Koordinatenfreies Lokationsbewusstsein

Localization without Coordinates

Sándor P. Fekete, Alexander Kröller, Technische Universität Braunschweig,
Carsten Buschmann, Stefan Fischer, Dennis Pfisterer, Universität Lübeck

Zusammenfassung Eines der fundamentalen Probleme in Sensornetzwerken besteht darin, ein Bewusstsein für die Position eines Knotens im Netz zu entwickeln. Dabei wird fast immer davon ausgegangen, dass dies durch die Zuweisung von Koordinaten zu erfolgen hat. In diesem Artikel wird auf theoretischer, praktischer und simulativer Ebene ein kritischer Blick auf entsprechende Verfahren geworfen, und es werden Grenzen aufgezeigt. Es wird ein Ansatz vorgestellt, mit dem in der Zukunft eine abstrakte Form von Lokationsbewusstsein etabliert werden kann, und es wird gezeigt, wie Anwendungen davon massiv profitieren können. Er basiert auf einer graphenbasierten Modellierung des Netzes: Knoten mit bestimmten topologischen oder Umwelteigenschaften werden zu Clustern zusammengefasst, und Clusternachbarschaften dann als Graphen modelliert. ►►► **Summary** Localization is one of the fundamental issues in sensor networks. It is almost always assumed that it must be solved by assigning coordinates to the nodes. This article discusses positioning algorithms from a theoretical, practical and simulative point of view, and identifies difficulties and limitations. Ideas for more abstract means of location awareness are presented and the resulting possible improvements for applications are shown. Nodes with certain topological or environmental properties are clustered, and the neighborhood structure of the clusters is modeled as a graph.

KEYWORDS C.2.2 [Network Protocols], F.2 [Analysis of Algorithms and Problem Complexity], G.2.2 [Graph Theory] sensor networks, localization, cluster, graphs, routing

1 Einleitung

Sensornetze stellen einen vergleichsweise jungen Forschungsbereich dar, in dem noch viele grundlegende Fragen ungeklärt sind. Eine dieser Fragen ist, wie Sensorknoten ein Bewusstsein für ihre Lage im Netz entwickeln können, und wie dieses in Anwendungen ausgenutzt werden kann. Dieses Bewusstsein kann beispielsweise darin bestehen, dass jeder Knoten seine Koordinaten in einem gemeinsamen System kennt. Da diese Form intuitiv sinnvoll erscheint, existiert eine Fülle von heuristischen Verfahren, die Koordinaten berechnen. Andererseits gibt es derzeit praktisch keine alternativen Ansätze. Ziel dieses Artikels ist es, diese intuitive Entscheidung in Frage zu stellen und Alternativen aufzuzeigen.

1.1 Überblick

Inhaltlich lässt sich dieser Artikel durch drei Kernthesen charakterisieren, die im Folgenden näher ausgeführt werden:

- Die Zuweisung von Koordinaten stellt ein im komplexitätstheoretischen Sinne schweres Problem dar, sodass es voraussichtlich keine fehlerfreien Algorithmen für Sensornetze geben kann. (Abschnitt 3.1)
- Gängige Lokationsalgorithmen schlagen auch in realistischen Anwendungsszenarien fehl. (Abschnitt 3.2)
- Es ist möglich, Lokationsinformation zu generieren, die unabhängig von Koordinaten ist. Für bestimmte Anwendungen ist dieses abstrakte Lokationsbewußtsein dem klassischen Ansatz sogar überlegen. (Abschnitt 4)

Im Folgenden wird zunächst in Abschnitt 1.2 ein Anwendungsszenario vorgestellt, das durchgehend als Beispiel dient. Abschnitt 2 geht auf koordinatenbasierte Lo-

kationsalgorithmen ein. Dazu werden Klassifikationen für Verfahren vorgestellt und die Anforderungen an die zu berechnenden Lösungen diskutiert. In Abschnitt 3 wird das zugrunde liegende Problem zunächst theoretisch analysiert und auf Auswirkungen in der Praxis eingegangen. Außerdem werden vier Algorithmen vorgestellt und simulativ untersucht, um nachzuweisen, dass die beschriebenen Probleme in der Realität tatsächlich auftreten. Abschnitt 4 skizziert, wie neue Ansätze genutzt werden können, um Alternativen zu den bekannten Algorithmen zu entwickeln. Es wird ein Verfahren umrissen, das topologische oder Umwelteigenschaften zum Clustern von Knoten nutzt, und sich daraus ergebende Clusternachbarschaften mit Hilfe eines Graphen modelliert.

1.2 Szenario

In der Literatur ist es üblich, Algorithmen auf einem Sensornetz zu testen, bei dem die Knoten gleichverteilt auf einem konvexen Gebiet, etwa einem Kreis oder Quadrat, platziert sind. Viele Verfahren verwenden Anker, das heißt Knoten, die ihre Koordinaten bereits im Vorfeld kennen. Diese folgen im Allgemeinen derselben Verteilung wie die übrigen „normalen“ Knoten. Viele anwendungsorientierte Szenarien sind weniger gutartig. Wie wir im Folgenden zeigen, liefern existierende Verfahren selbst unter leichten Abschwächungen der obigen Annahmen, insbesondere über die topologische Struktur des betrachteten Gebiets, unbrauchbare Ergebnisse.

Im Folgenden wird ein derartiges Szenario verwendet. Darin sind die Sensorknoten zufällig über ein Gebiet verteilt, das aus Straßen besteht. Es wird keine Gleichverteilung verwendet – einige Straßen sind dichter besetzt als andere. Ankerknoten sind nur in einigen Bereichen vertreten. Wir sind davon überzeugt, dass dieser Aufbau praktisch relevant ist, da es eine Vielzahl von Szenarien abdeckt. Abgesehen von Straßen einer Stadt sind dies z. B.:

- Kanalisationen.
- Von wenigen Fahrzeugen ausgebrachte Knoten.
- Über einer Seenlandschaft abgeworfene Knoten.
- Sensornetze, in denen in mehreren Gebieten alle Knoten ausgefallen sind und Löcher erzeugt haben.

Bild 1 zeigt das Sensornetz, das wir für Tests verwenden. Es sind nicht alle Kanten des Kommunikationsgraphen dargestellt, sondern aus Gründen der Übersichtlichkeit nur ein darunter liegender Teilgraph. Das Netzwerk besteht aus 2200 Knoten mit durchschnittlich jeweils 50 Nachbarn. Die schwarzen Kreise markieren die 200 Ankerknoten.

