerhalb (sie können dies am von ihrer eigenen Uhr abweichenden Versandstempel erkennen), rechnen sie alle Zeitdaten in ihr eigenes und somit das Zeitsystem in der Flutungszone um. Erhalten Knoten innerhalb des Ringes Datenpakete mit zusätzlichem Zeitstempel, können sie diesen ignorieren, da das Paket nur aus dem Ring stammen kann. Knoten innerhalb des Ringes verhalten sich genau so, als wäre das gesamte Sensornetz synchronisiert. Außerhalb des Ringes wird das Verfahren von Römer angewandt.

Es ist zu erwarten, dass TICTAC sich sowohl hinsichtlich des Overheads durch das Fluten als auch hinsichtlich des Overheads pro Kommunikationsvorgang außerhalb der Flutungszone zwischen der proaktiven Zeitsynchronisation und dem Verfahren von Römer einordnet. Der Flutungsoverhead hängt maßgeblich davon ab, auf welchen Wert der Zeitgeber das TTL-Feld setzt, das heißt wie groß die Region ist, in der die Uhren synchronisiert werden. Dabei steigt der entstehende Overhead in etwa quadratisch mit der Anzahl der Hops. Je größer dieser Wert ist, desto größer ist die Wahrscheinlichkeit, dass die Hochverkehrsregion komplett von der Flutungszone abgedeckt wird. Der Grad dieser Abdeckung ist wiederum ein Gradmesser für die mögliche Ersparnis im Verlauf des Betriebs des Sensornetzes nach einer lokalen Synchronisation. Je besser die Abdeckung, desto mehr Overhead fiele durch versendete Zeitstempel prozentual in diese Region und kann dadurch potenziell in der Flutungszone eingespart werden. Darüber hinaus hängt die mögliche Ersparnis davon ab, wie intensiv das Kommunikationsaufkommen in der Hochverkehrsregion ist.

---

**Algorithmus 1** On initialization

---

```
state:= RÖMER;
```

---

---

**Algorithmus 2** To trigger local synchronization

---

```
state:= SYNCED;
packet.type := SYNC;
packet.ttl := hopcount;
packet.time := localtime;
send packet;
```

---

---

**Algorithmus 3** On reception of a packet

---

```
if packet.type = SYNC then
  local time := packet.time;
  if packet.ttl>0 then
    state:= SYNCED;
    packet.ttl := packet.ttl-1;
    packet.time := local time;
    send packet;
  else
    state:=RING;
  end if
else
  if packet.type=RDATA then
    timedelta := localtime - p.time;
    convert all time stamps in packet.data using time delta;
    pass packet.data to application;
  else
    if packet.type=DATA then
      pass packet.data to application;
    end if
  end if
end if
```

---

---

**Algorithmus 4** To send a packet with data

---

```
packet.data := data;
if state=RÖMER or state=RING then
  packet.type := RDATA;
  packet.time := localtime;
else
  packet.type := DATA;
end if
send packet;
```

---

## 4.3. Simulative Evaluation

Um diese analytischen Vorüberlegungen zu untermauern, wurde zunächst simulativ untersucht, welches der Verfahren unter welchen Bedingungen besonders effizient arbeitet. Als Repräsentant für die proaktiven Verfahren kam das Fluten der Zeit durch das gesamten Netz in Anlehnung an [112] zum Einsatz.

Alle drei Verfahren wurden so modifiziert, dass sie die in [72] vorgeschlagene Technik der Zeitstempel auf Ebene der Medienzugriffs- beziehungsweise Sicherungsschicht einsetzen.

### 4.3.1. Simulationsumgebung

Zur Ermittlung der entstehenden Overheads wurde mit dem Netzwerksimulator ns-2 [134] eine Serie von Simulationen durchgeführt. Der in den Simulationen betrachtete Zeitraum beginnt jeweils, wenn in der Hochverkehrsregion ein erhöhtes Kommunikationsaufkommen entsteht. Folglich ist das Netz in zwei Regionen aufgeteilt: Im Großteil des Netzes herrscht über die gesamte Simulationsdauer ein gleich bleibender Verkehr, es werden Daten mit einer niedrigen Rate von Paketen pro Zeiteinheit (low traffic rate,  $r_{LT}$ ) ausgetauscht. In einem kleinen Teil des Netzes herrscht zu Beginn für eine begrenzte Zeit (high traffic time,  $t_{HT}$ ) ein stärkerer Verkehr mit hoher Paketrate (high traffic rate,  $r_{HT}$ ), danach herrscht auch hier wie im Rest des Netzes der geringere Verkehr.

Um die Verkehrsraten  $r_{HT}$  und  $r_{LT}$  festzulegen, werden mittlere Intervalle zwischen dem Versenden von zwei Paketen vorgegeben.

Soll eine der Raten beispielsweise ein Paket pro 30 Sekunden betragen, entnimmt ein Knoten nach erfolgtem Paketversand eine Zufallszahl zwischen 0 und 60 Sekunden aus einer Zufallsverteilung und versendet das nächste Paket nach der entsprechenden Zeit. Auf diese Weise versenden die Knoten im Schnitt alle 30 Sekunden ein Datenpaket.

Es kommen drei Typen von Zufallsverteilungen zum Einsatz: Normal-, Poisson- und Gleichverteilungen. Verteilungen, die wie die ersten beiden eine Häufung aufweisen, führen durch die steigende Wahrscheinlichkeit eines ähnlichen Sendezeitpunktes zu einer sprunghaften Entwicklung des resultierenden Overheads über die Zeit. Das erschwert den Vergleich der Ergebnisse sowie die spätere Modellbildung. Darüber hinaus hat die Wahl des Verteilungstyps allerdings keine erkennbaren Auswirkungen auf die Aussagen der Ergebnisse. Aus diesem Grund werden hier Simulationsergebnisse gezeigt, bei denen Gleichverteilungen für die Interpaketintervalle verwendet wurden.

Die Simulationen fanden auf einem quadratischen Areal der Größe  $A$  mit 1000 Metern Seitenlänge statt. In dessen Zentrum befindet sich ein Quadrat, das einen Anteil  $p_{HT}$  von 10% der Gesamtfläche einnimmt. Es hat folglich eine Seitenlänge von 316 Metern und ist Abbildung 4.1 gestrichelt dargestellt. In diesem Quadrat herrscht zu Beginn der Simulationen stärkerer Verkehr.

Auf der gesamten Simulationsfläche wurden Sensorknoten zufällig uniform verteilt. Die Anzahl der Knoten wurde so gewählt, dass sich bei einem eingestellten Kommunikationsradius von 100 Metern eine mittlere Netzdichte von etwa 10 Nachbarn pro Knoten ergibt. Dieses ist die minimale Dichte, die in etwa erforderlich ist um sicherzustellen, dass das Netz nicht partitioniert ist [83]. Diese Wahl stellt sicher, dass ein bei dieser Dichte funktionierendes Verfahren auch bei höheren Werten wie etwa den typischerweise notwendigen 15 Nachbarn reibungslos arbeitet. Aufgrund der gewählten Dichte ergibt sich eine Knotenzahl  $n$  von 318.

Zu Beginn einer Simulation, in der TICTAC zur Anwendung kommt, wird einer der Knoten im mittleren Quadrat zufällig ausgewählt, der als Zeitgeber fungiert und die lokale Zeitsynchronisation anstößt (dicker Punkt in Abbildung 4.1). Für die in dieser Arbeit präsentierten Ergebnisse wurde der Radius der Flutungszone passend zur Größe der Hochverkehrsregion auf vier Hops festgelegt. Auf diese Weise wird im Schnitt eine Überdeckung des Hochverkehrsbereiches durch die Flutungszone von etwa 75% erreicht. Das bedeutet, dass knapp drei Viertel aller Knoten, die zu Beginn einer Simulation mit der höheren Datenrate senden, nicht das Verfahren von Römer einsetzen, sondern ihre Uhren synchronisiert haben und deshalb auf die Umrechnung der Zeitangaben in den Datenpaketen verzichten können. Ein Nebeneffekt ist, dass auch etwa 50 Knoten außerhalb der Hochverkehrsregion synchronisiert werden.

Bei der lokalen Zeitsynchronisation mittels Fluten wird die Menge der Daten, die für die Synchronisation erforderlich ist, in Bytes mitgezählt.

Um die Länge eines versendeten Synchronisationspaketes in Bytes zu bestimmen, muss neben den eigentlichen Nutzdaten die Länge der von Medienzugriffs- und Sicherungsschicht hinzugefügten Headerdaten berücksichtigt werden. Zum Zeitpunkt dieser Arbeiten stand die ESB 430/2 Sensornetzplattform [51] der FU Berlin als Hardwareplattform zur Verfügung, so dass die Werte dieser Plattform auch für die Simulationen verwendet wurden. Für ein Synchronisationsdatenpaket ergibt sich somit eine Gesamtlänge  $l_{msg}$  von 20 Byte. Bei anderen Transceivern wird dieser Wert ähnlich ausfallen, kann aber prinzipiell sowohl nach oben als auch nach unten abweichen.

In Simulationsläufen, in denen das Verfahren von Römer zur Anwendung kommt, wird darüber hinaus der Overhead, der durch das Versenden des zusätzlichen Zeitstempels in jedem Paket entsteht, mitgezählt. Da ein Zeitstempel auf der ESB 430/2 Plattform eine Länge  $l_{TimeStamp}$  von 5 Byte hat, wurde auch dieser Wert für die Simulationen übernommen.

Folglich ergeben sich für alle drei Verfahren Daten, die widerspiegeln, wie sich die Datenmenge, die für Errichtung und Erhalt des konsistenten Zeitverständnisses versendet werden muss, über die Simulationszeit entwickelt. Um statistisch aussagekräftige Ergebnisse zu erhalten, wurden für jede Kombination von Parametern die Ergebnisse von 100 Simulationsläufen gemittelt.

### 4.3.2. Simulationsergebnisse

Legt man die resultierenden Daten der drei Verfahren für ein festgelegtes Set von Parametern in Form von Verlaufskurven in einem Diagramm übereinander, lassen sich Aussagen darüber ableiten, zu welchem Simulationszeitpunkt welches Verfahren kumulativ wie viel Overhead verursacht hat.

Die Frage, welches der Verfahren den geringsten Overhead erzeugt, ist stark von applikationsspezifischen Parametern des Szenarios abhängig:

1. die Dauer ( $t_{HT}$ ), während der in einem Teil des Netzes höherer Verkehr herrscht,
2. die dazugehörige Verkehrsrate bei hohem Verkehr ( $r_{HT}$ ) sowie
3. die Verkehrsrate bei geringem Verkehr ( $r_{LT}$ ).

Um diese Zusammenhänge zu untersuchen, wurden diese Parameter variiert und die Varianten zunächst simulativ untersucht. Dabei wurden alle Parameter gezielt einzeln und auch gemeinsam variiert, um die Auswirkungen studieren zu können.

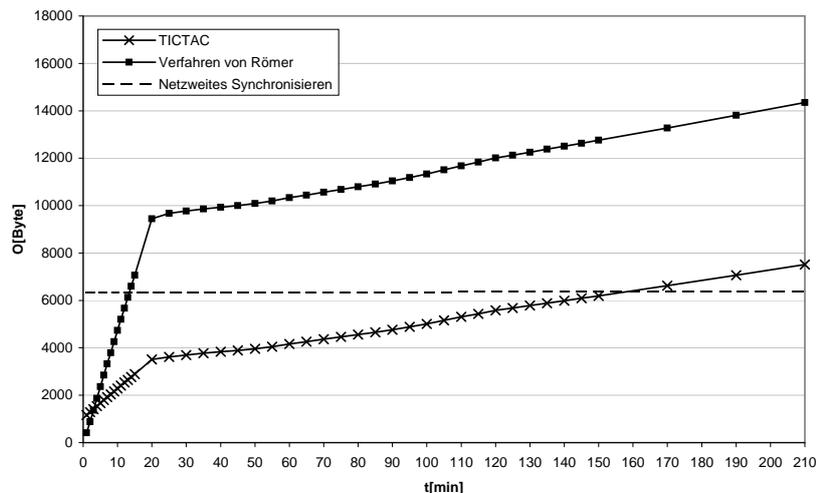


Abbildung 4.2.: Entwicklung des Overheads O in den Simulationen.

Abbildung 4.2 zeigt beispielhaft ein Diagramm, in dem die Entwicklung des Overheads über die Zeit veranschaulicht wird. Bei dem zugrunde liegenden Beispielszenario senden die Sensorknoten in der Hochverkehrszone für 20 Minuten ( $t_{HT} = 20$ ) im Schnitt alle 20 Sekunden ein Datenpaket ( $r_{HT} = 3$ ). Danach reduzieren sie den erzeugten Verkehr auf das geringere Maß von einem Paket pro 60 Minuten ( $r_{LT} = 1/60$ ), das auch für die übrigen Knoten zutrif. Die simulierte Zeitdauer betrug 210 Minuten.

Diese speziellen Parameter sind sicherlich nicht typisch für alle Arten von Sensornetzanwendung. Sie sind jedoch aufgrund des aus den drei Verfahren resultierenden Overheads besonders geeignet, prinzipielle Eigenschaften und Un-

terschiede deutlich zu machen, da sich bei dieser Parameterwahl sehr deutliche Unterschiede zeigen.

Es ist erkennbar, dass bei alleiniger Anwendung des Verfahrens von Römer während dieser ersten 20 Minuten der entstandene Overhead besonders stark steigt. Nachdem danach im gesamten Netz nur noch geringer Verkehr herrscht, steigt die Kurve merklich langsamer. Bereits nach 15 Minuten hat das Verfahren von Römer genauso viel Daten versendet wie bei einer netzweiten Zeitsynchronisation anfallen. Bei letzterer entsteht nach dem Synchronisieren aller Sensorknoten zu Beginn der Simulation bis zu einer eventuell später notwendigen Resynchronisation kein weiterer Overhead über die Simulationszeit. Die Kurve verläuft waagrecht, der Aufwand für die netzweite Synchronisation ist vom Nutzdatenaufkommen im Netzwerk unabhängig. Auch die Kurve von TICTAC verläuft wie erwartet: Nach geringem anfänglichen Aufwand (bei  $t = 0$ ) für die partielle Synchronisation steigt die Kurve merklich langsamer als die des Verfahrens von Römer. Sie ist insbesondere wesentlich weniger sensibel hinsichtlich der Verkehrsraten.

Es ist klar zu erkennen, dass sich über die Simulationszeit grundsätzlich 3 zeitliche Bereiche identifizieren lassen. Etwa bis zum Zeitpunkt von 3 Minuten hat das Verfahren von Römer am wenigsten Overhead verursacht. Danach lohnt sich der Einsatz von TICTAC, da der Overhead hier langsamer ansteigt: Im Bereich bis etwa 160 Minuten werden am wenigsten zusätzliche Daten versendet. Danach ergibt sich für die Zeitsynchronisation im gesamten Netzwerk die günstigste Bilanz, wenn nicht bereits vorher eine Resynchronisation nötig wurde.

Bei der Interpretation der Ergebnisse muss zusätzlich die Zeit in Betracht gezogen werden, nach der sowohl beim netzweiten Fluten als auch bei TICTAC resynchronisiert werden müsste. Diese Zeit ist nicht direkt Bestandteil der Simulationen, da sie von der Genauigkeit der Uhren, der für die Applikation notwendigen Genauigkeit der Synchronisation und anderen Faktoren abhängt.

Dieses Intervall muss nachträglich berücksichtigt werden: Müsste zum Beispiel nach 120 Minuten resynchronisiert werden, kann man in Abbildung 4.2 ablesen, dass zu diesem Zeitpunkt TICTAC am wenigsten Daten gesendet hat. Bei Resynchronisation nach 180 Minuten hingegen würde das netzweite Fluten die günstigste Bilanz aufweisen.

Um die notwendige Resynchronisation zu berücksichtigen, müsste man folglich das Diagramm zum betreffenden Zeitpunkt abschneiden, und die Kurven ab Zeitpunkt  $t = 0$  an ihren Verlauf wieder ansetzen. Daher wird das Verfahren, aus dem direkt vor der Resynchronisation am wenigsten Overhead resultiert, auch insgesamt über mehrere Resynchronisationen hinweg am wenigsten Verkehr erzeugen.

Zu diskutieren bleibt die Frage, wie sich ein Fehlalarm von TICTAC auswirken würde, das heißt wenn eine partielle Synchronisation eingeleitet würde, obwohl kein erhöhter Verkehr stattfindet. Das würde dazu führen, dass durch das Fluten

erneut Kommunikationsaufwand entsteht (und zwar im Umfang des Ordinatenabschnittes der entsprechenden Kurve). Folglich würde an der entsprechenden Stelle im Diagramm die Kurve sprunghaft um 960 Bytes ansteigen. Danach verlief sie etwas flacher, da in einem weiteren Teil des Netzes auf die Umrechnung der Zeitstempel verzichtet werden könnte.

In den Simulationen wurde auch untersucht, wie weit die lokale Flutwelle vom Initiator aus gesehen fortschreitet und welches Gebiet sie folglich abdeckt. Es wurde festgestellt, dass ab dem zweiten Hop die Ausbreitung nur noch etwa 60% der Kommunikationsreichweite beträgt. Die Ursache ist, dass die Knoten nicht immer in maximaler Entfernung des Knotens liegen, von dem sie das geflutete Paket empfangen.

Insgesamt ergibt sich aus der starken Abhängigkeit der Simulationsergebnisse von den verwendeten Parametern, dass sich keine allgemein gültigen Aussagen darüber treffen lassen, welches Verfahren den geringsten Overhead erzeugt und folglich am geeignetsten ist. Vielmehr müssen die Applikationsparameter für ein spezielles Szenario bei der Entscheidung über den Einsatz eines der Verfahren berücksichtigt werden. Um nicht für jedes Anwendungsszenario Simulationen durchführen zu müssen, bevor eine Entscheidung über das zu wählende Verfahren getroffen werden kann, wurde aus den durch die Simulationen gewonnenen Erkenntnissen ein mathematisches Modell entwickelt, das Aussagen über die Effizienz der drei Verfahren für beliebige Kombinationen der drei oben genannten Parameter erlaubt.

## 4.4. Modellbildung

Bei der Entwicklung des Modells wurde darauf geachtet, dass seine Berechnungskomplexität gering ist, damit es auch direkt auf Sensorknoten implementiert werden kann. Die Geräte können das Modell dann verwenden, um zur Laufzeit unter Berücksichtigung von Applikationsparametern und der Situation im Netz Entscheidungen über das zu wählende Verfahren zu treffen.

Aus dem Diagramm in Abbildung 4.2 ist zu erkennen, dass sich der Verlauf der Kurven als Gerade mit einem Knick zum Zeitpunkt  $t_{HT}$ , somit also mit zwei Steigungen  $\alpha$  und  $\beta$  sowie einem potentiellen Ordinatenabschnitt  $b$ , modellieren lässt (siehe Abbildung 4.3).

Ausgehend von dieser Beobachtung wurde für den Overhead  $O(t)$  zunächst eine allgemeine Geradengleichung mit  $\alpha$ ,  $\beta$ ,  $t_{HT}$  und  $b$  als Parametern aufgestellt:

$$O(t) = \begin{cases} \alpha t + b & t \leq t_{HT} \\ \beta(t - t_{HT} + 1) + \alpha t_{HT} + b & t > t_{HT} \end{cases} \quad (4.1)$$

Es gilt nun, die drei Gleichungsparameter  $\alpha$ ,  $\beta$  und  $b$  für alle drei Verfahren zu bestimmen. Setzt man dann diese Parameter, die im Weiteren jeweils die Indizes  $F$  für die Zeitsynchronisation mittels Fluten,  $R$  für das Verfahren von

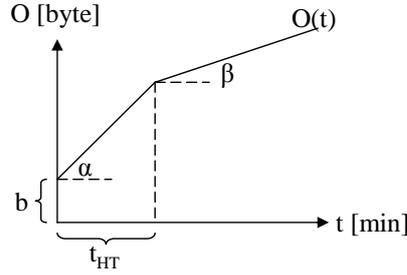


Abbildung 4.3.: Allgemeine Funktion zur Beschreibung des Overheads.

Römer und  $T$  für TICTAC erhalten, in Gleichung 4.1 ein, erhält man den entsprechenden Overhead über die Zeit.

#### 4.4.1. Modellbildung für Synchronisation mittels Fluten

Am einfachsten ist die Bestimmung der Gleichungsparameter für die netzwerkweite Zeitsynchronisation mittels Fluten. Da nach dem initialen Fluten keinerlei Mehraufwand mehr entsteht, gilt

$$\alpha_F = 0, \quad (4.2)$$

$$\beta_F = 0. \quad (4.3)$$

Da jeder Knoten während der Flutungsphase genau ein Paket versendet, ergibt sich

$$b_F = n l_{msg}. \quad (4.4)$$

#### 4.4.2. Modellbildung für das Verfahren von Römer

Für das Verfahren von Römer gilt  $b_R = 0$ , da kein initialer Aufwand anfällt. Weiterhin ergibt sich

$$\alpha_R = n p_{HT} r_{HT} l_{TimeStamp} + n (1 - p_{HT}) r_{LT} l_{TimeStamp}, \quad (4.5)$$

$$\beta_R = n r_{LT} l_{TimeStamp}. \quad (4.6)$$

Der erste Summand von  $\alpha_R$  beschreibt den Aufwand für den Netzbereich, in dem erhöhter Verkehr herrscht, und ergibt sich aus dem Produkt aus der Gesamtanzahl der Knoten  $n$ , dem Anteil der Knoten  $p_{HT}$  mit erhöhter Verkehrsrate  $r_{HT}$  sowie dem Zusatzaufwand pro Paket für den Zeitstempel der Länge  $l_{TimeStamp}$ . Der zweite Summand beschreibt den Aufwand im Netzbereich mit normalem Verkehr und besteht aus dem Produkt aus der Gesamtknotenanzahl

$n$ , dem prozentualen Anteil der Knoten mit normaler Verkehrsrate  $(1 - p_{HT})$ , der geringeren Verkehrsrate  $r_{LT}$  und der Größe des Zeitstempels. Die Aufwandsentwicklung in der Zeit nach Ablauf von  $t_{HT}$  wird von  $\beta_R$  beschrieben und setzt sich aus dem Produkt der Knotenanzahl, der Verkehrsrate  $r_{LT}$  und der Größe des Zeitstempels zusammen.

Der Gleichungsparameter  $\beta_R$ , der die Overheadentwicklung nach Ende des lokal erhöhten Kommunikationsaufkommens beschreibt, ergibt sich als Produkt der Gesamtknotenanzahl  $n$ , der geringeren Verkehrsrate  $r_{LT}$  sowie der Zeitstempelgröße.

### 4.4.3. Modellbildung für TICTAC

Am komplexesten ist das Modell für TICTAC, da hier die Situationen innerhalb der Flutungszone sowie im äußeren Netzbereich, in dem das Verfahren von Römer angewendet wird, unterschieden werden müssen. Dazu muss zunächst beschrieben werden, wie viele Knoten durch das partielle Fluten nach  $h$  Hops erreicht werden. Dabei ist eine Modellierung über die Fläche des Ausbreitungsgebiets zielführend. Es wird mittels konzentrischer Kreise abgeschätzt, wie groß die Fläche ist, die durch das Fluten nach  $h$  Hops abgedeckt wird. Kennt man die Anzahl der Knoten pro Fläche, kann man die Menge der erreichten Knoten errechnen. Die Gleichung

$$F(h) = \frac{\pi((1 + 0,6(h-1))r)^2}{\frac{A}{n}} \quad (4.7)$$

beschreibt diese Knotenzahl in Abhängigkeit von  $h$ . Die Front der Flutwelle bewegt sich bei jedem Hop nicht um einen gesamten Kommunikationsradius  $r$  voran, sondern um einen geringeren Wert. Daher wurde der Korrekturfaktor 0,6 eingefügt, der simulativ ermittelt wurde (vergleiche Abschnitt 4.3.2). Für den ersten Hop wird dieser Faktor allerdings nicht angewandt. TICTAC flutet die Synchronisationsnachrichten statisch über eine Distanz von 4 Hops, da sich die Distanz 4 simulativ als günstig relativ zur Größe der Hochverkehrsregion erwiesen hat. Ist die Größe dieser Region bekannt, ließen sich aus diesen Erfahrungen prinzipiell Heuristiken für einen geeigneten Hopcount ableiten. In den folgenden Formeln findet sich somit der Term  $F(4)$  für die von der Synchronisation erreichten Knoten. Die Zahl der Knoten, die im Rahmen des Synchronisierens entsprechende Pakete senden, lässt sich mittels  $F(3)$  abschätzen. Auf dieser Grundlage ergibt sich für TICTAC

$$\alpha_T = n p_{HT} (1 - p_{TICTAC}) r_{HT} l_{TimeStamp} + \quad (4.8)$$

$$[n(1 - p_{HT}) - (F(4) - n p_{HT} p_{TICTAC})] r_{LT} l_{TimeStamp},$$

$$\beta_T = (n - F(4)) r_{LT} l_{TimeStamp}, \quad (4.9)$$

$$b_T = F(3) l_{msg}. \quad (4.10)$$

Auch hier beschreibt der erste Summand von  $\alpha_T$  den Aufwand für den Hochverkehrsbereich. Zusätzlich zu  $\alpha_R$  (vergleiche Abschnitt 4.4.2) muss hier noch berücksichtigt werden, dass ein Anteil  $p_{TICTAC}$  der Knoten in der Hochverkehrszone von der initialen Flutwelle erreicht wird und deshalb das Verfahren von Römer nicht anwendet.

Der Minuend in der eckigen Klammer im zweiten Summanden beschreibt die Anzahl der Knoten außerhalb der Hochverkehrsregion, die von der Flutwelle erreicht werden. Er ergibt sich aus der Anzahl der insgesamt erreichten Knoten abzüglich derer, die in der Hochverkehrsregion liegen.  $\beta_T$  ergibt sich analog zum Verfahren von Römer mit der Ausnahme, dass von der Gesamtzahl der Knoten diejenigen abgezogen werden müssen, die durch das anfängliche partielle Fluten erreicht werden.

Abschließend bleibt der initiale Aufwand  $b_T$  zu beziffern. Da alle Knoten die Flutwelle weiterleiten, die bis zum dritten Hop erreicht werden, ergibt sich der Aufwand aus dem obigen Produkt.

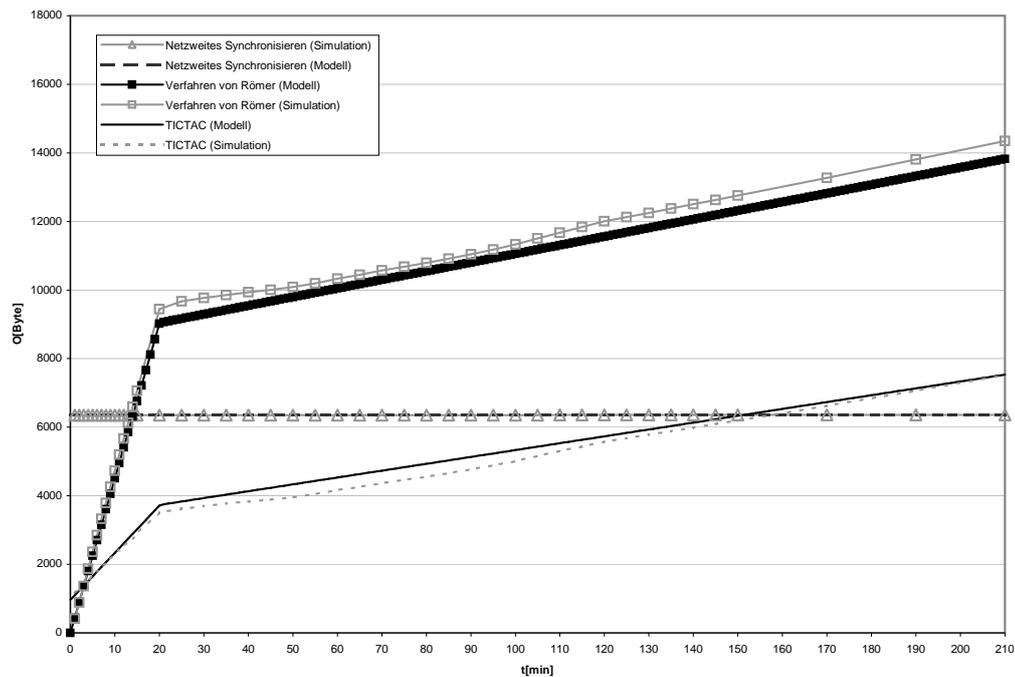


Abbildung 4.4.: Entwicklung des Overheads  $O$  gemäß dem mathematischen Modell.

Der Vergleich des mit dem Modell ermittelten Verlaufes des Kommunikationsaufkommens der drei Verfahren mit den Ergebnissen aus den Simulationen zeigt, dass sich Modell und Simulationsergebnisse sehr gut decken. Abbildung 4.4 zeigt die Daten des Modells (schwarze Kurven), zum Vergleich sind die Simulationsergebnisse aus Abbildung 4.2 grau dargestellt. Zum Simulationseende weichen die Daten des Modells weniger als 5% von den Daten der Simulation ab.

Anhand des Modells lassen sich nun auch diejenigen Parameterkombinationen

identifizieren, die die Nutzung eines bestimmten der drei Verfahren nahe legen. Erhöhter Verkehr innerhalb eines relativ begrenzten Netzbereiches führt dazu, dass sich TICTAC gewinnbringend einsetzen lässt. Je intensiver und lang andauernder dieser ist, desto größer wird der Vorteil gegenüber dem Verfahren von Römer. Geringer Verkehr im Rest des Netzes sowie häufig erforderliche Resynchronisation begünstigen TICTAC gegenüber der netzweiten Synchronisation. Diese Ergebnisse sind dabei weitestgehend unabhängig von der Größe des Netzes.

### 4.5. Zusammenfassung

In diesem Kapitel wurden die konzeptionellen Unterschiede von proaktiv und post-facto arbeitenden Verfahren zur Errichtung eines konsistenten Zeitverständnisses sowie der aus ihnen resultierende Kommunikationsoverhead diskutiert. Es wurde dargelegt, dass sich proaktive Verfahren eher bei starkem Kommunikationsaufkommen eignen, während Post-facto-Verfahren bei wenig Verkehr besonders effizient sind. Außerdem wurde mit TICTAC eine neues Verfahren vorgestellt, das die Vorteile beider Ansätze vereinen kann.

In Simulationen wurde der Overhead aller drei Varianten untersucht, und im Anschluss ein Modell entwickelt, das die Abschätzung des Overheads anhand von applikationsabhängigen Parametern ermöglicht.

Da die Berechnungskomplexität des Modells gering ist, kann es sogar direkt auf den Sensorknoten implementiert werden. Die Geräte könnten das Modell dann verwenden, um zur Laufzeit unter Berücksichtigung von Applikationsparametern und der Situation im Netz Entscheidungen über das zu wählende Verfahren zu treffen.

Nachdem sich dieses Kapitel mit dem Kontextaspekt Zeit beschäftigt hat, wenden sich die beiden nun folgenden Kapitel der Distanz und dem Ort als Kontextelementen zu.



## 5. Nachbarschaftsbasierte Distanzschätzung

Neben der Zeit ist der Ort die wichtigste Kontextkomponente in Sensornetzwerken. Wie bereits in Kapitel 3 erläutert, lassen sich Verfahren zur Lokationsbestimmung nach dem Prinzip der Multilateration im Allgemeinen in die Phasen Distanzschätzung, Positionsberechnung und Verfeinerung gliedern.

Das Abschätzen von Distanzen steht im Mittelpunkt dieses Kapitels. Neben dem Einsatz in den verschiedensten Lokationsverfahren ist die Kenntnis von Distanzen zwischen Geräten jedoch auch anderweitig verwendbar. So können Informationen über Entfernungen zu anderen Sensorknoten, etwa beim Verfolgen von Objekten (*object tracking*), hilfreich sein. Auch die Fusion von auf einem Sensorknoten gemessenen Werten mit denen von anderen Geräten kann auf Basis der räumlichen Entfernung geschehen, um so der Lokalität der gemessenen Eigenschaften Rechnung zu tragen.

Dieses Kapitel gliedert sich in fünf Unterabschnitte. Zunächst wird auf verwandte Arbeiten eingegangen, um daran anschließend ein neues Verfahren zur Distanzschätzung vorzustellen. Es basiert auf der Tatsache, dass Sensorknoten, die nahe beieinander liegen, mehr Nachbarn miteinander gemein haben als solche, die weiter voneinander entfernt sind. Nachdem die mathematischen Grundlagen erläutert und die Funktionsweise beschrieben wurde, werden die Ergebnisse der simulativen Evaluation der Distanzschätzung zwischen benachbarten Geräten dargestellt. Dann wird aufgezeigt, wie das Grundprinzip zu einer Multihop-Distanzschätzung erweitert werden kann, und erneut Simulationsergebnisse vorgestellt.

Teile der hier vorgestellten Ergebnisse sind in gemeinsamer Arbeit mit Dennis Pfisterer, Horst Hellbrück, Christian Werner, Alexander Kröller, Stefan Fischer und Sándor Fekete entstanden. Nachdem das Grundprinzip des Distanzschätzverfahrens in [31, 32, 66] erstmals vorgestellt wurde, folgten Veröffentlichungen mit Verallgemeinerungen [26] und Erweiterungen [33].

### 5.1. Verwandte Arbeiten

Dieser Abschnitt gibt einen Überblick über aus der Literatur bekannte Verfahren zur Distanzschätzung.

Zunächst werden Anforderungen an Distanzschätzungsverfahren aufgestellt, um die vorgestellten Verfahren geeignet diskutieren zu können. Danach wird auf

Verfahren zur Schätzung von Distanzen zwischen benachbarten Geräten eingegangen. Dabei werden Funktionsweise und Eigenschaften der unterschiedlichen Ansätze beschrieben sowie Vorzüge und problematische Aspekte hervorgehoben.

Abschließend wird erläutert, wie sich diese Schätzungen auf größere Reichweiten ausdehnen lassen, indem Schätzungen über mehrere Hops hinweg miteinander kombiniert werden.

### 5.1.1. Anforderungen an Distanzschätzverfahren

In der Vergangenheit wurden verschiedenste Techniken entwickelt, die der Ermittlung von Distanzen zwischen Geräten in einem verteilten System dienen. Ihnen gemein ist, dass sie nicht direkt Distanzen messen. Vielmehr misst man eine physikalische Größe, die dann mittels eines Modells in eine Distanz umgerechnet wird. Diese Tatsache ist auch der Grund dafür, dass häufig nicht von Distanzmessung, sondern von Distanzschätzung oder Distanzermittlung gesprochen wird. Die Genauigkeit der ermittelten Distanzen hängt maßgeblich von den Eigenschaften sowie der Eignung des verwendeten Modells ab.

Bei der Bewertung von Distanzschätzverfahren spielt neben der Genauigkeit, also der Korrektheit der ermittelten Distanzen, die technische Eignung für den Einsatz in realen Sensornetzen eine wesentliche Rolle. Daraus resultieren die folgenden Anforderungen, die ein praxistaugliches Verfahren erfüllen sollte:

- **Verzicht auf Spezialhardware:** Einige Verfahren erfordern den Einsatz eigener Hardwarekomponenten, speziell für die Distanzschätzung. Jeder Sensorknoten muss mit den entsprechenden Funktionseinheiten ausgestattet sein. Ein Beispiel hierfür sind Ultraschallsensoren, wie sie auch im Automobil als Einparkhilfen zum Einsatz kommen. Derartige zusätzliche Hardware wirkt sich negativ auf die Baugröße, die Entwicklungs- und Herstellungskosten sowie den Energieverbrauch der Sensorknoten aus.
- **Keine speziellen Anforderungen an die zum Einsatz kommende Hardware:** Einige Verfahren erfordern zwar keine dedizierte Hardware für die Distanzschätzung, stellen aber spezielle Anforderungen an die ansonsten auf den Sensorknoten verbauten Komponenten. Beispiele hierfür sind eine hohe Rechenleistung oder spezielle Eigenschaften des Funkchips. Folgen sind auch hier negative Auswirkungen, insbesondere auf den Preis und den Energieverbrauch der Knoten.
- **Geringer Energieverbrauch:** Um einem Sensornetz im realen Einsatz eine lange Lebensdauer zu verleihen, ist es zwingend erforderlich, sparsam mit der zur Verfügung stehenden Energie umzugehen. Dieses gilt natürlich auch für Verfahren zur Distanzermittlung. Es ist somit geboten, den Nachrichtenaustausch und die benötigte Rechenzeit soweit wie möglich zu begrenzen.
- **Geringe Dauer der Distanzermittlung:** Um ein Verfahren zur Distanzer-

mittlung auch in mobilen Sensornetzen einsetzen zu können ist es erforderlich, dass die Distanzermittlung nur einen kurzen Zeitraum in Anspruch nimmt. Ist dies nicht der Fall, kann sich die Situation im Sensornetz bei Abschluss der Schätzung bereits soweit verändert haben, dass die Ergebnisse nicht mehr mit der tatsächlichen Situation übereinstimmen. Eine geringe Dauer reduziert darüber hinaus den Energieverbrauch eines Verfahrens.

- **Verzicht auf manuelle Kalibrierung:** Da Sensornetze für den Einsatz ohne manuelle Konfiguration vorgesehen sind, sollten auch die Verfahren zur Distanzermittlung auf Konfiguration und Kalibrierung durch den Nutzer verzichten. Insbesondere eine geräteindividuelle Kalibrierung ist aufgrund der avisierten Netze mit hunderten und mehr Geräten nicht akzeptabel.
- **Reichweite:** Die Verfahren sollten in der Lage sein, möglichst große Distanzen korrekt zu schätzen. So können längere Messabschnitte durch weniger Einzelmessungen abgedeckt werden. Auf diese Weise kann die Multihop-Distanzschätzung seltener zum Einsatz kommen; durch sie bedingte Fehler bekommen geringeren Einfluss.
- **Robustheit gegen Variationen der Hardware und der Umwelt:** Da Sensornetze zum Teil unter a priori unbekanntem Umgebungsbedingungen zum Einsatz kommen, ist es wichtig, dass die Distanzermittlung möglichst robust gegen externe Einflüsse ist. Auch Toleranzen der zum Einsatz kommenden Hardware sollten die Ergebnisse möglichst wenig negativ beeinflussen. Ebenfalls wichtig ist, dass ein Verfahren möglichst immer zu einer Distanzschätzung gelangt.

### 5.1.2. Verfahren zur Distanzermittlung

Die verschiedenen Verfahren zur Distanzermittlung lassen sich anhand der ihnen zugrunde liegenden Funktionsweise in Gruppen einteilen. Jede einzelne Technik basiert auf unterschiedlichen Hardwarekomponenten, Messtechniken und Modellen, um autonom Distanzen zwischen Geräten zu ermitteln. Daraus resultieren unterschiedliche Eigenschaften der Verfahren. Fünf verschiedene Prinzipien sind heute bekannt:

- Signallaufzeit
- differenzielle Signallaufzeit
- Einstrahldauer
- Signalabschwächung
- Konnektivität
- Interferometrie

Das vielleicht bekannteste Verfahren ist die Messung von Distanzen auf Basis von *Signallaufzeiten*. Es kommt beispielsweise beim Global Positioning System (GPS) [87] zum Einsatz. Die Grundidee der Distanzermittlung ist hier, dass der Sender zu einem dem Empfänger bekannten oder in der Nachricht kodierten Zeitpunkt ein Signal aussendet. Der Empfänger ermittelt den Zeitpunkt, zu dem das Signal ihn erreicht, und kann aus der Differenz der beiden Zeitpunkte die Signallaufzeit errechnen. Ist die Ausbreitungsgeschwindigkeit des Signals bekannt, kann so die Distanz zwischen Sender und Empfänger ermittelt werden.

Problematisch an diesem Ansatz ist, dass der Zeitpunkt des Aussendens dem Empfänger bekannt sein muss. Dieses erfordert eine enge Zeitsynchronisation (vergleiche Abschnitt 3.1.1) zwischen beiden Geräten: Fehler in der Synchronisation führen direkt zu Fehlern in der Laufzeitberechnung. Vorteilhaft wirkt sich die kurze Verfahrensdauer aus.

Weiterhin hat das Verfahren den Vorteil, dass keine spezielle Hardware benötigt wird, wenn Funksignale zum Einsatz kommen. Aufgrund der hohen Ausbreitungsgeschwindigkeit ist allerdings eine extrem genaue Synchronisation erforderlich. So bedingt eine gewünschte Distanzauflösung von einem Meter eine erforderliche Synchronisationsgenauigkeit im Bereich von Picosekunden. Dies führt dazu, dass das Verfahren in Sensornetzen praktisch kaum verwendbar ist. Ein gravierender Nachteil des Verfahrens ist außerdem die starke Anfälligkeit gegen Reflexionen. Gelangt das Signal nicht auf direktem Weg vom Sender zum Empfänger, so resultieren aus dem längeren Signalweg unmittelbar zu große Distanzschätzungen. Am ehesten scheint sich Schall zur Laufzeitmessung zu eignen, da er sich langsam ausbreitet. Die Autoren von [74] berichten mittlere Genauigkeiten von 9cm. Allerdings werden hier separate Sender und Empfänger benötigt.

Eine neue Methode zur Distanzermittlung auf Basis von Signallaufzeiten realisiert ein nanoLOC TRX genannter Funkchip von Nanotron Technologies. Er arbeitet mit der Chirp-Modulationstechnik Chirp Spread Spectrum (CSS) nach dem IEEE-Standard 802.15.4a [90]. Das verwendete Verfahren trägt den Namen *Symmetrical Double-Sided Two Way Ranging (SDS-TWR)*. Es gestattet eine funkbasierte Abstandsmessung anhand der Signallaufzeiten ohne zusätzliche Zeitsynchronisation, da der Empfänger eines Datenpaketes zur Distanzmessung direkt ein Antwortpaket an den ursprünglichen Sender verschickt. Letzterer misst die Zeitdauer zwischen dem Versand seines Paketes und dem Empfang der Antwort. Diese Zeit setzt sich aus der doppelten Signallaufzeit und der Verarbeitungszeit zusammen. Da die gesamte Verarbeitung der Pakete in Hardware auf einem einzigen Chip realisiert ist, kann Verarbeitungszeit als bekannt angenommen werden, so dass sich die Signallaufzeit bestimmen lässt.

