



Freie Universität Berlin

Fachbereich Mathematik und Informatik

Institut für Informatik

## Diplomarbeit

# **Experimentelle Analyse verteilter Ereigniserkennung in Sensornetzen**

Christian Wartenburger - [wartenbu@inf.fu-berlin.de](mailto:wartenbu@inf.fu-berlin.de)

19. Dezember 2008

Betreuer: Prof. Dr.-Ing. Jochen Schiller  
Georg Wittenburg, M. Sc.  
Dipl.-Inform. Norman Dziengel



## **Danksagung**

Ich möchte Herrn M. Sc. Georg Wittenburg und Herrn Dipl.-Inform. Norman Dziengel meinen größtmöglichen Dank für ihre stets motivierte Betreuung meiner Diplomarbeit aussprechen. Auf ihre konstruktive Unterstützung konnte ich zu jeder Zeit zählen. Besonders möchte ich ihnen für die vielen Stunden und Tage danken, die Sie mit mir während der Versuche am Bauzaun verbracht haben, wodurch die Ergebnisse der Arbeit erst zustande gekommen sind.

Einen ganz herzlichen Dank möchte ich meinen Eltern sagen, die mich während meines gesamten Studiums mit liebevoller Fürsorge, großem Interesse, guten Ratschlägen und jedweder Unterstützung begleitet haben.



## Kurzfassung

Die verteilte Anordnung von Sensorknoten in einem drahtlosen Sensornetz ermöglicht es, Eigenschaften und Veränderungen der Umwelt an unterschiedlichen Stellen zu beobachten. Durch ein Zusammenwirken der Sensorknoten können in der Umwelt auftretende Ereignisse verteilt erkannt werden. Es existiert ein verteiltes Erkennungssystem, welches Ereignisdaten mehrerer Sensorknoten im Sensornetz fusioniert und anhand einer lokal durchgeführten Mustererkennung klassifiziert. Dieses System wurde unter Laborbedingungen entwickelt und getestet. Mit dieser Arbeit wird eine Weiterentwicklung des verteilten Erkennungssystems vorgestellt, welches den Anforderungen einer praktischen Anwendbarkeit gerecht wird.

Während des Mustertrainings können Ereignisdaten von einer dynamischen Anzahl von Sensorknoten verarbeitet werden. Die Daten werden dabei auf die für die Erkennung geeignetsten und aussagekräftigsten Werte reduziert. Die verteilte Ereigniserkennung ist nicht auf den Ort des Trainings beschränkt. Ein unbekanntes Ereignis kann an jedem Punkt des Sensornetzes von den das Ereignis wahrnehmenden Sensorknoten unter Austausch von Merkmalswerten klassifiziert werden. Ort und Typ des Ereignisses werden auf einer Basisstation angezeigt.

Das in der vorliegenden Arbeit entwickelte System wird anhand eines praktischen Anwendungsfalls getestet. Dabei handelt es sich um die Erkennung sicherheitsrelevanter Ereignisse an einem Bauzaun. Die Evaluation erfolgt für die Erkennung der Ereignisse am Ort gleich und ungleich des Trainings. Es wird untersucht, wie sich die Änderung des Ortes auf die Erkennungsraten auswirkt und wie viel Einfluss die Bedingungen der realen Welt auf die Ergebnisse im Vergleich zu denen unter Laborbedingungen besitzen. Am Ort des Trainings wird eine Korrektklassifikationsrate von 74,8% erzielt. Im Vergleich zum Projekt Fence Monitoring [Wit07] , welches sich ebenfalls mit der verteilten Erkennung von Ereignissen an Bauzäunen beschäftigt, wird eine Steigerung von 15,9 Prozentpunkten erzielt. In Bezug auf das dieser Arbeit zugrunde liegenden Erkennungssystem [Dzi07] , dessen Ergebnisse durch Versuche unter Laborbedingungen entstanden, ist ein Rückgang von 21,5 Prozentpunkten zu verzeichnen. Für den Fall der Erkennung am Ort ungleich des Trainings ergibt sich eine Korrektklassifikationsrate von 57,7%.



# Inhaltsverzeichnis

<b>1 Einleitung.....</b>	<b>1</b>
1.1 Motivation, Restriktionen und Anforderungen .....	1
1.2 Problemstellung und Lösungsansatz .....	2
<b>2 Verwandte Arbeiten.....</b>	<b>4</b>
2.1 Klassifikation und Detektion in der Überwachung von Lebensräumen .....	4
2.2 Verteilte Klassifikation menschlicher Aktionen .....	5
2.3 Gruppenbasierte Ereigniserkennung in Unterseenetzwerken .....	6
2.4 Fence Monitoring.....	7
2.5 VigilNet.....	8
2.6 Fahrzeugklassifikation in Sensornetzwerken .....	9
<b>3 Grundlagen .....</b>	<b>11</b>
3.1 Hardware- und Softwareplattform .....	11
3.1.1 Hardware.....	11
3.1.2 Software .....	12
3.1.3 Beschleunigung .....	12
3.2 Modell der allgemeinen Mustererkennung .....	12
3.2.1 Komponenten des allgemeinen Mustererkennungsmodells .....	13
3.2.2 Phasen des allgemeinen Mustererkennungsmodells .....	14
3.3 Modell der verteilten Mustererkennung .....	15
3.3.1 Abstraktionsebenen der Datenfusion .....	15
3.3.2 Fusionsmodell „Omnibus“.....	15
3.4 Lokale Mustererkennung .....	16
3.4.1 Modellkomponenten der lokalen Mustererkennung.....	16
3.4.2 Modellphasen der lokalen Mustererkennung .....	21
3.5 Verteilte Mustererkennung.....	22
3.5.1 Methoden .....	22
3.5.2 Phasen und Komponenten der verteilten Mustererkennung.....	23
<b>4 Erweiterung der verteilten Ereigniserkennung.....</b>	<b>25</b>
4.1 Grenzen des bestehenden Systems .....	25
4.2 Anforderungen.....	26
4.3 Konzeptionelle Lösungsansätze .....	27
4.3.1 Ortsunabhängigkeit.....	27

4.3.2	Merkmalsauswahl .....	32
4.3.3	Systemarchitektur und Anpassung des Mustererkennungsmodells .....	42
4.4	Externes System .....	44
4.4.1	Datenspeicherung – Training .....	47
4.4.2	Datenverifikation .....	47
4.4.3	Datenfusion.....	49
4.4.4	Merkmalsauswahl .....	51
4.4.5	Klassengenerierung .....	53
4.4.6	Knotenauswahl.....	53
4.4.7	Zurückweisung .....	54
4.4.8	Verteilung .....	54
4.4.9	Datenspeicherung – Erkennung .....	56
4.4.10	Klassifizierung .....	57
4.4.11	Anzeige .....	57
4.5	Anpassung des Sensorknotensystems.....	58
4.5.1	Merkmalsverteilung – Training .....	59
4.5.2	Parameterverteilung .....	62
4.5.3	Merkmalsverteilung – Erkennung .....	63
4.5.4	Klassifizierung .....	67
4.5.5	Klassifikationsverteilung .....	67
<b>5</b>	<b>Anwendungsspezifische Anpassung und Erweiterung .....</b>	<b>69</b>
5.1	Beschreibung des Anwendungsfalls .....	69
5.2	Anpassungen .....	71
5.2.1	Beschleunigungsachsen .....	71
5.2.2	Filttereinstellung .....	72
5.2.3	Segmentierung .....	73
5.2.4	Sonstige Anpassungen .....	74
5.3	Erweiterungen .....	75
<b>6</b>	<b>Auswertung .....</b>	<b>78</b>
6.1	Versuchsbeschreibung .....	78
6.1.1	Aufbau des Bauzauns und Anbringung der Sensorknoten .....	78
6.1.2	Ereignisse .....	79
6.1.3	Training .....	81
6.1.4	Systemparameter .....	81