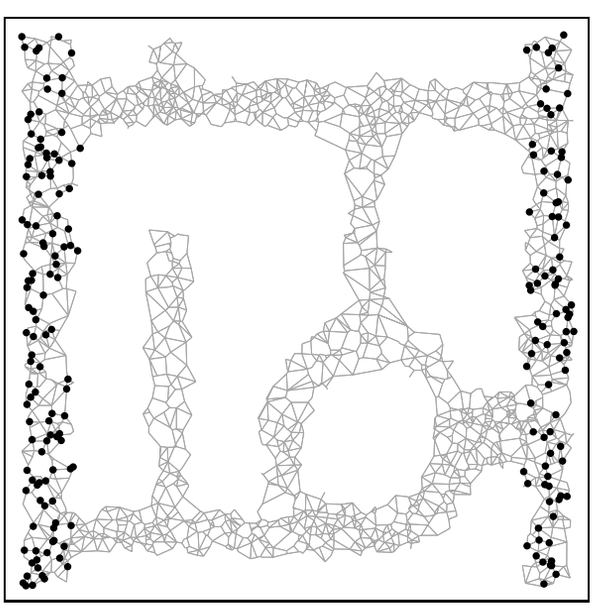

Bild 1 Beispielnetzwerk mit markierten Ankerknoten.

Die Ankerknoten befinden sich an zwei Rändern. Da Lokationsalgorithmen oft die Position eines Knotens anhand einiger umliegender Anker bestimmen, bilden sich zwischen Ankern zwangsläufig Gruppen von Knoten, die anhand der selben Ankermenge verortet werden. Da diese Gruppen algorithmisch vollständig unabhängig sind, ist es sinnvoll, die Vorgänge innerhalb einer solchen Gruppe zu untersuchen. Das verwendete Beispielnetz steht für eine einzelne Gruppe in einem größeren Netzwerk, in dem Ankerknoten aus globaler Sicht regelmäßig verteilt sind.

Dieser Artikel konzentriert sich auf Verfahren, die ein Sensornetz selbst ausführen kann. Algorithmen, die eine zentrale Recheninstanz benötigen, werden daher nicht betrachtet. Fragen der Laufzeit und Nachrichtenkodierung werden ebenfalls ignoriert, da es hier ausschließlich um die Güte der produzierten Lösungen geht.

2 Lokationsbewusstsein durch Koordinaten

Das klassische Lokationsproblem besteht darin, jedem Knoten eine Position im zwei- oder dreidimensionalen Raum zuzuordnen. Wenn ein externes „echtes“ Koordinatensystem vorgegeben ist, sollen die errechneten Positionen den wirklichen möglichst genau entsprechen. Im anderen Fall spricht man von virtuellen Koordinaten, die nur Konsistenzbedingungen erfüllen sollen.

Die Rahmenbedingungen für ein Lokationsverfahren entsprechen denen, die bei Sensornetzen üblicherweise für jeden Algorithmus gelten: Ein Verfahren sollte

- möglichst wenig externe Infrastruktur benötigen,

- mit möglichst wenig Kommunikation auskommen,
- möglichst geringe Anforderungen an die Ressourcen der Knoten, etwa Prozessor und Speicher, stellen und
- ohne externe Konfiguration im Rahmen des Ausbringens der Knoten auskommen.

Ein triviales Beispiel für ein Positionierungsverfahren besteht also darin, jeden Sensorknoten mit einem GPS-Empfänger auszustatten. Der Kommunikationsaufwand ist dabei optimal, da die Knoten keinerlei Nachrichten austauschen.

2.1 Klassifikation

Lokationsverfahren lassen sich anhand mehrerer Kriterien kategorisieren. Eine Gliederung wird nach dem Kriterium möglich, ob ein Verfahren die Messung der Abstände zwischen benachbarten Knoten vorsieht. So könnten die Knoten mit spezieller Hardware ausgestattet sein, die die Abstandsbestimmung zu anderen Geräten in Kommunikationsreichweite ermöglicht, z. B. mittels Ultraschall oder Infrarot. Solche Abstandsmessungen sind jedoch im Allgemeinen mit Messfehlern von bis zu 30% behaftet [4; 15]. Während die eine Gruppe der Verfahren solche Messungen vorsieht, um dennoch differenziertere Abstandsinformationen in der Größenordnung unterhalb des Kommunikationsradiuses einzusetzen, verzichtet die andere darauf und beschränkt sich auf Abstandsabschätzungen mithilfe des Kommunikationsradiuses.
Durch die Zahl der Ankerknoten wird eine weitere Unterteilung möglich. Anker sind spezielle Knoten, die ihre Position kennen, z. B. durch GPS. Sie dienen anderen Knoten als Referenzpunkte. Drei Gruppen von Verfahren lassen sich so unterscheiden:

- Diejenigen, die völlig ohne Anker arbeiten, die also nur eine konsistente Einordnung der Knoten in ein gemeinsames, virtuelles Koordinatensystem anstreben,
- sowie diejenigen, in denen jeder Knoten mehrere Anker mit bekannten Koordinaten in seinem Kommunikationsradius hat.
- Dazwischen liegen die Verfahren, in denen die meisten Knoten keinen Anker direkt hören, sondern Nachrichten von Ankern über mehrere *Hops* zu den Knoten gelangen.

Die meisten auf Ankern basierenden Lokationsverfahren lassen sich in drei Phasen unterteilen:

(1) Bestimmung des Abstandes zu verschiedenen Ankern, d. h. zu bestimmten Koordinatenpunkten.
(2) Berechnung der eigenen Position aus diesen Entfernungen und Punkten oder Winkeln dazwischen.
(3) Verfeinerung der eigenen Positionsschätzung durch lokale Heuristiken.

In einigen Algorithmen fallen einzelne Phasen weg, insbesondere die dritte. Andere verwenden ausschließlich diese, da sie sehr gut parallelisierbar ist. Im Allgemeinen ist es möglich, die einzelnen Phasen verschiedener Algorithmen auszutauschen [8].