Die Methode hat den Vorteil, dass der Transceiver auch zur sonstigen Datenkommunikation geeignet ist und folglich keine zusätzliche Hardware zum Einsatz kommen muss. Untersuchung haben gezeigt, dass das System Genauigkeiten von etwa 1 m erreicht, wenn Sichtkontakt zwischen Sender und Empfänger herrscht [29]. Ist dieses nicht der Fall, so neigt diese Technik zu deutlichen

Überschätzungen der Distanzen. Dieses ist dadurch zu erklären, dass die Signalübertragung nicht auf direktem Wege, sondern mittels Reflektionen stattfindet. Somit verlängert sich der Signalwert, der vom System gemessen wird, und der Messwert repräsentiert nicht mehr die Distanz zwischen Sender und Empfänger. Auf diese Weise werden Ergebnisse ermittelt, die durchaus dem Doppelten der Distanz entsprechen können [29].

Ebenfalls ohne Synchronisation kommen Verfahren aus, die auf dem Prinzip der Messung der *differenziellen Signallaufzeit* beruhen. Beispiele hierfür sind Calamari [192, 193], Cricket [145], AHLoS [159] und andere [75, 78, 155, 194, 195]. Hier werden zwei Signale ausgesandt, die sich mit unterschiedlichen Geschwindigkeiten ausbreiten. Kennt man diese Ausbreitungsgeschwindigkeiten, kann man aus der Zeitdifferenz zwischen der Ankunft beider Signale die Distanz zwischen Sender und Empfänger errechnen.

Die meisten der Systeme setzen hörbaren oder Ultraschall einerseits und Funksignale andererseits ein. Bloße Messungen der Empfangsintervalle liefern dabei zunächst Fehler von etwa 74% [192, 193]. Techniken der Signalverarbeitung wie Rauschunterdrückung, digitale Filterung, Maximalwertbestimmung und Kalibrierung ermöglichen jedoch erheblich höhere Genauigkeiten. Nachteilig wirkt sich aus, dass aufgrund der hohen Dämpfung des Schalls nur kurze Distanzen geschätzt werden können. Während einige Autoren mittlere Fehler von 10% ermittelten [192, 193], berichten andere von Fehlern von etwa 1% bei einer maximalen Distanz von 9 m [155]. Die Autoren von [195] erzielten Reichweiten von 12 m bei einem Fehler von 0,5%.

Diese Systeme liefern recht genaue Ergebnisse, doch dieser Vorteil ist teuer erkauft. Da sie auf Basis differenzieller Laufzeitmessungen arbeiten, benötigen sie inhärent einen zusätzlichen Sender und Empfänger. Dies wirkt sich negativ auf die Entwicklungs- und Herstellungskosten, die Baugröße und den Energieverbrauch der Sensorknoten aus. Darüber hinaus bringt der Einsatz von Schall zusätzliche Nachteile mit sich. So führen lokale Temperaturdifferenzen zu variierenden Ausbreitungsgeschwindigkeiten. Hindernisse in der direkten Sichtlinie können darüber hinaus dazu führen, dass die Laufzeitmessung verfälschte oder gar keine Ergebnisse liefert. Auch sind Schallabstrahlung und -wahrnehmung eines jeden Emitters beziehungsweise Detektors auf einen bestimmten Winkel begrenzt, so dass jeweils mehrere Sender und Empfänger zum Einsatz kommen müssen um omnidirektionale Messungen zu ermöglichen. Weiterhin benötigen Sender und Empfänger überproportional viel Energie. Der Einsatz von Schall limitiert auch die maximale Reichweite des Messsystems. Sie liegt typischerweise bei 3–15 m [159], das heißt einem Bruchteil der Reichweite der Funkschnittstelle. Neben der Kalibrierung ist die geeignete Verarbeitung der empfangenen Signale problematisch. Dauert zum Beispiel die Interruptverarbeitung unterschiedlich lange, so werden die Ergebnisse verfälscht. Vorteilhaft ist, dass die Distanzschätzung nur kurze Zeit in Anspruch nimmt.

Eine Sonderstellung nimmt die Distanzschätzung auf Basis der *Einstrahldauer* ein. Sie richtet sich an eine spezielle Unterkategorie der Sensornetze, die

das Laserlicht einer zentralen Basisstation zur Kommunikation nutzt [93, 190]. Bei diesen Netzen leuchtet ein Laserstrahl die Sensorknoten nacheinander an. Diese können einerseits den Laser detektieren und ihn andererseits mittels dreier senkrecht zueinander stehender Spiegel reflektieren. Da einer der drei Spiegel beweglich ist, können die Sensorknoten das Laserlicht in zeitlicher Abfolge zurückwerfen oder nicht, und so ein Signal auf das zurückgeworfene Licht modulieren.

Das in [150] vorgestellte Verfahren nutzt die Fähigkeiten der Sensorknoten zur Laserdetektion aus. Es basiert auf einer Art rotierendem Leuchtturm, der einen horizontal perfekt parallelen und vertikal sehr breiten Laserstrahl aussendet. Ein lasersensibler Sensorknoten nimmt diesen rotierenden Laser als ein kurzes Aufblitzen wahr, wenn er von dem vorbeiwandernden Strahl angeleuchtet wird. Die Einstrahldauer des Lasers hängt dabei von der Distanz des Sensorknotens zum Leuchtturm ab. Somit kann der Knoten aus der Dauer der Beleuchtung und der a priori bekannten Drehungsgeschwindigkeit auf die Distanz zum Leuchtturm schließen.

Der in der Publikation beschriebene Prototyp erreichte eine Genauigkeit von etwa 2% bei einer maximalen Reichweite von 14 m, die sich allerdings mittels einer ausgereifteren Mechanik auf 120 m bis 140 m erweitern ließe. Der Hauptnachteil des Verfahrens ist, dass zur Distanzschätzung eine ununterbrochene Sichtverbindung zwischen Sensorknoten und Basisstation erforderlich ist. Andernfalls kommt keinerlei Schätzung zu Stande. Vorteile des Verfahrens sind, dass in Sensornetzen, die ohnehin auf Basis von Laserlicht kommunizieren, keine zusätzliche Hardware erforderlich ist. Die Verarbeitung der entsprechenden Signale sowie die Signalverarbeitung sind einfach und verursachen nur geringen Rechenaufwand. Die heute üblichen Sensorknoten mit Funkkommunikation hingegen benötigen zusätzlich eine spezielle Laserdetektionseinheit, um dieses Verfahren anwenden zu können. Das wirkt sich negativ auf den Energieverbrauch, den Preis und die Baugröße aus.

### 5.1.3. Distanzschätzungen auf Basis der Funkschnittstelle

Um den Nachteilen der Systeme, die auf der Messung von (differenzieller) Laufzeitmessung oder Einstrahldauer basieren, aus dem Weg zu gehen, wurden andere Verfahren zur Distanzschätzung entwickelt. Dabei stand die Beschränkung auf die Nutzung der Funkschnittstelle, mit der die Geräte ohnehin ausgerüstet sind, im Mittelpunkt.

Diese Ansätze lassen sich nochmals in zwei Kategorien unterteilen:

- Qualitative Verfahren (range-free): Die hier eingeordneten Verfahren prüfen lediglich, ob der Empfang der Signale eines bestimmten Senders möglich ist. Falls ja, leitet man daraus ab, dass die Distanz unterhalb einer a priori bekannten maximalen Sendereichweite liegt.
- Quantitative Verfahren (range-based): Die Verfahren dieser Gruppen ver-

suchen, aus an der Funkschnittstelle messbaren Signaleigenschaften konkrete Entfernungen zwischen Sender und Empfänger abzuleiten.

Die frühesten Ansätze zur Distanzschätzung im Bereich des Ubiquitous Computing gehören zur Kategorie der *qualitativen Verfahren*. Sie beschränken sich auf die Detektion räumlicher Nähe. Einige Geräte mit bekannten Positionen senden Beacons aus. Die Empfänger dieser Signale – entweder Teile einer fest installierten Infrastruktur oder aber mobile Geräte – leiten daraus räumliche Nähe zum Sender ab. Mit einer großen Anzahl dicht angeordneter Sender lässt sich auf diese Weise eine feine Auflösung erzielen. Die wesentlichen Nachteile sind der Bedarf einer festen Infrastruktur und der daraus resultierende Installationsaufwand.

Erste Beispiele für diese Art der Distanzschätzung sind das „Active Badge System“ [188] und das „Hybrid Indoor Navigation System“ [34]. Beide nutzen infrarotes Licht, um Beacons mit einer eindeutigen Identifikation auszusenden.

Um diese Technik überhaupt in Verbindung mit Sensornetzwerken einsetzen zu können, wurden verschiedene Anstrengungen unternommen, qualitative Verfahren so zu gestalten, dass sie ohne feste Infrastruktur auskommen. Dazu wurde meist angenommen, dass es ausgezeichnete Knoten mit bekannten Positionen – die Anker – gibt, zu denen Abstände bestimmt werden. Diese senden mittels der vorhandenen Funkschnittstelle spezielle Funknachrichten aus, deren Empfänger auf die Nähe der Anker schließen können [22, 178]. Geblieben ist der Nachteil, dass Distanzen unterhalb des Kommunikationsradius nicht differenziert werden können. Daher sind die resultierenden Abstandsschätzungen eher grobgranular, unterhalb des Kommunikationsradius beschränken sie sich auf die Erkennung von Nähe. Verbesserungen der Granularität werden möglich, indem man derartig viele Anker einsetzt, dass jeder Knoten in Hörweite von mehreren Ankern ist [22, 178]. In [56] wird vorgeschlagen, Konnektivitätsbeziehungen zwischen Geräten als Restriktionen räumlicher Distanz aufzufassen. Aus diesen sowie den Ankerpositionen werden dann im Rahmen einer netzwerkweiten konvexen Optimierung die Positionen aller Geräte bestimmt. Andere Verfahren beschränken sich auf das Zählen von Hop-Abständen und werden daher im sich anschließenden Abschnitt über Multihop-Distanzschätzung besprochen.

Um differenziertere und feingranularere Aussagen über Distanzen treffen zu können, wurden unterschiedliche *quantitative Verfahren* entwickelt. Praktisch alle nutzen zur Abstandsschätzung die Abschwächung der Funksignale, die Sensorknoten im Rahmen ihrer Kommunikation austauschen. Die meisten Funkinterfaces stellen zu diesem Zweck einen numerischen Wert zur Verfügung, der die Signalstärke der empfangenen Funkdaten und somit indirekt die Verbindungsqualität widerspiegelt. Dieser Wert wird daher im Allgemeinen „received signal strength indicator“ (RSSI) oder „link quality indicator“ (LQI) genannt. Im Folgenden wird der erste der beiden Begriffe stellvertretend für alle derartigen Verfahren verwendet.

Systeme, die Distanzschätzungen auf Basis des RSSI-Wertes durchführen [22, 36, 157], liefern für kurze Distanzen relativ genaue Werte, wenn umfangreiche Nach-

verarbeitung der einzelnen Werte erfolgt. Für größere Distanzen sind die Ergebnisse jedoch ungenau [113]. Bei Distanzen von weniger als 20 m liefern einfache Funkchips Fehler zwischen 50% und 100% ihrer Kommunikationsreichweite, während aufwändigere Modelle den Fehler hier auf etwa 10% begrenzen können, letzteres jedoch nur nach aufwändiger in-situ Kalibrierung [195]. Negativ auf die Genauigkeit wirken sich Reflektion, Mehrwegeausbreitung und Abschattung durch Hindernisse aus. Diese Effekte erschweren die Entwicklung eines konsistenten Modells zur Umsetzung der gemessenen Werte in Distanzen [159]. Die Folge ist, dass Verfahren, die ausschließlich auf Signalabschwächung beruhen, lediglich mittelmäßige Genauigkeiten erreichen [60].

Den Nachteilen mittelmäßiger Genauigkeit und der erforderlichen Kalibrierung stehen verschiedene Vorteile gegenüber. So wird keine zusätzliche Hardware benötigt, da die bestehende Funkschnittstelle genutzt wird. Das Verfahren arbeitet schnell und benötigt nur wenig Energie, wenn die ohnehin anfallende Kommunikation zur Messung der RSSI-Werte herangezogen wird. Grundsätzlich können Distanzen bis zum Kommunikationsradius geschätzt werden; mit zunehmender Distanz nimmt die Genauigkeit jedoch ab.

Etwas anders arbeitet das in [113] vorgestellte, *Radiointerferometrie* genannte Verfahren, das sehr gute Genauigkeiten liefert. Es basiert auf der Idee, dass verschiedene Knoten, die Sinussignale leicht unterschiedlicher Frequenz abstrahlen, bei einem Empfänger in Abhängigkeit von der Distanz in verschiedenen Intervallen an- und abschwellige Überlagerungsergebnisse erzeugen. Kennt man mehrere solcher Intervalle, kann man ohne hochgenaue Zeitsynchronisation Rückschlüsse auf die Distanzen ziehen. Maroti et. al berichten von einer mittleren Abweichung von 3 cm und einer Reichweite von 160 m. Dabei betrug der maximal festgestellte Fehler 6 cm, was etwa 0,04% entspricht [113].

Dieser hervorragenden Genauigkeit stehen verschiedene Nachteile gegenüber. Obwohl die ohnehin vorhandene Funkschnittstelle genutzt werden kann, kann das Verfahren nur mit speziellen Funkchips zur Anwendung kommen, die die Emission von Sinuswellen dicht beieinander liegender Frequenzen unterstützen. Diese sind zumindest zurzeit eher die Ausnahme am Markt. Die hohe algorithmische Komplexität stellt darüber hinaus hohe Anforderungen an die Rechenleistung der Sensorknoten. Des Weiteren sind die Messungen langwierig, wodurch sich das Verfahren nicht für mobile Szenarien eignet. Während der Messung wird der Funkkanal belegt, was sich negativ auf den sonstigen Betrieb des Sensornetzes auswirkt und darüber hinaus viel Energie verbraucht. Wie bei den meisten Verfahren, die auf Funkmessungen zu einzelnen Knoten beruhen, ist eine relativ große Anfälligkeit gegen Einflüsse von Reflektion, Mehrwegeausbreitung und Abschattung durch Hindernisse zu erwarten. Diese wurde jedoch in [113] nicht untersucht.

#### 5.1.4. Multihop-Distanzschätzungen

Allen bisher betrachteten Verfahren ist gemein, dass sich hiermit lediglich Abstände zwischen Geräten schätzen lassen, die direkt miteinander kommunizieren können. Bei etlichen Sensornetzanwendungen ist es jedoch wünschenswert, dass sich auch Abstände zwischen Geräten bestimmen lassen, deren Entfernung von einander größer ist als die Kommunikationsreichweite. Auch hierzu wurden bereits Arbeiten durchgeführt.

Der einfachste Ansatz besteht darin, die Entfernungen, die zwischen zwei direkt benachbarten Knoten geschätzt werden, über die einzelnen Hops entlang eines Multihop-Pfades zu addieren: Damit alle Geräte die Distanz zu einem Knoten ermitteln können, sendet dieser eine Flutwelle aus. Jedes Gerät addiert beim Weiterleiten der Nachricht seine Distanz zum Vorgänger auf die in der Nachricht enthaltene Distanz auf. Lernt ein Gerät auf diese Weise einen Pfad mit einer kürzeren Distanz zum ursprünglichen Absender kennen, leitet es die Nachricht weiter, sonst verwirft es sie. Ein entsprechendes Verfahren wird in [160] beschrieben. Im Folgenden wird dieser Ansatz mit *Sum-Dist* bezeichnet – die Autoren von [102] verwenden diese Bezeichnung ebenfalls.

Ein Nachteil von *Sum-Dist* besteht darin, dass sich die Schätzfehler über mehrere Hops akkumulieren können: Wird bei einem Verfahren stets über- oder unterschätzt, zum Beispiel weil Umwelteinflüsse die Messungen zeitweilig beeinträchtigen, summieren sich die Schätzfehler. Weiterhin schließen die aufsummierten Segmente meist einen Winkel ein, so dass die resultierende Summe der Einzeldistanzen selbst bei perfekten Distanzschätzungen die Gesamtdistanz überschreitet. Dieser Aspekt wird detailliert in Abschnitt 5.4.1 diskutiert.

Niculescu und Nath greifen diese Aspekte auf und schlagen in [127] ein Verfahren namens *Distance-Vector-Hop (DV-Hop)* vor. Bei diesem Verfahren wird lediglich die minimale Anzahl der Weiterleitungen (statt der geschätzten Abstände) durch das Netzwerk propagiert. Kennt man die mittlere Hoplänge, kann man diese mit der Anzahl der benötigten Hops multiplizieren und erhält so eine Abstandsschätzung. Zur Ermittlung der mittleren Hoplänge ist ein separater Kalibrierungsschritt erforderlich, bei dem man davon ausgeht, dass im Netzwerk mehrere Geräte vorhanden sind, die ihren Abstand zueinander kennen. Diese können nach Empfang der ersten Flutwelle die mittlere Hoplänge berechnen und im Netz propagieren. Auf diese Weise kann auf quantitative Distanzmessung zwischen benachbarten Geräten verzichtet werden.

Sowohl Sum-Dist wie auch DV-Hop berücksichtigen keine geometrischen Abhängigkeiten, das heißt die geschätzten Distanzen werden nicht dahingehend überprüft, ob sie in der 2D-Ebene überhaupt geometrisch möglich sind. Diesen Aspekt greift das Verfahren *Euclidean* [127] von Niculescu und Nath auf. Hier werden geometrische Beziehungen ausgenutzt, um diejenigen Singlehop-Distanzen zu Multihop-Distanzschätzungen zusammenzuführen, die in der 2D-Ebene in plausiblen Ergebnissen resultieren. Dieses Vorgehen hat den Nachteil, dass das Verfahren sensibel auf Fehler in den Distanzschätzungen zwischen be-

nachbarten Geräten reagiert. Solche Ungenauigkeiten führen die geometrischen Grundlagen des Verfahrens ad absurdum, und resultieren in extremen Fehlern oder gar darin, dass Knoten nicht zu einer Distanzschätzung gelangen. Bereits geringe Fehler in den Singlehop-Distanzschätzungen können somit zum kompletten Versagen des Euclidean-Verfahrens führen [102].

Für alle hier vorgestellten Verfahren gilt, dass sie meist als Bestandteil von Lokationsverfahren, die auf Multilateration basieren, in der Literatur zu finden sind. Somit finden sich keine separaten Angaben über die Genauigkeit der Multihop-Distanzschätzungen.

### 5.1.5. Defizite bestehender Verfahren

Zusammenfassend ist festzustellen, dass es weiterhin an umfassend geeigneten Ansätzen zur Distanzschätzung mangelt, da keines der vorgestellten Verfahren alle Anforderungen erfüllt.

Das DV-Hop-Verfahren hat zwei wesentliche Defizite: Es bedingt einerseits durch den Kalibrierungsschritt zusätzlichen Nachrichtenaustausch. Andererseits kann es nur Distanzwerte ermitteln, die Vielfache der mittleren Hoplänge im Netz sind, da es nicht auf Singlehop-Distanzschätzungen zurückgreift. Daher sind die von DV-Hop gelieferten Distanzwerte inhärent mit einem erheblichen Fehler behaftet.

Sum-Dist und Euclidian umgehen dieses Problem, indem sie auf Singlehop-Distanzschätzungen zurückgreifen, die dann zu Multihop-Schätzungen zusammengeführt werden. Euclidian berücksichtigt zwar geometrische Abhängigkeiten, ist jedoch sehr anfällig gegenüber Fehlern der als Eingangsdaten dienenden Singlehop-Distanzschätzungen. Sum-Dist ist diesbezüglich deutlich robuster, letztendlich hängt jedoch auch hier die Genauigkeit der Ergebnisse von den Singlehop-Distanzschätzungen ab.

Neben der Genauigkeit ergeben sich auch viele der übrigen Eigenschaften von Euclidian und Sum-Dist maßgeblich aus den in den Abschnitten 5.1.2 und 5.1.3 vorgestellten Singlehop-Verfahren. Problematisch ist, dass keines dieser Verfahren alle in Abschnitt 5.1.1 vorgestellten Anforderungen erfüllt.

Die Verfahren auf Basis von Signallaufzeiten erfordern für die Anwendung in Sensornetzen unrealistische Synchronisationsgenauigkeiten oder lassen sich nur mit spezialisierten Transceivern umsetzen. Die Verwendung von differentieller Laufzeit und Einstrahldauer bedingt jeweils den Einsatz zusätzlicher Spezialhardware, die Kosten, Baugröße und Energieverbrauch der Sensorknoten steigert.

Besser geeignet sind Verfahren, die ausschließlich auf die Funkschnittstelle zurückgreifen, da so auf zusätzliche Hardware verzichtet werden kann. Die Distanzschätzung auf Basis der Signalabschwächung arbeitet schnell und besitzt eine geringe Berechnungskomplexität, ist aber anfällig gegenüber Reflektionen und Mehrwegeausbreitung. Das führt dazu, dass das Verfahren ohne Kalibrie-

nung nur sehr ungenaue Ergebnisse liefert. Entsprechende in-situ Kalibrierungen sind jedoch sehr aufwendig und mit der für Sensornetze gewünschten spontanen Einsatzbereitschaft unvereinbar. Die Radiointerferometrie hingegen arbeitet auch ohne Kalibrierung sehr genau, hat aber ebenfalls gravierende Nachteile: Das Verfahren dauert lange und die notwendigen Berechnungen sind komplex. Darüber hinaus kann die Interferometrie nur mit speziellen Transceivern durchgeführt werden.

## 5.2. NIDES: Nachbarschaftsbasierte Singlehop-Distanzschätzung

Nachdem auf den vorangegangenen Seiten verschiedene aus der Literatur bekannte Techniken zur Distanzschätzung besprochen wurden ist deutlich geworden, dass die bestehenden Verfahren sämtlich zum Teil gravierende Schwächen aufweisen. Aus diesem Grund stellt dieser Abschnitt ein neues Verfahren zur Distanzschätzung zwischen benachbarten Geräten vor. Es basiert auf dem Vergleich von Nachbarschaftslisten und leitet aus dem Anteil der gemeinsamen Nachbarn, das heißt aus dem Schnitt der Nachbarschaften, eine Distanzschätzung ab. Das Verfahren trägt daher den englischen Namen *Neighborhood Intersection Distance Estimation Scheme (NIDES)*.

Zunächst wird die Idee, auf der NIDES basiert, schematisch erläutert und mögliche Einflussfaktoren auf die Distanzschätzung diskutiert. Es folgt die mathematische Herleitung der Schätzfunktion von NIDES. Abschließend wird ausgeführt, wie die nachbarschaftsbasierte Distanzschätzung in einem Sensornetz implementiert werden kann.

### 5.2.1. Grundlegende Funktionsweise

Die NIDES zugrunde liegende Idee kann man geometrisch veranschaulichen (siehe Abbildung 5.1): Je geringer der Abstand zwischen zwei Geräten (bei näherungsweise gleichartiger, omnidirektionaler Funkausbreitung) ist, desto stärker überlappen sich ihre Kommunikationbereiche. Liegen zwei Knoten direkt nebeneinander, umfasst die Überlappung nahezu ihre gesamten Kommunikationsbereiche (a). Ist ihr Abstand hingegen so groß, dass sie gerade noch miteinander kommunizieren können, überlappen sich die Bereiche nur zu etwa 40% (b). Selbst wenn der Abstand zwischen den Geräten so groß ist, dass keine direkte Kommunikation möglich ist, schneiden sich die Bereiche zum Teil noch (c). Wird zusätzlich ein drittes Gerät in Betracht gezogen, so könnten aus der Überlappung sogar Rückschlüsse über Winkel zwischen Geräten gezogen werden (d).

Leider kann die Überlappung des Kommunikationsbereiches nicht direkt gemessen werden. Geht man jedoch davon aus, dass die betrachteten Knoten nicht einzeln vorliegen, sondern gleichmäßig von vielen Nachbarknoten umgeben sind, wird eines deutlich: Die Größe des Schnittes der Kommunikationsbe-

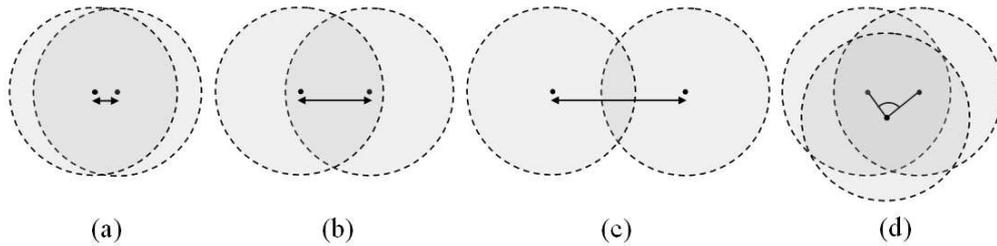


Abbildung 5.1.: Nachbarschaftüberschneidung bei verschiedenen Distanzen und Winkeln.

reiche drückt sich darin aus, wie viele Nachbarn zwei Geräte miteinander teilen. Dicht beieinander liegende Geräte haben viele Nachbarn gemeinsam, während weit voneinander entfernt liegende Geräte nur wenige oder keine Nachbarn gemeinsam haben.

In [37] präsentieren erstmalig Cartigny et al. die Idee, weit entfernt liegende Nachbarn zu erkennen, indem man die Ähnlichkeit von Nachbarschaften untersucht. Allerdings unternehmen sie nicht den Versuch, die Distanz zwischen Knoten durch den Vergleich von Nachbarschaftslisten zu ermitteln. Vielmehr nutzen sie den Anteil gemeinsamer Nachbarn, um zu entscheiden, ob ein Knoten eine geflutete Nachricht weiterleitet oder nicht. Je weniger Nachbarn Sender und Empfänger gemein haben, umso größer ist in dem von ihnen vorgeschlagenen Verfahren die Wahrscheinlichkeit, dass der Empfänger die Nachricht weiterleitet. Auf diese Weise ist die Wahrscheinlichkeit, dass weiter entfernte Knoten Nachrichten weiterleiten, besonders hoch, und die Anzahl neuer Knoten, die das Paket erhalten, wird gesteigert. Cartigny et al. beschränken den Vergleich von Nachbarschaften auf die Flutungssteuerung.

Im Gegensatz dazu wird in der vorliegenden Arbeit ein Verfahren vorgeschlagen, dass unter Verwendung eines Modells konkrete Distanzen zwischen Geräten aus dem Anteil der gemeinsamen Nachbarn ableitet.

NIDES basiert also auf dem Zählen von Nachbarn, seien es solche, die ein Knoten mit anderen Knoten teilt oder solche, die nur Nachbarn dieses Knotens sind. Ob die Kommunikation mit einem nahen Knoten möglich ist, er also ein Nachbar ist, hängt letztendlich von der Signalstärke der Nachrichten ab, die von diesem Knoten empfangen werden. Liegt der RSSI-Wert oberhalb einer bestimmten Schwelle, kann die Funkschnittstelle die Signale entschlüsseln, und ein Empfang ist möglich. Im Gegensatz zu RSSI-basierten Verfahren hat NIDES jedoch bis zu einem gewissen Maß die Möglichkeit, die Ungenauigkeiten der Messung von RSSI-Werten auszugleichen. Der Grund dafür ist, dass viele Geräte in die Distanzschätzung einbezogen werden, statt die Werte eines einzigen Senders zu messen. So können sich z.B. interindividuelle Hardwareunterschiede gegenseitig kompensieren, und die Distanzschätzung wird robuster.

Im Gegensatz zu vielen der in Abschnitt 5.1.2 besprochenen Verfahren zur Distanzschätzung benötigt NIDES keinerlei zusätzliche Hardware auf den Geräten. Vielmehr macht es von der Broadcasteigenschaft der ohnehin vorhandenen

drahtlosen Kommunikationsschnittstelle Gebrauch. In den folgenden Abschnitten wird darüber hinaus belegt, dass auch keine speziellen Anforderungen an die vorhandene Hardware der Knoten gestellt werden, und dass das NIDES schnell und energiesparend arbeitet.

### 5.2.2. Einflussfaktoren

Kern der in diesem Kapitel vorgestellten Überlegungen ist es, ein Modell zu entwickeln, mit dessen Hilfe sich aus dem Anteil der Nachbarn, die zwei Knoten gemeinsam haben, auf ihre Distanz schließen lässt.

Offensichtlich hat die Anzahl der Nachbarn, die ein Knoten besitzt, maßgeblichen Einfluss auf dieses Verfahren. Daher soll an dieser Stelle zunächst die Nachbarschaft von Knoten definiert werden: Ein Knoten B wird als Nachbar eines Knotens A angesehen, wenn A innerhalb eines bestimmten Zeitraumes von B versendete Funkpakete empfangen kann. Dafür ist es nicht erforderlich, dass A alle Pakete von B empfängt, es reicht ein einziges. Insbesondere ist es nicht erforderlich, dass auch B Pakete von A empfangen kann. Das bedeutet, dass es unerheblich ist, ob eine Verbindung uni- oder bidirektional ist. Somit spielt der Anteil der bidirektionalen Verbindungen keine Rolle. Dies ist wichtig, weil insbesondere in drahtlosen Sensornetzen häufig der Anteil unidirektionaler Verbindungen sehr hoch ist (vergleiche Abschnitt 2.5), was viele andere Verfahren und Protokolle behindert.

Von wesentlichem Einfluss auf die Nachbarschaft eines Knoten sind selbstverständlich

- die Anzahl der auf einer bestimmten Fläche ausgebrachten Knoten
- die Verteilung der Knoten und
- die Ausbreitungseigenschaften der eingesetzten Funkschnittstelle.

Die Anzahl der Knoten lässt sich zumindest innerhalb bestimmter Grenzen beeinflussen. Sie hängt zum einen natürlich davon ab, wie dicht die Geräte für die ihnen zugeordnete Messaufgabe ausgebracht werden müssen. Zum anderen hängt sie vom gewünschten Grad des Netzwerkes ab, das heißt von der mittleren Anzahl von Nachbarn. Hier besteht selbstverständlich ein enger Zusammenhang zu den Eigenschaften der Funkschnittstelle, zum Beispiel zu ihrer (mittleren) Reichweite.

Zusätzlich zur Reichweite hat aber auch die Charakteristik der Funkschnittstelle als solche einen wichtigen Einfluss. Dies sei hier an einem Beispiel simulativ gezeigt. Dabei bleibt die Knotendichte gleich, ebenso wie die mittlere Reichweite der Funkschnittstelle. Variiert wird lediglich die Charakteristik, hier exemplarisch widerspiegelt durch die drei in Abschnitt 2.5 beschriebenen und in Abbildung 2.4 dargestellten Ausbreitungsmodelle.

Für die drei Modelle zeigt Abbildung 5.2 ein Beispielszenario:

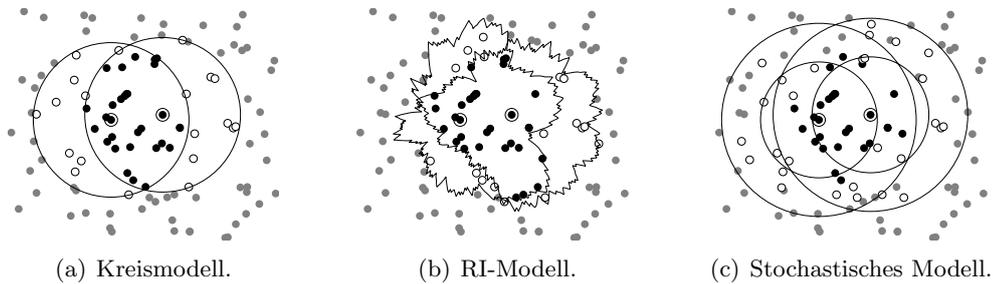


Abbildung 5.2.: Gemeinsame Nachbarn (schwarz) bei verschiedenen Funkmodellen.

- Zwei Bezugsknoten, die ihre Distanz schätzen wollen, sind als schwarze Kreise und mit zusätzlichem schwarzen Ring hervorgehoben dargestellt.
- Die Knoten, von denen beide Bezugsknoten Pakete empfangen können, sind als schwarze Kreise dargestellt. Sie sind gemeinsame Nachbarn der beiden Knoten.
- Solche Knoten, die Nachbarn nur eines der beiden Geräte sind, sind als weiße Kreise eingetragen.
- Alle übrigen Knoten des Szenarios sind grau dargestellt.

Zu beachten ist, dass alle Knoten in allen drei Szenarien an genau den gleichen Stellen liegen. Dennoch variiert die Anzahl der gemeinsamen Knoten aufgrund der Eigenschaften des verwendeten Funkmodells. In Teilabbildung 5.2(a) kam das Kreismodell zum Einsatz; die Kreise mit dem Radius  $r$  sind als dünne Ringe eingezeichnet. Hier gibt es 26 gemeinsame Nachbarn. Teilabbildung 5.2(b) zeigt die Situation bei Verwendung des RI-Modells; die beiden Knoten haben 27 gemeinsame Nachbarn. Das stochastische Modell kam für Teilabbildung 5.2(c) zum Einsatz. Die beiden konzentrischen Kreise kennzeichnen jeweils die beiden Distanzen von  $0,75r$  und  $1,25r$ . Hier haben die beiden Knoten 23 Nachbarn gemeinsam. Es sei nochmals betont, dass der Abstand der beiden Knoten in allen drei Szenarien gleich ist.

Es wird deutlich, dass die Anzahl der gemeinsamen Nachbarn vom Funkmodell abhängt. Dies gilt im Übrigen auch für die Gesamtzahl der Nachbarn jedes Knotens. Daraus resultiert die Variation des Anteils gemeinsamer Nachbarn in Abhängigkeit vom Funkmodell.

In der Folge wird klar, dass das Funkmodell bei der Distanzschätzung auf Basis von Nachbarschaften nicht außer Acht gelassen werden darf. Es muss vielmehr als integraler Bestandteil bei der Entwicklung der dem Verfahren zugrunde liegenden Modelle angesehen werden. Aus diesem Grund wählt das hier vorgestellte Verfahren das Funkmodell als Ausgangspunkt und berücksichtigt seine Eigenschaften.

Ein weiterer Faktor, der Einfluss auf die Distanzschätzung auf Basis von Nach-

barschaftsschnitten ausübt, ist die Verteilung der Nachbarn. Hat ein Gerät beispielsweise aufgrund einer ungleichmäßigen Verteilung nur in einem Teil des Bereiches, innerhalb dessen es kommunizieren kann, Nachbarn, kann dieses zu einer Verfälschung der Schätzergebnisse führen.

Ist dieser extreme Fall sicherlich die Ausnahme, so werden selbst unter günstigen Bedingungen gewisse Schwankungen in der Knotendichte nicht vermeidbar sein. Da die nachbarschaftsbasierte Distanzschätzung prinzipiell auf einer gleichmäßigen Knotenverteilung beruht, können Dichteschwankungen zu über- oder unterschätzten Distanzen führen. Dabei lassen sich grundsätzlich drei verschiedene Fälle unterscheiden. Diese werden in Abbildung 5.3 an extremen Beispielen schematisch für das Kreismodell gezeigt. Dabei zeigt der Pfeil von dem Knoten, der die Distanzschätzung vornimmt, zu demjenigen, zu dem die Distanz geschätzt wird.

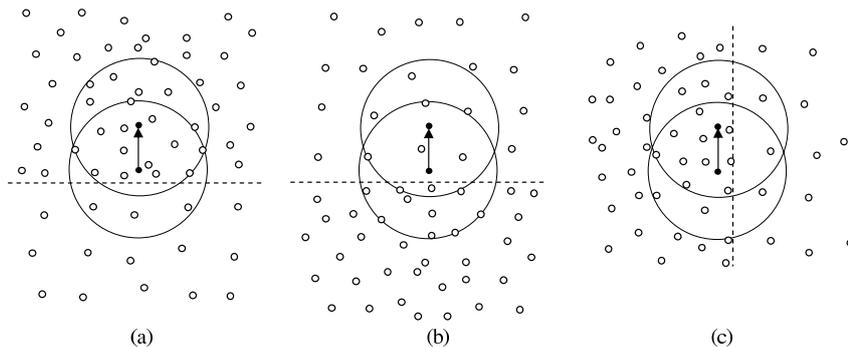


Abbildung 5.3.: Unterschiedliche Arten von Dichteschwankungen.

Ist die Knotendichte in der Schnittfläche höher als im Rest der Kommunikationsfläche (a), wird die geschätzte Distanz zu kurz sein, da der Anteil der Nachbarn in der Schnittfläche zu groß ist, um deren Größe angemessen widerzuspiegeln. Ist die Dichte in der Schnittfläche hingegen geringer als im Rest (b), wird es zu einer Überschätzung kommen. Das bedeutet, dass Dichtevariationen entlang der Schätzachse zu Fehlern führen, während Schwankungen orthogonal zu ihr ohne Einfluss bleiben (c).

Während sich die eher unrealistischen, hier lediglich zur Illustration herangezogenen großflächigen Dichteunterschiede an der unterschiedlichen Gesamtzahl der Nachbarn erkennen ließen, ist dieses für in der Realität auftretende lokale Schwankungen nicht möglich. Somit kann auch keine Kompensation der sich hieraus ergebenden Einflüsse erfolgen.

Insofern wird zu untersuchen sein, wie sich solche Schwankungen in der Netzdichte auf die Distanzschätzungen auswirken.

### 5.2.3. Mathematische Herleitung

Die Distanzschätzung auf Basis von Nachbarschaften erfordert eine Abbildung von dem Anteil der gemeinsamen Nachbarn zweier Knoten auf ihre Distanz. Dieser Abschnitt behandelt, wie diese unter Berücksichtigung eines Funkmodells ermittelt werden kann.

Zunächst wird hier beispielhaft für das Kreismodell die Berechnung einer Abbildungsfunktion dargestellt. Dieses Verfahren lässt sich jedoch nicht auf beliebige Funkmodelle anwenden. Daher wird im darauf folgenden Unterabschnitt ein Algorithmus beschrieben, der die entsprechende Abbildung für beliebige Funkmodelle ermittelt. Dieser verdeutlicht auch, wie die im Anschluss entwickelte allgemein gültige numerische Berechnung der Relation anschaulich funktioniert. In beiden Fällen ist das Ergebnis eine Abbildungstabelle zwischen dem Anteil der gemeinsamen Nachbarn  $s$  und der erwarteten Distanz  $d$ .

#### Vereinfachte Herleitung am Beispiel des Kreismodells

Um Distanzen auf Basis des Schnittes von Nachbarschaften abschätzen zu können, wird eine Abbildung zwischen dem Anteil der gemeinsamen Nachbarn der betrachteten Knoten und der Distanz zwischen ihnen benötigt.

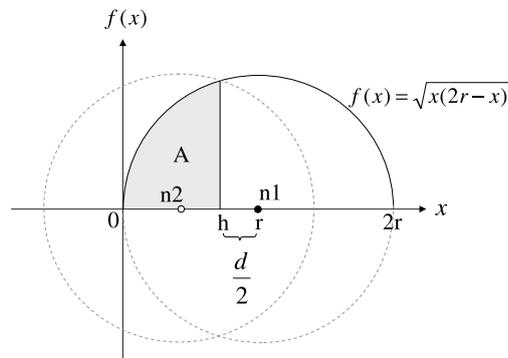


Abbildung 5.4.: Mathematische Grundlagen von NIDES beim Kreismodell.

Für Funkmodelle, die keine Wahrscheinlichkeitskomponenten einschließen, lässt sich  $d$  oft auf eine einfache, zweidimensionale Weise berechnen. Dieses soll hier am Beispiel des Kreismodells erläutert werden.

Abbildung 5.4 zeigt zwei Knoten  $n1$  und  $n2$ , die ihren Abstand bestimmen wollen, sowie ihre kreisförmigen Kommunikationsbereiche mit Radius  $r$ . Nun wird ein Koordinatensystem so angelegt, dass Knoten  $n1$  (schwarz gefüllter Punkt) an der Stelle  $(r, 0)$  zu liegen kommt. Aus dieser Konstruktion ergibt sich, dass der durchgezogene Halbkreis, der einen Teil des Kommunikationskreises von  $n1$  darstellt, durch folgende Kreisfunktion beschrieben wird:

$$f(x) = \sqrt{x(2r - x)}. \quad (5.1)$$

Die Fläche  $A$ , hier grau getönt dargestellt, entspricht einem Viertel der Fläche des Schnittes der beiden Kommunikationsbereiche von  $n_1$  und  $n_2$  und wird beschrieben von

$$F(h) = \int_0^h f(x) dx. \quad (5.2)$$

Die Stammfunktion von  $f$  ist

$$\begin{aligned} F(x) &= \int f(x) dx \\ &= \frac{r^2 \arcsin\left(\frac{x-r}{r}\right) + (x-r)\sqrt{x(2r-x)}}{2} + c, \end{aligned} \quad (5.3)$$

wegen  $F(0) = 0$  ergibt sich  $c = \frac{\pi r^2}{4}$ . Im Fall der Distanzschätzung ist die Situation so, dass  $A$  bekannt und der Punkt  $h$  gesucht ist, da sich aus  $h$  dann der Abstand zwischen  $n_1$  und  $n_2$  berechnen lässt. Mit anderen Worten: Gesucht ist derjenige Wert für  $x$ , für den  $F(x)$  einen bestimmten, gegebenen Wert annimmt. Unglücklicherweise kann  $F$  nicht nach  $x$  aufgelöst werden. Daher wird  $F$  durch eine Taylorreihe  $T$  um  $x = r$  approximiert, der wegen des Verlaufes von  $F$  kleinste sinnvolle Grad beträgt 3.

$$T(x) = -\frac{2x^3 - 6rx^2 - 6r^2x + r^3(10 - 3\pi)}{12r}. \quad (5.4)$$

$T$  lässt sich nun nach  $x$  auflösen. Dabei ergeben sich 3 Lösungen, wobei die relevante diejenige ist, die zwischen 0 und  $r$  liegt. Sie lautet

$$h = r + 2\sqrt{2}r \sin\left(\frac{\arcsin\left(\frac{3\sqrt{2}(4A - \pi r^2)}{16r^2}\right)}{3}\right). \quad (5.5)$$

Daraus lässt sich die Distanz zwischen den beiden Knoten nun leicht berechnen unter Verwendung von

$$d = 2(r - h). \quad (5.6)$$

Auf diese Weise kann man den Abstand zwischen zwei Knoten aus ihrem Kommunikationsradius und der Fläche  $A$  bestimmen, wobei  $A$  einem Viertel des Schnittes ihrer Kommunikationskreise entspricht. Dabei wird angenommen, dass  $r$  (oder wenigstens eine obere Schranke für  $r$ ) bekannt und für alle Knoten gleich ist. Daher wird  $r$  hier wie eine Konstante behandelt.

Dem folgenden Schritt liegt die Annahme zugrunde, dass die Sensorknoten gleichmäßig über die betrachtete Fläche verteilt sind und dass das Netzwerk

ausreichend dicht ist, damit die im Schnitt liegenden Knoten repräsentativ sind. Die Auswirkungen von lokalen Schwankungen in der Knotenverteilung wurden bereits in Abschnitt 5.2.2 diskutiert. Der Einfluss der Netzwerkdichte wurde simulativ untersucht; die entsprechenden Ergebnisse werden in Abschnitt 5.3.2 vorgestellt.