6.1.5	Untersuchte Szenarien .....	82
6.2	Kennwerte .....	83
6.3	Training .....	84
6.3.1	Datenverifikation .....	84
6.3.2	Merkmalsauswahl .....	85
6.4	Erkennung .....	90
6.4.1	Erkennung im Sensornetz .....	90
6.4.2	Zurückweisung über alle Sensorknoten .....	98
6.4.3	Erkennung auf dem externen System .....	100
6.4.4	Vergleiche .....	104
<b>7</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>109</b>
7.1	Zusammenfassung .....	109
7.2	Weiterführende Arbeiten .....	110
<b>8</b>	<b>Literaturverzeichnis .....</b>	<b>113</b>



# 1 Einleitung

*„Im 21. Jahrhundert wird die technologische Revolution das Alltägliche, Kleine und Unsichtbare sein“*

— Mark Weiser (1952-1999)

Fortschritte in der Mikrotechnologie und Entwicklungen im Bereich der drahtlosen Kommunikation erlauben die Herstellung kostengünstiger, energieeffizienter und multifunktionaler Sensorknoten. Einzeln betrachtet, ergeben sich für die Sensorknoten nur geringe Anwendungsmöglichkeiten. Wenn diese allerdings in einem drahtlosen Sensornetz (Wireless Sensor Network) zusammenarbeiten, ergibt sich die Möglichkeit, ihre Umwelt mit einer hohen Genauigkeit wahrzunehmen [Bha02].

## 1.1 Motivation, Restriktionen und Anforderungen

Sensorknoten können aufgrund ihrer geringen Größe, der Ausstattung mit multiplen Sensoren, der drahtlosen Kommunikation untereinander und der Leichtigkeit der Ausbringung in hoher Anzahl in vielen Anwendungsgebieten eingesetzt werden [Cho03]. Mit einem Sensornetz können Umweltbeobachtungen durchgeführt werden, wie etwa Überwachung von Lebensräumen, Erkennung seismischer Aktivitäten, Frühwarnsysteme oder Klimabeobachtungen. Im Gesundheitswesen kann der Gesundheitszustand von Patienten überwacht werden. Sensorknoten können in Alltagsgegenstände eingebaut werden, miteinander agieren und somit ein intelligentes Verhalten der Geräte erreichen. Sensornetze sind weiterhin besonders für militärische Anwendungen aufgrund ihrer Fehlertoleranz und Selbstorganisation geeignet. Dazu gehören die Observierung von Kampfgebieten, Truppen- und Materialüberwachung und die Erkennung von biochemischen oder nuklearen Angriffen.

Bei der Verwendung von Sensornetzen muss mit Restriktionen umgegangen werden, die aufgrund der Eigenschaften der Sensorknoten und Sensornetze vorhanden sind [Aky02]. Die geringe Größe der Sensorknoten geht damit einher, dass nur eine beschränkte Energieversorgung möglich ist. Da ein Sensornetz seinen Dienst möglichst für lange Zeit unbeaufsichtigt ausführen sollte, muss versucht werden, den Energieverbrauch im laufenden Betrieb möglichst gering zu halten. Energie, die für Funkübertragungen benötigt wird, ist im Vergleich zu der Energie, die bei Berechnungen verbraucht wird, deutlich höher. Da der Funkkontakt die einzige Möglichkeit der Sensorknoten ist, untereinander zu kommunizieren, müssen die Netzwerkprotokolle an die Anforderung angepasst werden.

Sofern die Sensorknoten in einer dichten Anordnung ausgebracht sind, kann sich bei Funkübertragungen die Kollisions- und Staugefahr im Netzwerk erhöhen [Tub03]. Es muss versucht werden, dem mit einer geringen Anzahl zusätzlicher Übertragungen oder nicht simultanen Senden zu begegnen.

Sensorknoten besitzen eine beschränkte Kapazität an Rechenleistung und Speicher. Daher muss versucht werden, den vorhandenen Speicherplatz möglichst optimal auszunutzen. Aufgrund der geringen Rechenleistung müssen sinnvolle und durchführbare Berechnungen abgeschätzt werden.

Sensornetze können aus einer hohen Anzahl von Sensorknoten bestehen. Es muss eine Skalierbarkeit gegeben sein, so dass Aufgaben des Sensornetzes auch bei Hinzunahmen von Sensorknoten ausgeführt werden können. Die Verwaltbarkeit der Sensorknoten sollte dabei auch weiterhin gegeben sein.

## 1.2 Problemstellung und Lösungsansatz

Sensorknoten dienen in ihrer einfachsten Funktion dazu, Daten zu sammeln und weiterzuleiten. Die Daten werden dabei mit Hilfe von Sensoren aufgenommen, vom Sensorknoten ausgelesen und zur weiteren Verarbeitung versendet. Das Datenvolumen kann dabei so umfangreich sein, dass eine ständige Weiterleitung aufgrund der beschriebenen Restriktionen nicht durchführbar ist. Es wäre also von Vorteil, wenn die Sensorknoten selbst auf die Ereignisse, die in ihrer Umwelt passieren, reagieren würden. Die Ereignisse in einem Sensornetz können dabei anhand von Sensormessdaten interpretiert werden. Eine Voraussetzung für eine Reaktion auf ein Ereignis ist, dieses erkennen und zuordnen zu können.

Die vorliegende Arbeit basiert auf der Arbeit von Norman Dziengel, „Verteilte Ereigniserkennung in Sensornetzen“ [Dzi07]. Dort wird die Auffassung vertreten, dass komplexe Ereignisse wie etwa Gesundheitszustände, Truppenaktivitäten oder eine Zaunüberwindung nur schwer von einem Knoten erkannt werden können. Es soll somit eine verteilte Ereigniserkennung ausgeführt, die mehrere Sensorknoten in die Entscheidungsfindung einbezieht, um eine höhere Erkennungsgenauigkeit zu erlangen.

Ereignisse, die in einem Sensornetz auftreten, werden dabei als Muster interpretiert. Die Erkennung basiert daher auf klassischen Mustererkennungsarchitekturen. Es existiert ein lokales Mustererkennungssystem, wodurch auf den einzelnen Sensorknoten verschiedene Ereignisse trainiert und erkannt werden können. Zur parallelen Erkennung eines Ereignisses mit mehreren Sensorknoten werden auf der lokalen Lösung aufbauende Methoden zur verteilten Erkennung auf verschiedenen Ebenen des Systems untersucht, implementiert und analysiert. Es wird ohne die Nutzung einer Basisstation oder anderen externen Rechenleistungen gearbeitet.