2.2 Konsistenz und Faltung

Ein entscheidendes Kriterium, das eine Koordinatenzuweisung erfüllen sollte, ist Konsistenz. Darunter fallen die folgenden zwei Forderungen:

(C1) Benachbarte Knoten bekommen nah beieinander liegende Koordinaten.
(C2) Nicht benachbarte Knoten bekommen Koordinaten, die einen Mindestabstand einhalten.

Von einer Faltung spricht man, wenn Knoten, die in Wirklichkeit weit entfernt sind, ähnliche Koordinaten bekommen. Dadurch werden in der räumlichen Struktur des Netzes zwei Bereiche „übereinander gefaltet“.

Das erste Kriterium lässt sich mit verteilten Algorithmen ohne Schwierigkeiten überprüfen, und es gibt sehr viele verteilte Verbesserungsheuristiken, wie beispielsweise *Spring Embedder* [12].

Das zweite Kriterium stellt im Kontext verteilter Systeme die eigentliche Herausforderung dar. Es ist den Knoten im Allgemeinen nicht möglich, die gesamte Netzwerkstruktur mit allen Koordinaten abzuspeichern oder auf energieeffiziente Weise zu übertragen. Da die Knoten also nicht wissen, ob weiter entfernte Knoten dieselben Koordinaten speichern, gibt es keinen einfachen Test, mit dem Faltungen entdeckt werden können.

Viele Algorithmen setzen auf Ankerknoten, um Faltungsfreiheit zu erzeugen. Oft wird im eigentlichen Verfahren ausschließlich das erste Kriterium berücksichtigt. Dadurch, dass die Ankerknoten ihre Position nicht verändern dürfen, wird das zweite Kriterium implizit eingebracht. Derartige Annahmen sind aber sehr fragil. Sie verlieren ihre Gültigkeit, wenn Abstandsschätzungen tendenziell immer etwas zu groß oder zu klein sind, es ankerfreie Gebiete im Netz gibt oder die Positionsbestimmung der Anker ungenau ist.

3 Problemanalyse

Der folgende Abschnitt untersucht drei wesentliche Punkte, die im Zusammenhang mit Lokationsalgorithmen beachtet werden müssen: Die theoretische Komplexität, praktische Erwägungen und die Qualität der von gängigen Algorithmen produzierten Koordinaten.

3.1 Komplexität

Gerade im Zusammenhang mit Konsistenz gibt es eine Reihe von Negativresultaten, die theoretisch bewiesen sind.

In [3] wird das Unit Disk Graph Recognition Problem untersucht. Dabei ist der Kommunikationsgraph ohne Abstandsschätzungen gegeben, und es wird eine

konsistente Koordinatenzuweisung gesucht. Dafür müssen Nachbarn unterhalb einer frei wählbaren Maximaldistanz M und Nichtnachbarn mit einer größeren Distanz platziert werden. Es wird gezeigt, dass bereits dieses grundlegende Problem $\mathcal{NP}$-vollständig ist.

Eine Möglichkeit, $\mathcal{NP}$-vollständige Probleme anzugehen, liegt in Approximationen. So kann zugelassen werden, dass nicht benachbarte Knoten nur einen Mindestabstand von $d \cdot M < M$ einhalten müssen. In [7] wird gezeigt, dass dieses Problem für $d \geq \sqrt{2/3} + \varepsilon$ $\mathcal{NP}$-vollständig ist, wobei ε für steigende Knotenzahlen gegen 0 konvergiert. Damit kann ein polynomielles Approximationsschema nur existieren, wenn $\mathcal{P} = \mathcal{NP}$ gilt.

Ein weiteres Problem ist UNIT DISK GRAPH RECONSTRUCTION. Hierbei wird nicht nur der Kommunikationsgraph vorgegeben, sondern zusätzlich das Ergebnis einer fehlerfreien Abstandsmessung benachbarter Knoten. Gesucht ist eine Koordinatenzuweisung, die diese Abstände einhält und nicht benachbarten Knoten einen gewissen Mindestabstand zuweist. Auch hier kann wieder gezeigt werden, dass dieses Problem $\mathcal{NP}$-vollständig ist [1] und somit voraussichtlich von einem Sensornetzwerk nicht zu lösen ist.

Zu letzterem Problem existiert auch ein positives Resultat: In [5] wird eine Klasse von Netzwerken identifiziert, für die eine Lokalisierung mit polynomiellem Aufwand möglich ist. Dabei muss unter anderem eine fehlerfreie Abstandsmessung vorliegen, bei der sämtliche zulässigen Lösungen durch Drehung und Translation auseinander hervorgehen. Zusätzlich wird gezeigt, dass die meisten Zufallsgraphen diese Eigenschaft haben. Leider ist dieses Resultat nicht auf praktische Szenarien anwendbar, da bereits kleinste Fehler in der Abstandsmessung die benötigten Voraussetzungen zerstören und dadurch das Lokalisierungsproblem wieder $\mathcal{NP}$-vollständig wird.

Insgesamt ist also davon auszugehen, dass auch weiterhin nur Heuristiken zur Verfügung stehen, die unter ungünstigen Umständen inkonsistente Resultate produzieren.

3.2 Praxis

Nach der theoretischen Untersuchung werden nun praktische Aspekte diskutiert. Zuerst wird auf die für Anker benötigten Geräte eingegangen. Danach wird beispielhaft gezeigt, wie Inkonsistenzen in der Positionierung Anwendungen empfindlich beeinträchtigen können.

3.2.1. Implementierung von Ankern Die Verwendung von Ankerknoten bringt einige praktische Probleme mit sich. Sie benötigen eine externe Quelle zur Bestimmung ihrer Position. Typischerweise werden dafür GPS-Empfänger eingesetzt. Die benötigte Hardware ist derzeit noch sehr groß, teuer und energiehungrig. Es ist nach allgemeiner Auffassung nicht davon auszugehen, dass zukünftige Hardware diese Probleme ausräumen kann. Außerdem benötigen GPS-Empfänger Verbindungen zu geostationären Satelliten. Damit scheiden sie für den Einsatz unter der Erde, im Meer, in Gebäuden oder auf anderen Planeten völlig aus. Somit sind mehrere relevante Einsatzgebiete nicht abgedeckt, was zeigt, dass ankerbasierte Verfahren für viele wichtige praktische Zwecke nicht ausreichend sind.