Unter Zuhilfenahme der Heuristik, dass die Größe der Fläche  $A$  proportional zu der Anzahl der sich im Schnitt befindlichen Nachbarknoten ist, kann  $A$  näherungsweise bestimmt werden.

Somit kann der Anteil der Nachbarn, die ein Knoten mit einem speziellen Nachbarn gemeinsam hat, benutzt werden, um die Größe der Fläche  $A$  auf

$$A = 0,25 \frac{n_{common}}{n_{total}} \pi r^2 \quad (5.7)$$

abzuschätzen, wobei  $n_{total}$  die Gesamtzahl der Nachbarn des Knotens  $n_1$  ist und  $n_{common}$  für die mit  $n_2$  geteilten Nachbarn steht.

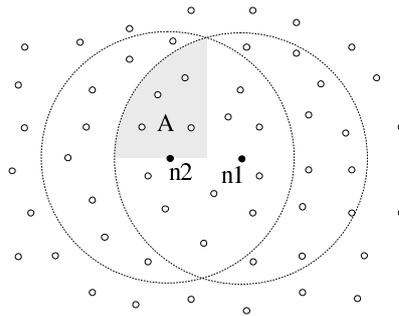


Abbildung 5.5.: Zusammenhang zwischen der Schnittfläche und der Anzahl gemeinsamer Nachbarn.

Abbildung 5.5 verdeutlicht die Approximation von  $A$  mit Hilfe des Anteils der gemeinsamen Nachbarn an einem Beispiel. Die Gesamtzahl  $n_{total}$  der Nachbarn von  $n_1$  liegt bei 24, die Zahl der gemeinsamen Nachbarn  $n_{common}$  bei 13. Somit beträgt der Schnitt  $\frac{13}{24}$  der Kreisfläche.  $A$  ist ein Viertel davon und ergibt sich zu  $4254,24$  für  $r = 100$ . Es resultiert eine geschätzte Distanz  $d$  von  $73,66$  bei einem wahren Abstand von  $73$ . Zu beachten ist, dass die beiden Knoten  $n_1$  und  $n_2$  selbst bei der Ermittlung des Schnittes nicht mitgezählt werden, da die erzielten Ergebnisse andernfalls verfälscht würden.

Allgemein gültig formuliert bedeutet das: Fasst man nun die Zahl gemeinsamer Nachbarn und die Gesamtzahl mittels

$$s = \frac{n_{common}}{n_{total}} \quad (5.8)$$

zu einem Anteil gemeinsamer Nachbarn  $s$  zusammen und führt die Gleichungen 5.5 bis 5.8 zusammen, ergibt sich insgesamt

$$d(s) = -4\sqrt{2}r \sin\left(\frac{1}{3} \arcsin\left(\frac{3}{16}\sqrt{2}(s-1)\pi\right)\right). \quad (5.9)$$

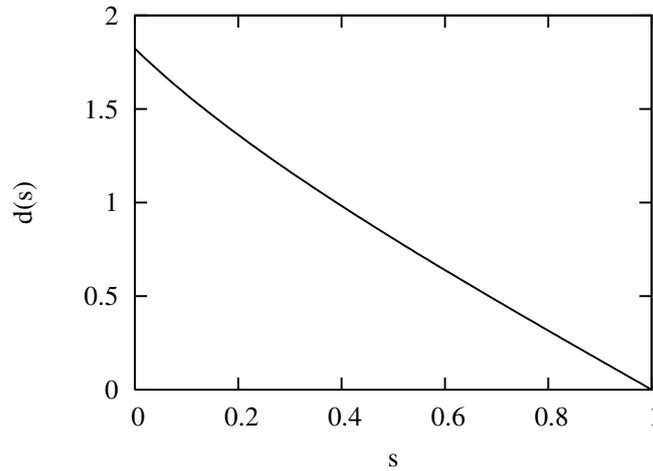


Abbildung 5.6.: Zusammenhang zwischen dem Anteil der Schnittfläche und der Distanz.

Abbildung 5.6 zeigt den Verlauf von  $d(s)$  graphisch. Dabei wurde der Klarheit wegen  $r = 1$  gesetzt. Zur Erinnerung: Die Funktion zeigt die erwartete Distanz über dem Anteil der gemeinsamen Nachbarn (beziehungsweise über dem Anteil der überlappenden Kommunikationskreisfläche). Ohne gemeinsame Nachbarn (gerade keine Überlappung, das heißt  $s = 0$ ) erwartet man eine Distanz von 2, sind alle Nachbarn gemeinsame Nachbarn (beziehungsweise überlappen die Kreise vollständig,  $s = 1$ ) ist eine Distanz von 0 zu erwarten.

Es ist zu erkennen, dass die Funktion für  $s = 0$  nicht wie erwartet den Wert 2 annimmt, sondern etwa 1,82. Dieses geht auf die Ungenauigkeit der Taylorreihe zurück. Bei perfekter Approximation müsste sich die Kurve für  $s \rightarrow 0$  der y-Achse anschmiegen. Der Fehler fällt jedoch bei der Distanzschätzung zwischen Nachbarn nicht ins Gewicht, da hier keine kleinen Werte für  $s$  auftreten können. Dieses würde darauf hindeuten, dass sich die Kommunikationsbereiche fast nicht überlappen, und somit die Knoten weit mehr als  $r$  auseinander liegen (vergleiche Abbildung 5.1 (c). Innerhalb des relevanten Bereiches  $s \in [0, 39; 1]$  jedoch ist die Approximation aufgrund des gewählten Entwicklungspunktes der Taylorreihe sehr genau.

### Allgemeine numerische Herleitung

Leider ist es nicht für alle Funkmodelle möglich, die Schätzfunktion wie oben beschrieben zu berechnen. Zum einen ist dies so nur für rotations-symmetrische Modelle ohne probabilistische Komponente möglich, zum anderen scheitert häufig schon der Schritt der symbolischen Integration an der Komplexität des

Funkmodells. Daher wird in diesem Abschnitt ein allgemeineres, numerisches Verfahren beschrieben, das aus dem Funkmodell die Schätzfunktion in Form einer Abbildungstabelle liefert.

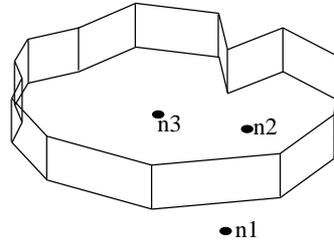


Abbildung 5.7.: Funkmodell von  $n3$ , sowie die Knoten  $n1$ ,  $n2$  und  $n3$ .

Um die Einträge in einer Abbildungstabelle zu berechnen bedarf es eines Algorithmus, der den erwarteten Anteil gemeinsamer Nachbarn zweier Knoten  $n1$  und  $n2$  für alle relativen Positionen von  $n1$  und  $n2$  zueinander berechnen kann.

Dabei ist zu beachten, dass ein Knoten  $n3$  Nachbar von  $n1$  und  $n2$  ist, wenn  $n1$  und  $n2$  Pakete von  $n3$  empfangen können. Das bedeutet, dass die Tatsache, ob  $n3$  zu den Nachbarschaften von  $n1$  und  $n2$  gehört, unabhängig von den Funkmodellen von  $n1$  und  $n2$  ist. Vielmehr hängt seine Nachbarschaft von  $n3$ 's Funkmodell ab, denn dieses bestimmt, ob  $n1$  und  $n2$  zum Beispiel eine Nachricht von  $n3$  empfangen können. Dieser Aspekt wird in Abbildung 5.7 verdeutlicht. Sie zeigt beispielhaft den Knoten  $n3$  mit seinem Funkmodell. Der Knoten  $n2$  liegt (unabhängig von seinem eigenen Funkmodell) innerhalb des Sendebereiches von  $n3$ , empfängt folglich seine Pakete und betrachtet ihn als Nachbarn, während  $n1$  den Knoten  $n3$  nicht als Nachbarn betrachtet. Dies bedeutet insbesondere, dass für Nachbarschaften keine bidirektionalen Verbindungen erforderlich sind.

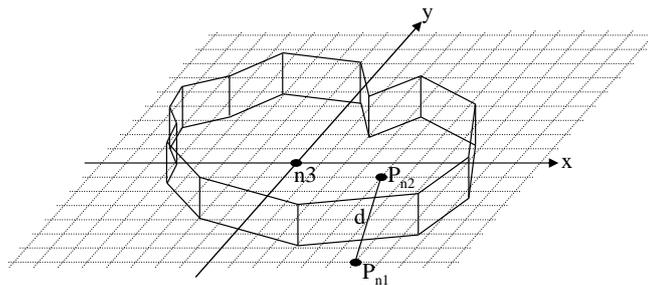


Abbildung 5.8.: Berechnung der Abbildungstabelle.

Der Algorithmus zur Berechnung der Abbildungstabelle wird nachfolgend beschrieben, seine Funktionsweise ist in Abbildung 5.8 veranschaulicht. Man platziert Knoten  $n3$  fix in der Mitte eines feinen Gitters und bewegt zwei Punkte  $P_{n1}$  und  $P_{n2}$  systematisch über alle Gitterpositionen, so dass alle möglichen Positionen abgedeckt werden. Die beiden Punkte repräsentieren so die jeweiligen Positionen zweier Knoten  $n1$  und  $n2$ . Für jede dieser Positionen wird nun die euklidische Distanz  $d$  zwischen  $P_{n1}$  und  $P_{n2}$  berechnet, und eine Variable  $n_{total}(d)$  inkrementiert, falls  $P_{n1}$  in Kommunikationsreichweite von  $n3$  liegt.

Diese Variable steht für die Fälle, in denen  $n_3$  ein Nachbar von  $n_1$  ist. Liegt  $P_{n_2}$  zusätzlich in Reichweite von  $n_3$ , wird eine zweite Variable  $n_{common}(d)$  inkrementiert. Sie repräsentiert die Fälle, in denen  $n_3$  ein gemeinsamer Nachbar von  $n_1$  und  $n_2$  wäre. Nachdem dieses für alle Gitterpositionen mit  $P_{n_1}$  und  $P_{n_2}$  durchgeführt wurde, ergibt  $s(d) = \frac{n_{common}(d)}{n_{total}(d)}$  die Abbildungstabelle zwischen der Distanz zweier Knoten und dem Anteil ihrer gemeinsamen Nachbarn. Auch die Umkehrung, das heißt die Relation zwischen dem Anteil der gemeinsamen Nachbarn und der Distanz  $d(s)$ , kann der Tabelle entnommen werden. Auf dieser Basis sind dann Distanzschätzungen möglich. Die diskreten Werte in der Tabelle werden dazu mittels linearer Interpolation in eine kontinuierliche Größe umgerechnet.

Während der soeben angegebene Algorithmus anschaulich zeigt, wie  $d(s)$  berechnet werden kann, wird nachfolgend ein systematischer, mathematisch-analytischer Ansatz zur Konstruktion von  $d$  aus einem gegebenen Funkmodell vorgestellt.

Startpunkt der Konstruktion ist eine Funktion  $f : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ , die das Funkmodell beschreibt. Sie gibt die Wahrscheinlichkeit an, dass ein Knoten, der an der Position  $(x, y)$  liegt, Funksignale eines im Ursprung positionierten Knotens empfangen kann. Als ein Beispiel dafür, wie eine solche Funktion aussehen kann, sei an dieser Stelle auf das in Abschnitt 2.5 vorgestellte und in Abbildung 2.4(b) dargestellte RI-Modell verwiesen.

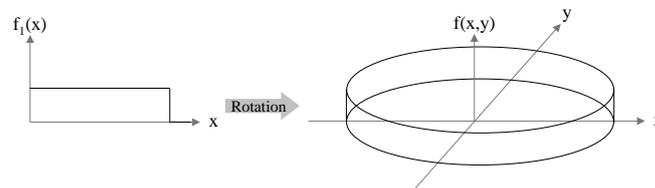


Abbildung 5.9.: Rotation von  $f_1$  um die Hochachse.

Einige Funkmodelle werden jedoch nur durch eine eindimensionale Funktion  $f_1 : \mathbb{R}^+ \rightarrow [0, 1]$  beschrieben, die die Empfangswahrscheinlichkeit in Abhängigkeit von der Distanz zwischen Sender und Empfänger angibt. Dies bedeutet, dass das beschriebene Funkmodell rotationssymmetrisch ist. Beispiele für derartig angegebene Funkmodelle sind das Kreismodell oder das stochastische Modell (vergleiche Abbildung 2.4(a) und Abbildung 2.4(c) in Abschnitt 2.5). Die eindimensionalen Funktionen solcher Modelle lassen sich durch eine Rotation um die Ordinate in eine zweidimensionale Funktion  $f(x, y)$  überführen, die dann wie folgt aussieht (siehe auch Abbildung 5.9):

$$f(x, y) = f_1(\sqrt{x^2 + y^2}) \quad (5.10)$$

Die sich ergebende Funktion beschreibt nun ebenfalls die Wahrscheinlichkeit, dass ein in  $(x, y)$  liegender Knoten Pakete eines in  $(0, 0)$  befindlichen Senders empfangen kann.

Der eigentliche Berechnungsschritt wird nun zunächst anschaulich beschrieben. Zur Vereinfachung wird hierzu eine rotationssymmetrische Funktion  $f$  herangezogen. Der voraussichtliche Anteil gemeinsamer Nachbarn  $s$  bei einer konkreten Distanz  $d$  lässt sich bestimmen, indem man  $f$  um  $d$  entlang der  $x$ -Achse verschiebt, sie mit ihrer nicht verschobenen Ausgangsfunktion multipliziert und das Volumen des Ergebnisses ins Verhältnis zu  $f$ 's Volumen setzt, also auf das Volumen von  $f$  normiert. Dieses ist in Abbildung 5.10 dargestellt, das Ergebnis der Multiplikation ist grau markiert. Um einen funktionalen Zusammenhang zwischen der Verschiebung (oder auch Distanz)  $d$  und dem Anteil der gemeinsamen Nachbarn  $s$  zu erhalten, muss man diesen Vorgang für alle  $d$  durchführen. Diesen Prozess nennt man Faltung (die abschließende Normierung ist allerdings nicht Teil der Faltung).

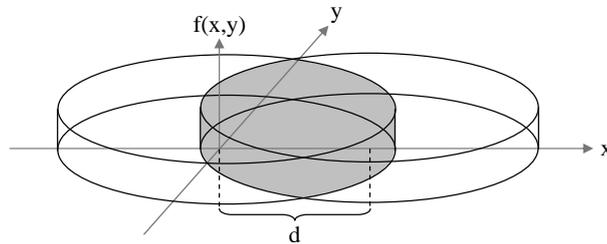


Abbildung 5.10.: Veranschaulichung einer Faltung.

Die Faltung von  $f$  entlang der  $x$ -Achse wird von der Gleichung

$$(f * f)(d) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - d, y) f(x, y) dx dy \quad (5.11)$$

beschrieben. Nimmt man die Normierung hinzu, ergibt sich

$$s(d) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - d, y) f(x, y) dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy}. \quad (5.12)$$

Da im Allgemeinen  $f$  nicht rotationssymmetrisch ist, muss während der Faltung auch eine Rotation durchgeführt werden.

$$f_{rot}(x, y, \alpha) = f(x \cos(\alpha) + y \sin(\alpha), -x \sin(\alpha) + y \cos(\alpha)) \quad (5.13)$$

gibt eine Funktion an, die aus  $f$  durch Drehung um die  $z$ -Achse um den Winkel  $\alpha$  entsteht. Durch Subtraktion von  $d$  wird sie entlang der  $x$ -Achse um  $d$  verschoben,  $d$  entspricht dabei dem Abstand zwischen Sender und Empfänger.

Der Anteil gemeinsamer Nachbarn über der Distanz lässt sich somit durch  $s : \mathbb{R}^+ \rightarrow [0, 1]$  beschreiben:

$$s(d) = \int_0^{2\pi} \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{rot}(x - d, y, \alpha) f_{rot}(x, y, \alpha) dx dy}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{rot}(x, y, \alpha) dx dy} d\alpha. \quad (5.14)$$

Während hier von  $-\infty$  bis  $\infty$  integriert wird, um die Formel maximal allgemein zu halten, können für den praktischen Einsatz die Grenzen an den Bereich angepasst werden, in dem die Funktion des Funkmodells Werte ungleich Null liefert.

In beiden Fällen gibt  $s(d)$  den zu erwartenden Anteil gemeinsamer Nachbarn für eine bestimmte Distanz  $d$  an.

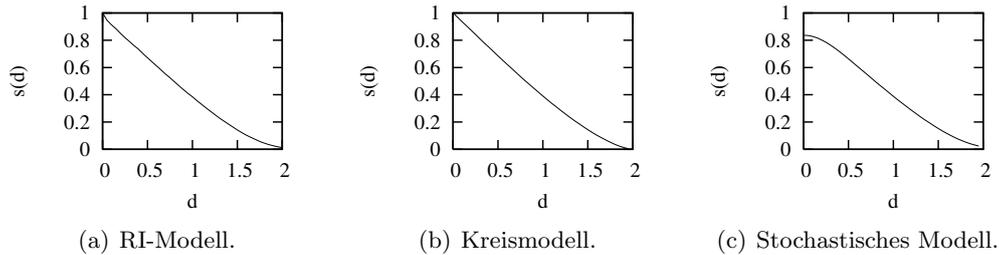


Abbildung 5.11.:  $s(d)$  für unterschiedliche Funkmodelle.

Abbildung 5.11 zeigt die resultierenden Verläufe von  $s(d)$  für unterschiedliche Funkmodelle. Für das RI-Modell ist der Verlauf in Teilabbildung (a) dargestellt, für das Kreismodell in (b) und das stochastische Modell in (c).

Da  $s$  den Anteil der gemeinsamen Nachbarn in Abhängigkeit von der Distanz beschreibt, wird für die Distanzschätzung ihre Umkehrfunktion benötigt:

$$d(s) = s^{-1}(d) \tag{5.15}$$

Um bei diesem Inversionsschritt eindeutige Ergebnisse zu erhalten ist es erforderlich, dass  $s$  bijektiv ist. Das bedeutet, dass sich nicht zwei Distanzen finden lassen, denen  $s$  den gleichen Anteil gemeinsamer Nachbarn zuordnet, oder spezieller formuliert, dass  $s$  streng monoton fallend ist. Wie in Abbildung 5.11 zu erkennen, ist dies für alle drei vorgestellten Funkmodelle der Fall.

### 5.2.4. Implementierung

In den vorhergehenden Abschnitten wurde aufgezeigt, wie aus einem Funkmodell die Schätzfunktion  $d(s)$ , das heißt die für die Distanzschätzung erforderliche Abbildung  $d(s)$  zwischen dem Anteil der gemeinsamen Nachbarn und der Distanz zweier Knoten ermittelt werden kann.

Dieser Abschnitt widmet sich nun der Frage nach der Umsetzung in realen Netzen. Es wird darauf eingegangen, wie sich das Verfahren auf die verschiedenen Phasen im Lebenszyklus eines Sensornetzes verteilt, wie Nachbarschaften ermittelt und gespeichert werden können und welche Komplexität die Berechnung der Distanz aufweist.

Um NIDES in einem realen Sensornetzwerk einsetzen zu können, müssen im Wesentlichen drei Schritte vollzogen werden:

1. Zunächst muss das Verhalten der Funkhardware ermittelt und beschrieben werden, das heißt das Funkmodell muss entwickelt werden. Dies kann durch Messungen geschehen oder zum Beispiel durch Konsultation der Datenblätter des Herstellers.
2. Ist das Funkmodell erstellt, kann die Schätzfunktion wie im vorigen Abschnitt beschrieben ermittelt werden. Entweder lässt sich die Funktion als Berechnungsvorschrift darstellen oder sie kann als Tabelle vorgehalten werden. In beiden Fällen kann sie in den Programmcode integriert werden, liegt dann im Code-Segment der Software der Geräte und benötigt in der Regel keinen Datenspeicher. In Tabellenform kann die Funktion alternativ auch im Datensegment abgelegt werden und könnte dann zur Laufzeit sogar noch verändert werden.
3. Zur Laufzeit müssen lediglich Nachbarschaftsinformationen zwischen den Knoten ausgetauscht werden. Die korrespondierenden Distanzinformationen können dann anhand der Schätzfunktion ermittelt und gespeichert werden.

Die ersten beiden Schritte können im Rahmen der Systementwicklung vor dem Ausbringen und dem Einsatz des Sensornetzwerkes erfolgen; die Schätzfunktion ist dann Teil der Software der Knoten. Auf dieser Basis können dann Distanzschätzungen erfolgen, sobald die Knoten ihre direkten Nachbarn sowie deren Nachbarn kennen. In diesem Abschnitt wird zunächst nur die Distanzschätzung zu direkt benachbarten Geräten betrachtet. Abschnitt 5.4 erläutert dann die Erweiterung auf Distanzen über mehrere Hops.

Um die benötigten Nachbarschaftsinformationen zu erlangen, muss jedes Gerät zwei Funkpakete broadcasten.

Zunächst müssen alle Knoten ein Paket aussenden, das es in der Nähe befindlichen Geräten ermöglicht, von ihnen Notiz zu nehmen. Jeder Empfänger fügt den Absender in seine Nachbarschaftsliste ein. Nach einiger Zeit haben alle Geräte eine vollständige Nachbarschaftsliste aufgebaut. Sie muss im Speicher gehalten werden und wird später um Distanzinformationen zu den jeweiligen Nachbarn ergänzt. Um schnelle Vergleiche von Nachbarschaftslisten zu ermöglichen, sollten die Listen nach den Adressen der Geräte sortiert vorgehalten werden. Im Allgemeinen sollte es für eine eindeutige Differenzierung von Nachbarn ausreichen, lediglich die unteren 16 Bit der Adressen zu betrachten. Selbst in extrem dichten Netzwerken mit 50 Nachbarn belegt eine solche Liste nur 100 Byte an Speicherkapazität, normalerweise deutlich weniger. Ist Speicherplatz extrem knapp, könnten die Adressen zumindest lokal eindeutig auf Achtbitidentifikatoren abgebildet werden.

Danach versendet jeder Knoten in einem zweiten Paket seine Nachbarschaftsliste. Wird diese von einem benachbarten Knoten empfangen, so kann dieser seine eigene Liste mit der gerade erhaltenen vergleichen und so die Anzahl  $n_{common}$

der gemeinsamen Nachbarn bestimmen. Da sowohl die lokale als auch die empfangene Liste sortiert sind, reicht ein paralleler Durchlauf der beiden Listen dazu aus.

Nachdem  $n_{common}$  nun bekannt ist, muss noch die Gesamtanzahl der Nachbarn  $n_{total}$  ermittelt werden um den Anteil gemeinsamer Nachbarn  $s$  bestimmen zu können. Hier sei vorgeschlagen, den Mittelwert der beiden Nachbarschaftsgrößen dafür heranzuziehen. Das hat den Vorteil, dass lokale Schwankungen in der Netzdichte und somit in der Anzahl der Nachbarn zumindest teilweise ausgeglichen werden können. Auf diese Weise können Distanzschätzungsfehler, die auf Schwankungen in der Netzdichte zurückgehen, verringert werden. Darüber hinaus ist so sichergestellt, dass beide Knoten zu der gleichen Distanzschätzung gelangen.

An dieser Stelle sei noch einmal auf die motivierenden Erläuterungen zur Ermittlung der Funktion  $s(d)$  zu Beginn der allgemeinen numerischen Herleitung erinnert. Das Verfahren an sich setzt nicht voraus, dass sich an den beiden Positionen, zwischen denen die Distanz ermittelt werden soll, auch tatsächlich physisch Knoten befinden. Tatsächlich würde das Verfahren auch für passive Knoten funktionieren, die ihre eigene Nachbarschaftsliste nicht aktiv aussenden, sondern statt dessen lediglich ihre Liste (ohne sich selbst) mit von anderen Knoten empfangenen Listen vergleichen. Es sei aber der Vollständigkeit halber darauf hingewiesen, dass der Verzicht auf die aktive Teilnahme natürlich die Schätzungen der anderen benachbarten Knoten verfälscht.

Somit ist bei der Bestimmung sowohl von  $n_{common}$  als auch von  $n_{total}$  zu beachten, dass die beiden an der Distanzschätzung beteiligten Knoten nicht mitgezählt werden. Würden zwei aktiv an der Distanzschätzung teilnehmende Knoten sich jeweils in den Listenvergleich einbeziehen, würde  $s$  von  $\frac{n_{common}}{n_{total}}$  zu  $\frac{n_{common}+1}{n_{total}+1}$  anwachsen und somit zu einer Distanzunterschätzung führen. Dieser Fehler nähme mit zunehmender Distanz zu, da  $n_{total}$  konstant bleibt während sich  $n_{common}$  verringert und so der Einfluss der falschen Zählung steigt.

Ist der Anteil gemeinsamer Nachbarn  $s$  bestimmt, kann die zugehörige Distanz durch Anwendung von  $d(s)$  ermittelt werden. Da dies im Allgemeinen nur numerisch geschehen kann, sei hier eine numerische Variante für  $d$  angeben:

$$d(x) = \min \{ d \in \mathbb{R}^+ \mid x \leq s(d) \}. \quad (5.16)$$

Die gesuchte Distanz ergibt sich zu dem kleinsten positiven Wert  $d$ , für den  $s(d)$  größer als oder gleich groß wie der Anteil der gemeinsamen Nachbarn der beiden Knoten ist. NIDES nutzt den in Abbildung 5.12 angegebenen Algorithmus zum Durchlaufen der Wertetabelle für  $s(d)$  und zur Interpolation zwischen den Werten. Als Argumente erhält er die nach Distanz geordnete Wertetabelle, den Anteil der gemeinsamen Nachbarn sowie die Schrittweite zwischen zwei Tabelleneinträgen. Die `while`-Schleife ermittelt zunächst den ersten Tabellenindex, an dem der entsprechende Tabelleneintrag kleiner ist als der übergebene Anteil gemeinsamer Nachbarn. Die Berechnung des `return`-Wertes führt dann noch

eine lineare Interpolation zwischen `table[index-1]` und `table[index]` durch, um das Ergebnis so genau wie möglich zu machen.

```
double
calculate_distance(
    double table[],
    double fraction,
    double step )
{
    index = 0;
    while (
        ( index < table.size() ) &&
        ( table[index] > fraction )
    )
        index++;

    if (index == 0 ) return 0;
    return
        step * ( index -
            ( fraction - table[index] ) /
            ( table[index-1] - table[index] )
        );
}
```

Abbildung 5.12.: Algorithmus zur Tabellensuche und Interpolation.

Der hier angegebene Algorithmus stellt eine direkte Implementierung von Gleichung 5.16 dar. Natürlich sind ausgefeiltere Interpolationsverfahren wie kubische Splines oder effizientere Suchtechniken wie beispielsweise ein logarithmisches Vorgehen denkbar. Allerdings liegt in der vorgestellten Variante ein günstiger Kompromiss zwischen Codekomplexität, effizienter Laufzeit und erzielter Genauigkeit vor.

Auf diese Weise kann nun der Empfänger einer Nachbarschaftsliste die Distanz zwischen dem Absender und ihm selbst ermitteln und diese in einer Distanzliste am Index des Absenders speichern. Beachtet sei, dass es nicht erforderlich ist, die empfangene Nachbarschaftsliste permanent zu speichern. Vielmehr kann diese Liste nach der Bestimmung des Anteils gemeinsamer Nachbarn verworfen werden. Lediglich die lokale Nachbarschaftsliste sowie die Distanzen zu diesen Nachbarn müssen dauerhaft im Speicher gehalten werden. Dabei sollte eine 8-Bit Repräsentation für die Distanzen ausreichend sein.

In einigen Fällen wird es sinnvoll sein den Austausch von Nachbarschaftsinformationen periodisch zu wiederholen. Dieses ermöglicht beispielsweise Distanzschätzungen in mobilen Szenarien. Je schneller sich die Knoten bewegen, umso häufiger müssen dann Nachbarschaften ausgetauscht werden, um sicherzustellen, dass die Informationen Gültigkeit behalten und korrekte Distanzen widerspiegeln.

Auch kann es sein, dass die Knoten bereits über Nachbarschaftslisten verfügen, ohne dass NIDES dazu explizit Nachrichten ausgetauscht hat. Beispiele für andere Verfahren und Protokolle, die ebenfalls Nachbarschaftsinformationen er-

heben, sind etwa Routingprotokolle wie AODV [140], Verfahren zur Topologiesteuerung wie SPAN [40] oder Clusteralgorithmen wie Tight Clubs [48]. Liegen bereits Nachbarschaftslisten vor, kann das Aussenden der ersten Nachricht entfallen. Der Austausch der Nachbarschaftslisten ist jedoch weiterhin notwendig, wenn dieses nicht ebenfalls bereits andere auf den Knoten laufende Protokolle übernehmen.

### 5.3. Simulative Evaluation der Distanzschätzung zwischen Nachbarn

Dieser Abschnitt widmet sich der simulativen Untersuchung der Genauigkeit von NIDES in Abhängigkeit von wechselnden Einflussfaktoren.

Um zunächst die Genauigkeit von NIDES bei Distanzschätzungen zwischen benachbarten Knoten zu untersuchen, wurde eine Vielzahl von Simulationen mit dem bekannten Netzwerksimulator ns-2 [134] durchgeführt.

#### 5.3.1. Simulationsumgebung

Wie bereits in Abschnitt 2.5 erläutert, wurden in der Vergangenheit verschiedene Funkausbreitungsmodelle daraufhin untersucht, inwiefern sie für drahtlose Sensornetze realistisch sind. Auch wurden spezielle Modelle wie das RI-Modell entwickelt. Da diese Modelle nicht Bestandteil von ns-2 sind, wurde der Simulator entsprechend ergänzt. Neben dem Freespace-Modell des Simulators, das der Wirkung nach dem Kreismodell entspricht, verfügt der Simulator nach dieser Erweiterung auch über Ausbreitungsmodelle, die das Verhalten des stochastischen Modells und des RI-Modells simulieren.

In ausnahmslos allen Simulationen wurde der mittlere Kommunikationsradius  $r$  auf 100 eingestellt.

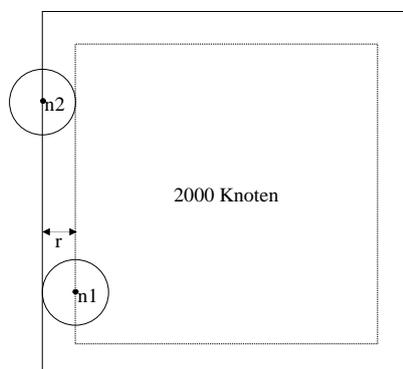


Abbildung 5.13.: Simulationsszenario.

Für alle Simulationen, die der Evaluation der Distanzschätzung zwischen be-

nachbarten Geräten dienen, wurde das in Abbildung 5.13 dargestellte Simulationsszenario verwendet.

Die Anzahl der Knoten in der gesamten Simulationsfläche wurde dabei stets so gewählt, dass im inneren Bereich 2000 Knoten zu liegen kommen. Um unterschiedlich dichte Netzwerke simulieren zu können, wurde die erforderliche Größe des inneren Bereiches für diese Dichte berechnet. Daraus ergibt sich, wie viele Knoten über der etwas größeren Gesamtfläche verteilt werden müssen.

Für nahezu alle Simulationen wurden Knoten gleichmäßig zufällig über die Simulationsfläche verteilt, allerdings wurden lediglich Distanzen zwischen solchen Knoten betrachtet, die im inneren Quadrat lagen. Alle Knoten darin verfügen mindestens über einen Abstand von  $r$  zum Rand der Simulationsfläche. Dies führt dazu, dass der gesamte Bereich, innerhalb dessen ein betrachteter Knoten kommunizieren kann, auch tatsächlich mit benachbarten Knoten besetzt ist (vergleiche Knoten n1). Dieses Vorgehen vermeidet Randeffekte, wie sie zum Beispiel bei n2 auftreten würden. Dieser verfügt auf der linken Seite über keinerlei Nachbarn, so dass die Schätzergebnisse verzerrt würden.

Wie bereits oben dargestellt, wurden die Knoten zufällig uniform auf der Simulationsfläche verteilt. Dies bedeutet, dass für jeden Knoten x- und y-Koordinaten aus einer Gleichverteilung entnommen wurden. Das führt zu einer Netzwerkdichte, die global betrachtet genau dem gewünschten Wert (z.B. 15) entspricht. Das bedeutet allerdings nicht, dass jeder Knoten exakt 15 Nachbarn besitzt, sondern lediglich, dass dies im Mittel gilt.

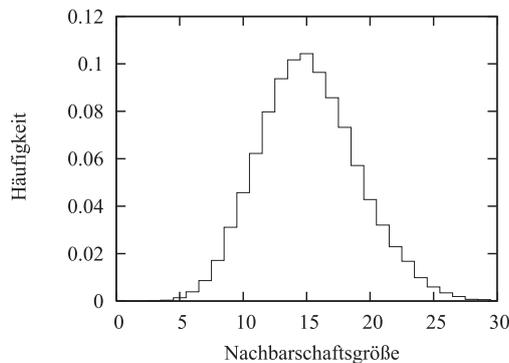


Abbildung 5.14.: Histogramm der Nachbarschaftsgrößen.

Dieser Aspekt wird in Abbildung 5.14 verdeutlicht. Sie zeigt exemplarisch ein Histogramm der Nachbarschaftsgrößen für eine eingestellte mittlere Dichte von 15. Es ist erkennbar, dass die meisten Knoten tatsächlich etwa 15 Nachbarn haben, allerdings wird auch klar, dass es ebenfalls Knoten mit nur 5 Nachbarn gibt. Andererseits treten Nachbarschaftsgrößen von bis zu 28 auf. Hier wird deutlich, dass bei einer mittleren Dichte von 15 erhebliche lokale Schwankungen der Dichte und somit der Gesamtzahl der Nachbarn bei den einzelnen Knoten auftreten. Es sei darauf hingewiesen, dass für dieses Histogramm nur Knoten und ihre Nachbarschaften betrachtet wurden, die im inneren Simulationsbereich

lagen. Somit sind die Schwankungen also nicht auf Randeffekte zurückzuführen, sondern sind vielmehr inhärenter Bestandteil zufälliger Knotenverteilungen und als solche nicht unerwünscht.

Um die Genauigkeit der nachbarschaftsbasierten Distanzschätzung zu untersuchen, wurde in jeder Simulation über alle Knoten im inneren Simulationsbereich iteriert, und dann die Distanzen zu allen Nachbarn (all jene Knoten, von denen Funkpakete empfangen werden können) mit Hilfe von NIDES abgeschätzt. Die geschätzten Distanzen wurden dann mit den echten, euklidischen Abständen verglichen. Die Differenz von geschätzter und tatsächlicher Distanz ergibt den Schätzfehler. Somit können also positive (das heißt aus Überschätzung resultierende) wie negative (aus Unterschätzung hervorgehende) Fehler entstehen.

Die Fehlerangaben werden im Weiteren stets auf den mittleren Kommunikationsradius normiert. Das bedeutet, dass der Fehler als Bruchteil des mittleren Kommunikationsradius  $r$  angegeben ist. Dieses Vorgehen ist in der Literatur durchaus üblich. Es hat gegenüber der Normierung auf die reale Distanz den Vorteil, dass ein im Wesentlichen konstanter absoluter Fehler für real sehr kleine Distanzen nicht ins Unendliche wächst und die Ergebnisse so leicht interpretierbar bleiben.

Um statistisch sichere Aussagen treffen zu können, wurden für jede im Anschluss beschriebene Untersuchungsreihe 100 gleichartige Simulationen durchgeführt. Dies bedeutet, dass sie sich lediglich durch zufällige Größen wie der Position der Knoten unterscheiden. Abschließend wurden die Ergebnisse gemittelt. Auf diese Weise kann sichergestellt werden, dass Ausreißer nicht fälschlicherweise als typisches Ergebnis interpretiert werden.

In der folgenden Evaluation liegt der Schwerpunkt auf der Verwendung des RI-Modells, da dieses speziell entwickelt wurde, um die Eigenschaften der Funkchnittstelle in drahtlosen Sensornetzen zu repräsentieren. Trotzdem werden im Weiteren ebenfalls mit Kreismodell und stochastischem Modell erlangte Ergebnisse präsentiert und mit denen des RI-Modells verglichen.

Es sei noch einmal darauf hingewiesen, dass in Abhängigkeit vom Ausbreitungsmodell nicht alle Kommunikationsverbindungen bidirektional sind. Diese Tatsache ist durchaus erwünscht, da dieser Effekt auch in der Realität auftritt (vergleiche Abschnitt 2.5) und, wie in Abschnitt 5.2.4 beschrieben, die Funktion des Verfahrens nicht einschränkt.

#### **5.3.2. Leistungsfähigkeit von NIDES in Abhängigkeit von der Netzdichte**

In diesem Abschnitt werden die Ergebnisse der Evaluation der Genauigkeit von NIDES in Abhängigkeit von der Netzdichte präsentiert. Dabei kam das RI-Modell zum Einsatz.

Abbildung 5.15 zeigt ein Interquartildiagramm, in dem der Schätzfehler über der mittleren Nachbarschaftgröße (das heißt der eingestellten Netzdichte) aufge-

tragen ist. Es wurden jeweils alle Schätzfehler innerhalb einer Simulationsserie mit einer bestimmten Netzwerkdichte betrachtet, und die Werte für Mittelwert, Median sowie oberes und unteres Interquartilbalkenende ermittelt. Alle Fehlerindikatoren sind als Bruchteile des mittleren Kommunikationsradius  $r = 100$  angegeben.

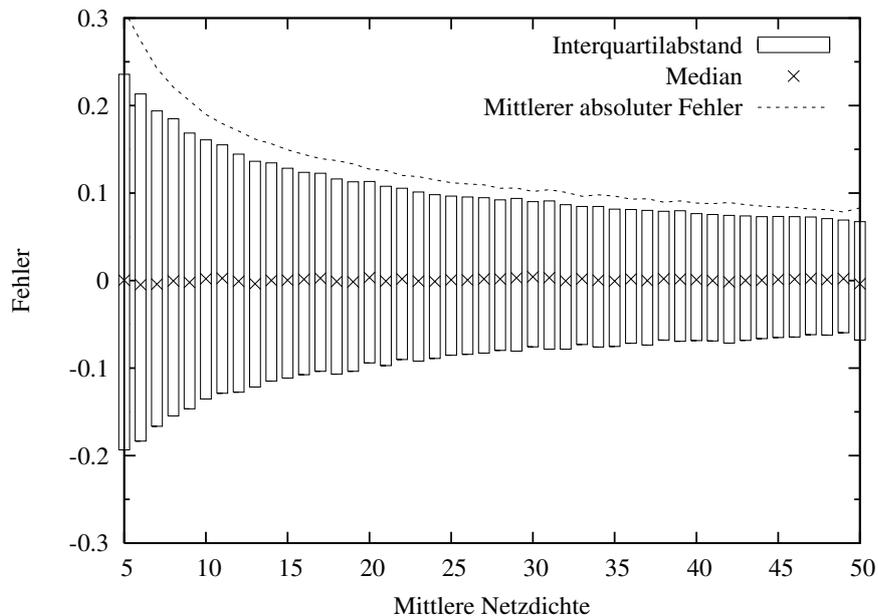


Abbildung 5.15.: Schätzfehler in Abhängigkeit von der Netzichte, RI-Modell.

In einem Interquartildiagramm gibt jeder Balken Aufschluss über Eigenschaften der Fehlerverteilung, hier für die Verteilung der Fehler für die betreffende Netzwerkdichte. Dabei beschreibt die Länge des Balkens denjenigen Bereich der Verteilung, innerhalb dessen die mittleren 50% der Werte liegen. Anders ausgedrückt liegt ein Viertel der Werte unterhalb des Balkens, ein weiteres Viertel oberhalb des Balkens, sowie die restliche Hälfte der Werte innerhalb des Balkens. Mit Hilfe eines Interquartildiagramms lassen sich also nicht nur Mittelwert und Median der Schätzfehler darstellen, sondern es ist auch erkennbar, wie stark die Fehlerstreuung ist.

Für die Angabe des mittleren absoluten Fehlers wurden die Beträge aller Fehler einer Simulationsserie summiert und durch die Anzahl der Werte geteilt. Der Mittelwert gibt somit an, um welchen Betrag die geschätzten Distanzen im Mittel von den tatsächlichen Distanzen abweichen.

Anders der Median: Hier werden unter Berücksichtigung des Vorzeichens alle ermittelten Fehler der Größe nach sortiert. Der Median ist derjenige Fehlerwert, der in der Mitte der resultierenden Reihung steht. Somit gibt er denjenigen Fehlerwert an, unter dem genauso viele Werte vorkommen wie über ihm. Er gibt folglich die Mitte der Verteilung an. Werden also genauso viele Distanzen unterwie überschätzt, ist ein Median von in etwa Null zu erwarten. Am Median lässt

sich somit ablesen, ob NIDES eine Tendenz zu Über- oder Unterschätzungen aufweist.

Hier wurde für alle Simulationsserien das RI-Modell mit einem Wert von 0,01 für den DOI-Parameter verwendet. Es kam die in Abbildung 5.11(a) dargestellte Schätzfunktion zum Einsatz.

In Abbildung 5.15 ist zu erkennen, dass die Spreizung der Interquartilsegmente mit zunehmender Netzwerkdichte abnimmt. Wie zu erwarten war, nimmt die Genauigkeit der Schätzungen mit steigender Nachbarschaftsgröße zu, die Streuung der Fehler reduziert sich. Bei sehr dichten Netzen liegt das Interquartil der Schätzungen innerhalb von  $0,07r$  um die korrekte Distanz, das heißt 50% der Schätzungen weisen einen Fehler von weniger als 7% des mittleren Kommunikationsradius auf. Für Netzdichten oberhalb von 10 liegt die Hälfte der Fehler innerhalb von  $\pm 0,15r$ . Der Mittelwert des Fehlers beträgt bereits hier weniger als 20% des mittleren Kommunikationsradius und fällt mit zunehmender Dichte auf etwa 8%.

Überall zeigt der Median praktisch keinerlei Abweichung von Null. Die Schätzungen weisen folglich unabhängig von der Netzhdichte keinerlei Tendenz zur Über- oder Unterschätzung auf.

Wie in Abschnitt 2.5 erläutert, stellt aus Sicht des Autors eine Netzwerkdichte von 15 eine sinnvolle untere Grenze für zufällig ausgebrachte Sensornetze dar. Daher soll die entsprechende Analyse der Schätzgenauigkeit für diesen Wert hier noch einmal separat erfolgen.

Bei einer mittleren Nachbarschaftsgröße von 15 ergibt sich ein Fehlermittelwert von etwa  $0,15r$ . Die Interquartilspreizung beträgt weniger als 0,2, somit weist die Hälfte der Schätzungen einen Fehler von weniger als  $0,1r$  auf.