Ziel dieser Arbeit ist es, das bestehende System, welches bisher nur unter Laborbedingungen verwendet und getestet wurde, bis zur Praxistauglichkeit weiterzuentwickeln. Zur Untersuchung der Genauigkeit der Ergebnisse der verteilten Ereigniserkennung soll ein groß angelegter Feldversuch durchgeführt werden. Bei dem Anwendungsfall handelt es sich um einen Bauzaun, an dem sicherheitsrelevante Ereignisse ausgeführt und erkannt werden sollen. Die Sensorknoten werden dazu an die einzelnen Bauzaunelemente angebracht und können die Bewegung des Zauns als Beschleunigung wahrnehmen. Eine Steigerung der Komplexität gegenüber dem Laborversuch ist darin zu sehen, dass ein Ereignis aufgrund der Verbindung der einzelnen Bauzaunelemente untereinander und der damit verbundenen Schwingungsausbreitung je nach Position der Sensorknoten unterschiedlich wahrgenommen wird. Weiterhin soll die Erkennung eines Ereignisses an einem beliebigen Ort des Bauzauns möglich sein.

Es muss eine Anpassung der Mustererkennungsalgorithmen vorgenommen werden, da diese nicht ohne weiteres übertragbar sind. Aufgrund der hohen und dynamischen Anforderungen an das System wird ein zusätzliches externes System eingeführt, welches Teile des Trainings und eine Anzeige von Erkennungsergebnissen vornimmt. Zusätzlich werden netzwerktechnische Komponenten integriert, wodurch das aufgrund der großen Ausdehnung des Sensornetzes notwendig gewordene Routing der Datenpakete durch das Sensornetz anwendungsspezifisch und energieeffizient realisiert wird.

## 2 Verwandte Arbeiten

Sensornetze sind mittlerweile in vielen Forschungs- und Anwendungsbereichen verbreitet. Dazu gehören Militär, Sicherheitsüberwachung, Gesundheitswesen, Umweltbeobachtung, Katastrophenschutz.

Es sollen im Folgenden Arbeiten betrachtet werden, welche sich mit der Mustererkennung in Sensornetzen beschäftigen. Alle Arbeiten etablieren eine verteilte Erkennung in irgendeiner Form. Die verteilte Entscheidungsfindung findet dabei entweder auf den Sensorknoten, auf einer Basisstation oder in Kombination statt.

### 2.1 Klassifikation und Detektion in der Überwachung von Lebensräumen

Es gibt zwei grundlegende Aufgaben in einem Lebensraumüberwachungssystem: die Erkennung und die Lokalisierung zu überwachender Tiere. Wang et al. beschreibt in [Wan03] Möglichkeiten zur Lösung dieser Herausforderungen.

Während der Systeminitialisierung ordnen sich alle Knoten selbstständig in Clustern an. In jedem Cluster gibt es einen Clusterkopf mit einer höheren Rechenleistung, der für die Zusammenarbeit der Knoten und die zentrale Datenauswertung zuständig ist. Die restlichen Knoten sind für Datenaufnahme und Datenvorverarbeitung verantwortlich.

Zur Erkennung und Lokalisierung von Tieren werden ihre Rufe analysiert. Auf jedem Knoten werden dazu ständig akustische Signale abgetastet und gepuffert. Die Ereignisbearbeitung wird in mehreren Stufen durchgeführt. In der schnellsten Stufe, welche nur auf dem Clusterkopf stattfindet, werden Intensitäten von aufgenommenen Signalen untersucht. Liegt die Intensität eines wahrgenommenen Signals über einem vordefinierten Schwellwert, werden für eine vom spezifizierten Ruf abhängige Zeit Daten gesammelt und diese zur Klassifikation verwendet. Dafür wird das Spektrogramm des aufgenommenen Wellenfeldes analysiert und dann mit einem Referenz-Spektrogramm korreliert (siehe Abbildung 2.1). Wenn der maximale Korrelationskoeffizient über einem Schwellwert liegt, wird das Signal dem spezifizierten Tier zugeordnet.

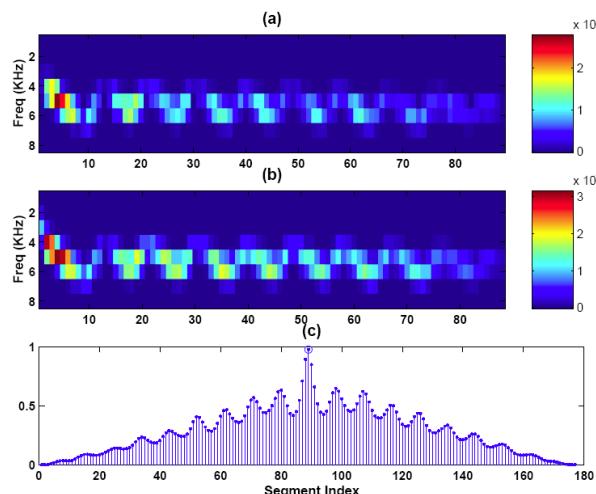


Abbildung 2.1: (a) Spektrogramm eines Froschrufs (b) Referenz-Spektrogramm (c) kreuz-korrelierte Koeffizienten [Wan03]

Nach einer erfolgreichen Zuordnung wird das Tier lokalisiert. Dafür werden die Daten von den anderen Knoten vom Clusterkopf abgefragt. Diese senden ihre gepufferten und auf die für die Lokalisierung nötigen reduzierten Wellenfelder an den Clusterkopf. Nachdem alle Daten eingetroffen sind, wird die Lokalisierung mittels eines Beamforming-Algorithmus durchgeführt. Beamforming ist eine Methode zur Ortung von Quellen in Wellenfeldern. Der Vorteil dieses mehrstufigen Verfahrens ist, dass nicht alle Stufen für jedes Signal durchlaufen werden müssen. Dadurch wird ein Echtzeitverhalten und zudem auch eine Energie- und Zeiteffizienz erreicht.

Die Klassifikation auf immer demselben Knoten mit lokalen Daten stellt einen Unterschied zur vorliegenden Arbeit dar. Erst bei der Lokalisierung werden verteilte Daten zentral ausgewertet. Das dafür verwendete Architekturmodell stellt einen weiteren gravierenden Unterschied dar. Während die Daten der Knoten in [Wan03] vom Clusterkopf abgerufen werden, werden diese in der vorliegenden Arbeit selbstständig von den Knoten gesendet.

## 2.2 Verteilte Klassifikation menschlicher Aktionen

In [Yan08] wird ein tragbares Sensornetzwerk dazu verwendet, um Bewegungen von Menschen zu erkennen. Die Klassifizierung wird hierbei verteilt auf den Knoten sowie auf einer Basisstation durchgeführt.

Die Sensorknoten sind an verschiedenen Stellen des Körpers angebracht und können Orientierungswerte an zwei Achsen mit einem Gyroskop sowie Beschleunigungswerte an drei Achsen messen.