3.2.2. Anwendung Eine Beispielanwendung, die Positionierungsinformationen verwendet, ist *GeoRouting*. Dabei werden Datenpakete nicht zu einem speziellen Knoten geschickt, sondern an eine bestimmte Position. Üblicherweise wird implizit vorausgesetzt, dass die Lokationsinformation grundsätzlich in Form von Koordinaten vorliegt. Eine Schwierigkeit, mit der viele koordinatenbasierte *GeoRouting*-Verfahren zu kämpfen haben, liegt in dem Umstand begründet, dass der geometrisch kürzeste Weg nicht unbedingt der beste ist, gerade wenn Fragen der Energieeffizienz, Latenz oder Ausfallsicherheit eine Rolle spielen. Die zweite offensichtliche Schwierigkeit liegt in Faltungen. Ein Datenpaket kann hier wie bei topologischen Löchern in einem lokalen Optimum enden, also bei einem Knoten, der keinen Nachbarn mehr hat, der näher am Ziel liegt als er selbst, vom Ziel aber noch weit entfernt ist. In solchen Situationen werden verschiedene Heuristiken eingesetzt, wie z. B. in [2] beschrieben. Ein denkbarer Ausweg könnte auch die lokale Verwendung globalen Wissens sein.

Ein weiterer wichtiger Verwendungszweck von Lokationsinformationen ist, mit Hilfe der Position des Knotens zusammen mit anderen Informationen auf seinen Kontext zu schließen, und so von ihm gelieferte Daten mit der Situation in Relation zu setzen. Wenn ein Sensornetz beispielsweise dazu eingesetzt wird, ein bestimmtes Phänomen zu überwachen und bei einem signifikanten Anstieg eines Sensorwertes einen Alarm abzusetzen, kann eine Faltung katastrophal sein: So würden etwa bei einer Mittelwertsberechnung über ein geographisches Gebiet auch Knoten einbezogen, die weit entfernt liegen und nur durch die Faltung ähnliche Koordinaten haben. Dadurch verändert sich der Mittelwert kaum, wenn das Phänomen nur im angefragten Gebiet auftritt, und kann so fälschlicherweise unter der vorgegebenen Alarmschranke bleiben.

Für viele Anwendungen ist es wichtiger, ob Informationen von einer abstrakten, symbolischen Lokation wie „Kreuzung X“ stammen, als von der Position (173, 25; – 35, 89). Auch sollte der Lokationsbegriff in der Lage sein, die Gruppierung von Knoten sowie die lokationsbasierte Adaption von Algorithmen und Anwendungen zu unterstützen.

Eine Diskussion grundlegender Lokationsmodelle sowie der Vorschlag für ein umfassendes Lokationsmodell, das symbolische und geometrische Lokationen kombiniert, findet sich in [9]. Das dort

beschriebene Modell eignet sich jedoch nicht für den Einsatz in Sensornetzwerken, da es aufgrund seiner Allgemeinheit sehr schwergewichtig ist. Wir stellen daher in Abschnitt 4.3 eine alternative, koordinatenfreie Form von Lokationsbewusstsein für Sensornetzwerke vor, bei der die verschiedenen Kritikpunkte und Anforderungen aufgegriffen werden.

3.3 Simulation

Um zu belegen, dass die beschriebenen Probleme keine theoretischen Sonderfälle sind, die in „realen" Netzen nicht auftreten, werden vier prominente Lokationsalgorithmen herangezogen. Wir simulieren das Beispielszenario (Bild 1) und zeigen, dass tatsächlich Faltungen und Inkonsistenzen entstehen.

3.3.1. Ein ankerbasierter zweiphasiger Algorithmus *Ad-Hoc Positioning* [10] ist ein Beispiel für einen zweiphasigen Algorithmus, der Ankerknoten und Distanzmessungen verwendet. Mangels dritter Phase stehen einem Knoten die Koordinaten seiner Nachbarn nicht zur Verfügung. Daher ist eine hohe Genauigkeit der Schätzung von Entfernungen zu Ankerknoten entscheidend. Hierbei wird, beginnend bei Ankerknoten, mittels Triangulierung über je zwei oder drei Zwischenknoten die eigene Distanz zu Ankern bestimmt. Eine perfekte Abstandsmessung und günstige Verteilung der Knoten vorausgesetzt, ist diese Schätzung fehlerfrei.

Nachdem ein Knoten seine Entfernungen zu mehreren Ankern kennt, berechnet er seine Position. Dafür wird *Multilateration* verwendet, wobei ein quadratisches Gleichungssystem linearisiert und dann eine *Least-Squares*-Lösung berechnet wird. Bei perfekten Messungen und günstigen Knotenverteilungen ergibt sich tatsächlich die korrekte Position.

Bild 2 zeigt das Ergebnis dieses Verfahrens bei einer Standardabweichung in Distanzmessungen von 1%, also weit weniger, als in der Praxis erreichbar ist. Es ist offensichtlich, dass dieses Verfahren in der Nähe der Anker gute Positionen berechnet; allerdings summieren sich Fehler so schnell auf, dass schon nach wenigen Schritten die berechneten Koordinaten völlig unbrauchbar sind. Keines der beiden Kriterien C1 und C2 wird erfüllt.

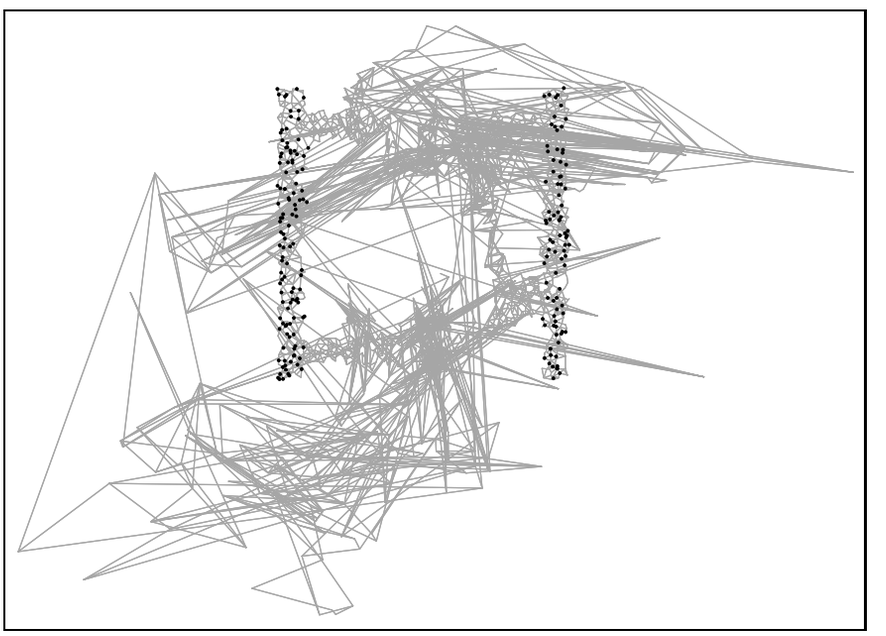

Bild 2 *Ad-hoc Positioning* [10].