Soweit nicht anders angegeben, wurden alle weiteren Fehlerauswertungen in Abschnitt 5.3 bei einer Netzhdichte von 15 durchgeführt.

#### 5.3.3. Einfluss des Abstandes der schätzenden Knoten

Ebenfalls wurde der Zusammenhang zwischen der Genauigkeit der Distanzschätzungen und dem tatsächlichen Abstand zwischen den beiden beteiligten Knoten untersucht. Die Resultate ergeben sich als Mittel aus einer Serie von 100 Simulationsläufen, dabei war die Netzhdichte auf 15 eingestellt.

Abbildung 5.16 zeigt ein Interquartildiagramm des Fehlers in Abhängigkeit von der euklidischen Distanz der Knoten. Sowohl der tatsächliche Abstand wie auch der Fehler werden dabei als Bruchteile des mittleren Kommunikationsradius  $r$  angegeben. Auch hier kam das RI-Modell zum Einsatz, der DOI-Wert betrug 0,01.

Es ist erkennbar, dass sowohl der mittlere absolute Fehler als auch die Interquartilspreizung mit der tatsächlichen Distanz der Knoten ansteigen. Liegt der mittlere absolute Fehler bei einer Distanz von  $0,1r$  noch bei 6% des mittleren

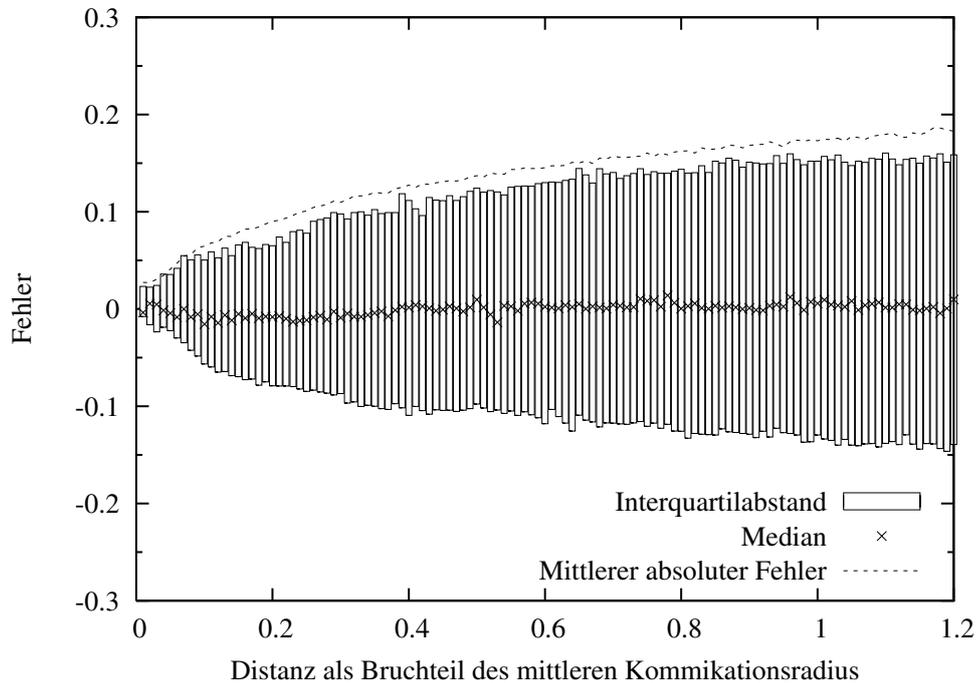


Abbildung 5.16.: Schätzfehler in Abhängigkeit von der Distanz, RI-Modell.

Kommunikationsradius, ist er bei einer Distanz von  $0,8r$  auf  $0,16r$  angestiegen.

Der Grund dafür ist, dass mit zunehmender Distanz der Knoten der Anteil und somit auch die absolute Anzahl gemeinsamer Nachbarn sinkt. Das führt dazu, dass lokale Schwankungen in der Knotenverteilung stärker ins Gewicht fallen und bei großen Distanzen vermehrt zu Fehlern bei der Distanzschätzung führen.

Dennoch bleibt die Interquartilspreizung unterhalb von  $\pm 0,15r$  und der mittlere absolute Fehler unterhalb von  $0,18r$ . Mittelt man Spreizung und Mittelwert über die gesamte Distanz, ergeben sich Werte von  $\pm 0,12r$  beziehungsweise  $0,15r$ . Wenig überraschend entspricht dieses den Ergebnissen in Abbildung 5.15 an der Stelle  $x = 15$ .

Bemerkenswert ist, dass der Fehler über alle Distanzen etwa dem Fehler bei einer Distanz von  $0,67r$  entspricht, also etwas nach oben aus der mittleren Distanz von  $0,6r$  herausgerückt ist. Das liegt daran, dass Schätzungen großer Distanzen denen kleiner Distanzen zahlenmäßig deutlich überlegen sind, da ein Knoten mehr weit entfernte Nachbarn hat als solche, die nahe bei ihm liegen. Somit gehen die etwas schlechteren Ergebnisse der Schätzungen über weite Distanzen stärker in das in Abschnitt 5.3.2 vorgestellte Gesamtergebnis ein.

Auch in Abbildung 5.16 wird deutlich, dass sich der Median stets sehr nahe der Null bewegt, NIDES also unabhängig von der tatsächlichen Distanz insgesamt keine Tendenz zur Über- oder Unterschätzung zeigt.

Es ist jedoch erkennbar, dass der Median für einzelne Distanzen nicht genau

zentriert ist, sondern einzelne Schwankungen aufweist. Tatsächlich ist es so, dass für einzelne Distanzen die Schätzungen leicht tendenziös sind. Die Ursache ist, dass die Distanz eine kontinuierliche Größe ist, während die Schätzergebnisse diskret sind. Der Grund liegt darin, dass beim Anteil gemeinsamer Nachbarn – selbst unter idealen Bedingungen – nur bestimmte Werte überhaupt auftreten können. So kann der Anteil beispielsweise bei  $\frac{7}{15}$  oder  $\frac{8}{15}$  liegen, nicht aber dazwischen, obwohl die perfekte Distanzschätzung dieses vielleicht erfordert hätte. Dieser Effekt ist beispielsweise bei einer Distanz von  $0,5r$  zu beobachten.

Die Intensität des Effektes hängt von der Netzdichte ab. Je größer die Dichte ist, desto größer wird der Nenner in demjenigen Bruch, der den Anteil gemeinsamer Nachbarn repräsentiert. Somit werden die Schwankungen geringer. Unabhängig von der Dichte liegen diese Unregelmäßigkeiten jedoch in einer Größenordnung, in der sie die Gesamtqualität der Distanzschätzungen nicht nennenswert beeinflussen.

#### 5.3.4. Einfluss schwankender Dichte

Alle bis zu diesem Punkt vorgestellten Evaluationen von NIDES beruhen auf der Annahme, dass die Sensorknoten zwar zufällig, aber gleichmäßig über die Simulationsfläche verteilt wurden. Diese Annahme ist jedoch für reale Sensornetze nicht besonders realistisch.

Denkt man an Szenarien, in denen die Sensorknoten zum Beispiel aus einem Flugzeug abgeworfen werden, ist vielmehr anzunehmen, dass die Knotenverteilung ungleichmäßig ist. Während in der Mitte der Fläche, auf der die Knoten zu liegen kommen, das Netz eher dicht sein wird, nimmt die Netzdichte zu den Rändern hin bis auf Null ab.

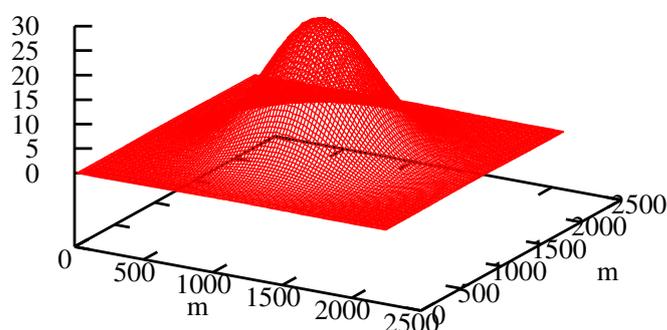


Abbildung 5.17.: Gaussverteilung der Knoten über die Simulationsfläche.

Um zu untersuchen, wie sich solche Verteilungen auf die Leistungsfähigkeit von NIDES auswirken, wurden auch Simulationen mit nicht-uniformen Knotenverteilungen durchgeführt. Zum Einsatz kam eine Gaussverteilung der Knoten über die Simulationsfläche, wie sie beispielhaft in Abbildung 5.17 dargestellt ist. Die Verteilung wurde so gewählt, dass die mittlere Netzdichte weiterhin bei 15 liegt, um eine Vergleichbarkeit mit den vorhergehenden Simulationen zu gewährleisten. Dabei ergibt sich in der Mitte der Simulationsfläche eine Netzdichte von etwa 27, die am Rand bis auf 0 sinkt.

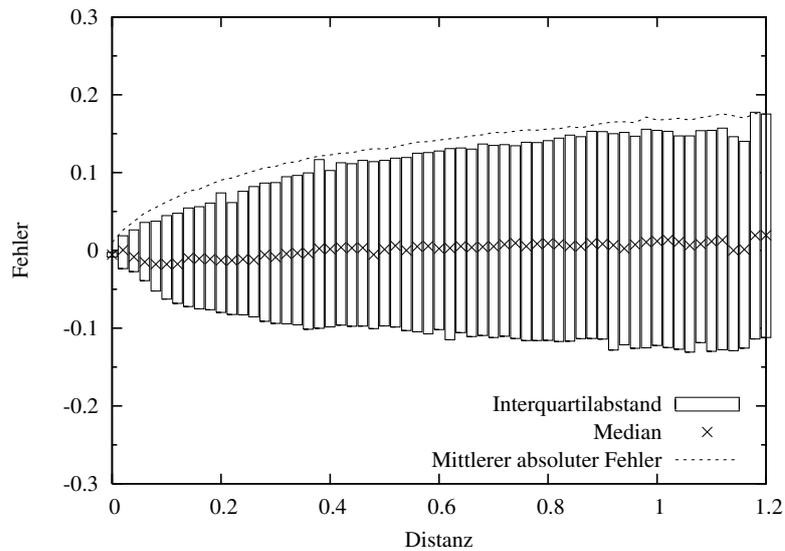


Abbildung 5.18.: Schätzfehler in Abhängigkeit von der Distanz, Gaussverteilung, RI-Modell.

Die sich ergebende Fehlercharakteristik wird in Abbildung 5.18 dargestellt. Sie zeigt die Schätzfehler in Abhängigkeit von der Distanz. Hier ist eine Darstellung über die Netzdichte nicht zielführend, da diese innerhalb der Simulation variiert. Angegeben sind wiederum der mittlere absolute Fehler, der Median und die Interquartilspreizung. Anders als in den bisher vorgestellten Simulationen wurden hier die Distanzschätzungen aller Knoten betrachtet, und nicht die Ränder der Simulationsfläche ausgeklammert.

Wie im vorhergehenden Abschnitt steigen der mittlere absolute Fehler wie die Spreizung der Interquartile mit zunehmender Distanz. Dabei gleichen die Ergebnisse denen in Abbildung 5.16. Lediglich die Streuung der Fehler ist geringfügig höher, was an der geringen Dichte des Netzes am Rand liegt. Der mittlere absolute Fehler ist nahezu gleich geblieben. Der Median liegt praktisch immer bei Null, die nicht-uniforme Verteilung der Knoten führt also nicht zu tendenziösen Distanzschätzungen.

Insgesamt lässt sich somit sagen, dass die ungleichmäßige Knotenverteilung nicht zu einer Verschlechterung der Genauigkeit von NIDES führt.

### 5.3.5. Einfluss des Radiomodells

In diesem Abschnitt wird untersucht, ob und wie das zum Einsatz kommende Funkmodell die Genauigkeit der Distanzschätzungen von NIDES beeinflusst. Dazu wurden Simulationen mit zwei weiteren Funkmodellen durchgeführt: dem stochastischen Modell und dem Kreismodell, die beide bereits in Kapitel 2.5 vorgestellt wurden.

Zunächst wurde der Einfluss des *linearen stochastischen Modells* untersucht, das bis zu einer Distanz von  $0,75r$  eine Kommunikationswahrscheinlichkeit von eins vorgibt, die dann bis  $1,25r$  linear auf Null zurückgeht. Die entsprechende Funktion, die die Wahrscheinlichkeit einer erfolgreichen Paketübertragung beschreibt, wurde bereits in Abbildung 2.4(c) gezeigt. Die sich daraus ergebende Schätzfunktion zeigte Abbildung 5.11(c).

Die Ergebnisse der Simulationen sind in Abbildung 5.19 zusammengefasst. Teilgrafik 5.19(a) zeigt die Schätzfehler in Abhängigkeit von der Netzdichte, der Fehler wird als Bruchteil der mittleren Kommunikationsreichweite angegeben. Der mittlere absolute Fehler liegt für eine Dichte von Zehn bei weniger als  $0,20r$  und sinkt bis auf 9% des Kommunikationsradius  $r$  bei einer Dichte von 50, während die Interquartilspreizung zwischen  $\pm 0,17r$  und  $\pm 0,07r$  liegt. Für eine Dichte von 15 liegt das absolute Fehlermittel bei 16% des Kommunikationsradius  $r$ . Die Hälfte der Schätzungen weist einen Fehler zwischen  $0,12r$  und  $0,14r$  auf. Der Median liegt stets in der Nähe von Null.

Diese Werte zeigen, dass die Ergebnisse marginal schlechter sind als diejenigen, die unter Verwendung des RI-Modells ermittelt wurden. Die Ursache dafür wird bei einem Blick auf Abbildung 5.19(b) deutlich. Es zeigt den Einfluss der euklidischen Distanz zwischen den schätzenden Knoten auf die Genauigkeit bei einer Netzdichte von 15.

Während sich oberhalb von  $0,3r$  eine starke Ähnlichkeit mit dem RI-Modell zeigt, ergeben die mit dem stochastischen Modell erzielten Distanzschätzungen darunter ein etwas anderes Verhalten. Dieses lässt sich unmittelbar aus den

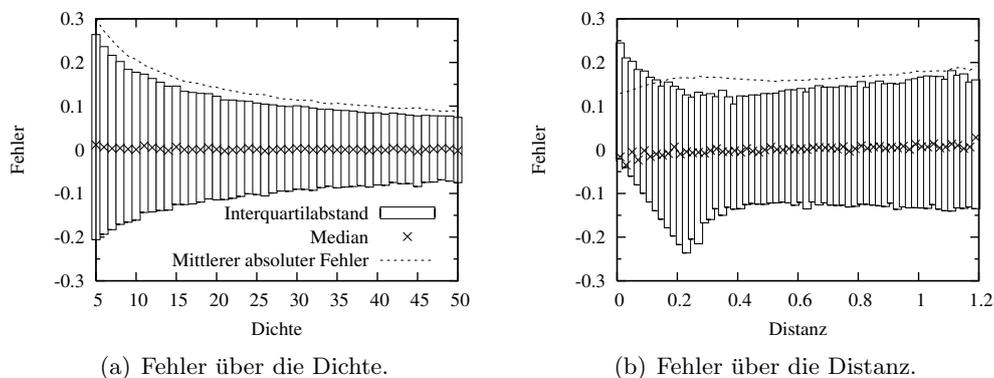


Abbildung 5.19.: Fehlercharakteristik mit dem stochastischen Funkmodell.

Eigenschaften des Funkmodells ableiten. Selbst wenn zwei Knoten sehr dicht beieinander liegen, kann ihre Nachbarschaft durchaus signifikant voneinander abweichen. Selbst wenn zwei Knoten an der selben Stelle liegen, das heißt einen Abstand von Null aufweisen, haben sie im Schnitt nur 83% ihrer Nachbarn gemeinsam (vergleiche die Schätzfunktion in Abbildung 5.11(c) an der Stelle  $d = 0$ ).

Da die Auswahl der Nachbarn aufgrund der stochastischen Komponente in der Beschreibung der Funkeigenschaften jedoch zufällig erfolgt, liegt die Überschneidung der Nachbarschaften zwar im Mittel bei 83%, kann jedoch im Einzelfall auch darüber oder darunter liegen. Die Schätzfunktion legt jedoch den Mittelwert zugrunde, sodass es zu Über- oder Unterschätzungen kommen kann, was zu einer verstärkten Fehlerstreuung führt.

Darüber hinaus verläuft die Schätzfunktion zu Beginn relativ flach. Somit führen bereits leichte Änderungen in der Zahl der gemeinsamen Nachbarn zu recht großen Variationen in der geschätzten Distanz. Teilen sich zwei dicht beieinander liegende Knoten weniger als 83% ihrer Nachbarn, schätzt NIDES somit einen überproportional großen Abstand.

Diese beiden Faktoren führen zu einer stärkeren Fehlerstreuung für kleine Distanzen.

Das lineare Abfallen des unteren Quartilendes für geringe Distanzen rührt daher, dass keine Distanzen unterhalb von Null geschätzt werden können, so dass die maximale Unterschätzung durch die wahre Distanz begrenzt ist. Dieser Effekt kann in Abbildung 5.19(b) für Distanzen unterhalb von  $0,23r$  beobachtet werden.

Die ausgeprägteren Fehler bei kleinen Distanzen beeinflussen das über alle Distanzen gemittelte Gesamtergebnis negativ, dennoch bleibt die Auswirkung auf einen etwa einprozentigen Fehlerzuwachs begrenzt.

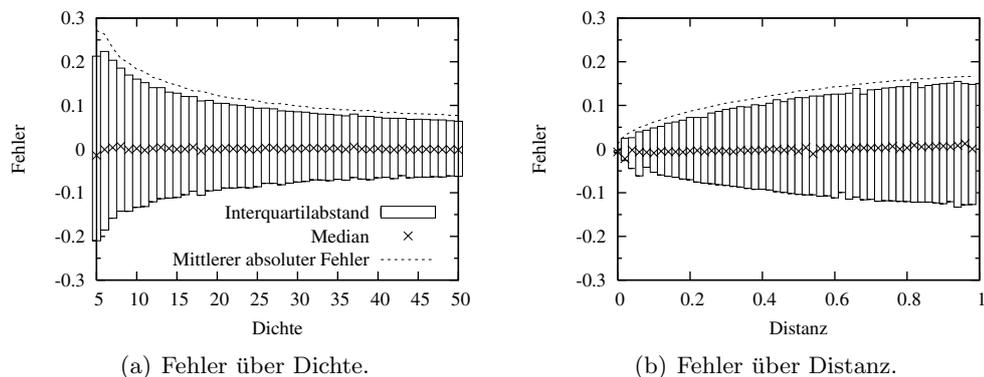


Abbildung 5.20.: Fehlercharakteristik mit dem Kreismodell.

Werden hingegen die Funkeigenschaften durch das *Kreismodell* beschrieben, zeigt sich ein Verhalten ähnlich dem des RI-Modells.

Die entsprechenden Diagramme sind in Abbildung 5.20 dargestellt. Hier kam die Schätzfunktion aus Abbildung 5.11(b) zum Einsatz.

Teilgrafik 5.20(a) zeigt den Einfluss der Netzdichte auf den Schätzfehler, der als Bruchteil des Kommunikationsradius  $r$  angegeben ist. Der mittlere absolute Fehler liegt für eine Dichte von 5 bei  $0,23r$  und sinkt auf 7% des Kommunikationsradius bei einer Dichte von 50, während die Interquartilspreizung von  $\pm 0,21r$  auf  $\pm 0,06r$  sinkt. Bei im Mittel 15 Nachbarn liegt der mittlere absolute Fehler bei etwa  $0,15r$ , während 50% der Schätzungen einen Fehler von  $\pm 0,12r$  aufweisen. Diese Werte sind insgesamt etwas günstiger als für das RI-Modell. Der Median der Fehler verläuft entlang der x-Achse.

In Teilgrafik 5.20(b) ist der Einfluss der tatsächlichen Distanz auf die Schätzgenauigkeit von NIDES bei einer Netzdichte von 15 dargestellt. Auch hier zeigt sich ein Verhalten, das dem der Funkeigenschaften des RI-Modell extrem ähnlich ist, von einer geringfügig höheren Genauigkeit abgesehen.

Insgesamt lässt sich also sagen, dass abgesehen von Unterschieden im Detail die Leistungsfähigkeit von NIDES weitgehend unabhängig davon ist, von welchem Funkmodell die Eigenschaften des Sensornetzes angemessen beschrieben werden.

## 5.4. Erweiterung von NIDES zur Multihop-Distanzschätzung

In den vorhergehenden Abschnitten wurde aufgezeigt, wie mit Hilfe von NIDES die Abstände zwischen benachbarten Knoten abgeschätzt werden können. Wie bereits in Abschnitt 3.2 dargelegt wurde, können so Sensorknoten, die über mehrere direkte Nachbarn mit bekannten Koordinaten (Ankerknoten) verfügen, ihre eigene Position ermitteln.

Es ist jedoch wünschenswert, dass auch dann Positionen bestimmt werden können, wenn der Anteil der Ankerknoten bei wenigen Prozent liegt. Ein derartig geringer Anteil von Ankerknoten hat zur Folge, dass nicht jeder Sensorknoten mehrere bereits positionsbewusste Knoten in seiner 1-Hop-Nachbarschaft hat.

Damit trotzdem möglichst viele Knoten ihre Position mit Hilfe eines Lokationsalgorithmus berechnen können, setzen viele der Verfahren voraus, dass auch Distanzen über mehrere Hops hinweg ermittelt werden können. So wird es möglich, Positionen auf Basis von wenigen Ankerknoten zu berechnen.

Aus diesem Grund war es erforderlich, NIDES so zu erweitern, dass Distanzen über mehrere Hops hinweg akkumuliert werden können. Die Wahl fiel auf die Kombination von nachbarschaftsbasierter Distanzschätzung mit dem Sum-Dist-Verfahren (vergleiche Abschnitt 5.1.4). Es hat die Vorteile, dass es schnell, umfassend und ohne weitere Voraussetzungen funktioniert.

Das *Euclidian*-Verfahren arbeitet mit sehr exakten Distanzschätzungen zwar

genauer, ist allerdings extrem sensibel gegen Fehler und kann Knoten zum Teil gar nicht mehr zu einer Distanzschätzung verhelfen. Das Verfahren DV-Hop verwendet keinerlei Singlehop-Distanzschätzungen und kam somit ebenfalls nicht in Frage.

Die Grundidee der Erweiterung von NIDES liegt darin, die zu den direkten Nachbarn mit Hilfe des Vergleiches von Nachbarschaftslisten ermittelten 1-Hop-Distanzen entlang des kürzesten Multihop-Pfades zwischen den Ankerknoten und den Knoten, die ihre Position ermitteln wollen, zu summieren.

#### 5.4.1. Auswirkungen von Dichteschwankungen

Die Analyse von Szenarien mit zufällig gleichverteilten Knoten zeigt, dass die Netzwerkdichte keineswegs gleichmäßig ist. Vielmehr ergibt sich eine Verteilung der Nachbarschaftsgrößen, die einer Normalverteilung um die eingestellte Netzsdichte gleicht. Diese Tatsache wurde bereits in Abschnitt 5.3.2 anhand von Abbildung 5.14 thematisiert, die ein Histogramm der Nachbarschaftsgrößen bei einer mittleren Netzsdichte von 15 zeigt.

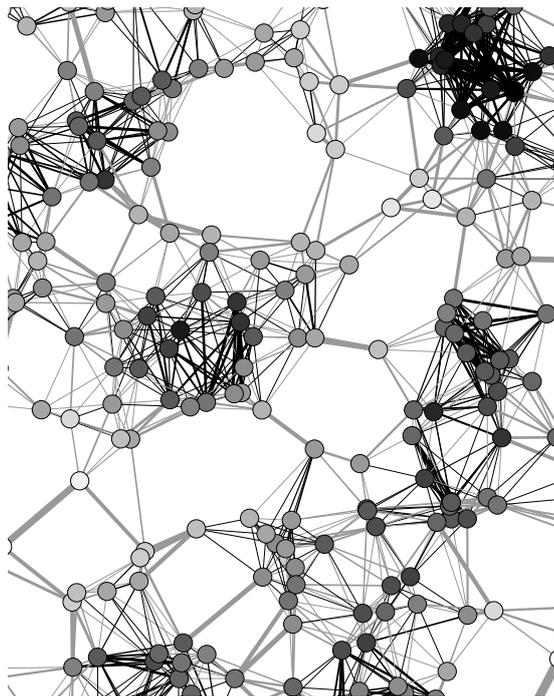


Abbildung 5.21.: Ausschnitt aus einem Szenario.

Die lokalen Schwankungen in der Anzahl der benachbarten Knoten sollen hier noch einmal aufgegriffen werden, um zwei weitere Folgen zu diskutieren, die speziell bei der Multihop-Distanzschätzung Einfluss ausüben.

In Abbildung 5.21 ist ein Ausschnitt aus einem Szenario der Dichte 15 zu se-

hen. Sie zeigt Knoten sowie die sich ergebenden Kommunikationsverbindungen zwischen ihnen. Der Grauton der Knoten repräsentiert dabei die Anzahl der Nachbarn: weiß steht für wenige, schwarz für viele Nachbarn, Grauschattierungen stehen für mittlere Dichten. Es ist klar zu erkennen, dass sich Cluster höherer Dichten und dazwischen Bereiche mit weniger Knoten bilden, die im Extremfall zu Löchern in der Topologie entarten können.

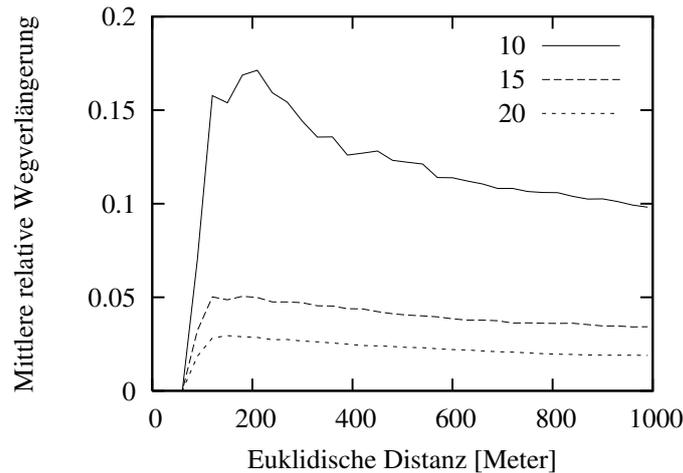


Abbildung 5.22.: Wegverlängerung bei Multihop-Pfaden.

Solche Löcher führen dazu, dass die Multihop-Pfade nicht gerade, sondern in Abhängigkeit von der Netzdichte mehr oder weniger stark im Zickzack verlaufen. Aufgrund dieser Tatsache ist die Summe der Distanzen zwischen den benachbarten Knoten im Pfad stets größer als die euklidische Distanz der Knoten an beiden Enden des Pfades. Diese Verlängerung des Weges ist von der Netzdichte abhängig: Je geringer die Dichte, desto länger sind die Pfade im Vergleich zur euklidischen Distanz. Abbildung 5.22 zeigt diesen Effekt beispielhaft für die Dichten 10, 15 und 20. Es ist weiterhin erkennbar, dass dieser Effekt bei kleinen Distanzen stärker ausgeprägt ist als bei großen. Ein Grund dafür ist, dass bei großen Distanzen mehr Hops zwischen den Knoten liegen und es somit mehr Möglichkeiten für günstige Pfade gibt. Weiterhin werden für Knoten mit geringerer Distanz zum Teil Pfade gefunden, die sehr viel länger sind als die euklidische Distanz, weil sie große Umwege nehmen. Für Knoten mit großer Distanz treten derartig lange Umwege praktisch niemals auf. Findet sich kein kurzer Pfad, so gibt es im Allgemeinen überhaupt keinen.

In Abbildung 5.21 ist erkennbar, dass einzelne Links die Löcher zwischen Clustern durchqueren und so die Knoten auf der einen Seite mit denen auf der anderen verbinden. Diese Links werden im Weiteren *Brückenlinks* genannt. Ihre verbindende Eigenschaft führt dazu, dass sie mit überdurchschnittlicher Wahrscheinlichkeit in Multihop-Pfaden enthalten sind. Das hat zur Folge, dass sich Fehler beim Schätzen von Distanzen entlang von Brückenlinks überproportional auf die Multihop-Schätzung auswirken.

In Abbildung 5.21 sind solche Links schwarz eingetragen, entlang derer es bei Distanzschätzungen mit NIDES zu Unterschätzungen kam, grau steht für einen Link, entlang dessen überschätzt wurde. Die Dicke der Linie steht für den Grad der Über- beziehungsweise Unterschätzung: Eine dicke Linie repräsentiert eine starke Verschätzung. Es fällt auf, dass die häufig verwendeten Brückenlinks fast immer (zum Teil erheblich) überschätzt werden. Die Ursache dafür ist, dass die Knoten an beiden Seiten des Links meist keine weiteren gemeinsamen Nachbarn haben. Daher gelangen sie zu einer Schätzung, die etwa beim Zweifachen ihrer mittleren Kommunikationsreichweite  $r$  liegt (vergleiche Abbildung 5.11(a)), obwohl sie auch beim RI-Modell im Mittel nicht weiter als  $r$  voneinander entfernt sind.

### 5.4.2. Adaption der Schätzfunktion an Multihop-Distanzen

Um zu ergründen, ob es sinnvoll ist, diesen Ausreißern entgegenzuwirken, werden im folgenden Abschnitt zwei unterschiedliche Varianten der Multihop-Erweiterung von NIDES untersucht und miteinander verglichen.

Die eine Variante basiert auf den in Abschnitt 5.2.3 beschriebenen und in Abbildung 5.11 dargestellten Schätzfunktionen, die auch bei den Simulationen der Singlehop-Variante von NIDES zum Einsatz kam.

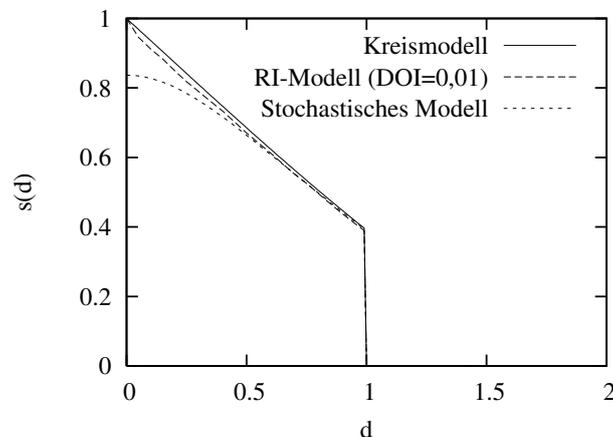


Abbildung 5.23.: Modifizierte Schätzfunktionen.

In der zweiten Variante kommen etwas veränderte Schätzfunktionen zum Einsatz. Diese Variation basiert auf der folgenden Überlegung: Geschätzte Distanzen, die oberhalb von  $r$  liegen, sind im Allgemeinen das Resultat von Überschätzungen, da es sehr unwahrscheinlich ist, dass zwei Knoten mit einem Abstand von mehr als  $r$  durch einen Link verbunden sind. Daher könnte es sinnvoll sein, die Schätzungen auf den mittleren Kommunikationsradius zu begrenzen. Die entsprechend angepassten Schätzfunktionen sind in Abbildung 5.23 dargestellt.

## 5.5. Simulative Evaluation von Multihop-Distanzschätzungen

Der folgende Abschnitt stellt die Ergebnisse der Simulationen zur Bewertung der Multihop-Erweiterung von NIDES für beide vorhergehend beschriebenen Varianten vor.

### 5.5.1. Simulationsumgebung

Um die Genauigkeit von Multihop-Distanzschätzungen auf Basis von Nachbarschaften zu untersuchen, wurde eine Reihe von Simulationen mit ns-2 [134] durchgeführt. Viele der Simulationsparameter gleichen denen in Abschnitt 5.3, allerdings gibt es Unterschiede in der Größe der Simulationsfläche und in der Auswahl der betrachteten Schätzungen.

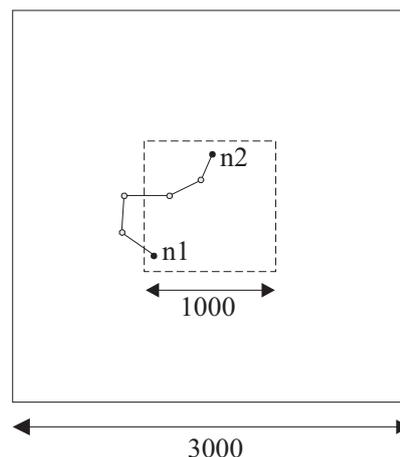


Abbildung 5.24.: Simulationsszenario.

In der 3000 m x 3000 m großen Simulationsfläche wurden die Knoten gleichverteilt angeordnet (Abbildung 5.24). Die mittlere Kommunikationsreichweite betrug 100 m. Die Anzahl der Knoten wurde so festgelegt, dass sich im Mittel die gewünschte Netzwerkdichte ergab. Dann wurden die Distanzen zwischen benachbarten Knoten mit Hilfe von NIDES geschätzt. Im Anschluss wurden für alle Knotenpaare die kürzesten Pfade untereinander ermittelt, wie es das Sum-Dist-Verfahren vorsieht. Entlang dieser Pfade wurden dann die Distanzschätzungen der einzelnen Hops addiert und mit der euklidischen Distanz der Knoten an den Pfadenden verglichen.

Für die Bewertung der Distanzschätzungen wurden nur Pfade zwischen Knotenpaaren herangezogen, die im zentral liegenden Neuntel der Simulationsfläche (gestrichelt) lagen.

Auf diese Weise können spezielle Randeffekte ausgeblendet werden. So könnte es beispielsweise bei Knoten am Rand passieren, dass außerhalb der Simulati-

onsfläche ein günstigerer Pfad hätte gefunden werden können. Durch die verwendete Pufferzone kann dieses weitgehend ausgeschlossen werden (wie bei dem Pfad zwischen  $n_1$  und  $n_2$ ).

Um statistisch verlässliche Aussagen zu erhalten, wurden wiederum jeweils die Ergebnisse von 100 Simulationsläufen gemittelt.

### 5.5.2. Auswertung

Im Rahmen der simulativen Evaluation wurde untersucht, wie sehr die Multihop-Distanzschätzungen von der euklidischen Distanz der Knoten an den beiden Enden des Multihop-Pfades abweichen. Dabei wurde das RI-Funkausbreitungsmodell verwendet. Die Netzwerkdichte betrug 15, das heißt jeder Knoten hatte im Mittel 15 Nachbarn.

Um den Erfolg der Adaption der Schätzfunktion bewerten zu können, wurden sowohl Simulationen mit der modifizierten Version von NIDES mit Begrenzung der Schätzungen auf den mittleren Kommunikationsradius wie auch ohne durchgeführt. Als Referenz wurden darüber hinaus Simulationen mit einer *perfekten* Distanzmessung entlang der Multihop-Pfade durchgeführt. Diese drei Messverfahren wurden dann jeweils mit der euklidischen Distanz verglichen.

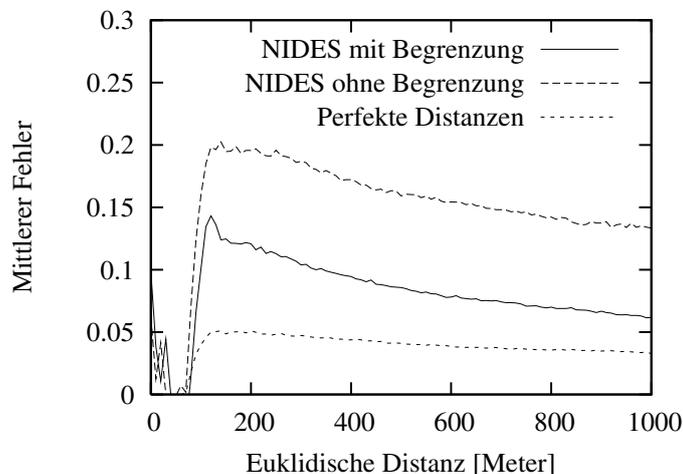


Abbildung 5.25.: Prozentualer Schätzfehler.

Abbildung 5.25 zeigt die Ergebnisse. Auf der Abszisse ist die euklidische Distanz der betrachteten Knoten aufgetragen. Die Ordinate zeigt die jeweiligen mittleren Messfehler der drei Verfahren als Bruchteil der tatsächlichen Distanz. Diese von den Singlehop-Distanzschätzungen abweichende Normierung ist sinnvoll, da extrem kurze Distanzen hier naturgemäß nicht im Mittelpunkt des Interesses stehen.

Es ist zu erkennen, dass die Messfehler ab einer Distanz von etwa 100 m stark

ansteigen. Das liegt daran, dass ab hier ein zweiter Hop zur Überwindung der Distanz erforderlich ist. Die Kurve der perfekten Distanzmessung zeigt, dass selbst ohne Fehler in der Singlehop-Distanzschätzung circa 5% zu lange Multihop-Distanzschätzungen auftreten. Die Ursache hierfür ist, wie im vorigen Abschnitt erläutert, dass die Distanz entlang des Multihop-Pfades tatsächlich länger ist als die euklidische, da der Pfad nicht gerade verläuft.

Der mittlere Fehler für NIDES mit Begrenzung steigt auf etwa 13% an, um dann kontinuierlich auf 6% für große Distanzen zurückzugehen. Am Verlauf der Kurve der perfekten Distanzen wird erkennbar, dass ein Drittel bis die Hälfte dieses Fehlers auf die Eigenschaften der Multihop-Pfade zurückgehen und selbst mit fehlerfreier Distanzschätzung nicht vermeidbar sind.

Ebenfalls wird deutlich, dass der Fehler für NIDES ohne Begrenzung auf 20% ansteigt, und dann sukzessive auf 13% abfällt. Er liegt damit etwa 7% höher als der der Variante mit Begrenzung. Dies macht deutlich, dass die Begrenzung der Distanzschätzungen auf den mittleren Kommunikationsradius ein äußerst effektives Mittel zur Fehlerreduzierung ist.

Für Distanzen unterhalb von 100 m (das heißt im Singlehop-Fall) liegt der Fehler für beide Varianten von NIDES bei etwa 5%. Im direkten Vergleich mit Abbildung 5.16 fällt ein anderer Verlauf der Fehlerkurve auf. Dies ist durch die etwas andere Art der Normierung begründet. Während der Fehler in Abbildung 5.16 relativ zum Kommunikationsradius dargestellt ist, wurde hier auf die tatsächliche Distanz normiert. Somit fallen Fehler bei kurzen Distanzen natürlich stärker ins Gewicht.

## 5.6. Eigenschaften von NIDES

Die nachbarschaftsbasierte Distanzschätzung erfüllt alle in Abschnitt 5.1.1 beschriebenen Anforderungen, um eine Eignung für den praktischen Einsatz in Sensornetzen sicherzustellen:

- Verzicht auf Spezialhardware beziehungsweise auf spezielle Anforderungen an die eingesetzte Hardware: Da NIDES auf Basis der ohnehin in drahtlosen Sensornetzen verwendeten Funkschnittstelle arbeitet, kann auf Spezialhardware vollständig verzichtet werden. Weder an den Transceiver werden besondere Anforderungen gestellt, noch an Prozessor und Speicher, da Berechnungskomplexität und Speicherbedarf gering sind.
- Geringer Energieverbrauch: Um die Distanzschätzungen zu ermitteln, werden lediglich zwei Nachrichten mit den direkten Nachbarn ausgetauscht, so dass der Kommunikationsaufwand und der daraus resultierende Energieverbrauch gering sind. Die niedrige Berechnungskomplexität und der begrenzte Speicherbedarf ermöglichen den Einsatz eines leistungsschwachen Prozessors, was, ebenso wie der Verzicht auf zusätzliche Hardware, den Energiebedarf minimiert.

- Geringe Dauer der Distanzermittlung: Die Dauer der Distanzschätzung ergibt sich aus dem Austausch zweier Nachrichten und der folgende Berechnung, und ist folglich auf Sekundenbruchteile begrenzt.
- Verzicht auf manuelle Kalibrierung: NIDES benötigt keinerlei Kalibrierung, lediglich ein Modell zur Beschreibung der Funkschnittstelle muss vorliegen.
- Reichweite: Die maximale Distanz, die NIDES direkt ermitteln kann, entspricht der mittleren Reichweite der Funkschnittstelle. Dieses ist zugleich das Maximum, das bei Distanzschätzungen auf Basis der Funkschnittstelle erzielbar ist. In Abschnitt 5.4 wurde darüber hinaus aufgezeigt, wie NIDES zu einer Multihop-Distanzschätzung erweitert werden kann, so dass die Reichweite beliebig gesteigert werden kann.
- Robustheit gegen Variationen der Hardware und der Umwelt: Da NIDES letztendlich auf dem Zählen von Nachbarn basiert, werden anders als bei Messungen, bei denen lediglich ein Sender und ein Empfänger beteiligt sind, viele Geräte in die Distanzschätzung einbezogen. Auf diese Weise können sich Hardwareunterschiede zwischen den Geräten oder unterschiedliche Umfeldbedingungen gegenseitig ausgleichen, so dass die Schätzungen robuster werden.

Aufgrund dieser Eigenschaften hebt sich NIDES deutlich von den meisten bestehenden Verfahren zur Distanzschätzung ab. Verfahren auf Basis von (differenziellen) Signallaufzeiten und der Messung der Einstrahldauer sowie die Radiointerferometrie benötigen zusätzliche Spezialhardware oder stellen spezielle Anforderungen an den Transceiver, auf die NIDES verzichten kann. Gegenüber der Interferometrie bietet NIDES darüber hinaus die Vorteile der deutlich kürzeren Dauer und der geringeren Komplexität.

Einzig Distanzschätzungen auf Basis von Konnektivität und Signalabschwächung bieten ähnliche Eigenschaften wie NIDES. Erstere gestatten jedoch lediglich qualitative Abschätzungen. Letzteren gegenüber bietet die nachbarschaftsbasierte Distanzschätzung nennenswert verbesserte Genauigkeiten.

Mit Hilfe von Simulationen wurde die Genauigkeit von NIDES untersucht. Wichtigster Einflussfaktor ist die Dichte des Sensornetzes, das heißt die mittlere Anzahl der Nachbarn. Während der durchschnittliche Fehler für eine Dichte von fünf bei etwa 30% des Kommunikationsradius liegt, sinkt er für eine Dichte von 50 auf nur noch 7%.

Bereits bei einer Dichte von 15 Nachbarn beträgt der Fehler im Mittel nur noch 15% der Kommunikationsreichweite.