Die Klassifizierung wird zuerst auf jedem Knoten vorgenommen. Da menschliche Aktionen verschieden lang sein können, müssen die aufgenommenen Daten normiert werden. Um auf eine einheitliche Darstellung zu kommen, wird eine lineare Interpolation oder Fast Fourier Transformation (FFT) verwendet. Weiterhin wird die Annahme gemacht, dass all Muster in einer Bewegungsklasse einem Modell eines Unterraums genügen. Die Trainingsdaten jeder Klasse spannen einen Unterraum auf, so dass jedes Testmuster derselben Klasse als Linearkombination der Trainingsdaten dargestellt werden kann. Die Zugehörigkeit eines Testmusters kann durch Lösen eines linearen Gleichungssystems, Projektion der gefundenen Koeffizienten auf jeden Unterraum und nachfolgender Untersuchung der größten Zugehörigkeit gefunden werden. Ein Problem stellt hierbei die hohe Dimensionalität des Datenraums dar. Gelöst wird dies durch eine lineare Projektion, wobei sich ein Dimensionsraum kleiner als die Anzahl der Trainingsdaten ergibt. Das neue lineare Gleichungssystem ist unterbestimmt und kann eindeutig gelöst werden. Die lokalen Knoten können zudem auch mögliche Ausreißer, welche nicht eindeutig einer Klasse zuzuordnen sind, erkennen. Dazu wird die Verteilung der Koeffizienten über alle Klassen berechnet. Sind die Koeffizienten gleich verteilt, ergibt sich ein niedriger Konzentrationswert, sind jedoch die Koeffizienten auf eine Klasse beschränkt, ergibt das einen hohen Wert. Für jeden Knoten wird ein Schwellwert definiert. Nur wenn berechnete Wert über diesem liegt, ist die Erkennung gültig und die Daten werden zur Basisstation gesendet. Es wird angemerkt, dass sich durch dieses Verfahren die Kommunikation zwischen Sensor und Basisstation um bis zu 50% reduzieren kann.

Nachdem ein oder mehr Sensorknoten ihre Daten an die Basisstation gesendet haben, kann dort eine globale Klassifikation vorgenommen werden. Wie auf den Knoten wird auch hier ein lineares Gleichungssystem gelöst, als Testmuster werden alle empfangenen Daten, zusammengefasst in einem Vektor, verwendet. Auch hier kann das Ergebnis zurückgewiesen werden, allerdings geschieht das weniger restriktiv, weil das Senden eines falscher Daten nicht so entscheidend ist wie eine globale Zurückweisung.

Das untersuchte System verwendet ebenso wie die vorliegende Arbeit eine lokale Klassifizierung. Diese unterscheidet sich aber in entscheidenden Punkten. Beim beschriebenen System werden nur Daten verwendet, die von den Knoten selbst aufgenommen wurden. Für die Klassifikation werden Trainingsdaten benutzt, die auf den Knoten gespeichert sind. Zudem ist auch der Berechnungsaufwand für den zugehörigen Algorithmus durch das Lösen eines linearen Gleichungssystems mit vorheriger Projektion deutlich höher als beim vorliegenden System. Ein weiterer Unterschied ist, dass eine zweite Klassifikation auf der Basisstation durchgeführt wird, wofür die Daten der Knoten dorthin übermittelt werden müssen. Als Gemeinsamkeit ist zu sehen, dass eine Zurückweisung auf den Knoten vorgenommen wird, die eine unnötige Übermittlung von Daten an die Basisstation verhindert.

## 2.3 Gruppenbasierte Ereigniserkennung in Unterseenetzwerken

Ein Verfahren zur Realisierung eines Überwachungssystems in Unterseenetzwerken wird in [Tav05] beschrieben. Hierbei wird eine gruppenbasierte Ereigniserkennung eingesetzt, um in ein überwachtes Gebiet eindringende Ziele zu erkennen.

In der untersuchten Arbeit wird die Möglichkeit verworfen, eine Erkennung durchzuführen, indem die Daten aller Sensorknoten an eine Basisstation gesendet werden und dort mit hoher Genauigkeit eine Klassifizierung vorgenommen wird. Gründe, die dagegen angeführt werden, sind der hohe Energieverbrauch, der auftritt, wenn der gesamte aufgenommene Datensatz versendet werden müsste. Des Weiteren könnte beim zuverlässigen Senden aller Daten eine Verzögerung entstehen, die eine Reaktion gemäß den Zeitbeschränkungen des Überwachungssystems verhindert. Die Erkennung sollte also im Sensornetzwerk durchgeführt werden, um die Verzögerung und Energiekosten gering zu halten.

Die Architektur des Systems wird in drei Schichten aufgeteilt: Sensorschicht, Knotenschicht und Gruppenschicht. In der Sensorschicht werden sensorspezifische Aufgaben wie Kalibrierung, Filterung und Erkennung ausgeführt. In der Knotenschicht werden die Daten verschiedener Sensoren gesammelt und daraus eine Klassifikation erstellt. Vorteilhaft ist dabei, dass dies zu einer Ergebnisverbesserung führt, da diese nicht von einzelnen Sensormessungen abhängt. Das Ergebnis der Knotenschicht ist die Eingabe der Gruppenschicht. Während der Ausbringung der Sensorknoten werden diese in Gruppen unterteilt, für jede Gruppe wird ein Leiter gewählt. Dieser führt die Klassifizierung der Gruppenschicht aus. Anstatt vieler Sensordaten werden also nur wenige prägnante Werte zwischen dem Gruppenführer und den anderen Mitgliedern der Gruppe ausgetauscht. Haben die Gruppenführer eine Klassifikation durchgeführt, senden sie diese an eine Basisstation zur weiteren Verarbeitung.

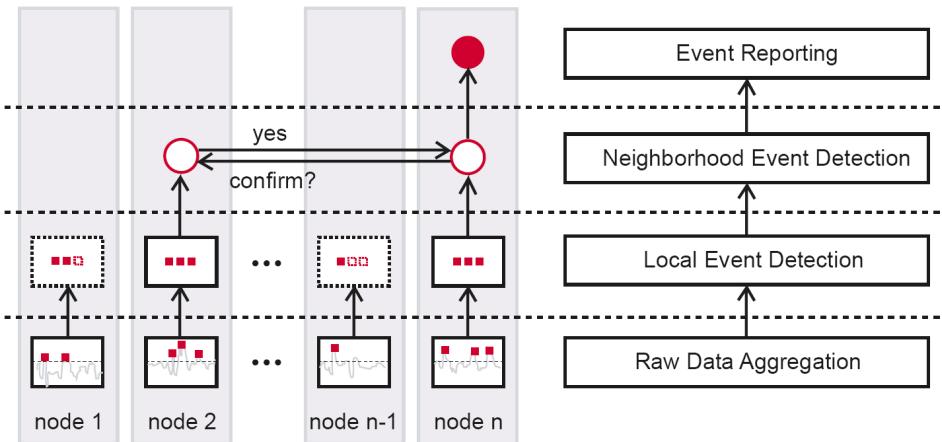
Zur Erhöhung des Vertrauensniveaus der erhaltenen Meldungen werden räumliche und historische Informationen herangezogen. Es steht in Bezug zu diesen Informationen und wird als Funktion der beiden Werte berechnet. Die räumliche Information gibt Auskunft darüber, wie viel Sensorknoten im Bereich des Ziels dieses auch erkennen. Umso höher dieser Wert ist, desto aussagekräftiger ist auch die Meldung. Die Historie entspricht den Erkennungen, welche Sensorknoten in zurückliegender Zeit gemacht haben. Haben diese für eine Zeit Ziele korrekt erkannt, ist die Wahrscheinlichkeit einer Falschmeldung gering. Je nach Gebiet kann die Gewichtung der Informationen variieren. Ist die Dichte der Sensorknoten sehr hoch, ist die räumliche Information eventuell ausreichend, um das Vertrauensniveau zu berechnen. Sind aber nur sehr wenige Knoten in einem Gebiet vorhanden, ist die historische Information höher einzuschätzen.