3.3.2. Ankerbasierte dreiphasige Algorithmen Als Verbesserung bietet sich die Verwendung einer dritten Phase an. Zwei prominente Vertreter dreiphasiger Algorithmen sind *Robust Positioning* [13] und *N-Hop Multilateration* [14]. Bei beiden berechnen Knoten ihre Position über wiederholte *Multilateration* zu ihren direkten Nachbarn, um verteilt zu einer konsistenten Lösung zu konvergieren.

Die Unterschiede liegen im Wesentlichen in der Berechnung der Startlösung. *Robust Positioning* benötigt keine Distanzmessungen,

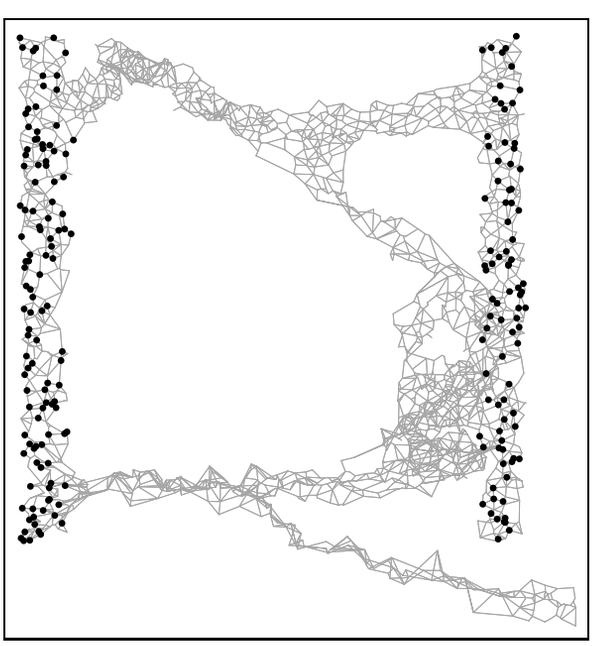

Bild 3 *Robust Positioning* [13].

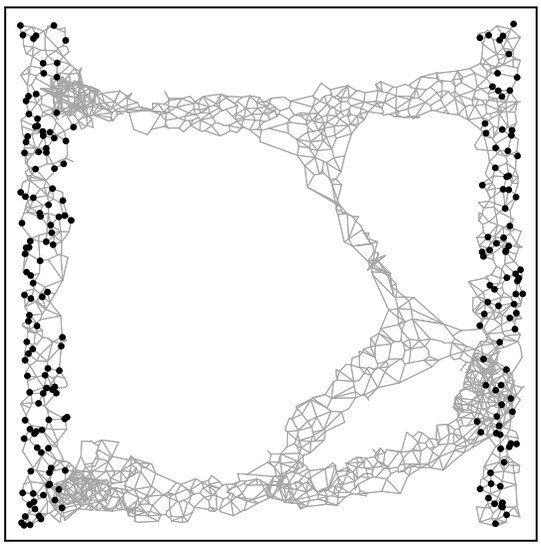

Bild 4 *N-Hop Multilateration* [14].

sondern verwendet *DV-Hop* in der ersten Phase. Dabei bestimmen die Ankerknoten untereinander ein durchschnittliches Verhältnis von euklidischem und *Hop*-Abstand. Dieses Verhältnis verwenden die übrigen Knoten, um aus einem *Hop*-Abstand einen euklidischen Abstand zu berechnen. Die Berechnung der Position läuft danach über *Multilateration*. *N-Hop Multilateration* berechnet den Abstand zu einem Anker als die Summe von Distanzmessungen über einen kürzesten Weg. Die Position wird berechnet, indem ein Knoten um jeden bekannten Ankerpunkt ein Quadrat (bzw. einen Würfel) mit der Distanzschätzung als halber Kantenlänge legt, und einen Punkt bestimmt, der im Schnitt aller Quadrate liegt.

Die berechneten Lösungen dieser beiden Algorithmen sind in den Bildern 3 und 4 dargestellt. Die dritte Phase eliminiert Ausreißer effektiv, wodurch das Kriterium C1 bei beiden Verfahren weitgehend erfüllt wird. Im Bereich der Ankerknoten ist der Positionierungsfehler sehr gering. Beide Verfahren haben massive Schwierigkeiten mit der Sackgasse im Netzwerk. Da sie nicht berechnen können, in welcher Richtung sie verläuft, ist sie in beiden Fällen falsch platziert. Mangels direkter Berücksichtigung des Kriteriums C2 entstehen hier fast zwangsläufig Faltungen. Diese Probleme verstärken sich, wenn die Knoten eine ungeschickte Auswahl an Ankern für die Positionierung treffen, oder die Verbindung zu Ankern nicht gradlinig verläuft, sodass Entfernungsschätzungen sehr ungenau sind.

Diese Ergebnisse legen die Schlussfolgerung nahe, dass die Verwendung von Ankern und entsprechenden Verfahren nur dann sinnvoll ist, wenn sichergestellt werden kann, dass jeder Knoten hinreichend viele Ankerknoten in näherer Umgebung hat. Sobald ankerfreie Bereiche existieren, ist Faltungsfreiheit nicht mehr gewährleistet.

3.3.3. Ein ankerfreier Algorithmus

Für den Fall, dass keine Ankerknoten verfügbar sind, sind weniger Algorithmen bekannt. *Anchor-Free Localization* [12] sucht zunächst heuristisch zwei Knotenpaare. Dabei sind die Knoten jedes Paars möglichst weit voneinander entfernt, und die Paare definieren zwei aufeinander senkrecht stehende Achsen. Diese Achsen werden dann als Koordinatenachsen verwendet, um eine möglichst faltungsfreie Startlösung zu erzeugen. Als Verbesserungsheuristik wird dann ein verteilter *Spring Embedder* eingesetzt. Dafür stehen Distanzmessungen zur Verfügung.