Verfahren auf Basis von Signalabschwächung hingegen erreichen ohne aufwändige in-situ Kalibrierung lediglich Genauigkeiten zwischen 50% und 100% ihrer Kommunikationsreichweite [195].

## 5.7. Zusammenfassung

In diesem Kapitel wurde mit NIDES ein Verfahren zur Distanzschätzung auf Basis des Vergleiches von Nachbarschaftslisten vorgestellt. Es nutzt die Tatsache, dass nahe beieinander liegende Sensorknoten mehr Nachbarn teilen als solche, die weiter voneinander entfernt sind. Mit Hilfe eines Modells, das die Eigenschaften der Funkausbreitung beschreibt, können so Distanzen zwischen benachbarten Geräten abgeleitet werden.

Aufgrund seiner Eigenschaften hebt sich NIDES deutlich von den meisten anderen Verfahren ab. Gegenüber Signalstärkemessungen, die ansonsten identische Eigenschaften aufweisen, bietet NIDES eine um den Faktor drei bis sieben verbesserte Genauigkeit.

In Kombination mit dem aus der Literatur bekannten Sum-Dist-Verfahren [160] können mit NIDES auch Distanzschätzungen über die Kommunikationsreichweite hinaus durchgeführt werden. Dazu wurde die Schätzfunktion an die Anforderungen der Multihop-Distanzschätzung in Adhoc-Netzen angepasst. Die simulative Untersuchung hat gezeigt, dass die Genauigkeit der Schätzung über mehrere Hops nur fünf bis zehn Prozent schlechter ist als die, die mit einem perfekten Distanzschätzer möglich wäre.

Das Haupteinsatzgebiet von Distanzschätzungen in Sensornetzen sind Lokationsverfahren, die mittels Lateration arbeiten (vgl. Abschnitt 3.2.2). Sie nutzen Distanzschätzungen zu Ankerknoten, um daraus Positionen für alle anderen Sensorknoten zu berechnen. Da sich Fehler der Distanzschätzungen bei den meisten Laterationsverfahren direkt in Form von Positionsfehler auswirken, kann NIDES hier zu verbesserten Genauigkeiten beitragen.

Auf dieser Basis können Sensornetze ihre Messungen mit Ortsinformationen verknüpfen und so auch die räumliche Ausdehnung der beobachteten Phänomene bestimmen. Ein Verfahren, das aus einzelnen Messwert-Positions-Paaren die Phänomenkontur verteilt bestimmen kann, wird im folgenden Abschnitt vorgestellt.



## 6. Räumliche Approximation von Phänomenkonturen

Ein typischer Anwendungsbereich von Sensornetzen ist die Überwachung ihrer Umgebung. Ihre besondere Stärke spielen sie dabei aus, wenn es darum geht, für Menschen gefährliche oder schwer zugängliche Gegenden zu kontrollieren. Dort können Sensornetze dann autonom Daten erheben und verarbeiten. In vielen Fällen ist es dabei gewünscht, dass das Netz die räumliche Ausdehnung eines physikalischen Phänomens überwacht und diese Information gegebenenfalls an eine Kontrollstation weiterleitet.

Vielzitierte Anwendungsszenarien sind der Einsatz in feuergefährdeten Waldgebieten, die Überwachung von austretenden, gefährlichen Chemikalien oder die Kontrolle von Überflutungen. Die Sensorknoten könnten beispielsweise aus Flugzeugen oder Hubschraubern über dem betroffenen Gebiet abgeworfen werden, sich dort dann in ein Netzwerk organisieren und mit der Messung der relevanten Umweltdaten beginnen. Heiße, verseuchte oder überflutete Gebiete könnten so detektiert und die Daten gemeinsam mit Informationen über Gerätschaften und Katastrophenbekämpfungskräfte von den Sensorknoten an eine Datensenke wie die Einsatzleitung weitergeleitet werden.

Dabei ist es jedoch nicht sinnvoll, die Messwerte jedes einzelnen Gerätes zur Senke zu routen, um dann dort die Phänomenkontur zu bestimmen. Dieses naive Vorgehen würde den Energiehaushalt der Geräte überfordern, die Kapazität des Funknetzes übersteigen und somit nicht skalieren.

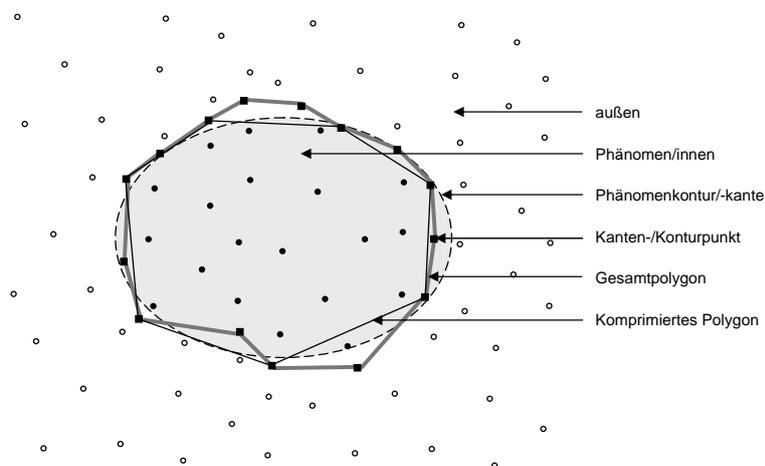


Abbildung 6.1.: Beispielhafte Darstellung der Konturapproximation.

Vielmehr sollte die räumliche Ausdehnung der Phänomenregion verteilt im Sensornetz ermittelt und die Konturdaten gebündelt übertragen werden. Dieses Kapitel beschäftigt sich damit, wie ein Verfahren, das diese Aufgabe übernimmt, gestaltet werden kann.

Die Ausgangssituation ist beispielhaft in Abbildung 6.1 dargestellt. Innerhalb des Gebietes, in dem das Sensornetz ausgebracht wurde, befindet sich eine Region, die besondere Eigenschaften (das sogenannte *Phänomen*) wie beispielsweise eine höhere Temperatur aufweist. Dieses ist in der Abbildung durch seine graue Färbung erkennbar. Zunächst muss die Kante des Phänomens (gestrichelte Linie) detektiert werden, das heißt es müssen Punkte (schwarze Quadrate) ermittelt werden, die auf oder nahe der Kante liegen. Diese werden schließlich zu einem möglichst kompakten Polygon zusammengefasst. Dazu wird aus dem Polygon, das alle ermittelten Punkte enthält (graue Linie), eines abgeleitet, das nur noch die wichtigsten Punkte enthält, das komprimierte Polygon (schwarze Linie).

Der Prozess lässt sich in drei Schritte unterteilen: Zunächst müssen die einzelnen Punkte der Kontur ermittelt werden, die sogenannte *Kantenerkennung* oder englisch *edge detection*. Dazu führen die Sensorknoten zunächst Messungen mit einem oder mehreren Sensoren durch. Sie entscheiden anhand der Messwerte und eines Prädikates, ob sie innerhalb oder außerhalb des gesuchten Phänomens liegen. Gemeinsam mit Knoten auf den anderen Seiten der Phänomenkontur können dann entsprechende Kantepunkte bestimmt werden. Diese Punkte müssen nun mit Hilfe der *Kantenaggregation* oder *boundary detection* zu einer geschlossenen Kontur, ausgedrückt als Polygonzug, zusammengeführt werden. Ohne weitere Vorverarbeitung kann eine solche Konturbeschreibung jedoch leicht aus vielen hundert Punkten bestehen, die sich weder effizient verarbeiten noch per Funk übertragen lassen. Somit bedarf es der *Konturapproximation* (englisch *boundary approximation*), die die Kantenbeschreibung auf wenige, repräsentative Punkte zusammenfasst.

Im Folgenden wird zunächst auf Anforderungen und verwandte Arbeiten aus der Literatur eingegangen. Danach wird dann ein neues Verfahren vorgestellt, das auf die Bedürfnisse von drahtlosen Sensornetzen zugeschnitten ist und insbesondere den Anforderungen genügt. Abschließend werden die Ergebnisse einer intensiven simulativen Evaluation des neuen Verfahrens vorgestellt.

Teile der in diesem Kapitel vorgestellten Ergebnisse sind in gemeinsamer Arbeit mit Daniela Krüger und Stefan Fischer entstanden und wurden in [27] veröffentlicht.

### 6.1. Verwandte Arbeiten

Dieser Abschnitt beschäftigt sich mit den Anforderungen an Verfahren zur Konturapproximation in Abschnitt 6.1.1 einerseits und in den folgenden beiden Abschnitten mit den Vorarbeiten anderer Autoren andererseits. Diese werden da-

bei kurz in ihrer Funktion erläutert, insbesondere jedoch mit den verschiedenen Anforderungen verglichen und daran gemessen.

Es fällt auf, dass es in der Literatur eigentlich keine Arbeiten gibt, die den gesamten Prozess der Konturapproximation abdecken. Während eine Gruppe von Verfahren sich lediglich mit der Kantendetektion befasst, beschäftigt sich eine andere mit der verteilten Berechnung von kompakten Repräsentationen der Kontur aus einzelnen Eckpunkten. Daher werden diese beiden Gruppen separat in den Abschnitten 6.1.2 und 6.1.3 besprochen.

### 6.1.1. Anforderungen an Verfahren zur Phänomenapproximation

Um für den praktischen Einsatz in einem Sensornetz geeignet zu sein, muss ein Verfahren zur Konturapproximation einer Reihe von Anforderungen genügen. Neben solchen, die – wie geringem Energie- und Ressourcenbedarf, eine verteilte Arbeitsweise oder die Robustheit gegen Kommunikationsfehler – generell für alle in Sensornetzen eingesetzten Verfahren gelten, gibt es auch speziell für die Konturapproximation geltende Anforderungen:

- **Robustheit gegen Messfehler:** Wie bereits in Abschnitt 2.4.3 thematisiert, arbeiten die eingesetzten Sensoren zum Teil nur mit begrenzter Zuverlässigkeit. Für ein Verfahren zur Konturapproximation heißt das, dass der Algorithmus fehlertolerant zum Beispiel auf Messfehler reagieren muss.
- **Kommunikationsstruktur:** Das Verfahren sollte nicht auf bestimmte Kommunikationsstrukturen wie beispielsweise Bäume, Cluster oder Overlays innerhalb des Sensornetzes angewiesen sein. Solche Strukturen aufzubauen und zu unterhalten ist aufgrund der fehleranfälligen und energieaufwendigen Kommunikation schwierig und ineffizient.
- **Nutzung von innen liegenden Knoten:** Eine zusätzlich wünschenswerte Eigenschaft ergibt sich aus den Phänomenen, die von Sensornetzen beobachtet werden. Einige von ihnen wie Wasser oder Feuer können dazu führen, dass Sensorknoten innerhalb der Phänomenregion Schaden nehmen, so dass diese Geräte besonders ausfallgefährdet sind. Insofern sollte der Algorithmus nach Möglichkeit auf den Sensorknoten im Randbereich des Phänomens ablaufen.

### 6.1.2. Verfahren zur Kantenerkennung

Einer der ersten Ansätze zur Kantenerkennung basiert auf Algorithmen, die ursprünglich aus der Bildverarbeitung stammen. Zur Nutzung in Sensornetzen werden diese adaptiert, indem die Sensornetzfläche in viele kleine Quadrate unterteilt wird, die jeweils als Pixel interpretiert werden. Auf diese virtuellen Pixel wird dann ein Filter zur Kantenerkennung angewendet [52]. Obwohl die Autoren diesen als Prewitt-Filter bezeichnen, differiert die verwendete Filter-

matrix von der des Prewitt-Filters [92]. Sensorknoten erklären sich zur Kante gehörig, wenn der Filterwert für ihr Quadrat eine Schwelle überschreitet. Die Schwierigkeit bei diesem Ansatz liegt darin, die Dichte des Sensornetzes so zu wählen, dass trotz zufälliger Verteilung der Sensorknoten möglichst genau ein Gerät in jedem Pixel zu liegen kommt.

In [42] vergleichen Chintalapudi und Govindan drei verschiedene Ansätze zur Kantenerkennung miteinander: erstens den zuvor beschriebenen (allerdings mit korrektem Prewitt-Filter), zweitens einen statistischen Ansatz und drittens einen Klassifikationsansatz. Der statistische Ansatz prüft, ob die Verteilung der Messwerte in der Nachbarschaft eines Knoten eine bestimmte (applikationsspezifische) Form hat. Wenn ja, erklärt sich der Knoten zu einem Kantenknoten. Für die Funktion des Klassifikationsansatzes ist es Voraussetzung, dass sich die Messwerte klar in zwei Gruppen unterteilen lassen. Dieser Ansatz versucht, eine Linie so in seiner Nachbarschaft zu positionieren, dass sie die Knoten mit Messwerten der beiden Gruppen möglichst komplett voneinander trennt. Überschreitet der Abstand des Knotens zur Linie nicht ein gewisses Maß, erklärt sich der Knoten zu einem Randknoten. Im auswertenden Teil ihrer Arbeit kommen die Autoren zu dem Ergebnis, dass die ersten beiden Ansätze unbefriedigende Ergebnisse zeigen, und lediglich der dritte Ansatz angemessen funktioniert. Leider weist dieser eine extrem hohe Komplexität im Hinblick auf die Berechnung auf und ist somit für den Einsatz in Sensornetzen nur bedingt geeignet.

Die Autoren von [55] schlagen ein Kantendetektionsverfahren mit besonderem Augenmerk auf Fehlertoleranz vor. Die Erkennung sowohl von Kantenknoten wie auch von Messfehlern geschieht auf Basis von Vergleichen des lokalen Messwertes mit Median und Varianz der Messwerte in der Nachbarschaft. Allerdings ist das Verfahren nur für Sensornetze mit sehr hohen Dichten geeignet: Die Autoren führen aus, dass das Verfahren bereits bei „moderaten Dichten“ von nur „30 Nachbarn“ gute Ergebnisse erziele.

Allen in diesem Abschnitt beschriebenen Verfahren ist gemein, dass sie nicht im eigentlichen Sinne Punkte einer Kante detektieren. Vielmehr wählen sie eine Teilmenge von Sensorknoten aus, die in einem Streifen rund um die zu approximierende Phänomenregion liegen. Allerdings hat dieser Streifen eine zum Teil nicht unerhebliche Breite und umfasst dementsprechend viele Knoten. Problematisch daran ist, dass es praktisch keine Literatur zu der Frage gibt, wie aus einer Teilmenge von Knoten Punkte berechnet werden können, die dann zu einer Polygondarstellung verdichtet werden können. Dennoch gehen alle im folgenden Abschnitt beschriebenen Verfahren zur Kantenaggregation beziehungsweise -approximation davon aus, dass bereits Punkte bekannt sind, die nur noch zusammengefasst werden müssen.

Das in der vorliegenden Arbeit beschriebene Verfahren zur Kantendetektion ist nach Wissen des Autors somit das einzige, das direkt verwertbare Punkte als Ergebnis der Kantendetektion liefert.

### 6.1.3. Verfahren zur Konturapproximation

Die im vorangegangenen Abschnitt beschriebenen Bemühungen zur Kantendetektion werden von einer Reihe von Vorschlägen begleitet, wie sich eine Menge von verteilt in einer Vielzahl von Sensorknoten vorliegenden Punkten zu einem Polygonzug zusammensetzen lassen. Leider setzen alle von ihnen voraus, dass im Sensornetz eine hierarchische Kommunikationsstruktur bereits vorliegt. Das hat zur Folge, dass alle vorgeschlagenen Verfahren zum Beispiel ohne eine Clusterung der Knoten nicht oder nur noch ineffizient arbeiten.

Die Autoren von [132, 133] stellen ein Verfahren eher theoretischer Natur vor. Das Sensornetz wird durch zwei Schnitte in vier Teile geteilt, die dann rekursiv wiederum geteilt werden, bis sich die gewünschte Kachelgröße ergibt. Während dieser Teilung wird auch ein Routing-Quadtree, das heißt ein Baum, bei dem jeder Knoten vier Kinder hat, zur Kommunikation aufgebaut. Aus den Kachelkanten zwischen innerhalb und außerhalb der Phänomenregion liegenden Knoten wird dann sukzessive die Phänomenkontur entlang des Routingbaumes zusammengesetzt. Dieses Verfahren weist zwei Nachteile auf: Einerseits kann die Kontur nur aus treppenartigen Teilstücken bestehen, beispielsweise diagonale Teilkonturen sind nicht möglich. Andererseits werden die Kommunikationsverbindungen im Baum lediglich nach der Kachelung ausgewählt, so dass die Gefahr groß ist, dass die Links aufgrund der volatilen Natur der Kommunikation sehr instabil sind.

Ein weiterer hierarchischer Ansatz wird in [23] vorgestellt. Die Autoren schlagen zwei Verfahren zur Berechnung der Kontur vor: ein verteiltes Wavelet-Verfahren sowie den deutlich komplexeren Adaptive Group Merge Algorithmus. Letzterer überbietet ersteren deutlich hinsichtlich der Approximationsqualität der tatsächlichen Phänomenkontur. Neben der hierarchischen Kommunikationsstruktur treffen die Autoren für beide Verfahren eine weitere, schwer zu realisierende Annahme. Zu Beginn des Verfahrens wird die Kontur durch eine Sequenz von Sensorknoten repräsentiert. Dabei ist es erforderlich, dass jeder der Knoten seinen Vorgänger und Nachfolger zu seinen direkten Nachbarn zählt und eine bidirektionale Kommunikation möglich ist. Ist lediglich eine Multihop-Kommunikation zwischen Knoten in der Sequenz möglich oder sind die Links unidirektional, scheitert das Verfahren. Somit verhindern schon kleine Löcher in der Randstruktur des Phänomens eine erfolgreiche Konturapproximation.

Auch das in [58] vorgeschlagene Verfahren benötigt eine hierarchische Kommunikationsstruktur. Anders als die zuvor beschriebenen Ansätze gehen die Autoren hier jedoch davon aus, dass die einzelnen Sensorknoten nicht nur lokale physikalische Eigenschaften wie beispielsweise die lokale Temperatur erfassen können, sondern vielmehr Eigenschaften in einer gewissen Entfernung messen, wie zum Beispiel ein Radar es kann. Ihr Ansatz beruht nun darauf, dass die Sensorknoten Entfernung und Richtung der Grenze des zu beobachtenden Phänomens bestimmen können. Diese Informationen werden dann zu der Kontur zusammengefasst, so dass dieses Verfahren nur bedingt zu der hier thematisierten Aufgabenstellung passt.

Obwohl sich auch das Verfahren in [158] mit der Konturapproximation in Sensornetzen befasst, soll es hier nicht näher vorgestellt werden, da es den Einsatz selbständig mobiler Geräte erfordert und somit nicht mit Hilfe der hier im Mittelpunkt stehenden statischen Sensornetze realisierbar ist.

## 6.2. Ein Algorithmus zur Konturermittlung und -approximation

Dieser Abschnitt beschreibt ein neues Verfahren zur Konturermittlung sowie -approximation. Dabei wurde Wert darauf gelegt, dass es speziell auf die Anforderungen und Eigenschaften von Sensornetzen abgestimmt ist und die Schritte von Kantendetektion und der Zusammenführung der daraus resultierenden Punkte zu einem Polygon in einem integrierten Verfahren zusammenfasst.

Zunächst werden die einzelnen Schritte hier grob skizziert und drei zugrunde liegende Annahmen beschrieben, um die einzelnen Teile dann in den folgenden Unterabschnitten detailliert zu beschreiben.

Das vorgeschlagene Verfahren lässt sich in eine Folge von sechs Aktionen unterteilen, die jedoch natürlich miteinander verzahnt sind und eher der logischen Gliederung dienen:

1. Einzelmessung: Zunächst führt jeder Sensorknoten eine lokale Messung von physikalischen Eigenschaften durch. Die Messergebnisse werden dann einem Prädikat zugeführt, mit dessen Hilfe der Knoten entscheidet, ob er an einer Stelle liegt, an der das zu detektierende Phänomen vorherrscht, und er somit innerhalb der Phänomenregion liegt.
2. Voting: Die lokalen Prädikatwerte werden dann per Broadcast in die Nachbarschaft kommuniziert und einer Mehrheitsentscheidung zugeführt, um so die Robustheit gegen lokale Messfehler zu steigern. Anhand der Prädikatwerte seiner Nachbarn entscheidet jeder Knoten, ob er innerhalb des Phänomens liegt.
3. Kantendetektion: Indem die Sensorknoten die Ergebnisse des Votings austauschen, können innen liegende Knoten Kantepunkte zwischen sich und außen liegenden Nachbarn berechnen.
4. Planarisierung: In diesem Schritt werden solche Kommunikationsverbindungen identifiziert, die nicht geeignet sind im nächsten Schritt zur Weiterleitung von Nachrichten im Rahmen der Approximation verwendet zu werden.
5. Approximation: Indem eine Nachricht entlang der Phänomenkontur weitergereicht wird, werden sukzessive alle Punkte der Kontur zu einem Polygon zusammengefasst. Dabei werden die einzelnen Punkte anhand ihrer Signifikanz für die Gesamtform bewertet und gegebenenfalls ausgesondert, so dass eine kompakte Darstellung entsteht.

6. Distribution: Abschließend wird die approximierte Kontur im Netz verteilt oder an eine Datensinke weitergeleitet, abhängig davon, wie das ermittelte Polygon genutzt werden soll. Dieser Schritt ist nicht im eigentlichen Sinne Bestandteil des Verfahrens und wird hier nur der Vollständigkeit halber genannt.

Die Schritte 1, 2 und 4 laufen auf allen Knoten im Sensornetz parallel ab, während an den Schritte 3 und 5 nur Knoten am Rand des Phänomens teilnehmen. Die Planarisierung findet dabei ebenfalls parallel statt, während die Approximation sukzessive abläuft.

Wie bereits erwähnt, war ein wichtiges Ziel beim Entwurf des hier vorgestellten Verfahrens, möglichst wenige Annahmen zu treffen, die Bedingung für das Funktionieren der Konturapproximation sind. Daher müssen lediglich drei einfache Voraussetzungen gegeben sein:

1. Wie auch bei allen anderen in Abschnitt 6.1.3 vorgestellten Ansätzen ist es erforderlich, dass alle Sensorknoten lokationsbewusst sind, das heißt sie müssen ihre Position kennen. Dies ist für Sensornetze eine gängige Annahme, da praktisch alle Anwendungen ohnehin Lokationsbewusstsein voraussetzen. Um dieses Ziel zu erreichen, kann eines der in Abschnitt 3.2 beschriebenen Lokationsverfahren zum Einsatz kommen. Gelingt es einzelnen Sensorknoten nicht, ihre Position im Rahmen der Lokation zu bestimmen, können diese von der Teilnahme an der Konturapproximation ausgenommen werden.
2. Es wird weiterhin davon ausgegangen, dass die Sensorknoten zumindest über ein grobgranulares gemeinsames Zeitbewusstsein verfügen. Auf diese Weise können sie zum Beispiel einzelne Phasen des Verfahren in etwa gleichzeitig beginnen.
3. Abschließend ist es erforderlich, dass die Sensorknoten anhand von Messungen bestimmen können, ob sie innerhalb der gesuchten Phänomenausdehnung liegen oder nicht. Allerdings kann eine gewisse Quote von Fehlern bei dieser Messung problemlos toleriert werden.

Nachdem jetzt das Verfahren grob umrissen ist und die Annahmen skizziert wurden, bieten die beiden nun folgenden Abschnitte eine detaillierte Beschreibung der Funktionsweise.

### 6.2.1. Einzelmessung

Wenn die Kontur eines Phänomens in einem Sensornetz ermittelt werden soll, beginnen zunächst alle Sensorknoten mit dem Schritt der Messung: die Knoten entscheiden so zunächst lokal, ob sie sich innerhalb oder außerhalb der Region befinden, innerhalb derer das Phänomen vorherrscht. Dabei sind die Messwerte die Eingangsgrößen eines binären Prädikates, das diese Werte auf 1 (=innen) oder 0 (=außen) abbildet. Dieser Prozess kann beispielsweise von einer Daten-

senke initiiert werden, oder er läuft periodisch ab.

Das zu Beginn dieses Kapitels erwähnte Waldbrandszenario wieder aufgreifend sei hier beispielhaft angenommen, dass die Kontur derjenigen Teilfläche innerhalb des Sensornetzwerkes näherungsweise bestimmt werden soll, innerhalb der eine Temperatur von mindestens  $40^{\circ}\text{C}$  herrscht. In diesem Fall sei jeder Sensorknoten mit einem Temperatursensor ausgestattet. Misst ein Knoten zum Beispiel eine Temperatur von  $50^{\circ}\text{C}$ , so kann er daraus ableiten, dass er sich innerhalb der Phänomenkontur befindet.

### 6.2.2. Voting

Da Messwerte jedoch aufgrund von Sensorungenauigkeiten, Rauschen oder lokalen Temperaturschwankungen verfälscht werden können, wird die lokale Auswertung des Prädikates bei einem Teil der Knoten allerdings wahrscheinlich ein falsches Resultat ergeben.

Um die Robustheit gegenüber solchen falschen Entscheidungen zu erhöhen, wird die abschließende Entscheidung über die Phänomenzugehörigkeit nicht ausschließlich aufgrund dieser lokalen Auswertung getroffen. Vielmehr ist sie Ergebnis eines weiteren Schrittes, des Votings innerhalb der Nachbarschaft der Sensorknoten. Jedes der Geräte broadcastet seinen Prädikatwert zusammen mit seiner ID in seine  $n$ -Hop-Nachbarschaft, wobei  $n$  ein einstellbarer Parameter ist. In Abschnitt 6.3.2 werden verschiedene Werte für  $n$  in Verbindung mit unterschiedlichen lokalen Messfehlerraten auf die Qualität der resultierenden Ergebnisse untersucht. Als Resultat wird jeder Sensorknoten die lokalen Prädikate seiner  $n$ -Hop-Nachbarschaft empfangen. Diese werden über ein Zeitintervall  $t_{vote}$  hinweg gesammelt.

Jedes Gerät entscheidet nun, dass es innerhalb der Phänomenregion liegt, wenn mindestens ein bestimmter Bruchteil  $f$  seiner Nachbarn (einschließlich dem Knoten selbst) aufgrund der lokalen Auswertung zu dem Schluss kam, dass er innerhalb der Region liegen müsse. Dabei trägt die lokale Einschätzung eines Knoten lediglich eine der Stimmen zum Votingprozess bei.

Unter der Annahme einer in etwa gleichmäßigen Knotenverteilung scheint eine sinnvolle Wahl für  $f$  der Wert von 0,5 zu sein. Dieses ist leicht geometrisch zu motivieren: Geht man von einer abschnittsweise geraden Kante aus, liegt ein Gerät im Mittel dann gerade noch innerhalb der Region, wenn die Hälfte seiner Nachbarn innerhalb liegt.

Allerdings ist mit diesem Schritt noch nicht klar, ob das Voting der benachbarten Knoten zu der Entscheidung führte, dass diese innerhalb oder außerhalb des Phänomens liegen. Daher sendet jedes Gerät nach Ablauf von  $t_{vote}$  eine zweite Broadcastnachricht (wiederum über  $n$  Hops) mit seinem Votingergebnis, seiner Position und einer Liste seiner direkten Nachbarn aus. Jeder Knoten wartet ein Zeitintervall  $t_{collect}$  auf diese Nachrichten seiner Nachbarn. Nach Empfang dieser Nachrichten weiß ein Sensorknoten folglich, ob seine Nachbarn innerhalb der

**Algorithmus 5** Voting

---

```

message := (ID, local_predicate);
msg_list :=  $\emptyset$ ;
start timeout  $t_{vote}$ ;
send message to n-hop neighbors;
while timeout not expired do
    collect neighbor messages in msg_list;
end while
count:=1;
inside:=integer(local_predicate);
for all m  $\in$  msg_list do
    if m.local_predicate=1 then
        inside := inside + 1;
    end if
    count:= count + 1;
end for
if inside  $\geq \frac{1}{2}$  * count then
    local_predicate :=1;
else
    local_predicate :=0;
end if
message := (ID, position, local_predicate, neighbor_list);
msg_list :=  $\emptyset$ ;
send message to n-hop neighbors;
start timeout  $t_{collect}$ ;
while timeout not expired do
    collect incoming messages in msg_list;
end while
neighbor_list:= $\emptyset$ ;
for all m  $\in$  msg_list do
    neighbor:= (m.ID, m.position, m.local_predicate, unidirectional);
    if ID  $\in$  m.neighbor_list then
        neighbor.link := bidirectional;
    end if
    neighbor_list.add(neighbor);
end for

```

---

Phänomenfläche liegen und vermerkt dieses in seiner lokalen Nachbarschaftsliste. Anhand der jeweils von den Nachbarn empfangenen Nachbarschaftslisten kann darüber hinaus entschieden werden, ob die Kommunikationsverbindungen zu den Nachbarn unidirektional oder bidirektional sind. Da zu einem späteren Zeitpunkt Acknowledgements zur Bestätigung des Empfangs von versendeten Paketen eingesetzt werden sollen, werden die Einträge in der lokalen Liste mit Nachbarn entsprechend markiert. Danach können die empfangenen Nachbarschaftslisten verworfen werden.

Eine formalisierte Beschreibung des Votings wird in Algorithmus 5 angegeben.

### 6.2.3. Kantendetektion

Nach Ablauf von  $t_{collect}$  prüfen alle Sensorknoten, deren Voting mit dem Ergebnis endete, dass sie sich innerhalb des Phänomens befinden, ob sich unter ihren direkten Nachbarn Knoten mit gegenteiligem Ergebnis finden. Besitzt ein innen liegender Knoten Nachbarn außerhalb, berechnet er einen Konturpunkt (oder auch *Vertex*). Dessen Position ergibt sich als Mittelwert der eigenen Position und der des nächsten äußeren Nachbarn. Eine formalisierte Beschreibung der Kantendetektion wird in Algorithmus 6 angegeben.

---

#### Algorithmus 6 Kantendetektion

---

```

if local_predicate=1 then
    distance:=∞;
    vertex:=INVALID;
    for all n ∈ neighbor_list do
        p := n.position;
        d :=  $\sqrt{(p.x-position.x)^2 + (p.y-position.y)^2 + (p.z-position.z)^2}$ ;
        if d < distance and m.predicate=0 then
            distance := d;
            vertex := ( $\frac{position.x+p.x}{2}$ ,  $\frac{position.y+p.y}{2}$ ,  $\frac{position.z+p.z}{2}$ );
        end if
    end for
end if

```

---

Im weiteren Verlauf der Beschreibung des Verfahrens werden Knoten, die einen Konturpunkt berechnet haben, der Einfachheit halber *Kantenknoten* genannt.

Es sei darauf hingewiesen, dass diese Berechnung von Konturpunkten nicht nur auf Basis aktueller Messwerte erfolgen kann, sondern theoretisch auch historische Daten zur Berechnung von bisherigen Phänomenkonturen herangezogen werden könnten.

### 6.2.4. Netzwerkplanarisierung

Die im bisherigen Verlauf angelegte Nachbarschaftsliste dient nicht nur der Kantendetektion. Sie wird auch zu einem die Konturapproximation vorbereitenden Schritt verwendet, der Netzwerkplanarisierung. Begreift man das Sensornetz als einen Graphen aus Sensorknoten mit bidirektionalen Kommunikationsverbindungen als Kanten, so entfernt die Planarisierung einzelne Graphenkanten so, dass sich keine Kanten kreuzen. Dieser Schritt ist notwendig, da das spätere Herumreichen einer Nachricht zur Polygonkonstruktion ein planares Netz erfordert.

Es gibt eine ganze Reihe von Algorithmen zur Graphenplanarisierung. Hier kommt die Transformation zu einem Gabriel-Graphen [70] zum Einsatz, da sie rein lokal durchgeführt werden kann und einen planaren Graphen garantiert.

Aus Sicht eines Sensorknotens handelt es sich bei der Planarisierung um das Markieren und Entfernen von Einträgen innerhalb seiner Nachbarschaftsliste.

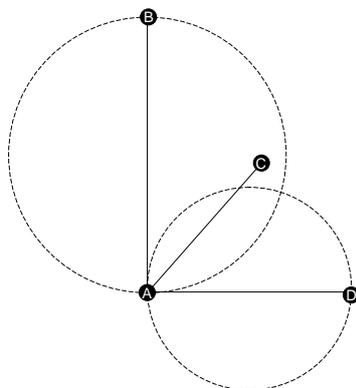


Abbildung 6.2.: Beispiel der Netzwerkplanarisierung.

Zunächst werden alle diejenigen Nachbarn aus der Liste entfernt, mit denen nur unidirektionale Kommunikation möglich ist. Dieses hat keinen direkten Zusammenhang mit der Planarisierung, sondern hat zur Folge, dass der Graph hinterher nur bidirektionale Kanten enthält. Mit den verbleibenden Einträgen wird wie folgt vorgegangen: Ein Nachbar ist nur dann auch Teil des planaren Netzes und wird als solcher markiert, wenn kein anderer Nachbar in dem Kreis liegt, der exakt zwischen den Knoten und den betrachteten Nachbarn passt. Dies wird durch Abbildung 6.2 verdeutlicht, die den Sensorknoten A und drei seiner Nachbarn zeigt. Während D Teil des planaren Netzes ist, weil kein anderer Nachbar von A in dem Kreis zwischen A und D liegt, fällt B wegen C im Kreis zwischen A und B heraus.

Nachdem der Vorgang für alle Listeneinträge durchgeführt wurde, können alle nicht markierten Einträge entfernt werden. Übrig bleiben genau diejenigen Nachbarn, die das planare, bidirektionale Netzwerk bilden.

Diese Vorgehensweise wird in Algorithmus 7 noch einmal formalisiert dargestellt.

### 6.2.5. Konturapproximation

Nach dieser Vorbereitung, die alle Knoten parallel zueinander nach Ablauf von  $t_{collect}$  und der eventuellen Berechnung von Konturpunkten durchführen, kann mit der Konstruktion des Polygons begonnen werden. Zu diesem Zweck soll der Kantenknoten mit der kleinsten ID beginnen, die entsprechende Polygonnachricht entlang des Randes der Region zu verschicken.

Dazu starten alle Kantenknoten einen Timer der Länge  $ID * c$ , wobei  $c$  eine Konstante zur Streckung des Zeitintervalls ist. Mit Hilfe der Konstanten kann das Intervall an die Zeit angepasst werden, die die Nachricht zum Umlauf der Phänomenregion benötigt. Läuft der Timer eines Knotens ab, bevor dieser eine Polygonnachricht erhalten hat, so sendet er eine entsprechende Nachricht mit seiner ID und seinem eigenen Konturpunkt an seinen im Uhrzeigersinn nächsten inneren Nachbarn. Dieser fügt der Nachricht wiederum seinen Konturpunkt an

---

**Algorithmus 7** Planarisierung

---

```

for all n ∈ neighbor_list do
  if n.link != bidirectional then
    neighbor_list.remove(n);
  end if
end for
for all n ∈ neighbor_list do
  p1 := n.position;
  center := ( (position.x+p1.x) / 2, (position.y+p1.y) / 2, (position.z+p.z) / 2 );
  r := √((center.x-position.x)2 + (center.y-position.y)2 + (center.z-position.z)2);
  for all n2 ∈ neighbor_list where n1 ≠ n2 do
    p2 := n2.position;
    r' := √((center.x-p2.x)2 + (center.y-p2.y)2 + (center.z-p2.z)2);
    if r' ≤ r then
      n.link := remove;
    end if
  end for
end for
for all n ∈ neighbor_list do
  if n.link = remove then
    neighbor_list.remove(n);
  end if
end for

```

---

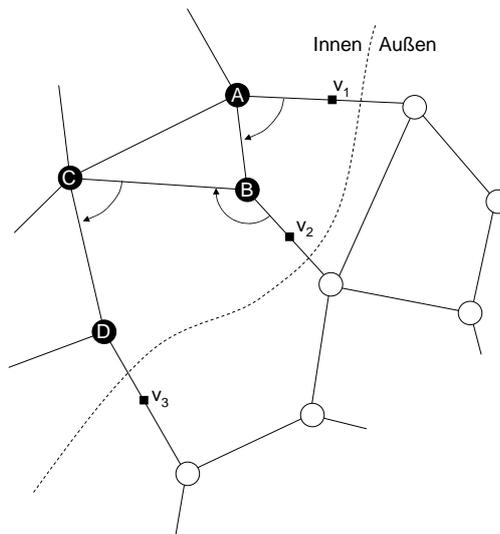


Abbildung 6.3.: Beispiel für die Weiterleitung der Polygongnachricht.

und leitet sie weiter.

Zur Verdeutlichung der Auswahl des jeweils nächsten Knotens sei das Beispiel in Abbildung 6.3 herangezogen: innen liegende Knoten sind als schwarze Kreise dargestellt, außen liegende als weiße. Sei A der Kantenknoten mit der kleinsten ID. Er erstellt eine Polygongnachricht mit einem Konturpunkt  $v_1$ . Dann sucht A seinen nächsten inneren Nachbarn, der – von einer Linie ( $pos_A, v_1$ ) aus gesehen – rechts liegt (wie durch den Pfeil angedeutet). Dieses Vorgehen ist

als Rechte-Hand-Regel bekannt und wird zum Beispiel vom Greedy Perimeter Stateless Routing (GPSR) [94] genutzt, um Nachrichten um Löcher in der Netzwerktopologie herumzurouten.

A leitet nun die Nachricht an B weiter, der  $v_2$  anfügt. Dann sucht B, ausgehend von seinem Konturpunkt, nach dem nächsten Nachbarn im Uhrzeigersinn und leitet die Nachricht an C weiter. Da C keinen Konturpunkt kennt und somit kein Kantenknoten ist, leitet er die Nachricht lediglich weiter. Dazu sucht C (statt von einem Konturpunkt aus) von seinem Vorgänger B aus nach dem nächsten Empfänger. Die Nachricht wird an D weitergeleitet und so fort.

Da die Polygonnachricht nicht verloren gehen darf, werden die Übertragungen mit Acknowledgements gesichert. Alternativ können auch passive Bestätigungen wie in [163] beschrieben genutzt werden. Ist die Übertragung zum nächsten Knoten auch nach mehreren Versuchen nicht geglückt, wird dieser Knoten als zumindest zeitweise ausgefallen angesehen und der nächste Nachbar im Uhrzeigersinn zur Weiterleitung ausgewählt.

Trotz des Konzeptes der Timer unterschiedlicher Länge auf Basis der Knoten-ID kann es passieren, dass mehrere Nachrichten um das Phänomen kreisen. Empfängt ein Knoten eine zweite Polygonnachricht, vergleicht er die ID der Knoten miteinander, die die Nachricht ursprünglich auf den Weg gebracht haben. Ist die ID der zweiten Nachricht kleiner als die der ersten, wird diese Nachricht weitergeleitet, ansonsten wird sie verworfen. Auf diese Weise wird sichergestellt, dass nur eine Nachricht das Phänomen komplett umrundet. Trifft diese Nachricht wieder bei ihrem ursprünglichen Absender ein, ist die Polygonkonstruktion vollendet.

Wie bereits erwähnt, kann das Polygon jedoch unter Umständen durch sehr viele Punkte beschrieben werden, so dass die Nachricht durch das sukzessive Anfügen von Konturpunkten ebenfalls stark anwachsen kann. Dieses hat verschiedene Nachteile: Zum einen ist die Länge von Nachrichten stets technisch begrenzt, so dass unter Umständen mehrere Nachrichten verschickt werden müssten. Zum anderen überfordern Polygodarstellungen mit mehr als einhundert Punkten leicht die Speicherkapazität der Sensorknoten.

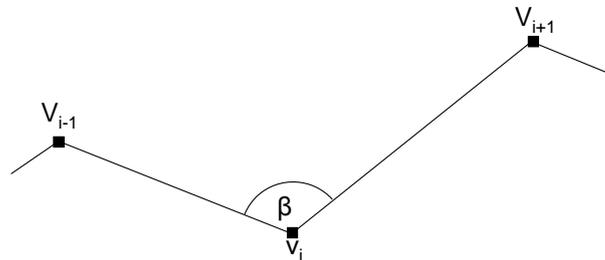


Abbildung 6.4.: Beurteilung der Signifikanz einzelner Polygonpunkte.

Daher ist es erforderlich, die Anzahl der Polygonpunkte bereits im Verlauf der Polygonkonstruktion zu reduzieren. Somit wurde ein Mechanismus vorgesehen, der es ermöglicht eine maximale Anzahl von Punkten anzugeben, die das kon-

struierte Polygon enthalten darf. Bis zu dieser Anzahl werden sukzessive Punkte angefügt. Wird die Grenze dann erreicht, wird der am wenigsten signifikante Punkt aus dem Polygon entfernt. Das Kriterium zur Signifikanzbewertung der Punkte basiert auf der Länge der beiden adjazenten Kanten des Punktes sowie auf deren Winkel (vergleiche Abbildung 6.4) und ist [103] mit leichten Variationen entnommen:

$$\sigma(v_i) = \frac{(180^\circ - \beta) |\overline{v_{i-1}v_i}| |\overline{v_iv_{i+1}}|}{|\overline{v_{i-1}v_i}| + |\overline{v_iv_{i+1}}|} \quad (6.1)$$

Die Berücksichtigung der Kantenlängen im Produkt des Zählers steht dabei im Wesentlichen für die Flächenänderung im Falle des Entfalls des Punktes. Durch die Winkeldifferenz wird zusätzlich sichergestellt, dass spitze Ecken am Polygon weniger leicht entfallen als stumpfe. Je flacher der Winkel ist, desto mehr nähert sich  $\beta$  dem Winkel von  $180^\circ$  an, und die Differenz wird klein. Ein spitzer Winkel nahe  $0^\circ$  hingegen sorgt für einen großen Differenzwert und steigert folglich die Signifikanz  $\sigma$  insgesamt.

Mit Hilfe dieses Kriteriums kann der Speicherbedarf für das Verfahren ebenso wie die Nachrichtengröße begrenzt werden, indem eine maximale Anzahl von Polygonpunkten spezifiziert wird. So kann insbesondere eine Abwägung zwischen voraussichtlich erreichter Genauigkeit und dem Aufwand hinsichtlich Speicher und Kommunikation getroffen werden. Es ist jedoch ebenso möglich, auf eine Approximation zu verzichten und alle Punkte in das Polygon aufzunehmen. Die Auswirkung der Begrenzung der zugelassenen Polygonpunkte wird in Abschnitt 6.3.5 detailliert untersucht.

Kommt die Polygongnachricht wieder bei ihrem Initiator an, ist die Kantenapproximation komplett. Das resultierende Polygon kann nun innerhalb oder in der Nähe der Region, oder im gesamten Netz verteilt werden. Alternativ ist aber auch das Weiterleiten an eine Datensenke möglich, zum Beispiel um dort einer Leitstelle zur Kenntnis zu gelangen.