Die in der Arbeit von [Tav05] beschriebene Verwendung der Daten mehrerer Sensoren auf der Knotenebene stellt eine Datenfusion dar, ist aber nicht vergleichbar mit der Datenfusion der vorliegenden Arbeit, bei der Daten mehrerer Knoten verwendet werden. Dies geschieht auf der Gruppenschicht, wo allerdings nur noch reduzierte Informationen verwendet werden. Der geschilderte Ansatz der Berechnung des Vertrauensniveaus ist schwer übertragbar, da die Sensorknoten im vorliegenden System räumlich relativ gleichverteilt sind und historische Informationen meist nicht ermittelbar sind.

## 2.4 Fence Monitoring

In [Wit07] wird der Einsatz einer verteilten Ereigniserkennung am Beispiel eines Bauzauns im Kontext des Gebäudeschutzes untersucht. Dabei werden Ergebnisse in Vergleich zu einer lokalen Erkennung gestellt.

Die Sensorknoten sind mit Beschleunigungssensoren ausgestattet und jeweils an einem Bauzaunelement befestigt. Es werden sechs Typen von Ereignissen betrachtet: gegen den Zaun lehnen, gegen den Zaun treten, kurzes Schütteln, langes Schütteln, klettern bis man über den Zaun sieht und über den Zaun klettern. Zur Ereigniserkennung wird eine Architektur verwendet, welche aus vier Schichten besteht. In der untersten Schicht werden Daten lokal auf den Sensorknoten aggregiert. Hierbei werden periodisch Beschleunigungsdaten ausgewertet. Liegt ein Wert über einem vordefinierten Schwellwert, wird die Abtastrate erhöht und alle nachfolgenden Daten gespeichert. Fällt der Wert danach unter einen zweiten Schwellwert, wird Rate wieder gesenkt und prägnante Merkmale der gespeicherten Werte an die zweite Schicht weitergeleitet. Rohdaten müssen nur zwischen den beiden Zeitpunkten gespeichert werden, wo die Schwellwerte auftreten. Dadurch wird der Speicher- und Energiebedarf minimiert. Die Aufgabe der zweiten Schicht besteht darin, lokal einen Ereigniskandidaten aus einer aufgezeichneten Ereignissequenz auszuwählen. Dies wird regelbasiert durchgeführt. Als einziges Ereignis soll hier das Überqueren des Zauns gemeldet werden, andere Ereignisse könnten durch zusätzliche Regeln hinzugenommen werden. In der dritten Schicht wird eine Nachbarschafterkennung durch Verteilung von Ereigniskandidaten vorgenommen. Erhält ein Sensorknoten einen Kandidaten, bewertet er diesen mit Hilfe seiner eigenen Daten und sendet gegebenenfalls eine positive Rückmeldung an den sendenden Knoten. Bei genügenden positiven Meldungen wird das Ereignis durch die vierte Schicht an die Basisstation gesendet.



**Abbildung 2.2: Softwareschichten des verteilten Ereigniserkennungsalgorithmus [Wit07]**

Der im Fence Monitoring verwendete Anwendungsfall entspricht dem der vorliegenden Arbeit. Die erzielten Erkennungsraten der verteilten Ereigniserkennung beider Arbeiten können somit verglichen werden. Im Gegensatz zur beschriebenen Klassifikation auf zwei Ebenen wird bei dieser Arbeit nur eine Erkennung mit den Daten aller anderen Sensorknoten, die am Ereignis beteiligt sind, durchgeführt. Die Entscheidung, ob ein Ereignis an die Basisstation gesendet werden soll, ist nicht abhängig von den positiven Meldungen der Nachbarknoten, sondern muss von jedem Knoten mit Hilfe von Zurückweisungswerten entschieden werden. Weiterhin unterscheidet sich die Erkennung in dieser Arbeit durch einen linearen Mustererkennungsalgorithmen von der beschriebenen regelbasierten Schwellwerterkennung.

## 2.5 VigilNet

Die Überwachung von Kampfgebieten stellt ein risikoreiches Element in der Arbeit von Militärs dar. Es ist von großer Bedeutung, eindringende Ziele unbemannt mit Hilfe von großflächigen Sensornetzen erkennen zu können. VigilNet ist ein solches System für militärische Zwecke, mit dem Fahrzeuge, Personen und Personen, die eisenhaltige Gegenstände mit sich tragen, erkannt werden können. Gu et. al beschreibt den Ansatz dafür in [Gu05].

Eine der Forderungen des Systems ist, ein Ergebnis mit geringer Latenz zu liefern. Da die Sensorknoten viele Daten mit einer hohen Abtastrate aufnehmen, ist es unrealistisch, alle Daten an eine Basisstation zu senden, da dies bei vielen Knoten zu einer zu hohen Verzögerung führen würde. Die Erkennung und Klassifikation muss also innerhalb des Netzwerks stattfinden. Um dies zu erreichen, müssen die Sensorknoten kooperative Entscheidungen treffen. Dies motiviert die Verwendung einer hierarchischen Architektur, in diesem Fall bestehend aus vier Stufen, wie in Abbildung 2.3 zu sehen ist. Die untersten beiden Schichten, die Sensor- und die Knotenschicht, werden lokal auf den Knoten ausgeführt. Die Sensorschicht nimmt Daten mittels Bewegungssensoren, Mikrofonen und Magnetometern auf. Die Knotenschicht aggregiert diese und errechnet daraus eine lokale Klassifikation. In der darüber liegenden Gruppenschicht wird eine Klassifikation mit mehreren

Knoten durchgeführt. Eine Middleware verwaltet die Gruppen und bestimmt deren Leiter. Gruppenleiter sammeln die Klassifikationen ihrer Mitglieder und berechnen daraus erneut eine Klassifikation. Ein wichtiges Element dieser Schicht ist der minimale Grad der Aggregation (MDOA). Dieser sagt aus, dass eine bestimmte Menge an Informationen von den Mitgliedern gesammelt werden muss, damit die Aussage der Gruppenschicht gültig ist. In der obersten Schicht, der Basisschicht, werden die Aufgaben ausgeführt, die ein globales Wissen erfordern. Hier werden Daten für jedes bisherige erkannte Ziel bereitgehalten, unter anderem mit Werten für das Datum der letzten Änderung, Meldungen über das Ziel und letzte bekannte Position. Die Basisstation wählt das gespeicherte Ziel, wessen Position am nächsten zu der Position der ankommenden Meldungen ist. Liegen diese zu weit auseinander, wird das Ziel als neu eingestuft. Mit Hilfe der gespeicherten Daten und genügend eingetroffenen Meldungen kann durch eine Regressionsanalyse die Geschwindigkeit und Richtung des Ziels berechnet werden.