Bild 5 zeigt das Ergebnis dieses Verfahrens. Da der Algorithmus ein eigenes Koordinatensystem verwendet, ist dieses gegenüber den vorigen Lösungen gedreht und gespiegelt. Abgesehen davon ist die produzierte

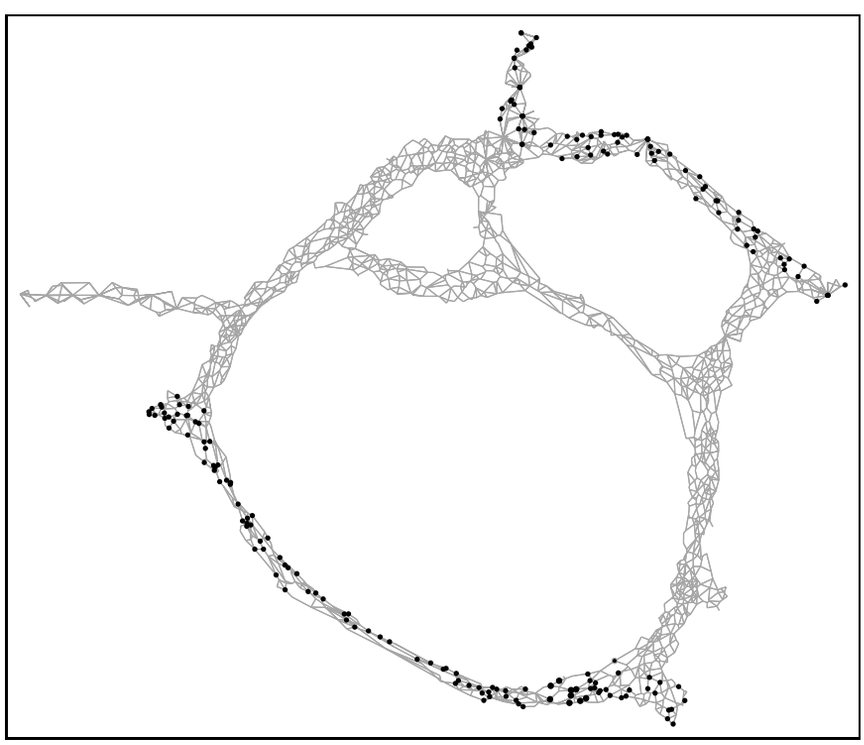

Bild 5 *Anchor-Free Localization* [12].

Lösung fast faltungsfrei und somit den zuvor vorgestellten überlegen. Auch dieser Algorithmus produziert Faltungen bei ungünstigen Topologien, gerade in größeren Netzen; das verwendete Beispielszenario löst er aber befriedigend. Es ist also abzuwarten, welche Verbesserungen in zukünftigen Versionen noch erzielt werden können.

Da die Faltungsfreiheit nicht garantiert ist, sind die eingangs angesprochenen Probleme auch hier nicht ausgeräumt. Im nächsten Abschnitt wird daher ein völlig neuer Ansatz vorgestellt.

4 Abstraktes Lokationsbewusstsein

Im Folgenden soll eine alternative Form von Lokationsbewusstsein vorgestellt werden. Unseres Wissens nach ist sie trotz ihres großen Potenzials kaum erforscht. Die einzelnen Bestandteile sind dabei seit langem bekannt und untersucht: Symbolische Lokation, topologische oder geometrische Cluster zur Abstraktion der Topologie und gewichtete Graphen zur Abbildung des Sensornetzes. Da es sich hierbei um unerforschtes Neuland handelt, befinden sich entsprechende Verfahren noch in der Entwicklungsphase.

Auf unterster Ebene werden Graphen zur Modellierung von Netzwerken genutzt: Jeder Sensorknoten wird als Knoten modelliert. Kanten verbinden je zwei Knoten, die miteinander direkt kommunizieren können.

Die Idee besteht nun darin, Sensorknoten zu großen Clustern zu gruppieren und jeden Cluster als einzelnen Knoten des Graphen aufzufassen. Clusternachbarschaften erzeugen dann die Kanten. Der Vorteil ist, dass die resultierenden Datenstrukturen sehr viel kleiner als vollständige Kommunikationsgraphen sind.

Diese reduzierten Graphen können dann an alle Sensorknoten verteilt werden. Das Wissen um den eigenen Cluster und seine Lage im Graphen ist dann eine Art von Lokationsbewusstsein.

4.1 Cluster und Topologie

Ein wichtiger Schritt besteht darin, eine Clusterstruktur aufzubauen, die die topologische Struktur des Netzes gut abbildet.

Im bereits erwähnten Beispiel des Straßensystems besteht ein gutes Clustering etwa darin, jede Straße und jede Kreuzung zu jeweils einem Cluster zusammenzufassen. Dazu muss das Netzwerk die lokale Topologie erkennen können. Eine Möglichkeit dafür wurde in [6] skizziert. Die Sensorknoten entscheiden anhand eines lokalen Kriteriums, ob sie am Rand des Netzes liegen, und bilden zusammenhängende Randstreifen. Eine Straße ergibt sich dann als zusammenhängende Knotenmenge, die demselben Randpaar folgt. Kreuzungen sind Bereiche, in denen mehrere Straßen zusammentreffen.

Besonders wichtig ist dabei, dass es keine Rolle spielt, ob die Straßenerkennung perfekt funktioniert. Entscheidend ist nur, dass die Cluster die wesentliche topologische Struktur wiedergeben. Hierfür existieren derzeit nur erste Ansätze, die noch weiterentwickelt werden müssen.

Eine zweite Möglichkeit verwendet die Sensoren der Knoten. Dabei schliessen sich Knoten zusammen, wenn sie in einer ähnlichen Umgebung liegen. Sofern die Knoten in der Lage sind mit ihren Sensoren abstrakte Umgebungseigenschaften wie beispielsweise „Wasser“, „Acker“ oder „Schatten“ zu erkennen, kann eine Clusterstruktur aufgebaut werden, die auf natürliche Weise die Umgebung des Netzes abbildet. Allerdings liegen auch hier noch sehr viele ungelöste Probleme und Fragestellungen, die erforscht werden müssen.