### 6.2.6. Illustration anhand eines Beispiels

Im Folgenden wird das Verfahren anhand eines Beispiels veranschaulicht. Dazu dient das in Abbildung 6.5 gezeigte Szenario. Es wird davon ausgegangen, dass innerhalb der als gestricheltes Rechteck gekennzeichneten Region eine Temperatur von  $50^\circ\text{C}$  und außerhalb nur eine Temperatur von  $20^\circ\text{C}$  herrscht. Die als Kreise dargestellten Sensorknoten sind mit Temperatursensoren ausgestattet. Die Ziffern geben die Geräteadressen an, Kommunikationslinks werden durch die grauen Linien symbolisiert. Im Beispiel wird davon ausgegangen, dass alle Links bidirektional sind.

Die Sensorknoten beginnen parallel mit der Konturapproximation. Dieses kann beispielsweise durch einen Koordinationsschritt ausgelöst werden, der nicht Bestandteil des Verfahrens selbst ist.

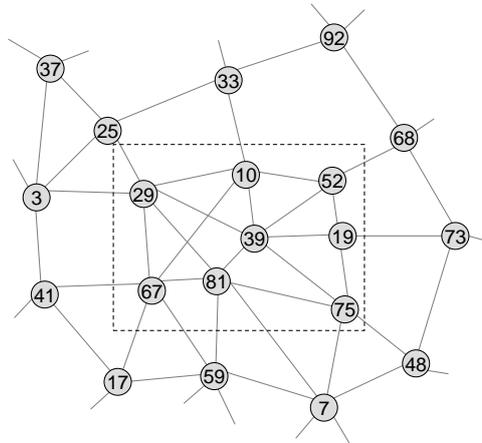


Abbildung 6.5.: Beispielszenario.

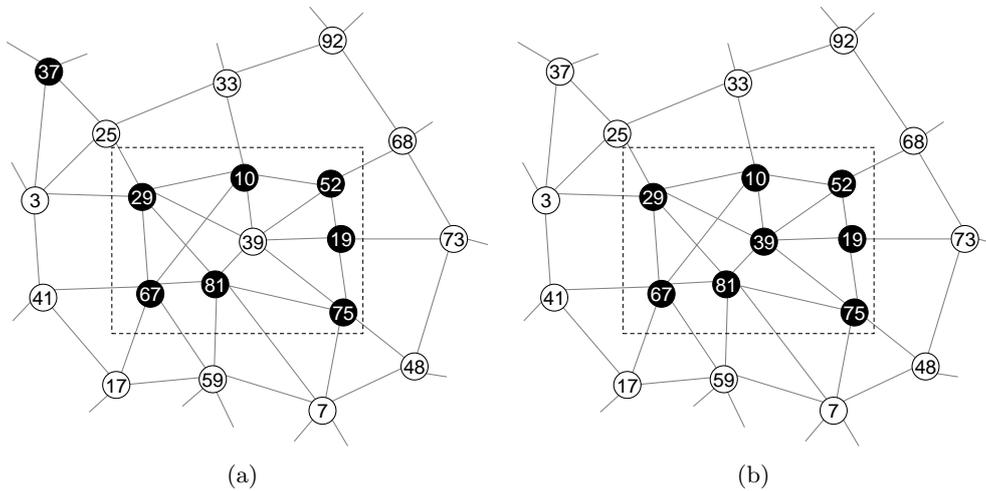


Abbildung 6.6.: Einzelmessung und Voting.

Zunächst nehmen alle Knoten eine Temperaturmessung vor und verwenden den resultierenden Messwert als Eingangsdatum für ein unäres Prädikat. Als Kriterium für die Regionszugehörigkeit sei eine Temperatur oberhalb von  $40^\circ\text{C}$  angenommen, das entsprechende Prädikat  $p : \mathbb{N} \rightarrow \{0, 1\}$  lautet

$$p(temp) = temp \geq 40. \tag{6.2}$$

Alle Knoten innerhalb des Rechteckes messen eine Temperatur von  $50^\circ\text{C}$ , die einzige Ausnahme bildet Knoten 39, der beispielsweise aufgrund eines Messfehlers nur  $30^\circ\text{C}$  ermittelt. Die Knoten außerhalb des Rechteckes messen  $20^\circ\text{C}$ , hier bildet jedoch der Knoten 37 eine Ausnahme ermittelt eine Temperatur von  $45^\circ\text{C}$ . Die aus diesem ersten Verfahrensschritt resultierende Situation ist in Abbildung 6.6(a) dargestellt. Knoten mit einem Prädikatwert von 1 sind schwarz gefärbt, die übrigen weiß. Es ist zu erkennen, dass zwei der Sensor-

knoten aufgrund lokaler Messfehler zu einem falschen Ergebnis hinsichtlich der Regionszugehörigkeit gelangt sind.

Um solche Fehler zu konsolidieren, folgt das Voting, bei dem sich die Sensorknoten mit ihren Nachbarn abstimmen. In diesem Beispiel findet das Voting nur innerhalb der direkten Nachbarschaft statt, das heißt mit einem Hopcount von  $n = 1$ . Die Knoten starten einen Timeout der Länge  $t_{vote}$  und broadcasten ein Datenpaket mit ihrer Adresse und ihrem lokalen Prädikatwert. Bis zum Ablauf des Timeouts sammeln die Geräte die empfangenen Adressen und Prädikatwerte ihrer Nachbarn in einer Liste. Dann passen die Knoten 37 und 39 das Ergebnis des lokalen Prädikats an, da mindestens die Hälfte ihrer Nachbarn (einschließlich ihnen selbst) zu einem gegenteiligen Ergebnis kam. Das Votingresultat ist in Abbildung 6.6(b) dargestellt. Im Anschluss broadcasten alle Sensorknoten ein Datenpaket mit ihrem endgültigen Prädikat, ihrer Position und der Liste ihrer Nachbarn und starten einen Timeout der Länge  $t_{collect}$ , während dessen sie die von ihren Nachbarn empfangenen Nachbarlisten- und Prädikate und Positionen sammeln.

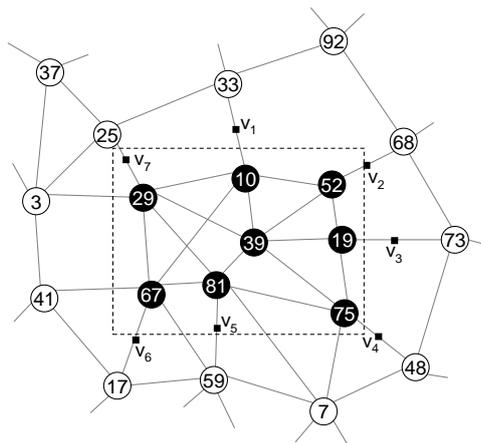


Abbildung 6.7.: Kantendetektion.

Nach Ablauf des Timeouts beginnt die Phase der Kantendetektion. Anhand der empfangenen Datenpakete kann jeder Knoten in der Region (jeder Knoten mit positivem endgültigen Prädikat) prüfen, ob er Nachbarn außerhalb der Region (also mit negativem Prädikat) besitzt. Ist das der Fall, wird der am nächsten liegende unter ihnen ermittelt und ein entsprechender Konturpunkt  $v$  berechnet (siehe Abbildung 6.7, die Konturpunkte sind als schwarze Quadrate dargestellt).

Im Anschluss folgt die Netzplanarisierung. Die Vorgehensweise wird in Abbildung 6.8(a) anhand von Knoten 67 dargestellt. Mittels der Nachbarschaftsliste und den darin enthaltenen Positionen wird für jeden Nachbarn geprüft, ob in dem Kreis zwischen Knoten und Nachbarn weitere Nachbarn liegen. Die entsprechenden Kreise sind in Abbildung 6.8(a) dargestellt, sie sind mit Ausnahme des Kreises zwischen Knoten 67 und Knoten 10 leer. Folglich entfernt Knoten 67 nur Knoten 10 aus seiner Nachbarschaftsliste.

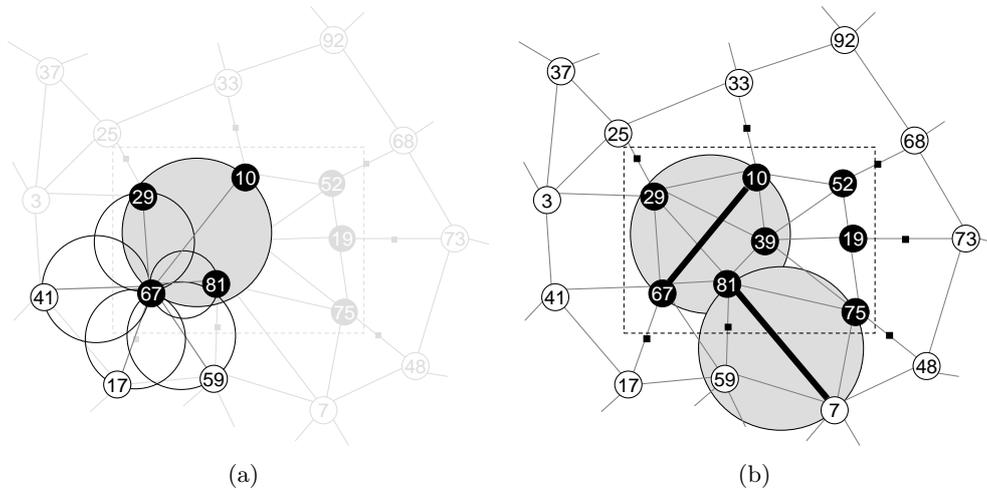


Abbildung 6.8.: Netzplanarisierung.

Alle anderen Knoten gehen ebenso vor, dabei wird Knoten 67 aus der Liste von 10 gelöscht, und auch die Knoten 81 und 7 entfernen sich jeweils gegenseitig aus ihren Listen. In Abbildung 6.8(b) sind die entfernten Links als dicke Linien gekennzeichnet und auch die entsprechenden Kreise dargestellt. Die resultierende Netzstruktur ist beispielsweise in Abbildung 6.9(a) erkennbar.

Es folgt die Konturapproximation. Als Obergrenze für die Anzahl der im Approximationspolygon zugelassenen Konturpunkte sei 5 angenommen. Um den Initiator der Approximation zu bestimmen, starten alle Sensorknoten, die einen Konturpunkt berechnet haben, einen Timeout. Seine Länge hängt von ihrer Adresse ab. Da Knoten 10 die kleinste Adresse der entsprechenden Knoten hat, läuft der Timeout bei ihm zuerst ab. Er ermittelt folglich gemäß der Rechte-Hand-Regel ausgehend von der Richtung von  $v_1$  Knoten 52 als Adressaten und sendet ihm eine Nachricht mit seiner Adresse und seinem Konturpunkt  $v_1$  (Abbildung 6.9(a)). Knoten 52 fügt seinen Konturpunkt  $v_2$  an, wendet ebenfalls die Rechte-Hand-Regel an und leitet das Paket an Knoten 19 weiter (b), der wiederum  $v_3$  anfügt und das Paket an Knoten 75 sendet (c). Dieses Vorgehen setzt sich fort, bis Knoten 67 die Nachricht erhält (Abbildungen 6.9(d) und 6.10(a)).

Dieser kann seinen Konturpunkt  $v_6$  nicht ohne Weiteres anfügen, weil die Nachricht bereits 5 Punkte enthält und damit die Obergrenze erreicht ist. Er wendet daher auf die Konturpunkte  $v_2$  bis  $v_4$  in der Nachricht das in Gleichung 6.1 angegebene Signifikanzkriterium  $\sigma$  an und entfernt mit  $v_3$  den am wenigsten signifikanten Punkt. Daraufhin kann er seinen eigenen Punkt der Nachricht anfügen und leitet sie an Knoten 29 weiter (Abbildung 6.10(b)).

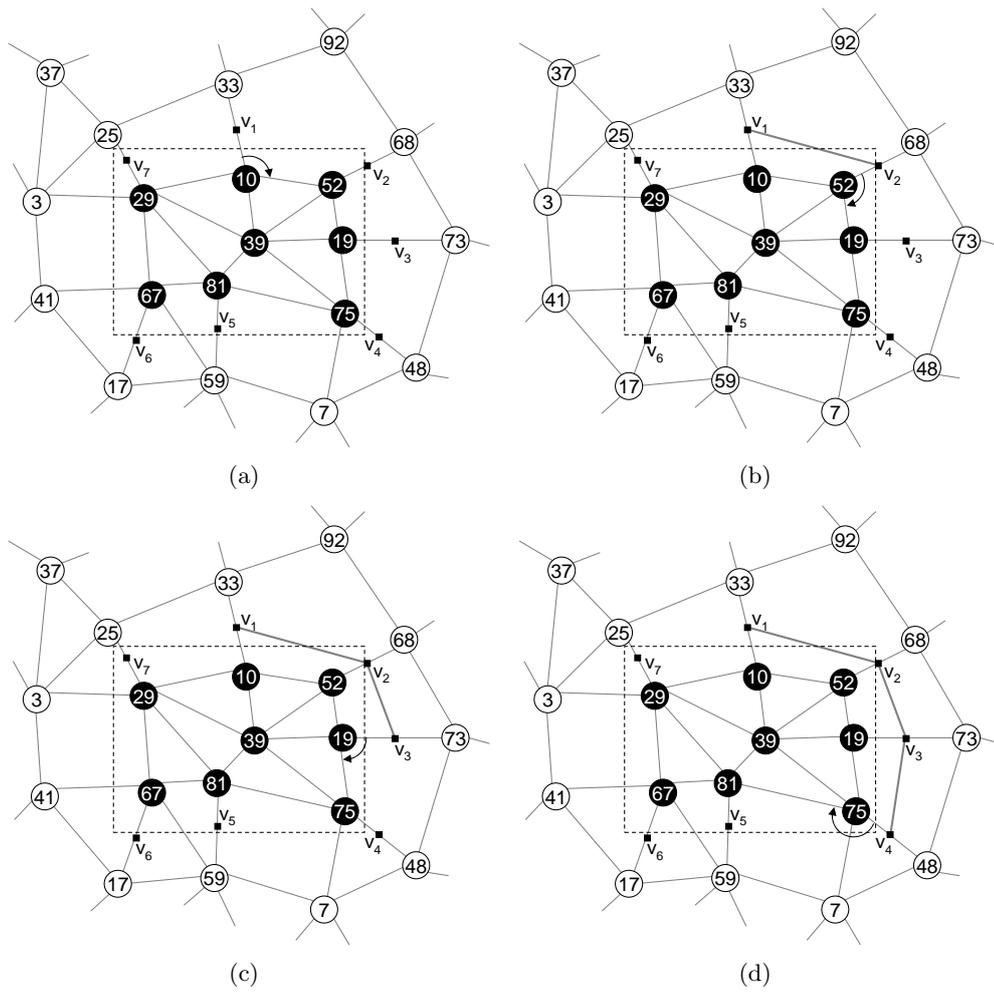


Abbildung 6.9.: Konturapproximation (Teil 1).

6.2. Ein Algorithmus zur Konturermittlung und -approximation

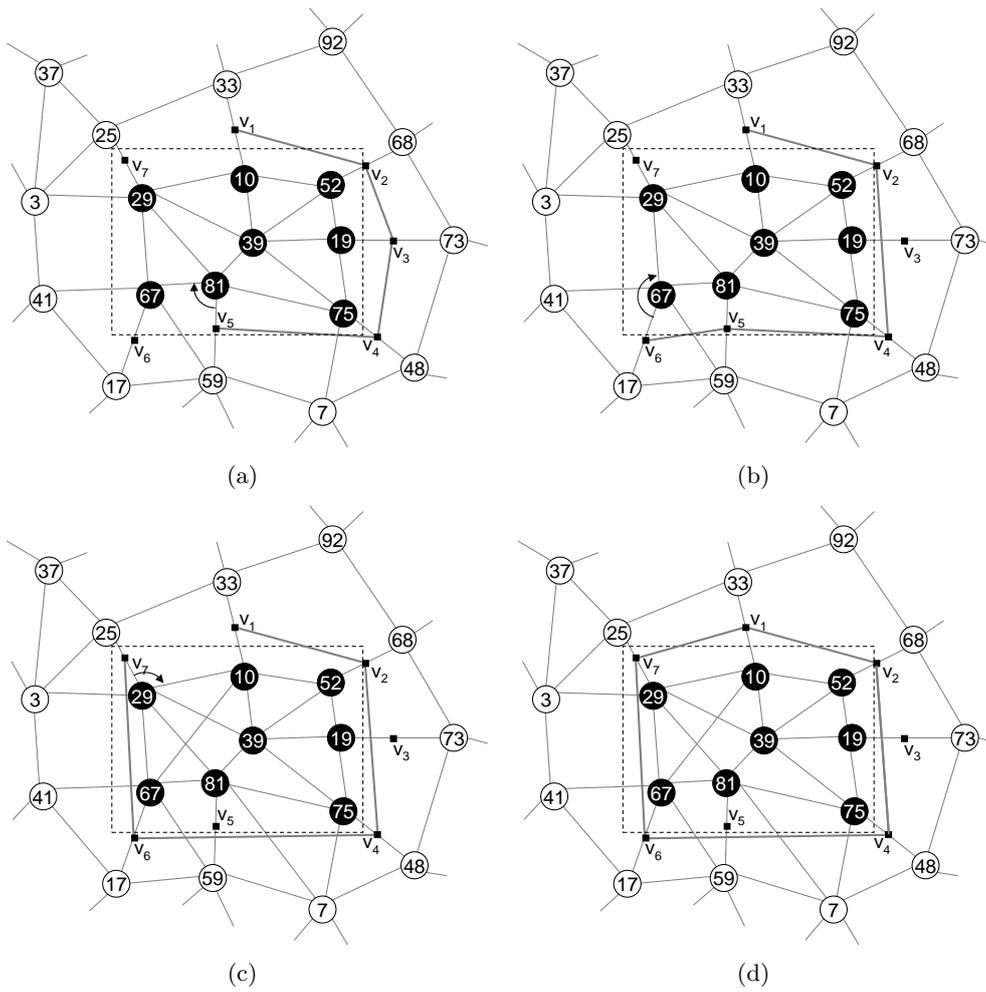


Abbildung 6.10.: Konturapproximation (Teil 2).

Dieser muss ebenfalls den jetzt am wenigsten signifikanten Punkt  $v_5$  entfernen, fügt  $v_7$  an und sendet die Nachricht an Knoten 10 (c). Dieser erkennt an seiner eigenen Adresse in der Nachricht, dass er ihr Urheber ist. Damit ist die Konturapproximation abgeschlossen (d), das Approximationspolygon lautet  $[v_1, v_2, v_4, v_6, v_7]$ .

### 6.3. Simulative Evaluation

Dieser Abschnitt beschreibt die simulative Untersuchung des Verfahrens zur Approximation von Phänomenkonturen in Sensornetzwerken. Dabei wird zunächst auf die Simulationsumgebung eingegangen, um dann die Ergebnisse der einzelnen Untersuchungen vorzustellen. Besonderes Augenmerk bei der Evaluation erfuhren die Robustheit gegenüber Messfehlern und Paketverlusten sowie die Genauigkeit von Konturermittlung und -approximation.

#### 6.3.1. Simulationsumgebung

Für die Simulation des Verfahrens hat sich der Autor entschieden, nicht den gesamten Netzwerkstack zu simulieren. Anstelle dessen werden die Auswirkungen der entsprechenden Prozesse simuliert, die tatsächlich Einfluss auf den Ablauf des Verfahrens haben, also zum Beispiel Paketverluste als Folge instabiler Links. Aus diesem Grund kam der Simulator SHAWN [98] für die Untersuchungen zum Einsatz. Er wurde speziell für die Simulation großer Sensornetze entwickelt, legt den Schwerpunkt eher auf algorithmische Untersuchungen anstelle detailgenauer Simulation des Netzwerkstacks und kann den Austausch von Nachrichten oder das Einlesen von Messwerten extrem effizient durchführen. In Voruntersuchungen wurde belegt, dass die Ergebnisse von SHAWN denen von ns-2 [134] gleichen.

Anstelle des eher unrealistischen Kreismodells zur Beschreibung der Funkschnittstelle wurde das in 2.5 beschriebene RI-Modell auch in SHAWN implementiert. Dieses Funkmodell führt insbesondere zu einem gewissen Anteil unidirektionaler Links. Der mittlere Kommunikationsradius betrug 100 m.

Darüber hinaus wurden in Abhängigkeit vom Untersuchungsgegenstand verschiedene Paketverlustraten vorgegeben. Details können den Beschreibungen der betreffenden Simulationen in Abschnitt 6.3.3 entnommen werden.

In den Simulationen wurde der Einfluss von verfälschten Positionsinformationen, wie sie durch Fehler von Lokationsverfahren entstehen können, ausdrücklich nicht betrachtet. Um die Genauigkeit des Konturapproximationsverfahrens als solchem beurteilen zu können, wurde davon ausgegangen, dass alle Knoten ihre korrekte Position kennen.

Für alle Simulationen kam eine Simulationsfläche von 2500 m x 2500 m zum Einsatz. Auf dieser wurden 2984 Sensorknoten zufällig uniform verteilt, so dass

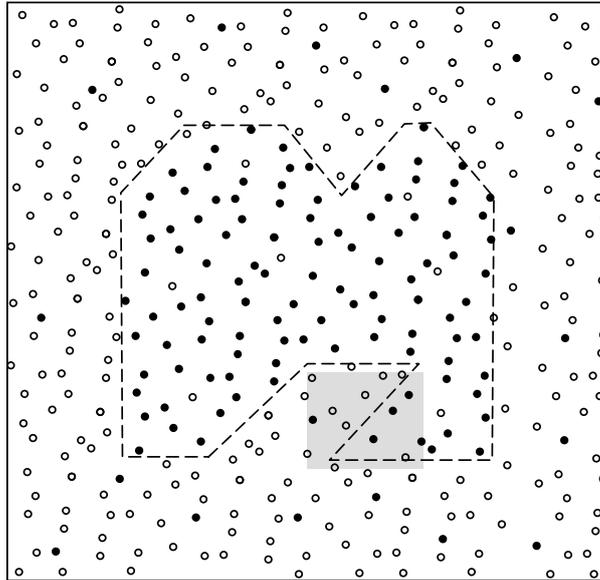


Abbildung 6.11.: Simulationsszenario mit Phänomenkante (gestrichelt) und Position des in Abbildung 6.17 vergrößert dargestellten Ausschnittes (grau).

alle Knoten im Mittel 15 Nachbarn haben.

Auf der Simulationsfläche wurde dann ein Phänomen platziert, das von den Sensorknoten detektiert werden kann. Seine äußere Form ist in Abbildung 6.11 durch eine gestrichelte Linie kenntlich gemacht und wird durch ein Polygon mit 14 Punkten (Anfangs- und Endpunkt tauchen separat auf) beschrieben. Diese spezielle Polygonform wurde gewählt, weil sie nicht konvex ist und spitze, stumpfe sowie rechte Winkel enthält. Auch besitzt sie unten rechts eine Art Halbinsel, die nur über einen relativ schmalen Steg mit dem Rest der Form verbunden ist.

Sensorknoten, deren lokale Einzelmessung ergab, dass sie innerhalb der Form liegen, sind symbolisch schwarz eingezeichnet. Weiß eingetragene Knoten hingegen deuten an, dass diese lokal zu dem Ergebnis kamen, dass sie außerhalb des Phänomens lägen. Einige schwarze Knoten außerhalb der Phänomenregion beziehungsweise weiße innerhalb symbolisieren solche Knoten, die lokal zu einem falschen Ergebnis kamen.

In diesem Szenario wurde das Verfahren zur Kantenapproximation durchgeführt und die Ergebnisse der unterschiedlichen Phasen ausgewertet. Die Resultate werden in den folgenden Abschnitten vorgestellt und diskutiert. Um statistisch aussagekräftige Ergebnisse zu erzielen, wurden jeweils 100 Simulationen mit den gleichen Parametern durchgeführt. Diese unterscheiden sich lediglich durch zufällige Größen wie zum Beispiel die Knotenverteilung. Abschließend wurden die Ergebnisse gemittelt. Auf diese Weise kann sichergestellt werden, dass Ausreißer nicht fälschlicherweise als typisches Ergebnis interpretiert werden.

### 6.3.2. Robustheit gegen Messfehler

Als Erstes wurde der Einfluss von Fehlern bei den Einzelmessungen untersucht. Daher wurden Simulationen mit Fehlerquoten von 0%, 10%, 20% und 30% durchgeführt. Das bedeutet, dass in diesen Simulationsserien kein (beziehungsweise jeder zehnte, fünfte, dritte) Sensorknoten anhand seiner lokalen Messung zu dem Ergebnis kam, dass er innerhalb des Phänomengebietes läge, obwohl er tatsächlich außerhalb lag, beziehungsweise umgekehrt. Im Zentrum des Interesses stand einerseits, ob und inwieweit das Voting diese Fehler kompensieren kann. Andererseits wurde geklärt, welchen Einfluss dabei die Wahl von  $n$ , also der Distanz in Hops, innerhalb der das Voting durchgeführt wird, auf die Qualität der Fehlerkompensation hat.

Um die Einflüsse der Messfehler von denen von Paketverlusten zu isolieren, wurde in diesen Simulationen die Paketverlustrate auf Null eingestellt.

Abbildung 6.12 zeigt die resultierenden Fehlerraten nach dem Voting in Abhängigkeit von der Votingdistanz in Hops für verschiedene Ausgangsfehlerraten. Die Linien stehen für die mittleren Fehlerraten nach dem Voting, darüber hinaus sind die aufgetretenen Minima und Maxima der Fehlerquoten dargestellt. Dabei entspricht eine Hopdistanz von Null dem vollständigen Verzicht auf ein Voting. Somit stellen die entsprechenden Werte ganz links im Diagramm die initialen Fehlerraten vor einem eventuellen Voting zwischen 0% und 30% dar.

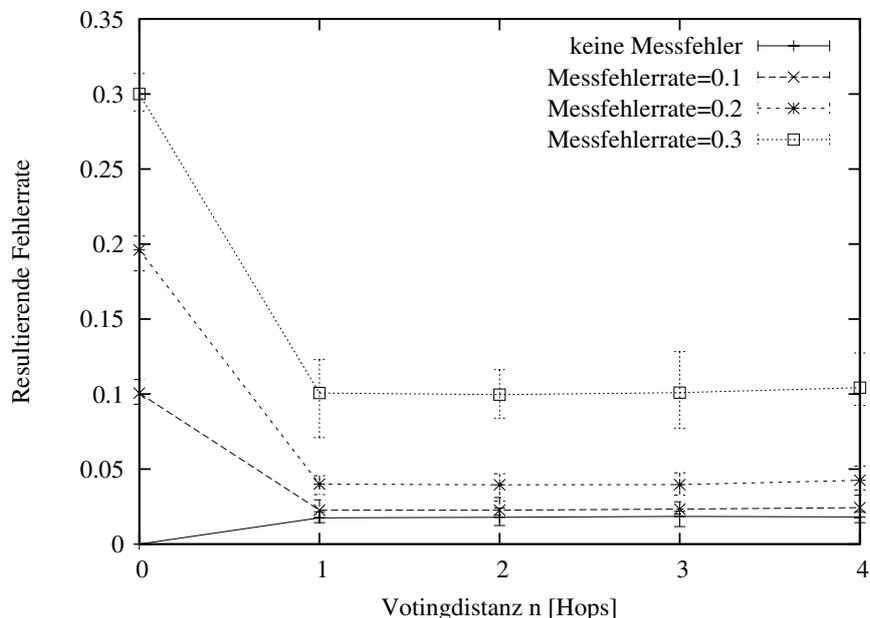


Abbildung 6.12.: Resultierende Fehlerrate nach dem Voting in Abhängigkeit des Hopcounts für initiale Messfehlerraten zwischen 0% und 30%.

Es wird deutlich, dass das Voting die Fehlerrate bereits bei einer Distanz von nur einem Hop drastisch reduziert. Betrug der Anteil der fehlerhaften Messungen vor dem Voting 10%, so sind es mit einem 1-Hop-Voting nur noch 2%.

Fehlerraten von 20% können auf 4% gedrückt werden. Selbst wenn jeder dritte Sensorknoten von einer falschen Messung ausgeht, reduziert sich die Quote auf unter 10%.

Darüber hinaus ist klar erkennbar, dass Multihop-Voting nicht sinnvoll ist. Während die Fehlerquote beim Voting über mehr als einen Hop nicht weiter zurückgeht (beziehungsweise sogar wieder etwas anwächst), entsteht durch die gesteigerte Weiterleitungsdistanz ein erhöhtes Datenaufkommen und somit letztlich ein zunehmender Energieverbrauch. Daher liefert ein Wert von  $n = 1$  für nahezu alle Fälle das beste Ergebnis.

Lediglich in einem Szenario ohne initiale Messfehler wäre es günstiger, auf das Voting zu verzichten. Hier führt dieser Schritt dazu, dass sich nach dem Voting etwa 1,8% der Sensorknoten falsch einordnen. Der Grund dafür ist, dass in einigen Fällen eigentlich korrekte Messungen überstimmt werden. Dies kann beispielsweise passieren, wenn ein Knoten in einer spitzen Ausstülpung des Phänomens liegt. Seine Messung kommt korrekt zu dem Schluss, dass er innerhalb des Phänomens liegt, da jedoch fast alle seiner Nachbarn außerhalb liegen, kommt er nach dem Voting zu einem falschen Ergebnis.

Konsequenter Weise sollte bei der Abwesenheit von Messfehlern auf das Voting verzichtet werden. Der Autor ist jedoch der Auffassung, dass ein solches Szenario ausgesprochen unrealistisch ist. Somit ist unter realen Bedingungen stets das Voting innerhalb der direkten Nachbarschaft die beste Wahl.

### 6.3.3. Robustheit gegen Paketverluste

Als nächster Schritt wurde die Robustheit des Votings gegenüber Paketverlusten untersucht. Da die im vorangegangenen Abschnitt beschriebenen Untersuchungen des Votings bereits gezeigt haben, dass  $n = 1$  die beste Wahl für die Votingdistanz darstellt, wurde diese Einstellung für alle folgenden Simulationsserien beibehalten.

Da das Voting jedoch mittels Broadcast-Kommunikation erfolgt und daher nicht durch Bestätigungen gesichert werden kann, könnten Paketverluste die Funktion des Votings potenziell stören. Daher wurde geprüft, wie Paketverlustraten zwischen 0% und 40% die Fehlerrate nach dem Voting beeinflussen.

Abbildung 6.13 zeigt die resultierenden Fehlerraten nach dem Voting in Abhängigkeit von der Paketverlustrate für verschiedene initiale Messfehlerraten. Wiederum sind die mittleren resultierenden Fehlerraten durch die Linien gekennzeichnet und darüber hinaus Minimal- und Maximalwerte angegeben, die insgesamt in den betreffenden Simulationen auftraten. Angemerkt sei, dass die Angaben für eine Verlustrate von Null den Werten in Abbildung 6.12 für  $n = 1$  entsprechen.

Offensichtlich ist das Voting äußerst robust gegen Paketverluste; die resultierenden Fehlerraten steigen mit zunehmenden Verlusten kaum an. Selbst extreme Paketverlustraten von 40% führen zu einer nur geringen Verschlechterung der

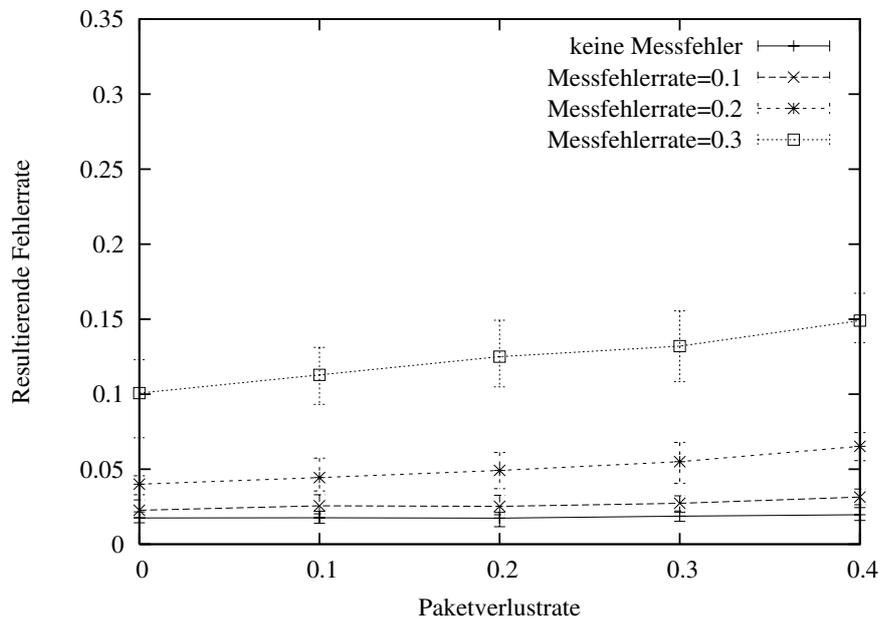


Abbildung 6.13.: Resultierende Fehlerrate nach dem Voting in Abhängigkeit von der Paketverlustrate.

Fehler von allerhöchstens 5%. Dieser Wert tritt bei einer initialen Messfehlerrate von 0,3 auf.

Bei ursprünglich 20% Fehlern steigt die resultierende Fehlerrate schlimmstenfalls um 3%, bei einer Rate von 10% initialer Messfehler ergibt sich sogar nur ein einprozentiger Anstieg. Am besten sind die Ergebnisse ohne initiale Messfehler, hier ist die resultierende Fehlerrate praktisch unabhängig von den Paketverlusten.

Somit zeigt sich zusammenfassend ein Zusammenhang zwischen dem Einfluss der Paketverlustrate und der ursprünglichen Messfehlerrate. Je geringer letztere ist, desto weniger wirken sich Paketverluste aus. Insgesamt lässt sich sagen, dass das Voting von Paketverlusten nur sehr geringfügig beeinflusst wird und sich als ausgesprochen robust zeigt.

### 6.3.4. Genauigkeit der Konturermittlung

Nachdem das bereits untersuchte Voting stattgefunden hat, berechnen innen liegende Sensorknoten mit außen liegenden Nachbarn einzelne Konturpunkte.

Dieser Abschnitt beschreibt die Ergebnisse der Untersuchung dieses Schrittes. Hier wird zunächst auf die Zusammenfassung von Polygonpunkten zu einer kompakteren Darstellung verzichtet. Es wird lediglich geprüft, wie exakt das sich aus allen Konturpunkten ergebende Polygon  $E$  der Originalkontur  $O$  des Phänomens entspricht. Auf die Kompression des Polygons wird dann im folgenden Abschnitt eingegangen.

Dabei ergab sich die Schwierigkeit, eine geeignete Metrik zu finden, die Aussagen über die Genauigkeit des ermittelten Polygons  $E$  erlaubt. Der Autor hat dazu die Metrik

$$M(P_1, P_2) = \frac{A(P_1 \setminus P_2) + A(P_2 \setminus P_1)}{A(P_1)} \quad (6.3)$$

entwickelt, die zwei Polygone  $P_1$  und  $P_2$  vergleicht, die Abweichung der von ihnen beschriebenen Flächen quantifiziert und ins Verhältnis zur Fläche von  $P_1$  setzt. Dabei steht  $A(P)$  für den Flächeninhalt eines Polygons  $P$ .

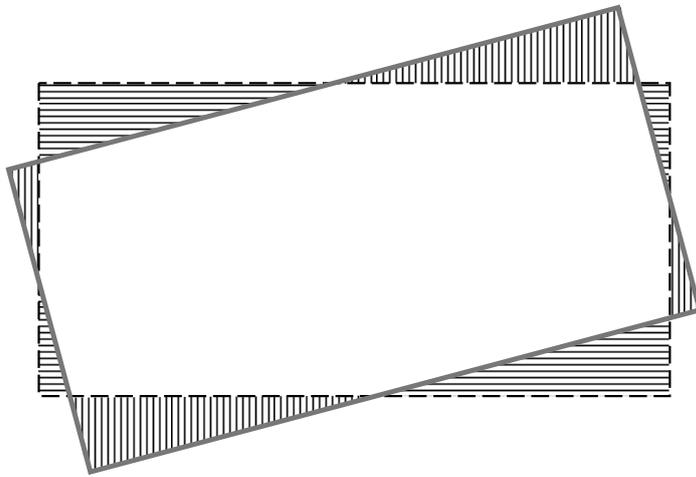


Abbildung 6.14.: Beispiel des Vergleiches von Polygonflächen.

Die Funktionsweise der Metrik sei anhand von Abbildung 6.14 beispielhaft erläutert. Gegeben sind zwei Polygone in Rechteckform. Das Polygon  $P_1$  liegt gerade und wird durch eine schwarze, gestrichelte Linie markiert. Das zweite Polygon  $P_2$  wird durch eine graue, durchgezogene Linie markiert; es ist leicht verdreht und beschreibt somit eine andere Fläche. Nun werden zunächst diejenigen Flächen bestimmt, die zwar innerhalb von  $P_1$  liegen, nicht aber innerhalb von  $P_2$ . Diese Flächen sind in der Abbildung horizontal gestreift und werden in der Metrik durch  $A(P_1 \setminus P_2)$  beschrieben. Dieser Schritt wird für solche Flächen wiederholt, die in  $P_2$ , nicht aber in  $P_1$  liegen. Sie sind vertikal gestreift eingezeichnet und werden durch  $A(P_2 \setminus P_1)$  beschrieben.

Die Metrik setzt nun die Summe der Flächen, die nur in einem der Polygone liegen, ins Verhältnis zur Gesamtfläche von  $P_1$ . Sie beschreibt somit den Fehler von  $P_2$  gegenüber  $P_1$  normiert auf die Fläche von  $P_1$ .

Anders als die in [23] herangezogene Hausdorff-Metrik [80], die lediglich die maximale Abweichung zweier Polygone heranzieht, berücksichtigt die hier zum Einsatz kommende Metrik  $M$  neben dem Grad der maximalen Abweichung auch, wie „oft“ abgewichen wurde. Somit kann sie eine einzelne Abweichung zwischen ansonsten deckungsgleichen Polygonen beispielsweise von einer Ab-

weichung gleichen Ausmaßes entlang des gesamten Polygonzuges differenzieren und ist somit deutlich aussagekräftiger.

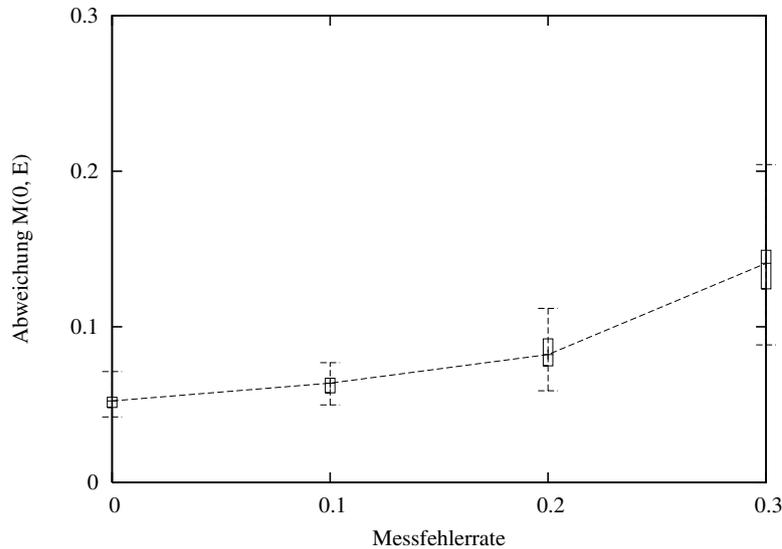


Abbildung 6.15.: Abweichung des aus den Konturpunkten ermittelten Polygons  $E$  von der Phänomenfläche  $O$  in Abhängigkeit von der Messfehlerrate.

Abbildung 6.15 zeigt die Abweichung des aus den Konturpunkten ermittelten Polygons  $E$  von der tatsächlichen Phänomenfläche  $O$  in Abhängigkeit von der initialen lokalen Messfehlerrate. Abweichend von den vorhergehenden Diagrammen sind hier nicht nur Mittelwert sowie in der Simulationsreihe aufgetretene Minima und Maxima angegeben, sondern darüber hinaus auch die Interquartilspreizung in Form eines Balkens. Innerhalb dieses Balkens liegen die mittleren 50% der aufgetretenen Werte. Weitere 25% der Werte liegen zwischen oberem Balkenende und dem Maximum und nochmals 25% zwischen Balken und Minimum.

Bei fehlerfreien Messungen liegt die mittlere Abweichung bei weniger als 5% der Phänomenfläche. Sie wächst auf 6,5%, 8% beziehungsweise 14% für Messfehlerraten von 0,1, 0,2 beziehungsweise 0,3 an. Dabei wächst auch die Streuung der Abweichungen an, was sowohl an den Quartilen als auch an der Spreizung von Minima und Maxima festzumachen ist.

Dennoch zeigen diese Ergebnisse die gute Genauigkeit der Konturermittlung. Selbst bei einer Messfehlerrate von 20%, das heißt wenn sich immerhin jeder fünfte Sensor-knoten aufgrund seiner lokalen Messung zunächst falsch einordnet, beträgt die Abweichung der ermittelten Phänomenfläche niemals mehr als 11% der Originalfläche (im Mittel sogar nur 8%). Auch hier wird die hohe Robustheit des Verfahrens gegenüber Fehlern deutlich.

### 6.3.5. Qualität der Approximation

Da davon auszugehen ist, dass die Phänomene mit großer räumlicher Ausdehnung zu derartig vielen Konturpunkten führen, dass diese von Sensorknoten weder gespeichert, verarbeitet noch per Funk kommuniziert werden könnten, schließt das Verfahren zur Konturapproximation auch einen Schritt Polygonkompression während der Zusammenführung der Punkte zu einem Polygon ein.

Abschließend werden in diesem Abschnitt Untersuchungen zur Qualität dieser Approximation vorgestellt. Dabei kam erneut die in Abschnitt 6.3.4 beschriebene Metrik  $M$  zur Quantifikation der Abweichung des komprimierten Polygons  $E_{approx}$  vom originalen Phänomenpolygon  $O$  zum Einsatz. Bestenfalls können somit die Genauigkeiten aus dem vorangegangenen Abschnitt erreicht werden, die sich jedoch mit zunehmender Kompressionsrate verschlechtern werden.

Es sei an dieser Stelle darauf hingewiesen, dass ein unkomprimiertes Polygon, wie es im Rahmen der Untersuchung der Konturermittlung betrachtet wurde, etwa 130 Punkte besitzt. Die hier betrachteten Polygone bestehen aus 20 bis hinunter zu 10 Punkten, was Kompressionsraten zwischen 0,15 und 0,07 (Reduktion auf 15% beziehungsweise 7% der Originalgröße, also Kompressionsfaktoren zwischen 6,5 und 13) entspricht.