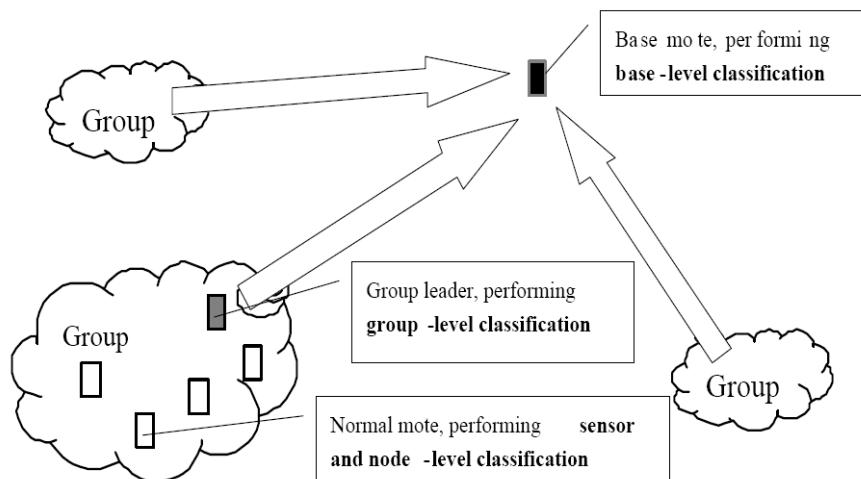


Abbildung 2.3: Hierarchische Architektur der Klassifikation [Gu05]

In der Arbeit werden keine genauen Aussagen darüber gemacht, wie die lokale und die verteilte Ereigniserkennung in den einzelnen Schichten ablaufen. Es werden Daten von mehreren Sensoren miteinander verknüpft und verarbeitet, was sich deutlich von der 1-Sensor-Datenaufnahme in der vorliegenden Arbeit unterscheidet und auf eine leistungsfähigere Hardware hindeutet. Die verteilte Ereigniserkennung wird in der beschriebenen Arbeit durch eine Erkennung auf mehreren Ebenen realisiert, da ein Ereignis von sehr vielen Knoten wahrgenommen werden kann. Im vorliegenden System ist diese Anzahl aber stark begrenzt, so dass eine Gruppenbildung nicht erforderlich ist und Daten zwischen den Sensorknoten ausgetauscht werden können.

## 2.6 Fahrzeugklassifikation in Sensornetzwerken

Die Klassifikation von Fahrzeugen bietet weitreichende Anwendungen im militärischen, aber auch zivilen Bereich, wie etwa bei Intelligenten Transportsystemen. In [Dua04] werden Methoden beschrieben, wie eine solche Klassifikation erreicht werden kann.

Die Sensorknoten benutzen akustische und seismische Sensoren, um Daten aufzunehmen. Daraus werden mit Hilfe einer Fast Fourier Transformation (FFT) 50-dimensionale Merkmalsvektoren extrahiert. Die Knoten, die in Cluster unterteilt sind, berechnen nach dem Auftreten eines Ereignisses lokal eine Klassifikation. Dazu werden verschiedene Methoden vorgestellt. Neben dem k-nächste-Nachbarn (KNN) und Support-Vektor-Maschinen (SVM) Klassifikator wird ein Maximum-Likelihood-Verfahren beschrieben. Hierbei wird eine Schätzfunktion verwendet, die mit Hilfe von bestehenden trainierten Daten die höchste Wahrscheinlichkeit der Zugehörigkeit zu einer bestimmten Klasse berechnet. Nach der Berechnung der Ergebnisse werden diese an den Kopf der Cluster gesendet. Dort wird eine Entscheidung getroffen, indem eine gewichtete Summe über die Wahrscheinlichkeiten oder Klassifikationen der Clusterknoten gebildet wird. Gewichte lassen sich auf verschiedene Weisen anhand der Distanzen der Sensorknoten zum Fahrzeug setzen. Möglichkeiten wären, das Gewicht eines Knotens auf 0 zu setzen, wenn die Distanz unter einem definierten Schwellwert oder kleiner als alle anderen Distanzen ist, sonst auf 1.

Die Hardware, die in [Dua04] verwendet wird, ist deutlich leistungsfähiger als die in der vorgestellten Arbeit benutzte Plattform. Die damit verbundenen Möglichkeiten lassen eine nur geringe Vergleichbarkeit zu. Es ist unter anderem möglich, hochdimensionale Merkmalsvektoren mit rechenintensiven Verfahren zu erstellen. Die vorgestellten lokalen Klassifikationsverfahren verwenden Trainingsdaten, die dazu auf den Sensorknoten gespeichert werden müssen. Dies ist bei dieser Arbeit aufgrund des geringen Speichers nicht möglich, so dass auf andere Methoden zurückgegriffen werden muss. Die verteilte Entscheidungsfindung findet in der vorliegenden Arbeit auf einer höheren Ebene statt, indem Ergebnisse der Klassifikationen der unteren Knotenebene aggregiert werden. Ein Austausch von Rohdaten oder Merkmalsvektoren wie in dieser Arbeit findet nicht statt.

### 3 Grundlagen

Die vorliegende Arbeit verwendet [Dzi07] als Grundlage und Ausgangspunkt, um das dort vorgestellte Erkennungssystem zu erweitern. In [Dzi07] wird ein Erkennungssystem beschrieben, welches Daten von Ereignissen durch verschiedene Fusionsmethoden anhand einer lokal auf den Sensorknoten durchgeführten Mustererkennung klassifiziert. Die dort beschriebenen Konzepte und Methoden sollen in diesem Kapitel zusammengefasst werden, um einen Überblick über das bestehende Erkennungssystem zu geben, der für das Verständnis der vorliegenden Arbeit notwendig ist.

Zuerst wird eine Übersicht über die verwendete Hardware- und Softwareplattform gegeben. Daraufhin wird der Sensor beschrieben, mit dem Ereignisdaten aufgenommen werden sollen. Weiterhin wird das Modell der allgemeinen und verteilten Mustererkennung erläutert. Im Anschluss daran wird die Anpassung und Umsetzung dieser Modelle in Bezug auf den Anwendungsfall geschildert. Dies teilt sich in die lokale und verteilte Mustererkennung auf.

#### 3.1 Hardware- und Softwareplattform

In dieser Arbeit wird die ScatterWeb Plattform eingesetzt, welche einen verteilten, heterogenen ad-hoc Einsatz von Sensornetzen ermöglicht. In [Sch05] wird die neueste, modular aufgebaute Generation von Sensorknoten beschrieben.

##### 3.1.1 Hardware

Es existieren drei Module, die wie in Abbildung 3.1 dargestellt, aufeinander aufgesetzt werden können.

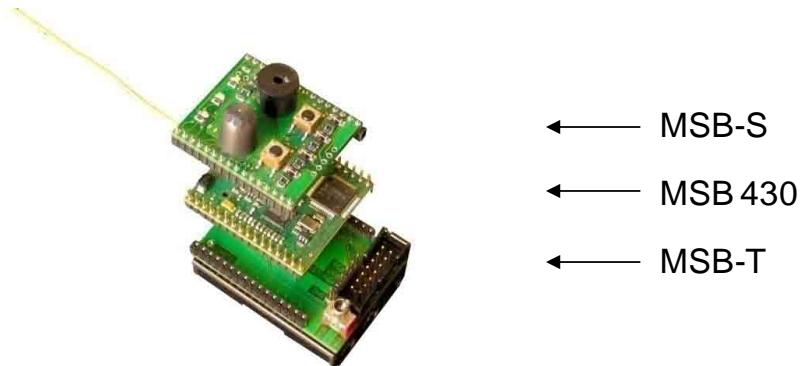


Abbildung 3.1: Sensorknotenmodule [Dzi07]

Das Basismodul MSB-430, verwendet den Texas Instruments MSP430F1612 [Tex06] Prozessor. Der Mikrokontroller besitzt einen Speicher von 60 KB, der in 5 KB flüchtigen Speicher (RAM) und 55 KB Festspeicher (Flash) aufgeteilt ist. Zur Funkübertragung dient der Chipcon CC1020 Transceiver [Chi05], welcher mit einer Datenrate von 19,2 kbit/s und auf einer Frequenz von 402-470 bzw. 804-970 MHz arbeitet. Auf dem MSB-430 ist neben einem Temperatur- und einem Luftfeuchtesensor ein Beschleunigungssensor an-

gebracht, welcher in Abschnitt 3.1.3 genauer beschrieben wird. Des Weiteren ist auf dem Modul eine LED zu finden.