4.2 Cluster und Graphen

Die Clusterstruktur wird in einen Graphen überführt, indem für jeden Cluster ein Knoten und für benachbarte Cluster Kanten eingeführt werden. Alternativ kann, wenn die skizzierten Cluster des Straßenszenarios verwendet werden, ein Knoten für jede Kreuzung und eine Kante für jede Straße erzeugt werden.

Ein hinreichend grobes Clustering vorausgesetzt, ist der entstehende Graph klein genug, um ihn an alle Sensorknoten zu verteilen. Da die Knoten wissen, in welchem Cluster sie sich befinden, bekommen sie dadurch ein Bewusstsein über die Struktur des Netzes sowie ihre eigene Position darin.

Bild 6 zeigt anhand des Straßenbeispiels, welche Sensorknoten sich zu Clustern zusammenschlie-

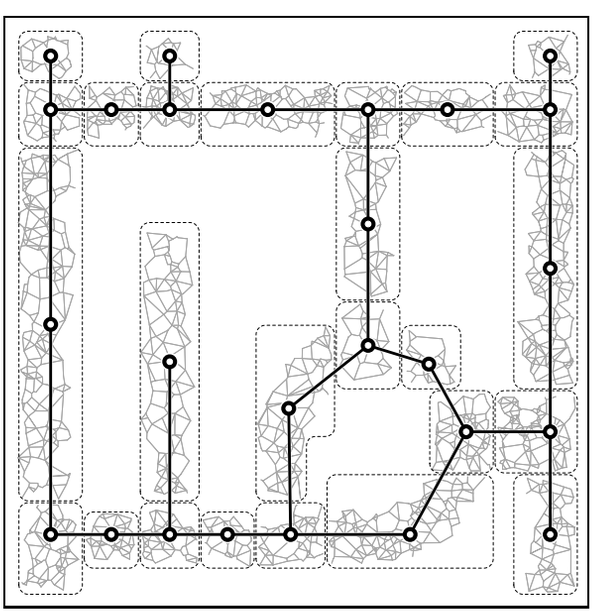

Bild 6 Cluster und Clustergraph.

ßen, und welcher Clustergraph daraus entsteht.

4.3 Anwendung

Da die Knoten über den erzeugten Graphen eine globale Sicht auf das Sensornetz haben, können in jedem Bereich, in dem üblicherweise lokale Heuristiken zur Erreichung globaler Ziele eingesetzt werden, Verbesserungen erzielt werden. Beispielhaft soll dies im Folgenden anhand von *GeoRouting* diskutiert werden.

Es gibt schon seit einiger Zeit Routingverfahren, die auf hierarchische Cluster aufsetzen. Ein Überblick findet sich z. B. in [11]. Viele dieser Verfahren sind in Verruf geraten, weil sie trotz des vorhandenen globalen Wissens nur auf lokale Greedyheuristiken setzen. Dadurch kann es passieren, dass zwei Knoten, die nahe beieinander liegen, aber in verschiedenen Clustern sind, nur über sehr große Umwege kommunizieren dürfen. Dennoch sind Cluster – insbesondere die hier verwendeten topologischen – sehr gut geeignet, um mit lokalen Entscheidungen global günstige Routingpfade aufzubauen.

Die Vorteile ergeben sich daraus, dass bereits der Sender die globale Topologie zur Verfügung hat. Daher kann er den Routingpfad als Sequenz benachbarter Cluster festlegen. Dabei können zusätzliche Informationen über den Graphen für weitere Verbesserungen genutzt werden.

So könnten die Kanten mit einer Schätzung der Restenergie oder Kommunikationsbandbreite der Sensorknoten im zugehörigen Cluster abgespeichert werden. Dadurch kann bereits im Vorfeld ein energieeffizienter Pfad hoher Bandbreite angelegt werden. Wenn zu erwarten ist, dass viele Daten zu übertragen sind, können *Load-Balancing*-Strategien über disjunkte Pfade implementiert werden. Dadurch wird gleichzeitig ausfallsichere Kommunikation ermöglicht.

Auf Basis einer solchen Graphenmodellierung lassen sich viele weitere Vorteile für die Anwendungen erzielen. So lassen sich beispielsweise Clustern Eigenschaften zuordnen, die wiederum eine Kategorisierung möglich machen. Diese kann dann als Kontextinformation genutzt werden. So ließen sich beispielsweise aufgrund der Nachbarschaftsbeziehungen Eigenschaften wie „Kreuzung“ für einzelne Cluster ableiten. Allgemein unterstützt die Clusterbildung die Kontextgewinnung im Netz, beispielsweise über die relative globale Clusterposition am Rand oder in der Mitte des Netzes. Auch ließen sich so Cluster abstrakteren Ordnungsrelationen wie Reihenfolgen für Flussbetrachtungen im Gesamtnetz unterwerfen. Auch Datenfusion lässt sich nicht nur auf Basis von Koordinaten durchführen, sondern auch clusterbasiert, da häufig Daten von Sensorknoten in ähnlichen Kontexten zusammengeführt werden sollen.

5 Zusammenfassung

In diesem Artikel wird die in der aktuellen Forschung verfolgte Vorgehensweise hinsichtlich der Lokalisierung diskutiert.

Es wird aufgezeigt, dass oft ins Feld geführte Annahmen und Voraussetzungen über die Gutartigkeit von Szenarien zu optimistisch sind. Dies führt in anwendungsorientierten Szenarien dazu, dass es vielmehr schwierig bis unmöglich ist, zu konsistenten Koordinatenzuweisungen zu gelangen. Es wird theoretisch wie simulativ belegt, dass ohne technische Hilfsmittel konsistente Lösungen voraussichtlich nicht berechenbar sind. Nicht nur bei Anwendungen wie beispielsweise dem *GeoRouting* können solche Inkonsistenzen jedoch zu erheblichen Problemen führen, sodass die Suche nach alternativen Ansätzen angeraten ist.

Die Autoren schlagen daher vor, das Problem gerade in komplizierten, realitätsnahen Szenarien auf eine neue Art und Weise anzugehen. Wir zeigen, wie Algorithmen mithilfe von geographischen Clustern und dem zugehörigen Verbindungsgraphen eine abstrakte Form von Lokationsbewusstsein aufbauen können, die von externer Infrastruktur vollständig unabhängig ist. Zusätzlich ist dieser aggregierte Graph so klein, dass er an alle Knoten im Netz verteilt werden kann. Anders als beim koordinatenbasierten Ansatz erlangt somit jeder Knoten ein globales Bild von der Netzstruktur.