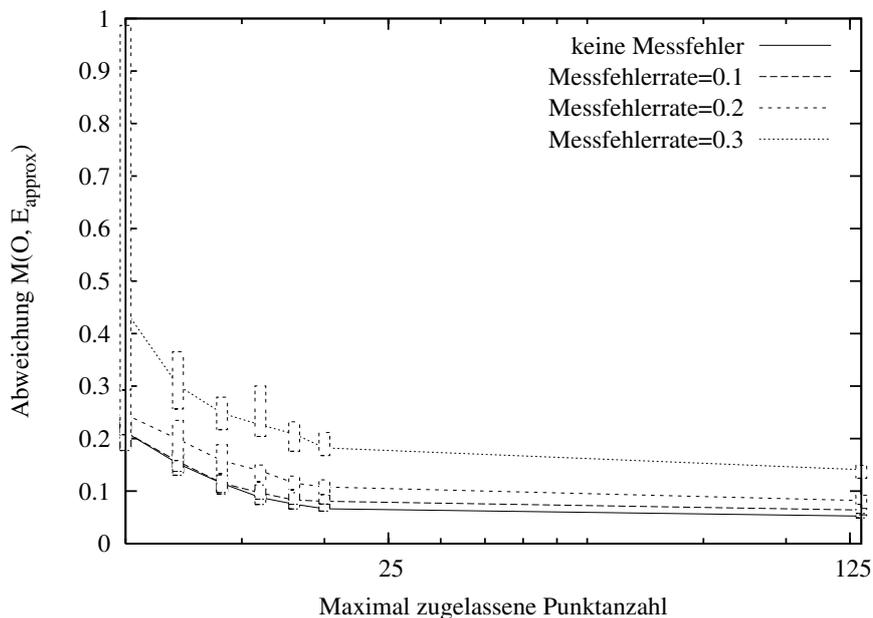


Abbildung 6.16.: Abweichung des approximierten Polygons in Abhängigkeit von der Anzahl seiner Punkte.

Abbildung 6.16 zeigt die Abweichung des approximierten, das heißt aus Konturpunkten zusammengesetzten und komprimierten, Polygons  $E_{approx}$  vom tatsächlichen Phänomenpolygon  $O$  in Abhängigkeit von der maximal zugelassenen Anzahl von Polygonpunkten für verschiedene Messfehlerraten. Dabei kam eine logarithmische x-Achse zum Einsatz, da die Ergebnisse für kleine Maxi-

malpunktzahlen aufgrund des breiten Wertebereiches sonst nicht voneinander unterscheidbar gewesen wären.

Abweichend von Abbildung 6.15 fehlen hier zur Verbesserung der Lesbarkeit die Minimal- und Maximalindikatoren. Angegeben sind der Mittelwert und die Interquartilspreizung. Gezeigt werden Ergebnisse für 130 Polygonpunkte (in etwa die Anzahl der Punkte des unkomprimierten Polygons) sowie für maximale Punktanzahlen von 10, 12, 14, 16, 18 und 20.

Zwei Einflussfaktoren werden hier deutlich: Die Abweichung des approximierten Polygons von dem Phänomenpolygon steigt einerseits mit zunehmender Messfehlerrate und andererseits mit abnehmender Maximalpunktzahl. Lässt man jedoch 20 Punkte zu, hat das approximierte Polygon nahezu die gleiche Genauigkeit wie das unkomprimierte. Ohne den Einfluss von Messfehlern liegt die Abweichung bei 6,6% (unkomprimiert: 5%), und selbst bei einer Messfehlerrate von 20% liegt die Abweichung bei nur 11% (unkomprimiert: 8%). Lediglich mit steigender Kompression nimmt die Qualität der Approximation dann sukzessive ab.

Allerdings ist zu bedenken, dass das speziell für die simulative Evaluation gewählte Phänomenpolygon nicht mit weniger als 14 Polygonpunkten dargestellt werden kann. Selbst wenn man diesen kritischen Wert als maximal zulässige Punktzahl wählt, ist die Abweichung des resultierenden Polygons nur doppelt so hoch wie mit 130 Punkten.

### 6.3.6. Zusammenfassende Veranschaulichung

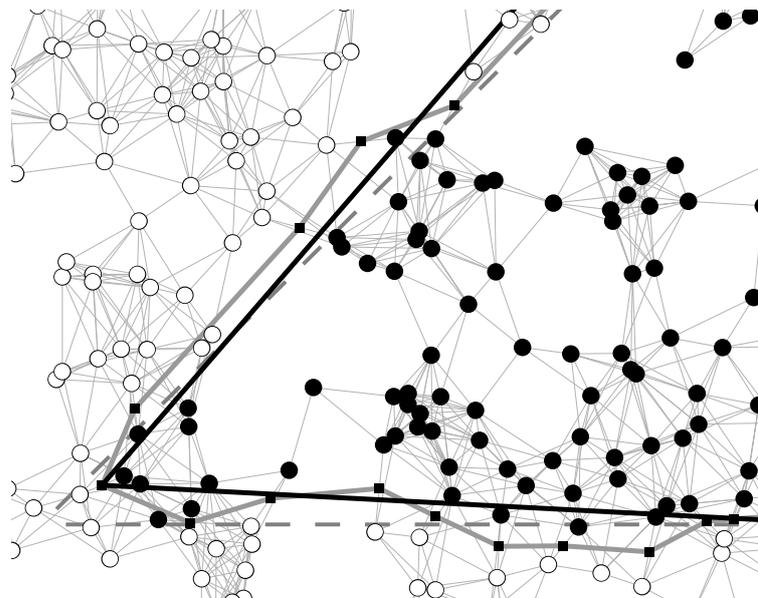


Abbildung 6.17.: Vergrößerter Ausschnitt aus der Simulationsfläche.

Abschließend sollen die Ergebnisse der vorangegangenen Abschnitte noch ein-

mal an einem typischen Beispiel illustriert werden. Dazu zeigt Abbildung 6.17 einen vergrößerten Ausschnitt aus einem Simulationsszenario. Seine Lage innerhalb der Simulationsfläche ist in Abbildung 6.11 grau gekennzeichnet. Knoten, die nach dem Voting zu dem Ergebnis kamen, dass sie innerhalb des Phänomens liegen, sind durch schwarze, die anderen durch weiße Punkte gekennzeichnet. Die Kommunikationsverbindungen zwischen den Knoten sind durch graue Linien angedeutet. Die gestrichelte Linie beschreibt die tatsächliche Phänomenkontur.

Die schwarzen Quadrate verdeutlichen die Lage aller berechneten Konturpunkte. Verbindet man diese durch eine Linie miteinander, ergibt sich das unkomprimierte Polygon (graue durchgezogene Linie). Das komprimierte Polygon ist durch eine schwarze, durchgezogene Linie kenntlich gemacht. In dieser Simulation war die Maximalzahl der zulässigen Punkte auf 20 gesetzt.

## 6.4. Eigenschaften des Verfahrens

Unter Einbeziehung der in Abschnitt 6.2 beschriebenen Funktionsweise und der in Abschnitt 6.3 dargestellten Simulationsergebnisse kann festgestellt werden, dass das Verfahren den in Abschnitt 6.1.1 aufgestellten Anforderungen genügt:

- Geringer Energie- und Ressourcenbedarf: Die Konturapproximation arbeitet sehr ressourcensparend. Unter der Annahme, dass der Hopcount für das Voting  $n$  auf den simulativ als optimal ermittelten Wert 1 festgesetzt ist, sendet jeder Knoten im Rahmen des Votings nur zwei Broadcastnachrichten an seine direkten Nachbarn. Lediglich die Knoten am Rand des Phänomens, die an der Konturapproximation beteiligt sind, senden in der Regel ein zusätzliches Paket aus. Während des Verfahrens müssen die Sensorknoten lediglich eine Liste ihrer Nachbarn mit deren Positionen und Prädikatwerten, ihren eigenen Prädikatwert sowie gegebenenfalls einen Konturpunkt speichern. Alle Berechnungsschritte weisen eine geringe Berechnungskomplexität auf. Aufgrund dieser Eigenschaften ist der Energiebedarf des Verfahrens ebenfalls niedrig.
- Verteiltheit und Skalierbarkeit: Das Verfahren arbeitet vollständig verteilt und ohne zentrale Berechnungen. Nachrichten werden ausschließlich zwischen benachbarten Geräten ausgetauscht. Einzig die Dauer der Konturapproximation variiert mit der Größe beziehungsweise der Dichte des Netzes, sie entwickelt sich linear mit dem Umfang des Phänomens beziehungsweise der Zahl der sukzessive an der Konturapproximation beteiligten Sensorknoten.
- Robustheit gegen Messfehler: In der in Abschnitt 6.3 beschriebenen simulativen Evaluation wurde gezeigt, dass das Verfahren ausgesprochen robust gegen Messfehler ist. Dieses ist in erster Linie auf das Voting zurückzuführen, das die Fehlerrate durch die Abstimmung mit den direkten Nachbarn effektiv reduziert.

- **Robustheit gegen Kommunikationsfehler:** Die Simulationen haben ebenfalls belegt, dass die Konturapproximation sehr robust gegen Kommunikationsfehler ist. So haben während der ersten Phasen der Verfahrens auch Paketverlusten von 40% kaum Einfluss auf die Qualität der Ergebnisse. Während der abschließenden Konturapproximation wird der Nachrichtenempfang jeweils mit Hilfe von Acknowledgements bestätigt. Bleibt eine Bestätigung aus, wird die Nachrichtenübertragung wiederholt. Ist ein Zielknoten nicht erreichbar, wird mit der Rechte-Hand-Regel auf den nächsten Nachbarn ausgewichen.
- **Kommunikationsstruktur:** Es werden keinerlei dauerhafte Strukturen wie Bäume oder Routen benötigt.
- **Nutzung von innen liegenden Knoten:** Fällt ein Teil der Sensorknoten innerhalb der Phänomenregion vor oder während des Verfahrensablaufs aus, schadet das keinem der Schritte des Verfahrens. Die Konturapproximation, die letztendlich zum Ergebnis führt, arbeitet ausschließlich auf Knoten am Rand des Phänomens, so dass bezüglich Knotenausfällen eine weitestgehende Robustheit sichergestellt ist.

### 6.5. Zusammenfassung

In diesem Kapitel wurde ein effizientes und robustes Verfahren zur Approximation von Phänomenkonturen vorgestellt. Anwendungsgebiete sind zum Beispiel die Detektion von Ausbreitungsgebieten von Bränden, Überflutungen oder Leckagen. Diese Informationen können dann beispielsweise an eine Einsatzstelle weitergeleitet werden.

Ein weiteres Anwendungsgebiet sind sogenannte *symbolische Lokationen*. Verteilt man die Polygoninformationen im Sensornetz, kann man darauf neue, programmiersprachliche Abstraktionen stützen. So könnten sich Geräte nicht nur in Abhängigkeit von ihren Koordinaten unterschiedlich verhalten, sondern in Abhängigkeit davon, ob sie sich zum Beispiel *innerhalb* oder *außerhalb* eines Feuers befinden.

Das vorgestellte Verfahren arbeitet ausschließlich auf Basis lokaler Interaktion und kombiniert dabei Kantenerkennung und Konturapproximation. Es ist extrem robust gegen Messfehler und Paketverluste. Dabei liefert es eine kompakte Polygonrepräsentation des Phänomens, die nur wenige Byte groß ist. Auch nicht-konvexe Strukturen können approximiert werden. Durch die Möglichkeit, eine Maximalpunktzahl der resultierenden Polygonbeschreibung festzulegen, lässt sich der Ressourcenbedarf des Verfahrens effektiv begrenzen. Es sind keinerlei hierarchische Kommunikationsstrukturen Voraussetzung, lediglich Lokationsbewusstsein und lose Zeitsynchronisation sind erforderlich, damit der Algorithmus ablaufen kann.

Sind die Messungen, auf denen die Detektion des Phänomens beruht, mit einer geringen Fehlerquote behaftet, kann das Verfahren die räumliche Ausbreitung

des Phänomens mit einer mittleren Abweichung von nur 5% gegenüber der tatsächlichen Phänomenkontur ermitteln. Es berechnet daraus eine Approximation mit nur 20 Punkten, die lediglich um 6,6% von der Realität abweicht. Der Algorithmus kann jedoch auch mit Messfehllerraten von 20% und mehr umgehen.



## 7. Zusammenfassung und Ausblick

Drahtlose Sensornetze können nicht nur Daten feingranular messen, verarbeiten und aufbereiten, sondern gestatten auch den drahtlosen Datenaustausch innerhalb des Netzes. Die von den Sensoren erfassten Daten jedoch sind praktisch bedeutungslos, solange sie für sich allein stehen. Sie erhalten ihre Aussagekraft erst, wenn ihnen Informationen über den Kontext ihrer Erhebung hinzugefügt werden. Für die Interpretation, Korrelation, Validierung und Aggregation von Sensordaten sind der Zeitpunkt und der Ort die wichtigsten Kontextaspekte.

Kontextinformationen sind jedoch nicht nur im Zusammenhang mit den gemessenen Daten von Interesse, sie sind auch für die Optimierung des Anwendungsverhaltens hilfreich. Hier treten weitere Kontextaspekte in Erscheinung, beispielsweise der verbleibende Energievorrat oder die Anzahl und Entfernung direkter Nachbarn. Solche Informationen können beispielsweise bei der Koordination der Aktivitäten der einzelnen Sensorknoten dazu dienen, durch Vergleich von Knotendistanz und Sensorreichweite die Sensorabdeckung sicherzustellen und gleichzeitig die Gesamtlaufzeit des Netzes zu maximieren, indem bevorzugt Knoten mit viel Restenergie aktiv sind.

Da viele Kontextaspekte nicht direkt von den Sensorknoten selbst erhoben werden können, wird vielfach so vorgegangen, dass zunächst die benötigten Eingangsdaten mit Hilfe von Protokollen im Netz verteilt werden und dann Algorithmen zur Berechnung der gewünschten Kontextinformationen aus diesen Daten zur Anwendung kommen.

In dieser Arbeit wurden Verfahren vorgestellt, die ebenfalls auf diese Weise arbeiten und es den Sensorknoten ermöglichen, Bewusstsein bezüglich der beiden wichtigsten Kontextaspekte für Sensornetze, Zeit und Raum, zu entwickeln. Die Erlangung eines konsistenten Zeitbewusstseins wurde – mit Schwerpunkt auf dem resultierenden Kommunikationsaufwand – in Kapitel 4 thematisiert. Kapitel 5 lieferte einen Beitrag zur Erlangung von Positionsbewusstsein, indem ein neues Verfahren zur Distanzschätzung vorgestellt wurde, das bei der Positionsbestimmung mit Hilfe der Multilateration angewendet werden kann. Auf Basis ihrer so bestimmbareren Koordinaten können die Sensorknoten mit dem in Kapitel 6 präsentierten Verfahren die Konturen aller im Sensornetz beobachtbaren Phänomene approximieren.

Um effizient arbeiten zu können, müssen die eingesetzten Protokolle und Algorithmen an die Erfordernisse von Sensornetzen angepasst sein. Eingangs wurden Fehlertoleranz, Ressourcensparsamkeit, minimierte Kommunikation und Skalierbarkeit als erforderliche Eigenschaften identifiziert.

Im Hinblick auf die Erlangung eines konsistenten Zeitbewusstseins wurde herausgearbeitet, dass sich mit proaktiven und post-facto-Verfahren zwei grundsätzliche Vorgehensweisen herausgebildet haben, und deren Kommunikationsoverhead untersucht. Es wurde deutlich gemacht, dass die post-facto-Verfahren bei geringem Datenaustausch weniger Overhead erzeugen als die proaktiven, die bei starkem Kommunikationsaufkommen im Netz allerdings weniger Overhead bedingen. Beide Ansätze haben somit ein bevorzugtes Einsatzgebiet, können aber nicht auf zeitliche und lokale Schwankungen des Datenaufkommens reagieren.

Mit TICTAC wurde ein Verfahren vorgestellt, das sein Verhalten dem Verkehrsaufkommen im Sensornetz anpassen kann und es somit ermöglicht, abhängig vom Anwendungskontext die am besten geeignete Strategie zu nutzen. Während TICTAC bei geringem Kommunikationsaufkommen post-facto arbeitet, führt es bei lokalen Zunahmen des Datenaustausches eine lokale proaktive Zeitsynchronisation durch.

In Simulationen wurde die Entwicklung des Overheads der drei Konzepte über die Zeit untersucht und verglichen. Es wurde gezeigt, dass TICTAC über bestimmte Zeitbereiche hinweg am wenigsten Overhead erzeugt, deren Länge von Applikationsparametern wie dem Datenaufkommen im Netz sowie der Intensität und Dauer von dessen Schwankungen abhängen. Darauf aufbauend wurde dann ein Modell entwickelt, das die quantitative Vorhersage des entstehenden Overheads erlaubt. Dieses Modell kann einerseits während der Phase der Protokollauswahl und Anwendungsentwicklung genutzt werden, andererseits hat seine geringe Berechnungskomplexität den Vorteil, dass es auch direkt auf den Sensorknoten implementiert werden kann. So kann das Modell zur Laufzeit verwendet werden, um auf Basis der Applikationsparameter Entscheidungen über die einzusetzende Strategie zu fällen.

Im Hinblick auf ortsbezogene Kontextinformationen wurde aufgezeigt, dass die auf Basis von Ankerpositionen und Abständen zu diesen arbeitende Multilateration gut zur Positionsberechnung in Sensornetzen geeignet ist. Mit NIDES wurde ein neues Distanzschätzverfahren vorgestellt, das solche Abstandsinformationen liefert. Es basiert auf der Tatsache, dass dicht beieinander liegende Sensorknoten mehr gemeinsame Kommunikationspartner haben als solche, die weiter voneinander entfernt liegen. Zunächst wurden verwandte Arbeiten anderer Forscher mit ihren Vor- und Nachteilen sowie der mit ihnen erzielbaren Genauigkeiten vorgestellt. Dann wurde erläutert, wie, ausgehend vom einem Modell der Funkausbreitung der drahtlosen Kommunikationsschnittstelle und dem Anteil der mit einem benachbarten Sensorknoten gemeinsamen Kommunikationspartner, die Distanz zu ihm ermittelt werden kann.

Mit Hilfe von Simulationen wurde gezeigt, dass sich das Verfahren für verschiedene Funkausbreitungsmodelle eignet und die Genauigkeit der ermittelten Distanzen mit der Netzdichte (das heißt der mittleren Anzahl von Nachbarn) zunimmt. Während der durchschnittliche Fehler für eine Dichte von fünf bei etwa 30% des Kommunikationsradius liegt, sinkt er für eine Dichte von 50 auf nur noch 7%. Bereits bei einer für Sensornetzwerke sinnvollen Dichte von 15

---

Nachbarn beträgt der Fehler im Mittel nur noch 15% der Kommunikationsreichweite.

Mit diesen Ergebnissen ordnet sich NIDES zwischen bereits existierenden Methoden ein. Es arbeitet genauer als Verfahren, die auf Signalstärkenmessungen beruhen. Wie diese hat es den Vorteil, dass keine zusätzliche Hardware für die Distanzschätzungen benötigt wird. Verfahren, die sich auf die Messung von Laufzeitdifferenzen stützen, arbeiten zwar genauer, erfordern jedoch erheblichen zusätzlichen Hardwareaufwand. Radioinfernometrie arbeitet ebenfalls genauer als NIDES, erfordert jedoch spezialisierte Funkchips, weist eine hohe Berechnungskomplexität auf und arbeitet sehr langsam. Diese Nachteile können bei Nutzung des nachbarschaftsbasierten Verfahrens vermieden werden.

Es wurde auch erläutert, wie in Kombination mit dem Sum-Dist-Verfahren [160] mit NIDES auch Distanzschätzungen über die Kommunikationsreichweite hinaus durchgeführt werden können. Die simulative Untersuchung hat gezeigt, dass die Genauigkeit der Schätzung über mehrere Hops nur fünf bis zehn Prozent schlechter ist als die, die mit einem perfekten Distanzschätzer möglich wäre. In Verbindung mit der in Abschnitt 3.2 vorgestellten Multilateration können Sensorknoten ihre Positionen in Form von Koordinaten bestimmen.

Ist es gewünscht, Ortsinformationen nicht in Form von Koordinaten, sondern symbolisch anzugeben, kann das in Kapitel 6 beschriebene Verfahren zum Einsatz kommen. Es ermittelt die räumliche Ausdehnung von Phänomenen, die von einem Sensornetz beobachtet werden können, approximiert sie zu einer kompakten Phänomenkontur und stellt sie in Form eines Polygons dar. Auf dieser Basis kann man Regionen im Sensornetz, die bestimmte Eigenschaften haben, beschreiben und auf dieser Basis Primitive wie „innerhalb“ oder „außerhalb“ definieren.

Das Verfahren kann auch als Teil der im Sensornetz laufenden Anwendung eingesetzt werden, indem die Ausdehnung der Regionen nach außen weitergeleitet wird. So könnte eine Anwendung zur Beobachtung von Temperaturunterschieden Bereiche mit Werten ober- und unterhalb von Schwellenwerten identifizieren und an eine Kontrollstelle weiterleiten.

Das Verfahren arbeitet ausschließlich auf Basis lokaler Interaktion und kombiniert dabei Kantenerkennung und Konturapproximation. Es ist extrem robust gegen Messfehler und Paketverluste. Dabei liefert es eine kompakte Polygonrepräsentation des Phänomens, die nur wenige Byte groß ist. Auch nicht-konvexe Strukturen können approximiert werden. Durch die Möglichkeit, eine Maximalpunktzahl der resultierenden Polygonbeschreibung festzulegen, lässt sich der Ressourcenbedarf des Verfahrens effektiv begrenzen.

Sind die Messungen, auf denen die Detektion des Phänomens beruht, mit einer geringen Fehlerquote behaftet, kann das Verfahren die räumliche Ausbreitung des Phänomens mit einer mittleren Abweichung von nur 5% gegenüber der tatsächlichen Phänomenkontur ermitteln. Es berechnet daraus eine Approximation mit nur 20 Punkten, die lediglich um 6,6% von der Realität abweicht.

Der Algorithmus kann jedoch auch mit Messfehlerraten von 20% und mehr umgehen.

Die in dieser Arbeit präsentierten Verfahren wurden mit Hilfe von Simulationen auf ihre Eigenschaften untersucht, da diese Methode verschiedene Vorteile aufweist. Zum Einen kann man auf diese Weise auch Netze untersuchen, die mehr Sensorknoten umfassen, als zum Experimentieren zur Verfügung stehen. Zum Anderen gestattet es der Simulator, jederzeit auf jede Art von Daten auf jedem Knoten zuzugreifen und auf diese Weise ein besonders umfassendes Bild des Verhaltens eines Protokolls zu erlangen. Drittens wird auch die Umgebung der Sensorknoten durch den Simulator modelliert, so dass man deren Eigenschaften gezielt variieren kann. Folglich kann man Protokolle über einen weiten Bereich von Umgebungsbedingungen untersuchen.

Bei den Simulationen wurden die tatsächlichen Eigenschaften der Sensorknoten durch Modelle repräsentiert, die meist von anderen Autoren entwickelt wurden. Beispiele hierfür sind [73,202,204]. Trotz ihrer sorgfältigen und wissenschaftlich fundierten Konstruktion stellen solche Modelle allerdings inhärent Vereinfachungen und Verallgemeinerungen der Realität dar.

Raum für zukünftige Arbeiten ergibt sich folglich insbesondere im Bereich der experimentellen Erprobung auf tatsächlicher Hardware. Das iSense-Softwaresystem bietet hierfür eine gute Grundlage, da es mehrere Plattformen unterstützt. Darunter befindet sich auch der Simulator Shawn, der mittels einer Adaptionsschicht den gleichen Code ausführen kann, der auch auf den tatsächlichen Geräten zum Einsatz kommt [141].

Für alle in dieser Arbeit vorgestellten Verfahren wurde argumentativ dargelegt, dass sie sparsam mit den auf Sensorknoten verfügbaren Ressourcen umgehen, indem sie wenig Daten für ihren Betrieb speichern müssen, eine geringe Berechnungskomplexität aufweisen und nur wenig Daten austauschen. Diese Eigenschaften würden sich durch ihre Realisierung auf Geräten zusätzlich belegen lassen.

Von besonderem Interesse ist eine reale Implementierung sicherlich für NIDES, da die Genauigkeit der Distanzschätzungen auch davon abhängt, wie gut sich das als Basis dienende Funkmodell mit den tatsächlichen Eigenschaften der Funkausbreitung deckt. Erste Untersuchungen in diese Richtung deuten jedoch bereits an, dass sich mit dem stochastischen Modell häufig eine gute Abbildung der vorgefundenen Realität erzielen lässt.

## A. Verzeichnisse



# Abbildungsverzeichnis

2.1. Schematischer Aufbau eines Sensorknotens. . . . .	7
2.2. Unterschiedliche Sensorknoten. . . . .	8
2.3. Schematische Darstellung eines Sensornetzes mit Gateway. . . . .	10
2.4. Funkmodelle. . . . .	18
3.1. Inkonsistentes Zeitbewusstsein. . . . .	23
3.2. Ermittlung von Zeitdifferenzen mit einzelnen Nachrichten. . . . .	25
3.3. Ermittlung von Zeitdifferenzen mit mehreren Nachrichten [104]. . . . .	25
3.4. Austausch von Nachrichten mit Zeitstempeln. . . . .	26
3.5. Der Approximate Point-In-Triangulation Test. . . . .	35
3.6. Positionsbestimmung mit dem Min-Max-Verfahren. . . . .	38
3.7. Positionsbestimmung mit der VOR-Methode. . . . .	39
3.8. Rückführung der Multiangulation auf die Multilateration. . . . .	40
4.1. Sensornetzbereich (durchgezogenes Quadrat), Hochverkehrsbe- reich (gestricheltes Quadrat), sowie die Entwicklung des vom dick eingezeichneten Knoten angestoßenen Flutens bei TICTAC (qua- litative Darstellung). . . . .	48
4.2. Entwicklung des Overheads O in den Simulationen. . . . .	53
4.3. Allgemeine Funktion zur Beschreibung des Overheads. . . . .	56
4.4. Entwicklung des Overheads O gemäß dem mathematischen Modell. . . . .	58
5.1. Nachbarschaftüberschneidung bei verschiedenen Distanzen und Winkeln. . . . .	72
5.2. Gemeinsame Nachbarn (schwarz) bei verschiedenen Funkmodellen. . . . .	74
5.3. Unterschiedliche Arten von Dichteschwankungen. . . . .	75
5.4. Mathematische Grundlagen von NIDES beim Kreismodell. . . . .	76
5.5. Zusammenhang zwischen der Schnittfläche und der Anzahl ge- meinsamer Nachbarn. . . . .	78
5.6. Zusammenhang zwischen dem Anteil der Schnittfläche und der Distanz. . . . .	79
5.7. Funkmodell von $n_3$ , sowie die Knoten $n_1$ , $n_2$ und $n_3$ . . . . .	80
5.8. Berechnung der Abbildungstabelle. . . . .	80
5.9. Rotation von $f_1$ um die Hochachse. . . . .	81
5.10. Veranschaulichung einer Faltung. . . . .	82
5.11. $s(d)$ für unterschiedliche Funkmodelle. . . . .	83
5.12. Algorithmus zur Tabellensuche und Interpolation. . . . .	86
5.13. Simulationsszenario. . . . .	87
5.14. Histogramm der Nachbarschaftsgrößen. . . . .	88

5.15. Schätzfehler in Abhängigkeit von der Netzdichte, RI-Modell. . . . .	90
5.16. Schätzfehler in Abhängigkeit von der Distanz, RI-Modell. . . . .	92
5.17. Gaussverteilung der Knoten über die Simulationsfläche. . . . .	93
5.18. Schätzfehler in Abhängigkeit von der Distanz, Gaussverteilung, RI-Modell. . . . .	94
5.19. Fehlercharakteristik mit dem stochastischen Funkmodell. . . . .	95
5.20. Fehlercharakteristik mit dem Kreismodell. . . . .	96
5.21. Ausschnitt aus einem Szenario. . . . .	98
5.22. Wegverlängerung bei Multihop-Pfaden. . . . .	99
5.23. Modifizierte Schätzfunktionen. . . . .	100
5.24. Simulationsszenario. . . . .	101
5.25. Prozentualer Schätzfehler. . . . .	102
6.1. Beispielhafte Darstellung der Konturapproximation. . . . .	107
6.2. Beispiel der Netzwerkplanarisierung. . . . .	117
6.3. Beispiel für die Weiterleitung der Polygonnachricht. . . . .	118
6.4. Beurteilung der Signifikanz einzelner Polygonpunkte. . . . .	119
6.5. Beispielszenario. . . . .	121
6.6. Einzelmessung und Voting. . . . .	121
6.7. Kantendetektion. . . . .	122
6.8. Netzplanarisierung. . . . .	123
6.9. Konturapproximation (Teil 1). . . . .	124
6.10. Konturapproximation (Teil 2). . . . .	125
6.11. Simulationsszenario mit Phänomenkante (gestrichelt) und Posi- tion des in Abbildung 6.17 vergrößert dargestellten Ausschnittes (grau). . . . .	127
6.12. Resultierende Fehlerrate nach dem Voting in Abhängigkeit des Hopcounts für initiale Messfehlerraten zwischen 0% und 30%. . . . .	128
6.13. Resultierende Fehlerrate nach dem Voting in Abhängigkeit von der Paketverlustrate. . . . .	130
6.14. Beispiel des Vergleiches von Polygonflächen. . . . .	131
6.15. Abweichung des aus den Konturpunkten ermittelten Polygons $E$ von der Phänomenfläche $O$ in Abhängigkeit von der Messfehlerrate. . . . .	132
6.16. Abweichung des approximierten Polygons in Abhängigkeit von der Anzahl seiner Punkte. . . . .	133
6.17. Vergrößerter Ausschnitt aus der Simulationsfläche. . . . .	134

# Abkürzungsverzeichnis

ADP .....	Asynchronous Diffusion Protocol
AoA .....	Angle of Arrival
APIT .....	Approximate Point-In-Triangulation Test
CSS .....	Chirp Spread Spectrum
DOI .....	Degree of Irregularity
DV-Hop .....	Distance-Vector-Hop
FTSP .....	Flooding Time Synchronization Protocol
GPS .....	Global Positioning System
GPSR .....	Greedy Perimeter Stateless Routing
LPS .....	Local Positioning System
LQI .....	Link quality indicator
LTS .....	Lightweight Tree-Based Synchronization
MDS .....	Multidimensionale Skalierung
NIDES .....	Neighborhood Intersection Distance Estimation Scheme
NTP .....	Network Time Protocol
OC .....	Organic Computing
RAM .....	Random Access Memory
RIM .....	Radio Irregularity Model
RSSI .....	Received signal strength indicator
SDS-TWR .....	Symmetrical Double-Sided Two Way Ranging
TDP .....	Time-Diffusion Synchronization Protocol
TICTAC .....	Traffic Induced Control of Time And Communication
TPSN .....	Timing-sync Protocol for Sensor Networks
TTL .....	Time-to-live
TTP .....	Time Transmission Protocol
UDG .....	Unit Disk Graph
UKW .....	Ultrakurzwelle
VOR .....	VHF Omnidirectional Radio Range



## Literaturverzeichnis

- [1] ADEBUTU, T., SACKS, L., AND MARSHALL, I. Simple position estimation for wireless sensor networks. In *Proc. of London Communications Symposium (LCS 2003)* (Sept. 2003).
- [2] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., AND CAYIRCI., E. A survey on sensor networks. *IEEE Communications Magazine* 40, 8 (Mar. 2002), 102–114.
- [3] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., AND CAYIRCI., E. Wireless sensor networks: A survey. *Computer Networks* 38, 4 (Mar. 2002), 393–422.
- [4] ALBOWICZ, J., CHEN, A., AND ZHANG, L. Recursive position estimation in sensor networks. In *IEEE International Conference on Network Protocols (ICNP 2001)* (Nov. 2001), pp. 35–41.
- [5] ARISHA, K. A. Energy-aware TDMA-based MAC for sensor networks. In *IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002)* (2002).
- [6] ARVIND, K. Probabilistic clock synchronization in distributed systems. *IEEE Transactions Parallel Distributed Systems* 5, 5 (1994), 474–487.
- [7] ASH, J. N., AND POTTER, L. C. Robust system multiangulation using subspace methods. In *IPSN '07: Proc. of the 6th international conference on Information processing in sensor networks* (New York, NY, USA, 2007), ACM, pp. 61–68.
- [8] AVANCHA, S., UNDERCOFFER, J. L., JOSHI, A., AND PINKSTON, J. Secure sensor networks for perimeter protection. *Computer Networks* 43, 4 (Nov. 2003), 421–435.
- [9] BAER, M. The ultimate on-the-fly network. *Wired Magazine*, 12 (2003). <http://www.wired.com/wired/archive/11.12/network.html>.
- [10] BAUER, P., SICHITIU, M., ISTEPANIAN, R., ENGLAND, U., AND PREMAMATNE, K. The mobile patient: wireless distributed sensor networks for patient monitoring and care. In *Proc. of the IEEE EMBS International Conference on Information Technology Applications in Biomedicine* (2000), pp. 17–21.
- [11] BEHNKE, R., AND TIMMERMANN, D. Awcl: Adaptive weighted centroid localization as an efficient improvement of coarse grained localization. In *5th Workshop on Positioning, Navigation and Communication 2008*

- (*WPNC'08*) (2008).
- [12] BERGAMO, P., AND MAZZINI, G. Localization in sensor networks with fading and mobility. In *Proc. of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (2002).
  - [13] BLUMENTHAL, J., GROSSMANN, R., GOLATOWSKI, F., AND TIMMERMANN, D. Weighted centroid localization in zigbee-based sensor networks. In *IEEE International Symposium on Intelligent Signal Processing (WISP 2007)* (2007).
  - [14] BOKAREVA, T., HU, W., KANHERE, S., RISTIC, B., GORDON, N., BESSELL, T., RUTTEN, M., AND JHA, S. Wireless sensor networks for battlefield surveillance. In *Proc. of The Land Warfare Conference (LWC)* (Brisbane, Queensland, Australia, 2006).
  - [15] BONABEAU, E., DORIGO, M., AND THERAULAZ, G. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.
  - [16] BORG, I., AND STAUFENBIEL, T. *Lehrbuch Theorien und Methoden der Skalierung*, 4 ed. Hübner, 2007.
  - [17] BOUAZIZI, I. ARA - the ant-colony based routing algorithm for MANETs. In *ICPPW '02: Proc. of the 2002 International Conference on Parallel Processing Workshops* (Washington, DC, USA, 2002), IEEE Computer Society.
  - [18] BRIESEMEISTER, L. Sensor data dissemination through ad hoc battlefield communications. In *In Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)* (2003).
  - [19] BROWN, P. J. The Stick-e document: A framework for creating context-aware applications. In *Electronic Publishing* (1995), pp. 259–272.
  - [20] BROWN, P. J., BOVEY, J. D., AND CHEN, X. Context-aware applications: from the laboratory to the marketplace. *IEEE Personal Communications* 4, 5 (Oct. 1997), 58–64.
  - [21] BULUSU, N., BYCHKOVSKIY, V., ESTRIN, D., AND HEIDEMANN, J. Scalable, ad hoc deployable, rf-based localization. In *Proc. of Computing Conference 2002* (Oct. 2002).
  - [22] BULUSU, N., HEIDEMAN, J., AND ESTRIN, D. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications* (2000).
  - [23] BURAGOHAIN, C., GANDHI, S., HERSHBERGER, J., AND SURI, S. Contour approximation in sensor networks. In *DCOSS* (2006), pp. 356–371.
  - [24] BUSCHMANN, C., FEKETE, S. P., FISCHER, S., KRÖLLER, A., AND PFISTERER, D. Koordinatenfreies Lokationsbewusstsein. *it - Information Technology, Themenheft Sensornetze* 47, 4 (Apr. 2005).

- 
- [25] BUSCHMANN, C., AND FISCHER, S. Analyse des Kommunikationsaufwands für konsistentes Zeitbewusstsein. Tech. rep., 4. Fachgespräche Drahtlose Sensornetze, GI/ITG KuVS, 2005.
- [26] BUSCHMANN, C., HELLBRÜCK, H., FISCHER, S., KRÖLLER, A., AND FEKETE, S. Radio propagation-aware distance estimation based on neighborhood comparison. In *Proc. of the 14th European conference on Wireless Sensor Networks (EWSN 2007)*, Delft, The Netherlands (Jan. 2007).
- [27] BUSCHMANN, C., KRÜGER, D., AND FISCHER, S. Lean and robust phenomenon boundary approximation. In *Proc. of the 12th IEEE International Conference on Emerging Technologies and Factory Automation*, Patras, Greece (Sept. 2007).
- [28] BUSCHMANN, C., AND PFISTERER, D. iSense: A modular hardware and software platform for wireless sensor networks. Tech. rep., 6. Fachgespräch Drahtlose Sensornetze der GI/ITG-Fachgruppe Kommunikation und Verteilte Systeme, 2007.
- [29] BUSCHMANN, C., AND PFISTERER, D. Potenziale und Grenzen spontan etablierter Lokationssysteme. *vfdB-Zeitung Forschung, Technik und Management im Brandschutz 3* (2008), 121–128.
- [30] BUSCHMANN, C., PFISTERER, D., AND FISCHER, S. Kommunikationsaufwand von Verfahren zur Errichtung eines konsistenten Zeitbewusstseins - Eine quantitative Analyse. *PIK - Praxis der Informationsverarbeitung und Kommunikation, Sonderheft Wireless Sensor Networks*, 4 (Apr. 2005).
- [31] BUSCHMANN, C., PFISTERER, D., AND FISCHER, S. Estimating distances using neighborhood intersection. In *11th IEEE International Conference on Emerging Technologies and Factory Automation, Prague, Czech Republic* (Sept. 2006).
- [32] BUSCHMANN, C., PFISTERER, D., AND FISCHER, S. On the relation of neighborhoods and distances. Tech. rep., 5. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, 2006.
- [33] BUSCHMANN, C., WERNER, C., HELLBRÜCK, H., AND FISCHER, S. Nides: Ein verfahren zur multihop-distanzschätzung mittels nachbarschaftsanalyse. In *Proc. of the 15. ITG/GI - Fachtagung Kommunikation in Verteilten Systemen (KiVS)* (2007).
- [34] BUTZ, A., BAUS, J., KRÜGER, A., AND LOHSE, M. A hybrid indoor navigation system. In *IUI '01: Proc. of the 6th international conference on Intelligent user interfaces* (New York, NY, USA, 2001), ACM Press, pp. 25–32.
- [35] CAO, L., TIAN, J., AND LIU, Y. Remote real time automatic meter reading system based on wireless sensor networks, 2008.
- [36] CAPKUN, S., HAMDI, M., AND HUBAUX, J.-P. GPS-free positioning in

- mobile ad hoc networks. In *Proc. of 34th Hawaii International Conference on System Sciences* (2001).
- [37] CARTIGNY, J., AND SIMPLOT, D. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In *HICSS '03: Proc. of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)* (Washington, DC, USA, 2003), IEEE Computer Society, p. 303.
- [38] CHAN, M., ESTÈVE, D., ESCRIBA, C., AND CAMPO, E. A review of smart homes-present state and future challenges. *Comput. Methods Prog. Biomed.* 91, 1 (2008), 55–81.
- [39] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Mobile Computing and Networking* (2001), pp. 85–96.
- [40] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wirel. Netw.* 8, 5 (2002), 481–494.
- [41] CHEN, H., MINENO, H., AND MIZUNO, T. An energy-aware routing scheme with node relay willingness in wireless sensor networks. In *ICICIC '06: Proc. of the First International Conference on Innovative Computing, Information and Control* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 397–400.
- [42] CHINTALAPUDI, K., AND GOVINDAN, R. Localized edge detection in sensor fields. *Ad Hoc Networks* 1, 2-3 (2003), 273–291.
- [43] CHINTALAPUDI, K. K. *Design of a wireless sensor network based structural health monitoring systems*. PhD thesis, University of Southern California, Los Angeles, CA, USA, 2006.
- [44] CHONG, S. K., LOKE, S. W., AND KRISHNASWAMY, S. Wireless sensor networks: from data to context to energy saving. In *International Workshop on Ubiquitous Data Management* (2005), pp. 33–40.
- [45] COALESENSES GMBH. Product brief: iSense solar power system. <http://www.coalesenses.com/uploads/pdf/ProductBriefSolarModule.pdf>.
- [46] COMMITTEE, I. C. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Standard 802.11-1997, New York, NY, 1997.
- [47] COORE, D. *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*. PhD thesis, MIT Department of Electrical Engineering and Computer Science, 1999.
- [48] COORE, D., NAGPAL, R., AND WEISS, R. Paradigms for structure in an amorphous computer. Tech. Rep. AIM-1614, Artificial Intelligence Lab,

- MIT, June 1997.
- [49] CRISTIAN, F. Probabilistic clock synchronization. *Distributed Computing* 3, 3 (Sept. 1989), 146–158.
  - [50] CROSSBOW TECHNOLOGY INC. Mica2Mote. <http://www.xbow.com>.
  - [51] CST GROUP, FU BERLIN. Website of the Embedded Sensor Board ESB 430/2. [http://www.inf.fu-berlin.de/inst/ag-tech/scatterweb\\_net/](http://www.inf.fu-berlin.de/inst/ag-tech/scatterweb_net/).
  - [52] DEVAGUPTAPU, D., AND KRISHNAMACHARI, B. Applications of localized image processing techniques in wireless sensor networks, 2003.
  - [53] DEY, A. K. Context-aware computing: The CyberDesk project. In *AAAI 1998 Spring Symposium on Intelligent Environments* (Palo Alto, 1998), AAAI Press., pp. 51–54.
  - [54] DIALL, H., RAJA, K., DASKALOPOULOS, I., HAILES, S., ROUSSOS, G., TORFS, T., AND VON HOOF, C. Sensor Cube: A modular, ultra-compact, power-aware platform for sensor networks. *Information Processing Society of Japan (IPSJ) Journal* 48, 6 (20070615), 2065–2075.
  - [55] DING, M., CHEN, D., XING, K., AND CHENG, X. Localized fault-tolerant event boundary detection in sensor networks. In *24th Annual IEEE Conference on Computer Communications* (2005).
  - [56] DOHERTY, L., PISTER, K., AND GHAOUI, L. E. Convex position estimation in wireless sensor networks. In *Proc. of the IEEE InfoCom* (2001).
  - [57] DORIGO, M., AND STÜTZLE, T. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
  - [58] DUTTAGUPTA, S., RAMAMRITHAM, K., AND RAMANATHAN, P. Distributed boundary estimation using sensor networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)* (Oct. 2006), pp. 316–325.
  - [59] ELNAHRAWY, E., FRANCISCO, J.-A., AND MARTIN, R. P. Bayesian localization in wireless networks using angle of arrival. In *SenSys '05: Proc. of the 3rd international conference on Embedded networked sensor systems* (2005).
  - [60] ELNAHRAWY, E., LI, X., AND MARTIN, R. P. The limits of localization using signal strength: A comparative study. In *IEEE SECON* (2004).
  - [61] ELSAIFY, A., PADHY, P., MARTINEZ, K., AND ZOU, G. GWMAC - a TDMA based MAC protocol for a glacial sensor network. In *4th ACM PE-WASUN 2007* (2007), Sheridan Printing Company, Inc.
  - [62] ELSON, J., GIROD, L., AND ESTRIN, D. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Oper. Syst. Rev.* 36, SI (2002), 147–163.
  - [63] ELSON, J. E. *Time synchronization in wireless sensor networks*. PhD

- thesis, 2003.
- [64] ESTRIN, D., CULLER, D., PISTER, K., AND SUKHATME, G. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing* 1, 1 (2002), 59–69.
  - [65] ESTRIN, D., GOVINDAN, R., AND HEIDEMANN, J. Embedding the internet: introduction. *Communications of the ACM* 43, 5 (2000), 38–41.
  - [66] FEKETE, S., KRÖLLER, A., BUSCHMANN, C., AND FISCHER, S. Geometric distance estimation for sensor networks and unit disk graphs. In *Geometric Networks and Metric Space Embeddings* (2007).
  - [67] FEKETE, S. P., KRÖLLER, A., PFISTERER, D., FISCHER, S., AND BUSCHMANN, C. Neighborhood-Based Topology Recognition in Sensor Networks. In *Proc. of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004)* (July 2004).
  - [68] FIALA, M., AND BASU, A. Robot navigation using panoramic tracking. *Pattern Recognition* 37, 11 (2004), 2195–2215.
  - [69] FRANKLIN, D., AND FLACHSBART, J. All gadget and no representation makes jack a dull environment. In *Proc. of AAAI 1998 Spring Symposium on Intelligent Environments* (1998). AAAI TR SS-98-02.
  - [70] GABRIEL, K., AND SOKAL, R. A new statistical approach to geographic variation analysis. *Systematic Zoology* 18 (1969), 259–278.
  - [71] GANERIWAL, S., KUMAR, R., ADLAKHA, S., AND SRIVASTAVA, M. Network-wide time synchronization in sensor networks. Tech. rep., Networked and Embedded Systems Lab (NESL), Electrical Engineering Department, UCLA, 2003.
  - [72] GANERIWAL, S., KUMAR, R., AND SRIVASTAVA, M. B. Timing-sync protocol for sensor networks. In *SenSys '03: Proc. of the 1st international conference on Embedded networked sensor systems* (New York, NY, USA, 2003), ACM, pp. 138–149.
  - [73] GANESAN, D., KRISHNAMACHARI, B., WOO, A., CULLER, D., ESTRIN, D., AND WICKER, S. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Tech. rep., UCLA, 2002.
  - [74] GIROD, L., BYCHKOVSKIY, V., ELSON, J., AND ESTRIN, D. Locating tiny sensors in time and space: A case study. In *Proc. of International Conference on Computer Design (ICCD)* (Sept. 2002).
  - [75] GIROD, L., AND ESTRIN, D. Robust range estimation using acoustic and multimodal sensing. In *IEEE/RSI Int. Conf. on Intelligent Robots and Systems (IROS)* (2001).
  - [76] GU, Y., AND HE, T. Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links. In *SenSys '07: Proc. of*

- the 5th international conference on Embedded networked sensor systems* (New York, NY, USA, 2007), ACM, pp. 321–334.
- [77] GUSELLA, R., AND ZATTI, S. The accuracy of the clock synchronization achieved by tempo in berkeley unix 4.3bsd. *IEEE Trans. Softw. Eng.* 15, 7 (1989), 847–853.
- [78] HARTER, A., HOPPER, A., STEGGLES, P., WARD, A., AND WEBSTER, P. The anatomy of a context-aware application. In *MobiCom '99: Proc. of the 5th annual ACM/IEEE international conference on Mobile computing and networking* (New York, NY, USA, 1999), ACM Press, pp. 59–68.
- [79] HASSANEIN, H. J. L. Reliable energy aware routing in wireless sensor networks. In *Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems* (2006).
- [80] HAUSDORFF, F. *Gesammelte Werke, Band 2, Grundzüge der Mengenlehre*, 6 ed. Springer, 2002.
- [81] HE, T., HUANG, C., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHHER, T. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proc. of the 9th annual international conference on Mobile computing and networking* (New York, NY, USA, 2003), ACM Press, pp. 81–95.
- [82] HEINZELMAN, W. R., CH, A., AND BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of the 33rd Annual Hawaii International Conference on System Sciences* (2000), pp. 3005–3014.
- [83] HELLBRÜCK, H., AND FISCHER, S. Towards analysis and simulation of ad-hoc networks. In *Proc. of the 2002 International Conference on Wireless Networks (ICWN02)* (Las Vegas, Nevada, USA, June 2002), IEEE Computer Society Press, pp. 69–75.
- [84] HILL, J., HORTON, M., KLING, R., AND KRISHNAMURTHY, L. The platforms enabling wireless sensor networks. *Commun. ACM* (2004).
- [85] HILL, J., SZEWCZYK, R., WOO, A., HOLLAR, S., CULLER, D., AND PISTER, K. System architecture directions for networked sensors. *SIGPLAN Not.* 35, 11 (2000), 93–104.
- [86] HILL, J. L. *System architecture for wireless sensor networks*. PhD thesis, University of California, Berkeley, 2003.
- [87] HOFMANN-WELLENHOF, B., LICHTENEGGER, H., AND COLLINS, J. *Global Positioning System: Theory and Practice*, 5 ed. Springer, 2001.
- [88] HOWARD, A., MATARIC, M. J., AND SUKHATME, G. Relaxation on a mesh: a formalism for generalized localization. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)* (Oct. 2001), IEEE, pp. 1055–1060.