Das Sensormodul MSB-S biete zusätzliche Sensoren und kann auf das Basismodul aufgesteckt werden. Es enthält ein Mikrofon, einen Piepser, einen Lichtsensor, drei LEDs sowie zwei programmierbare Knöpfe. Diese Zusatzausstattung wird nicht benötigt und somit wird auf die Verwendung des MSB-S Moduls verzichtet.

Zur Stromversorgung der anderen Module dient das Energieversorgungsmodul MSB-T. Als Stromquelle können sowohl drei 3V AAA Batterien als auch die auf dem Modul vorhandene USB-Schnittstelle verwendet werden. Diese Schnittstelle dient auch zur Kommunikation mit externen Geräten wie einem PC. Das Flashen des Speichers des Basismoduls geschieht über eine JTAG-Schnittstelle, welche ebenfalls auf dem Modul vorhanden ist.

### 3.1.2 Software

Die Software für die Sensorknoten ist zwei Bereiche unterteilt: Firmware und Anwendung. Die bestehende Firmware stellt Funktionen zur Steuerung der Sensorknoten, wie etwa das Setzen von Zeitgebern, Ein- und Ausgabe oder das Versenden von Nachrichten, dar. Auf diese kann über eine umfangreiche API zugegriffen werden. Weiterhin dient die Firmware zur Ausführung der Hauptschleife und Behandlung von Interrupts. Die Anwendung setzt auf der Firmware auf und muss ebenso wie diese in ANSI-C geschrieben werden.

### 3.1.3 Beschleunigung

Der auf dem MSB-430 angebrachte Dreiachsenbeschleunigungssensor MMA7260 von FreeScale [Fre06] liefert Spannungsdaten im Bereich von 0 bis 3 Volt. Spannungsveränderungen werden über eine geschaltete Kondensator-Technik ermittelt. Spannungswerte werden mittels eines Analog-Digital-Wandlers auf die Werte 0 bis 4095 linear abgebildet, was der maximal messbaren Beschleunigung entspricht. Diese wird durch Einstellung der Sensitivität des Sensors festgelegt. Die Ruheposition des Sensors entspricht einem erwarteten Wert von 2048. Praktisch gibt es verschiedene Faktoren, die diesen Wert beeinflussen können, wie etwa die Erdbeschleunigung oder baulich bedingte Schwankungen. Für jede Achse muss somit ein individueller Ruhepunkt ermittelt werden, was in Abschnitt 3.4.1.1 beschrieben wird.

## 3.2 Modell der allgemeinen Mustererkennung

Nach Theodoridis [The03] ist Musterkennung die wissenschaftliche Disziplin, dessen Ziel die Klassifikation von Objekten in eine Menge von Kategorien oder Klassen ist.

Dieses Ziel wird durch Lösen mehrerer Teilprobleme erreicht. Es existiert eine Reihe von Modellen, welche eine Lösung für die Zerlegung des Mustererkennungsproblems anbieten. Ein solches Modell, welches vielfach eingesetzt wird, zeigt Abbildung 3.2. Es muss den gegebenen Bedingungen und Anforderungen sowie dem Anwendungsfall angepasst werden.

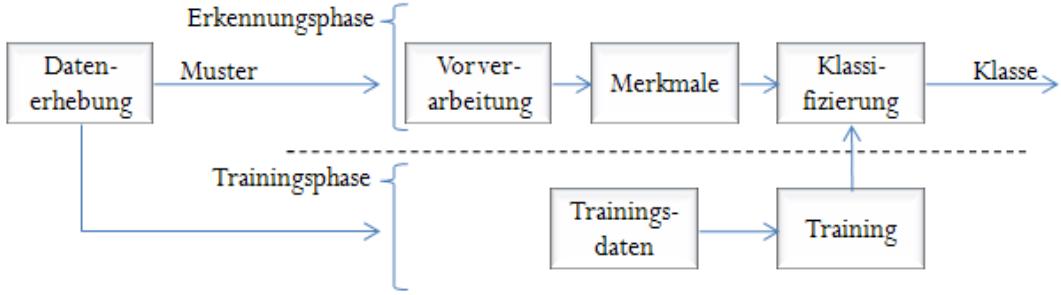


Abbildung 3.2: Modell der Mustererkennung [Dzi07]

### 3.2.1 Komponenten des allgemeinen Mustererkennungsmodells

Die Komponenten des oben abgebildeten Mustererkennungsmodells dienen dazu, aus aufgenommenen Daten eine Klassifikation zu erstellen. Eine umfassende Darstellung der Komponenten wird in vielen anerkannten Werken wie [Web02], [Fuk90], [Dud01], [Nie83], [The03] gegeben, ist jedoch hier nicht möglich, so dass nur eine kurze Übersicht präsentiert werden soll.

#### 3.2.1.1 Datenerhebung

Mittels Sensoren werden Daten oder Signale aufgenommen und digitalisiert. Die digitalisierten Daten werden Rohdaten genannt. Die Datenaufnahme stellt den Ausgangspunkt für die weitere Mustererkennung dar.

#### 3.2.1.2 Vorverarbeitung

Die Vorverarbeitung dient dazu, Muster in andere Muster zu transformieren, ohne Kerninformationen zu verlieren. Das Ziel dabei ist, dass das entstandene Muster für die weitere Verarbeitung besser geeignet ist.

Eine Verringerung von Rauschen und die damit verbundene Signalglättung können durch Einsatz von Filtern realisiert werden. Mit Hilfe der Segmentierung lassen sich Muster voneinander abgrenzen oder aus einem Datenstrom extrahieren. Die Normierung bildet die Muster auf einen einheitlichen Wertebereich ab, um verschiedene Muster vergleichbar zu machen.

#### 3.2.1.3 Merkmale

Die Durchführung einer Klassifikation mit allen Werten der (vorverarbeiteten) Rohdaten würde zu einem unzweckmäßigen Aufwand führen. Es muss also eine Komprimierung der Datenmenge auf repräsentative Merkmale stattfinden.

Die Merkmalsextraktion erzeugt Merkmale aus Rohdaten, so dass diese die zugehörigen Klassen so genau wie möglich beschreiben. Insbesondere sollen sich Merkmale verschiedener Klassen unterscheiden, Merkmale gleicher Klassen aber ähneln. Als eine der Grundideen geht mit der Merkmalsextraktion oft eine Dimensionsreduktion bzw. Datenkomprimierung einher. Redundanzen werden ausgenutzt und entfernt und klassenbezogene Informationen werden in eine verdichtete Anzahl von Merkmalen überführt.