Zur Umsetzung dieser Ansätze sind noch viele Fragen offen, da der Forschungsbereich kaum erschlossen ist. An der TU Braunschweig und der Universität zu Lübeck wird zurzeit an verschiedenen Methoden zur Errichtung abstrakten Lokationsbewusstseins gearbeitet. Es ist daher zu erwarten, dass sich in naher Zukunft noch viele Erkenntnisse ergeben und Verfahren entwickelt werden.

Literatur

[1] J. Aspnes, D. Goldenberg, and Y.R. Yang. On the computational complexity of sensor network localization. In: *ALGOSENSORS 2004, Lecture Notes in Computer Science*, Vol. 3121, pp. 32–44. Springer Verlag, 2004.

[2] L. Blazevic, J.-Y. Le Boudec, and S. Giordano. A scalable routing scheme for self-organized terminode network. In: *Communication Networks and Distributed systems modelling and Simulation conference (CNDS)*, San Antonio, Texas, 2002.

[3] H. Breu and D.G. Kirkpatrick. Unit disk graph recognition is NP-hard. In: *Computational Geometry: Theory and Applications* 9(1–2):3–24. Elsevier, 1998.

[4] C. Buschmann und S. Fischer. Eigenschaften der Funkschnittstelle in Sensornetzen und Auswirkungen für Anwendungen. In: *2. GI/ITG Fachgespräch „Sensornetze“, 2004.*

[5] T. Eren, D. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. In: *Int'l Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2004), Hong Kong*, pp. 2673–2684, 2004.

[6] S.P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology recognition in sensor networks. In: *ALGOSENSORS 2004, Lecture Notes in Computer Science*, Vol. 3121, pp. 123–136. Springer Verlag, 2004.

[7] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Unit disk graph approximation. In: *ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC 2004), Philadelphia*, ACM Press, 2004.

[8] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: A quantitative comparison. In: *Computer Networks* 43(4):499–518, Elsevier, 2003.

[9] U. Leonhardt. Supporting location-awareness in open distributed systems. *Dissertation, Dep. of Computing, Imperial College, London*, 1998.

[10] D. Niculescu and B. Nath. Ad-hoc positioning system. In: *IEEE Global Telecommunications Conf. (GLOBECOM 2001)*, Vol. 5, pp. 2926–2931, 2001.

[11] C.E. Perkins. Ad Hoc Networking, Addison Wesley, 2001.

[12] N.B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-Free Distributed Localization in Sensor Networks. *MIT Technical Report MIT-LCS-TR-892*, 2003.

[13] C. Savarese, J. Rabaey, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX Technical Annual Conf. 2002, Monterey, CA*, pp. 317–327, 2002.

[14] A. Savvides, H. Park, and M.B. Srivastava. The bits and flops of the N-hop multilateration primitive for node localization problems. In *First ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA 2002), Atlanta, GA*, pp. 112–121, ACM Press, 2002.

[15] K. Whitehouse. The Design of Calamari: An ad-hoc localization system for sensor networks. *Master thesis, University of California at Berkeley*, 2002.


1


2


3


4


5

1 Prof. Dr. Sándor P. Fekete ist seit 2001 Professor für Mathematische Optimierung an der TU Braunschweig. Frühere Stationen waren die Universität zu Köln, die University of Waterloo in Kanada (Ph.D. 1992), die SUNY Stony Brook, Köln (Habilitation 1998) sowie die TU Berlin. Sein Arbeitsgebiet sind Algorithmen im weitesten Sinne, etwa in der diskreten Optimierung, der algorithmischen Geometrie oder für autonome und verteilte Systeme.
Adresse: Institut für Mathematische Optimierung, Pockelsstraße 14, 38106 Braunschweig,
Tel.: +49-531-3917551,
E-Mail: sandor.fekete@tu-bs.de

2 Alexander Kröller studierte bis 2003 Mathematik an der Technischen Universität Berlin. Seitdem forscht er an der TU Braunschweig über Sensornetzwerke. Sein Interesse gilt dabei insbesondere kombinatorischen Algorithmen und Heuristiken, die die geometrischen Eigenschaften der Netze ausnutzen.
Adresse: Institut für Mathematische Optimierung, Pockelsstraße 14, 38106 Braunschweig,
Tel.: +49-531-3917410,
E-Mail: a.kroeller@tu-bs.de

3 Carsten Buschmann erhielt 2002 das Diplom und den Master of Science in Informatik von der Technischen Universität Braunschweig. Gegenwärtig arbeitet er als wissenschaftlicher Mitarbeiter an der Universität Lübeck im Bereich Sensornetze. Im SWARMS-Projekt forscht er an der Entwicklung von *Middleware*-Konzepten zur Unterstützung der Entwicklung von Anwendungen im *Pervasive Computing*.
Adresse: Institut für Telematik, Ratzeburger Allee 160, 23538 Lübeck,
Tel.: +49-451-500-5384,
E-Mail: buschmann@itm.uni-luebeck.de

4 Prof. Dr. Stefan Fischer ist Professor für Informatik an der Universität Lübeck und leitet dort das Institut für Telematik. Zuvor arbeitete er an der TU Braunschweig, am Zentrum für Netzwerkforschung der IBM in Heidelberg, an der Universität Mannheim (wo er 1996 promovierte), der Universität Montréal sowie der International University in Bruchsal. Seine wissenschaftliche Arbeit konzentriert sich derzeit auf Anwendungen und Basissoftware in verteilten Systemen.
Adresse: Institut für Telematik, Ratzeburger Allee 160, 23538 Lübeck,
Tel.: +49-451-500-5380,
E-Mail: fischer@itm.uni-luebeck.de

5 Dipl.-Ing. (FH) Dennis Pfisterer studierte Nachrichtentechnik an der Fachhochschule für Technik und Gestaltung in Mannheim. Seit 2001 arbeitet er am European Media Lab in Heidelberg im Bereich Ressourcen-adaptiver Systeme. Zurzeit forscht er parallel dazu im Projekt SwarmNet als wissenschaftlicher Mitarbeiter an der Universität Lübeck im Bereich Sensornetze.
Adresse: Institut für Telematik, Ratzeburger Allee 160, 23538 Lübeck,
Tel.: +49-451-500-5384,
E-Mail: pfisterer@itm.uni-luebeck.de