- [89] HWAN CHANG, J., AND TASSIULAS, R. Energy conserving routing in wireless ad-hoc networks. In *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)* (2000), pp. 22–31.
- [90] IEEE 802.15 WORKING GROUP. Part 4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 1: Add Alternate PHYs. <http://www.ieee802.org/15/pub/TG4.html>.
- [91] IYENGAR, R., AND SIKDAR, B. Scalable and distributed GPS free positioning for sensor networks. In *Proc. of IEEE ICC 2003* (May 2003).
- [92] JÄHNE, B. *Digitale Bildverarbeitung*, 6 ed. Springer, 2005.
- [93] KAHN, J. M., KATZ, R. H., AND PISTER, K. S. J. Emerging challenges: Mobile networking for smart dust. *Journal of Communications and Networks* 2, 3 (Sept. 2000), 188–196.
- [94] KARP, B., AND KUNG, H. T. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proc. of the 6th annual international conference on Mobile computing and networking* (New York, NY, USA, 2000), ACM Press, pp. 243–254.
- [95] KIM, S., PAKZAD, S., CULLER, D., DEMMEL, J., FENVES, G., GLASER, S., AND TURON, M. Wireless sensor networks for structural health monitoring. In *SenSys '06: Proc. of the 4th international conference on Embedded networked sensor systems* (New York, NY, USA, 2006), ACM, pp. 427–428.
- [96] KIM, S., PAKZAD, S., CULLER, D., DEMMEL, J., FENVES, G., GLASER, S., AND TURON, M. Health monitoring of civil infrastructures using wireless sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on* (2007), pp. 254–263.
- [97] KITASUKA, T., NAKANISHI, T., AND FUKUDA, A. Location estimation system using wireless ad-hoc network. In *The 5th International Symposium on Wireless Personal Multimedia Communications* (Oct. 2002).
- [98] KRÖLLER, A., PFISTERER, D., BUSCHMANN, C., FEKETE, S. P., AND FISCHER, S. Shawn: A new approach to simulating wireless sensor networks. In *Design, Analysis, and Simulation of Distributed Systems 2005, Part of the SpringSim 2005* (Apr. 2005).
- [99] KRÖLLER, A., FEKETE, S. P., PFISTERER, D., AND FISCHER, S. Deterministic boundary recognition and topology extraction for large sensor networks. In *Proc. 17th ACM-SIAM Sympos. Discrete Algorithms* (2006), pp. 1000–1009.
- [100] KUHN, F., WATTENHOFER, R., ZHANG, Y., AND ZOLLINGER, A. Geometric ad-hoc routing: of theory and practice. In *PODC '03: Proc. of the twenty-second annual symposium on Principles of distributed computing*

- (New York, NY, USA, 2003), ACM, pp. 63–72.
- [101] LAMPORT, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM* 21, 7 (1978), 558–565.
  - [102] LANGENDOEN, K., AND REIJERS, N. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Netw.* 43, 4 (2003), 499–518.
  - [103] LATECKI, L. J., AND LAKÄMPER, R. Convexity rule for shape decomposition based on discrete contour evolution. *Comput. Vis. Image Underst.* 73, 3 (1999), 441–454.
  - [104] LEMMON, M. D., GANGULY, J., AND XIA, L. Model-based clock synchronization in networks with drifting clocks. In *PRDC '00: Proc. of the 2000 Pacific Rim International Symposium on Dependable Computing* (Washington, DC, USA, 2000), IEEE Computer Society, p. 177.
  - [105] LI, P., AND WEN, Y. Energy harvesting transducer by collecting electromagnetic energy based on ultrasonic horn. In *2006 IEEE International Conference on Information Acquisition* (2007), pp. 550–555.
  - [106] LI, Q., AND RUS, D. Global clock synchronization in sensor networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies* (2004), vol. 1, p. 574.
  - [107] LI, Q., SHI, Z., SHI, J., AND SHI, Z. Swarm intelligence clustering algorithm based on attractor. In *ICNC : International conference on advances in natural computation* (2005), ICNC (3), pp. 496–504.
  - [108] LIN, L., SHROFF, N. B., AND SRIKANT, R. Energy-aware routing in sensor networks: A large system approach. *Ad Hoc Netw.* 5, 6 (2007), 818–831.
  - [109] MAINWARING, A., CULLER, D., POLASTRE, J., SZEWCZYK, R., AND ANDERSON, J. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proc. of the 1st ACM international workshop on Wireless sensor networks and applications* (New York, NY, USA, 2002), ACM Press, pp. 88–97.
  - [110] MANN, R. P., NAMUDURI, K. R., AND PENDSE, R. Energy-aware routing protocol for ad hoc wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking* 5, 5 (2005), 635–644.
  - [111] MANNERMAA, J., KALLIOMAKI, K., MANSTEN, T., AND TURUNEN, S. Timing performance of various gps receivers. In *Proc. of the 1999 Joint Meeting of the European Frequency and Time Forum and the IEEE International Frequency Control Symposium* (1999).
  - [112] MAROTI, M., KUSY, B., SIMON, G., AND LEDECZI, A. The flooding time synchronization protocol. In *SenSys '04: Proc. of the 2nd international conference on Embedded networked sensor systems* (New York, NY, USA,

- 2004), ACM Press, pp. 39–49.
- [113] MAROTI, M., VÖLGYESI, P., DORA, S., KUSY, B., NADAS, A., LEDECZI, A., BALOGH, G., AND MOLNAR, K. Radio interferometric geolocation. In *SenSys'05: Proc. of the 3rd international conference on Embedded networked sensor systems* (2005).
  - [114] MARTINEZ, K., PADHY, P., ELSAIFY, A., ZOU, G., RIDDOCH, A., HART, J. K., AND ONG, H. L. R. Deploying a sensor network in an extreme environment. In *SUTC '06: Proc. of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing - Vol 1 (SUTC'06)* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 186–193.
  - [115] MATTERN, F., AND RÖMER, K. Drahtlose Sensornetze. *Informatik Spektrum* 26, 3 (2003), 191–194.
  - [116] MILLS, D. L. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications* 39 (1991), 1482–1493.
  - [117] MILLS, D. L. A brief history of NTP time: memoirs of an internet timekeeper. *SIGCOMM Comput. Commun. Rev.* (2003).
  - [118] MÜLLER-SCHLOER, C., VON DER MALSBERG, C., AND WÜRTZ, R. P. Organic Computing. Aktuelles Schlagwort. *Informatik Spektrum* 27, 4 (Aug. 2004), 332–336.
  - [119] MOCK, M., FRINGS, R., NETT, E., AND TRIKALIOTIS, S. Continuous clock synchronization in wireless real-time applications. In *SRDS '00: Proc. of the 19th IEEE Symposium on Reliable Distributed Systems (SRDS'00)* (Washington, DC, USA, 2000), IEEE Computer Society, p. 125.
  - [120] MOLINA, F. J., BARBANCHO, J., AND LUQUE, J. Automated meter reading and SCADA application for wireless sensor network. In *ADHOC-NOW* (2003), pp. 223–234.
  - [121] MOSES, R. L., KRISHNAMURTHY, D., AND PATTERSON, R. A self-localization method for wireless sensor networks. *Eurasip Journal on Applied Signal Processing, Special Issue on Sensor Networks* 4 (Mar. 2003), 358–358.
  - [122] MOSES, R. L., AND PATTERSON, R. M. Self-calibration of sensor networks. In *Proc. of SPIE Unattended Ground Sensor Technologies and Applications IV* (Apr. 2002), vol. 4743, pp. 108–119.
  - [123] MOTEIV CORPORATION. Tmote Sky. <http://www.moteiv.com/products-tmotesky.php>, 2005.
  - [124] NAGPAL, R. Organizing a global coordinate system from local information on an amorphous computer. Memo 1666, Massachusetts Institute of Technology (MIT), A.I. Laboratory, Cambridge, Massachusetts, USA,

Aug. 1999.

- [125] NAGPAL, R., SHROBE, H., AND BACHRACH, J. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proc. of 2nd International Workshop on Information Processing in Sensor Networks (IPSN 2003)* (Apr. 2003).
- [126] NASIPURI, A., AND LI, K. A directionality based location discovery scheme for wireless sensor networks. In *1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002)* (Sept. 2002), pp. 105–111.
- [127] NICULESCU, D., AND NATH, B. Ad hoc positioning system (APS). In *Proc. of GLOBECOM, San Antonio, November 2001.* (2001).
- [128] NICULESCU, D., AND NATH, B. Ad hoc positioning system (aps) using aoa. In *Proceedings of INFOCOM 2003* (2003).
- [129] NICULESCU, D., AND NATH, B. Localized positioning in ad hoc networks. *Journal of Ad Hoc Networks* (May 2003).
- [130] NICULESCU, D., AND NATH, B. Position and orientation in ad hoc networks. *Journal of Ad Hoc Networks* (2004).
- [131] NICULESCU, D., AND NATH, B. VOR base stations for indoor 802.11 positioning. In *MobiCom '04: Proc. of the 10th annual international conference on Mobile computing and networking* (New York, NY, USA, 2004), ACM, pp. 58–69.
- [132] NOWAK, R., AND MITRA, U. Boundary estimation in sensor networks: Theory and methods. In *IPSN* (2003), pp. 80–95.
- [133] NOWAK, R., MITRA, U., AND WILLETT, R. Estimating inhomogeneous fields using wireless sensor networks. *IEEE Journal on Selected Areas in Communications* 22, 6 (2004), 999–1006.
- [134] The network simulator ns-2 (v2.29). <http://www.isi.edu/nsnam/ns/>, Oct. 2001.
- [135] ORGANIC COMPUTING INITIATIVE. A novel computing paradigm. <http://www.organic-computing.org>.
- [136] PAEK, J., CHINTALAPUDI, K., AND GOVINDAN, R. A wireless sensor network for structural health monitoring: Performance and experience. In *Proc. of the Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)* (2005).
- [137] PATWARI, N., AND III, A. O. H. Location estimation accuracy in wireless sensor networks. In *Proc. of IEEE Asilomar Conference on Signals and Systems 2002* (Nov. 2002).
- [138] PATWARI, N., AND III, A. O. H. Using proximity and quantized rss for sensor localization in wireless networks. In *2nd ACM International*

- Workshop on Wireless Sensor Networks and Applications (WSNA 2003)* (Sept. 2003).
- [139] PATWARI, N., O'DEA, R. J., AND WANG, Y. Relative location in wireless networks. In *Proc. of IEEE Vehicular Technology Conference* (May 2001).
- [140] PERKINS, C. Adhoc on-demand distance vector (AODV) routing. Request for Comments 3561, Network Working Group, Internet Engineering Task Force, July 2003.
- [141] PFISTERER, D., AND BUSCHMANN, C. Coalescing simulation and embedded WSN application development. In *International Workshop on Sensor Network Engineering (IWSNE'08)* (2008).
- [142] PHYSIKALISCH-TECHNISCHE BUNDESANSTALT. DCF77, 1973. [http://www.ptb.de/en/org/4/44/442/dcf77\\_1\\_e.htm](http://www.ptb.de/en/org/4/44/442/dcf77_1_e.htm).
- [143] POLASTRE, J., SZEWCZYK, R., AND CULLER, D. E. Telos: enabling ultra-low power wireless research. In *IPSN '05: Proc. of the 4th international symposium on Information processing in sensor networks* (2005), pp. 364–369.
- [144] POTTIE, G. J., AND KAISER, W. J. Wireless integrated network sensors. *Commun. ACM* 43, 5 (2000), 51–58.
- [145] PRIYANTHA, N. B., CHAKRABORTY, A., AND BALAKRISHNAN, H. The cricket location-support system. In *MobiCom '00: Proc. of the 6th annual international conference on Mobile computing and networking* (New York, NY, USA, 2000), ACM Press, pp. 32–43.
- [146] RAGHUNATHAN, V., KANSAL, A., HSU, J., FRIEDMAN, J., AND SRIVASTAVA, M. Design considerations for solar energy harvesting wireless embedded systems. In *IPSN '05: Proc. of the 4th international symposium on Information processing in sensor networks* (Piscataway, NJ, USA, 2005), IEEE Press, p. 64.
- [147] RAYKAR, V. C., KOZINTSEV, I., AND LIENHART, R. Position calibration of audio sensors and actuators in a distributed computing platform. In *ACM Multimedia 2003* (Nov. 2003).
- [148] RAYKAR, V. C., KOZINTSEV, I., AND LIENHART, R. Self localization of acoustic sensors and actuators on distributed platforms. In *International Workshop on Multimedia Technologies in E-Learning and Collaboration (ICCV 2003)* (Oct. 2003).
- [149] RÖMER, K. Time synchronization in ad hoc networks. In *MobiHoc '01: Proc. of the 2nd ACM international symposium on Mobile ad hoc networking and computing* (2001).
- [150] RÖMER, K. The lighthouse location system for smart dust. In *ACM/USENIX Conference on Mobile Systems, Applications, and Services (MobiSys)*

- (May 2003), pp. 15–30.
- [151] ROBINSON, D. P., AND MARSHALL, I. W. An iterative approach to locating simple devices in an ad-hoc network. In *Proc. of London Communications Symposium (LCS 2003)* (Sept. 2003).
- [152] RODDEN, T., CHEVERST, K., DAVIES, K., AND DIX, A. Exploiting context in HCI design for mobile systems. In *Workshop on Human Computer Interaction with Mobile Devices* (1998).
- [153] ROUNDY, S., WRIGHT, P. K., AND RABAEY, J. M. *Energy Scavenging for Wireless Sensor Networks: With Special Focus on Vibrations*. Kluwer Academic Publishers, Norwell, MA, USA, 2004.
- [154] RYAN, N. S., PASCOE, J., AND MORSE, D. R. Enhanced reality fieldwork: the context-aware archaeological assistant. In *Computer Applications in Archaeology 1997* (Oxford, Oct. 1998), V. Gaffney, M. van Leusen, and S. Exxon, Eds., British Archaeological Reports, Tempus Reparatum.
- [155] SALLAI, J., BALOGH, G., MAROTI, M., LEDECZI, A., AND KUSY, B. Acoustic ranging in resource-constrained sensor networks. In *International Conference on Wireless Networks* (2004).
- [156] SAVARESE, C., RABAEY, J., AND LANGENDOEN, K. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proc. of the 2002 USENIX Technical Annual Conference* (June 2002), pp. 317–328.
- [157] SAVARESE, C., RABAEY, J. M., AND BEUTEL, J. Locationing in distributed ad-hoc wireless sensor networks. In *International Conference on Accoustic, Speech, and Signal Processing (ICASSP 2001)* (May 2001), vol. 4, pp. 2037–2040.
- [158] SAVVIDES, A., FANG, J., AND LYMBEROPOULOS, D. Using mobile sensing nodes for boundary estimation. In *Workshop on Applications of Mobile Embedded Systems (WAMES)* (June 2004).
- [159] SAVVIDES, A., HAN, C.-C., AND SRIVASTAVA, M. B. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proc. of the 7th annual international conference on Mobile computing and networking* (New York, NY, USA, 2001), ACM Press, pp. 166–179.
- [160] SAVVIDES, A., PARK, H., AND SRIVASTAVA, M. B. The bits and flops of the n-hop multilateration primitive for node localization problems. In *First ACM International Workshop on Wireless Sensor Networks and Application* (Atlanta, GA, USA, Sept. 2002).
- [161] SAVVIDES, A., PARK, H., AND SRIVASTAVA, M. B. The n-hop multilateration primitive for node localization problems. *Mobile Networks and Applications*, 8 (2003), 443–451.
- [162] SCATTERWEB GMBH. ScatterNode. <http://www.scatterweb.com/>, 2005.
- [163] SCHEUERMANN, B., LOCHERT, C., AND MAUVE, M. Implicit Hop-by-

- Hop Congestion Control in Wireless Multihop Networks. *Elsevier Ad Hoc Networks* (2007).
- [164] SCHILIT, B., ADAMS, N., AND WANT, R. Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications* (Santa Cruz, CA, US, 1994).
- [165] SCHILIT, B., AND THEIMER, M. Disseminating active map information to mobile hosts. *IEEE Network* 8, 5 (1994), 22–32.
- [166] SCHILLER, J., LIERS, A., RITTER, H., WINTER, R., AND VOIGT, T. ScatterWeb - low power sensor nodes and energy aware routing. In *HICSS '05: Proc. of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9* (Washington, DC, USA, 2005), IEEE Computer Society, p. 286.3.
- [167] SCHMECK, H. Organic computing – vision and challenge for system design. In *Proc. of the Parallel Computing in Electrical Engineering, International Conference on (PARELEC 2004)* (Los Alamitos, CA, USA, 2004), IEEE Computer Society, pp. 3–3.
- [168] SCHMECK, H. Organic computing - a new vision for distributed embedded systems. *isorc 00* (2005), 201–203.
- [169] SHANG, Y., RUML, W., ZHANG, Y., AND FROMHERZ, M. P. J. Localization from mere connectivity. In *Proc. of 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)* (June 2003), ACM.
- [170] SICHITIU, M., AND VEERARITTIPHAN, C. Simple, accurate time synchronization for wireless sensor networks. *IEEE Wireless Communications and Networking* 2 (2003), 1266–1273.
- [171] SIMIC, S. N., AND SASTRY, S. A distributed algorithm for localization in random wireless networks. *Discrete Applied Mathematics (to appear)* (2003).
- [172] SIMIC, S. N., AND SASTRY, S. Localization in dense random sensor networks. In *IPSN '04: Proc. of the 3rd international conference on Information processing in sensor networks* (2004).
- [173] SINGH, S., AND RAGHAVENDRA, C. S. PAMAS: Power aware multi-access protocol with signalling for ad hoc networks. *ACM Computer Communication Review* 28 (1998), 5–26.
- [174] SIXSMITH, A., AND JOHNSON, N. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Computing* 3, 2 (2004), 42–47.
- [175] SRIVASTAVA, A., SOBAJE, J., POTKONJAK, M., AND SARRAFZADEH, M. Optimal node scheduling for effective energy usage. In *Proc. of the IEEE Workshop on Integrated Management of Power Aware Communications Computing and Networking in Sensor Networks* (2002).

- 
- [176] STORDEUR, M., AND STARK, I. Low power thermoelectric generator-self-sufficient energy supply for micro systems. In *Thermoelectrics, 1997. Proc. ICT '97. XVI International Conference on* (1997), pp. 575–577.
- [177] SU, W., AND AKYILDIZ, I. F. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Trans. Netw.* 13, 2 (2005), 384–397.
- [178] SUNDARAM, N., AND RAMANATHAN, P. Connectivity based location estimation scheme for wireless ad hoc networks. In *Proc. of IEEE Globecom* (Nov. 2002).
- [179] SZEWCZYK, R., MAINWARING, A., POLASTRE, J., ANDERSON, J., AND CULLER, D. An analysis of a large scale habitat monitoring application. In *Proc. of the 2nd international conference on Embedded networked sensor system, SenSys 2004* (Nov. 2004).
- [180] SZEWCZYK, R., POLASTRE, J., MAINWARING, A., AND CULLER, D. Lessons from a sensor network expedition. In *Proc. of the First European Workshop on Sensor Networks (EWSN)* (Jan. 2004).
- [181] TAKENGA, C. M., AND KYAMAKYA, K. Robust positioning system based on fingerprint approach. In *MobiWac '07: Proc. of the 5th ACM international workshop on Mobility management and wireless access* (New York, NY, USA, 2007), ACM, pp. 1–8.
- [182] THULASIRAMAN, P., THULASIRAM, R. K., AND ISLAM, M. T. An ant colony optimization based routing algorithm in mobile ad hoc networks and its parallel implementation. *High performance scientific and engineering computing: hardware/software support* (2004), 267–284.
- [183] TSAI, H.-W., CHU, C.-P., AND CHEN, T.-S. Mobile object tracking in wireless sensor networks. *Comput. Commun.* 30, 8 (2007), 1811–1825.
- [184] TURAU, V., RENNER, C., VENZKE, M., WASCHIK, S., WEYER, C., AND WITT, M. The heathland experiment: Results and experiences. In *Proc. of the REALWSN'05 Workshop on Real-World Wireless Sensor Networks* (Stockholm, Sweden, June 2005).
- [185] TURAU, V., WITT, M., AND WEYER, C. Analysis of a real multi-hop sensor network deployment: The heathland experiment. In *Proc. of the 3rd International Conference on Networked Sensing Systems (INSS'06)* (Chicago, Illinois, USA, June 2006).
- [186] VAN DAM, T., AND LANGENDOEN, K. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *The First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)* (Los Angeles CA, Nov. 2003).
- [187] VAN GREUNEN, J., AND RABAEY, J. Lightweight time synchronization for sensor networks. In *WSNA '03: Proc. of the 2nd ACM international conference on Wireless sensor networks and applications* (New York, NY,

- USA, 2003), ACM Press, pp. 11–19.
- [188] WANT, R., HOPPER, A., FALCAO, V., AND GIBBONS, J. The active badge location system. *ACM Transactions on Information Systems* 10, 1 (1992), 91–102.
- [189] WARD, A., JONES, A., AND HOPPER, A. A new location technique for the active office. *IEEE Personal Communications* 4, 5 (Oct. 1997), 42–47.
- [190] WARNEKE, B., LAST, M., LIEBOWITZ, B., AND PISTER, K. S. J. Smart dust: Communicating with a cubic-millimeter computer. *Computer* 34, 1 (2001), 44–51.
- [191] WHITEHOUSE, C. D. The design of calamari: an ad-hoc localization system for sensor networks. Master’s thesis, University of California, Berkeley (UCB), 2002.
- [192] WHITEHOUSE, K., AND CULLER, D. Calibration as parameter estimation in sensor networks. In *WSNA ’02: Proc. of the 1st ACM international workshop on Wireless sensor networks and applications* (New York, NY, USA, 2002), ACM Press, pp. 59–67.
- [193] WHITEHOUSE, K., AND CULLER, D. Macro-calibration in sensor/actuator networks. *Mobile Networks and Applications Journal (MONET), Special Issue on Wireless Sensor Networks* (June 2003).
- [194] WHITEHOUSE, K., KARLOF, C., AND CULLER, D. A practical evaluation of radio signal strength for ranging-based localization. *ACM Mobile Computing and Communications Review (MC2R), Special Issue on Localization Technologies and Algorithms* (2007).
- [195] WHITEHOUSE, K., KARLOF, C., WOO, A., JIANG, F., AND CULLER, D. The effects of ranging noise on multihop localization: an empirical study. In *The Fourth International Conference on Information Processing in Sensor Networks (IPSN ’05)* (2005).
- [196] WITTENBURG, G., TERFLOTH, K., VILLAFUERTE, F. L., NAUMOWICZ, T., RITTER, H., AND SCHILLER, J. Fence monitoring - experimental evaluation of a use case for wireless sensor networks. In *Proc. of the 4th European Conference on Wireless Sensor Networks (EWSN’07)* (Delft, The Netherlands, Jan. 2007).
- [197] XUE, F., AND KUMAR, P. R. The number of neighbours needed for connectivity of wireless networks. *IEEE Wireless Networks* 10, 2 (2004), 169–181.
- [198] YARVIS, M. D., CONNER, W. S., KRISHNAMURTHY, L., CHHABRA, J., ELLIOTT, B., AND MAINWARING, A. Real-world experiences with an interactive ad hoc sensor network. *icppw 00* (2002), 143.
- [199] YOUNIS, M., YOUSSEF, M., AND ARISHA, K. Energy-aware routing in cluster-based sensor networks. In *MASCOTS ’02: Proc. of the 10th IE-*

- EE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'02)* (Washington, DC, USA, 2002), IEEE Computer Society, p. 129.
- [200] YU, L., WANG, N., AND MENG, X. Real-time forest fire detection with wireless sensor networks. In *Proc. of the 2005 International Conference on Wireless Communications, Networking and Mobile Computing (2005)*, vol. 2, pp. 1214–1217.
- [201] ZHANG, P., SADLER, C. M., LYON, S. A., AND MARTONOSI, M. Hardware design experiences in ZebraNet. In *Proc. of the 2nd international conference on Embedded networked sensor systems (2004)*, ACM Press, pp. 227–238.
- [202] ZHOU, G., HE, T., KRISHNAMURTHY, S., AND STANKOVIC, J. A. Impact of radio irregularity on wireless sensor networks. In *MobiSys '04: Proc. of the 2nd international conference on Mobile systems, applications, and services* (New York, NY, USA, 2004), ACM Press, pp. 125–138.
- [203] ZHOU, J., AND ROURE, D. D. Designing energy-aware adaptive routing for wireless sensor networks. In *The 6th International Conference on ITS Telecommunications (2006)*, pp. 680–685.
- [204] ZUNIGA, M., AND KRISHNAMACHARI, B. Analyzing the transitional region in low power wireless links. In *First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)* (Oct. 2004).
- [205] ZUSSMAN, G., AND SEGALL, A. Energy efficient routing in ad hoc disaster recovery networks. In *IEEE INFOCOM (2003)*, pp. 682–691.



## **B. Persönliche Informationen**



# Lebenslauf

## Privat

---

Carsten Buschmann Geboren am 27.10.1976 in Gifhorn, wohnhaft in Lübeck, ledig  
Eltern Horst und Helga Buschmann (geb. Wölm)

## Ausbildung

---

Aug. 1993 – Jul. 1994 Seirin High School, Osaka, Japan: einjähriger Rotary Schüleraustausch, Unterrichtssprache Japanisch  
Aug. 1989 – Mai 1996 Otto-Hahn-Gymnasium, Gifhorn, Deutschland: Leistungskurse Englisch und Mathematik, Abitur am 21. Mai 1996 (Note: 1,5)  
Okt. 1997 – Aug. 2002 Technische Universität Braunschweig, Deutschland: Diplom und Master in Informatik mit Auszeichnung (Note: 1,0) Nebenfach: Betriebswirtschaftslehre und Unternehmensführung, Auszeichnung für hervorragende akademische Leistungen im Oktober 2001

## Beruflicher Werdegang

---

Aug. 1996 – Sept. 1997 Lebenshilfe, Gifhorn, Deutschland: Zivildienst (Kindergarten für geistig und körperlich behinderte Kinder)  
Sept. 2001 – Dez. 2001 Philips Research Laboratories, Eindhoven, Niederlande: freiwilliges Auslandspraktikum: Konzeption und Implementierung eines Dokumentenmanagementsystems  
Okt. 2002 – Jan. 2005 Technische Universität Braunschweig, Institut für Betriebssysteme und Rechnerverbund, Deutschland: Wissenschaftlicher Mitarbeiter  
seit 01/2005 Universität zu Lübeck, Institut für Telematik, Deutschland: Wissenschaftlicher Mitarbeiter  
seit 07/2005 Coalesenses GmbH, Lübeck, Deutschland: Mitbegründer und Geschäftsführer



## Eigene Publikationen

- [1] BUSCHMANN, C., FEKETE, S. P., FISCHER, S., KRÖLLER, A., AND PFISTERER, D. Koordinatenfreies Lokationsbewusstsein. *it - Information Technology, Themenheft Sensornetze* 47, 4 (Apr. 2005).
- [2] BUSCHMANN, C., AND FISCHER, S. Eigenschaften der Funkschnittstelle in Sensornetzen und Auswirkungen für Anwendungen. Tech. rep., 2. GI/ITG Fachgespräch Sensornetze, Feb. 2004.
- [3] BUSCHMANN, C., AND FISCHER, S. Analyse des Kommunikationsaufwands für konsistentes Zeitbewusstsein. Tech. rep., 4. Fachgespräche Drahtlose Sensornetze, GI/ITG KuVS, 2005.
- [4] BUSCHMANN, C., AND FISCHER, S. Sensornetze: Vom Forschungsthema zum Start-up. *FOCUS MUL* 24, 2 (June 2007).
- [5] BUSCHMANN, C., FISCHER, S., KOBERSTEIN, J., AND LUTTENBERGER, N. Ein Ansatz zur effizienten Softwareentwicklung für drahtlose Sensornetze auf Basis von Shared Information Spaces. Tech. rep., GI/ITG Fachgespräche Systemsoftware für Pervasive Computing, Oct. 2004.
- [6] BUSCHMANN, C., FISCHER, S., KOBERSTEIN, J., AND LUTTENBERGER, N. Towards information centric application development for wireless sensor networks. In *Proc. of the System Support for Ubiquitous Computing Workshop (UbiSys) at the Sixth Annual Conference on Ubiquitous Computing (UbiComp 2004)* (Sept. 2004).
- [7] BUSCHMANN, C., FISCHER, S., KOBERSTEIN, J., LUTTENBERGER, N., AND REUTER, F. SWARMS - Software Architecture for Radio-Based Mobile Self-Organizing Systems. Tech. rep., TKN-03-012, 1. GI/ITG Fachgespräch Sensornetze, July 2003.
- [8] BUSCHMANN, C., FISCHER, S., LUTTENBERGER, N., AND REUTER, F. Middleware for swarm-like collections of devices. *IEEE Pervasive Computing Magazine* 2, 3 (2003).
- [9] BUSCHMANN, C., HELLBRÜCK, H., FISCHER, S., KROELLER, A., AND FEKETE, S. Radio propagation-aware distance estimation based on neighborhood comparison. In *Proc. of the 4th European conference on Wireless Sensor Networks (EWSN)* (2007).
- [10] BUSCHMANN, C., KRÜGER, D., AND FISCHER, S. Lean and robust phenomenon boundary approximation. In *Proc. of the 12th IEEE International Conference on Emerging Technologies and Factory Automation, Patras*,

Greece (Sept. 2007).

- [11] BUSCHMANN, C., MÜLLER, F., AND FISCHER, S. Grid-Based Navigation for Autonomous, Mobile Robots. In *Proc. of 1st Workshop on Positioning, Navigation and Communication 2004 (WPNC'04)* (Mar. 2004), Shaker Publishing.
- [12] BUSCHMANN, C., AND PFISTERER, D. iSense: A modular hardware and software platform for wireless sensor networks. Tech. rep., 6. Fachgespräch Drahtlose Sensornetze der GI/ITG-Fachgruppe Kommunikation und Verteilte Systeme, 2007.
- [13] BUSCHMANN, C., AND PFISTERER, D. Potenziale und Grenzen spontan etablierter Lokationssysteme. *vfdB-Zeitung – Forschung, Technik und Management im Brandschutz 3* (2008), 121–128.
- [14] BUSCHMANN, C., PFISTERER, D., AND FISCHER, S. Experimenting with Computer Swarms: a Mobile Platform based on Blimps. Poster, June 2004.
- [15] BUSCHMANN, C., PFISTERER, D., AND FISCHER, S. From diodes to insights. In *Proc. of the 2nd Workshop on Sensor Networks, INFORMATIK 2005* (Sept. 2005).
- [16] BUSCHMANN, C., PFISTERER, D., AND FISCHER, S. Kommunikationsaufwand von Verfahren zur Errichtung eines konsistenten Zeitbewusstseins - Eine quantitative Analyse. *PIK - Praxis der Informationsverarbeitung und Kommunikation, Sonderheft Wireless Sensor Networks*, 4 (Apr. 2005).
- [17] BUSCHMANN, C., PFISTERER, D., AND FISCHER, S. Estimating distances using neighborhood intersection. In *Proc. of the 11th IEEE International Conference on Emerging Technologies and Factory Automation, Prague, Czech Republic* (Sept. 2006).
- [18] BUSCHMANN, C., PFISTERER, D., AND FISCHER, S. On the relation of neighborhoods and distances. Tech. rep., 5. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, 2006.
- [19] BUSCHMANN, C., PFISTERER, D., FISCHER, S., FEKETE, S. P., AND KRÖLLER, A. SpyGlass: Taking a closer look into sensor networks. Poster, Nov. 2004.
- [20] BUSCHMANN, C., PFISTERER, D., FISCHER, S., FEKETE, S. P., AND KRÖLLER, A. SpyGlass: A Wireless Sensor Network Visualizer. *ACM SIGBED Review 2*, 1 (2005).
- [21] BUSCHMANN, C., WERNER, C., HELLBRÜCK, H., AND FISCHER, S. NIDES: Ein Verfahren zur Multihop-Distanzschätzung mittels Nachbarschaftsanalyse. In *Proc. of the 15. ITG/GI - Fachtagung Kommunikation in Verteilten Systemen (KiVS)* (2007).
- [22] FEKETE, S. P., KRÖLLER, A., PFISTERER, D., FISCHER, S., AND BUSCHMANN, C. Neighborhood-Based Topology Recognition in Sensor

- Networks. In *Proc. of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004)* (July 2004).
- [23] KOBERSTEIN, J., LUTTENBERGER, N., BUSCHMANN, C., AND FISCHER, S. Shared Information Spaces for Small Devices: The SWARMS Software Concept. In *Proc. of the Workshop on Sensor Networks at Informatik 2004* (Sept. 2004).
- [24] KRÖLLER, A., PFISTERER, D., BUSCHMANN, C., FEKETE, S. P., AND FISCHER, S. Shawn: A new approach to simulating wireless sensor networks. In *Proc. of the Conference on Design, Analysis, and Simulation of Distributed Systems 2005, Part of the SpringSim 2005* (Apr. 2005).
- [25] KRÜGER, D., BUSCHMANN, C., AND FISCHER, S. Discovering topologies in wireless sensor networks. Tech. rep., 5. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, 2006.
- [26] KRÜGER, D., BUSCHMANN, C., AND FISCHER, S. Topology recognition in Wireless Sensor Networks. Poster, June 2006.
- [27] KRÜGER, D., BUSCHMANN, C., AND FISCHER, S. Object tracking on graph structures. In *Proc. of the 3rd IEEE European Conference on Smart Sensing and Context (EuroSSC'08)* (2008).
- [28] PFISTERER, D., AND BUSCHMANN, C. Coalescing simulation and embedded wsn application development. In *Proc. of the International Workshop on Sensor Network Engineering (IWSNE'08)* (2008).
- [29] PFISTERER, D., BUSCHMANN, C., HELLBRÜCK, H., AND FISCHER, S. Data-type centric middleware synthesis for wireless sensor network application development. In *Proc. of the Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006)* (2006).
- [30] PFISTERER, D., BUSCHMANN, C., HELLBRÜCK, H., AND FISCHER, S. Supporting WSN application development using data type-centric middleware synthesis. Tech. rep., 5. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, 2006.
- [31] PÖHLSSEN, S., BUSCHMANN, C., AND WERNER, C. Integrating a decentralized Web Service discovery system into the internet infrastructure. In *Proc. of the 6th IEEE European Conference on Web Services (ECOWS'08)* (Dublin, Ireland, Nov. 2008).
- [32] SCHMIDT, S., BUSCHMANN, C., AND FISCHER, S. Sicherheit in Sensornetzen am Beispiel von SWARMS. Tech. rep., TKN-03-012, 1. GI/ITG Fachgespräch Sensornetze, July 2003.
- [33] WERNER, C., BUSCHMANN, C., BRANDT, Y., AND FISCHER, S. Compressing SOAP messages by using pushdown automata. In *Proc. of the IEEE International Conference on Web Services* (Sept. 2006).
- [34] WERNER, C., BUSCHMANN, C., BRANDT, Y., AND FISCHER, S. XML

- compression for Web Services on resource-constrained devices. *International Journal of Web Services Research* 5, 3 (2008).
- [35] WERNER, C., BUSCHMANN, C., AND FISCHER, S. Compressing SOAP messages by using differential encoding. In *Proc. of the IEEE International Conference on Web Services* (July 2004).
- [36] WERNER, C., BUSCHMANN, C., AND FISCHER, S. WSDL-driven SOAP compression. *International Journal of Web Services Research* 2, 1 (2005).
- [37] WERNER, C., BUSCHMANN, C., AND FISCHER, S. Advanced data compression techniques for SOAP Web Services. In *Modern Technologies in Web Services Research*, L.-J. Zhang, Ed. IGI Publishing, 2007, pp. 76–97.
- [38] WERNER, C., BUSCHMANN, C., AND FISCHER, S. Efficient encodings for web service messages. In *Web Services Research and Practices*, L.-J. Zhang, Ed. CyberTech Publishing, 2008.
- [39] WERNER, C., BUSCHMANN, C., JÄCKER, T., AND FISCHER, S. Bandwidth and latency considerations for efficient SOAP messaging. *International Journal of Web Services Research* 3, 1 (2006).
- [40] WERNER, C., BUSCHMANN, C., JÄCKER, T., AND FISCHER, S. Efficient transport bindings for Web Service messages. In *Web Services Research and Practices*, L.-J. Zhang, Ed. CyberTech Publishing, 2008.
- [41] WERNER, C., JÄCKER, C. B. T., AND FISCHER, S. Enhanced transport bindings for efficient SOAP messaging. In *Proc. of the IEEE International Conference on Web Services* (July 2005).