Mit Hilfe der Merkmalsextraktion kann man trotz Datenkomprimierung eine große Anzahl von Merkmalen erhalten. Mit steigender Anzahl von Merkmalen steigt der Aufwand für die Klassifikation, aber auch die Wahrscheinlichkeit einer höheren Fehlerrate ist gegeben. Die Aufgabe der Merkmalsauswahl ist es, aus einer gegebenen Anzahl von Merkmalen die geeigneten herauszusuchen, so dass zum Einen ihre Anzahl reduziert, zum Anderen so viel klassencharakterisierende Informationen wie möglich beibehalten werden. Die Merkmalsauswahl besteht aus einem Auswahlverfahren, womit die Menge an ausgewählten Merkmalen vergrößert oder verringert wird, sowie einer Bewertungsfunktion für diese Menge.

### 3.2.1.4 Klassifizierung

Die Klassifizierung ordnet ein Muster einer Klasse zu. Die Zuordnung basiert dabei auf den Informationen, die aus dem Muster erhalten wurden. Dementsprechend werden die aus den Rohdaten des Musters extrahierten Merkmale für die Klassifizierung verwendet. Die Definition eines Klassifikators, der einen Merkmalsvektor  $\underline{x}$  auf ein Element  $k$  aus einer Menge von Klassen  $K$  abbildet, ist in Gleichung (1) [Dzi07] gegeben.

$$\underline{x} \rightarrow k, \quad k \in K \quad (1)$$

Es existieren eine Reihe von theoretischen Ansätzen für Klassifizierungsmethoden, wie etwa statistische Klassifizierung, Neuronale Netze, Entscheidungsbäume, Support Vektor Maschinen und Polynom-Klassifizierung, um nur einige zu nennen.

## 3.2.2 Phasen des allgemeinen Mustererkennungsmodells

Eine Mustererkennung läuft in mehreren Phasen ab. Man unterscheidet zwischen der Trainingsphase, in welcher Repräsentanten für die Klassen erzeugt werden, und der Erkennungsphase, wo anhand dieser Repräsentanten eine Klassifizierung vorgenommen wird. Die Trainingsphase wird somit vor der Erkennungsphase durchgeführt.

### 3.2.2.1 Trainingsphase

Ein Muster kann einer Klasse während der Klassifizierung nur dann zugeordnet werden, wenn Informationen über alle Klassen vorhanden sind. Während der Trainingsphase werden diese Informationen ermittelt. Die während dieser Phase aufgenommenen Muster werden auch Trainingsdaten oder Stichproben genannt. Die daraus erhaltenen Merkmalsvektoren sollten die zugehörigen Klassen so genau wie möglich beschreiben. Neben der Merkmalsextraktion spielt hier also auch die Merkmalsauswahl eine große Rolle. Die Zuweisung einer Stichprobe zu einer Klasse kann auf verschiedene Arten geschehen. Bei einem überwachten Training wird festgelegt, zu welcher Klasse eine bestimmte Stichprobe gehört. Bei einem nicht überwachten Training steht nur die Anzahl der Klassen fest, die Stichproben werden mit Hilfe von Algorithmen wie Clustertechniken und Merkmalsauswahl in Gruppen mit hoher Ähnlichkeit der Merkmale zusammengefasst.

### 3.2.2.2 Erkennungsphase

Nachdem genügend Trainingsdaten erfasst und diese den Klassen zugeordnet wurden, startet die Erkennungsphase. Zu erkennende Muster werden hierbei mit den Trainingsda-

ten verglichen. Die Erkennungsphase verwendet bis auf die Merkmalsauswahl und einer veränderten Klassenzuordnung dieselben Komponenten wie die Trainingsphase. Die Merkmalsauswahl wird nicht verwendet, da die zu verwendenden Merkmale bereits während des Trainings bestimmt worden sind. Die Klassifizierung wird anhand eines Klassifikationsalgorithmus und dem dazu entsprechenden Vergleichsverfahren vorgenommen.

### 3.3 Modell der verteilten Mustererkennung

Eine verteilte Mustererkennung kann durch Fusion von Daten erreicht werden. Es existieren Modelle, wie die Mustererkennung in Zusammenhang mit einer verteilten Entscheidungsfindung gebracht werden kann.

#### 3.3.1 Abstraktionsebenen der Datenfusion

Fusionen von Daten können auf verschiedenen Ebenen eines Mustererkennungssystems vorgenommen werden. Allgemein gilt, dass mit einer höheren Abstraktionsebene weniger Daten verschickt werden müssen. Dies senkt den Kommunikationsaufwand. Eine niedrige Ebene bedeutet mehr Daten, wodurch eine präzisere Erkennungsaussage erreicht werden kann. Nach [Rus07] existieren folgende Abstraktionsebenen:

- *Signalebene*  
Aufgenommene Messwerte aller Sensoren werden ausgetauscht. Aufgrund des hohen Kommunikations- und Synchronisationsaufwands ist diese Möglichkeit zu vernachlässigen.
- *Merkmalsebene*  
Merkmalsvektoren, welche aus den Messwerten der Sensoren extrahiert wurden, werden ausgetauscht. Die Größe der Merkmalsvektoren kann angepasst werden, um das Verhältnis zwischen Kommunikation und Erkennung zu optimieren.
- *Klassifikationsebene*  
Mehrere Klassifikationsergebnisse werden verwendet, um daraus anhand von statistischen Verfahren eine Gesamtklassifikation zu generieren. Einzelne Ergebnisse können das Gesamtergebnis anhand von Bewertungsmethoden mit verschiedener Gewichtung beeinflussen.

#### 3.3.2 Fusionsmodell „Omnibus“

Das Omnibusmodell (siehe Abbildung 3.3) stellt einen typischen Ablauf einer Mustererkennung mit den Möglichkeiten einer Datenfusion dar. Es werden Transformationen zusammen mit Schnittstellen aufgezeigt, an denen die entstandenen Daten fusioniert werden können.

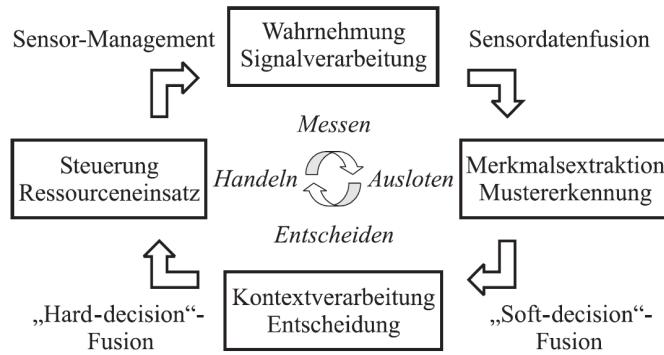


Abbildung 3.3: Fusionsmodell Omnibus [Dzi07]

### 3.4 Lokale Mustererkennung

Damit Muster in einem verteilten Sensornetzwerk erkannt werden können, muss auch jeder einzelne Knoten dazu in der Lage sein. Um den Kommunikationsaufwand für verteilte Erkennungen gering zu halten, sollen nur wenige Informationen wie Merkmalsvektoren oder Klassifikationsergebnisse ausgetauscht werden. Demzufolge ist die Fähigkeit einzelner Knoten, diese Informationen zu berechnen, vorteilhaft.

Das allgemeine Mustererkennungsmodell wird für eine konkrete Implementierung modifiziert, wie in Abbildung 3.4 abgebildet ist.

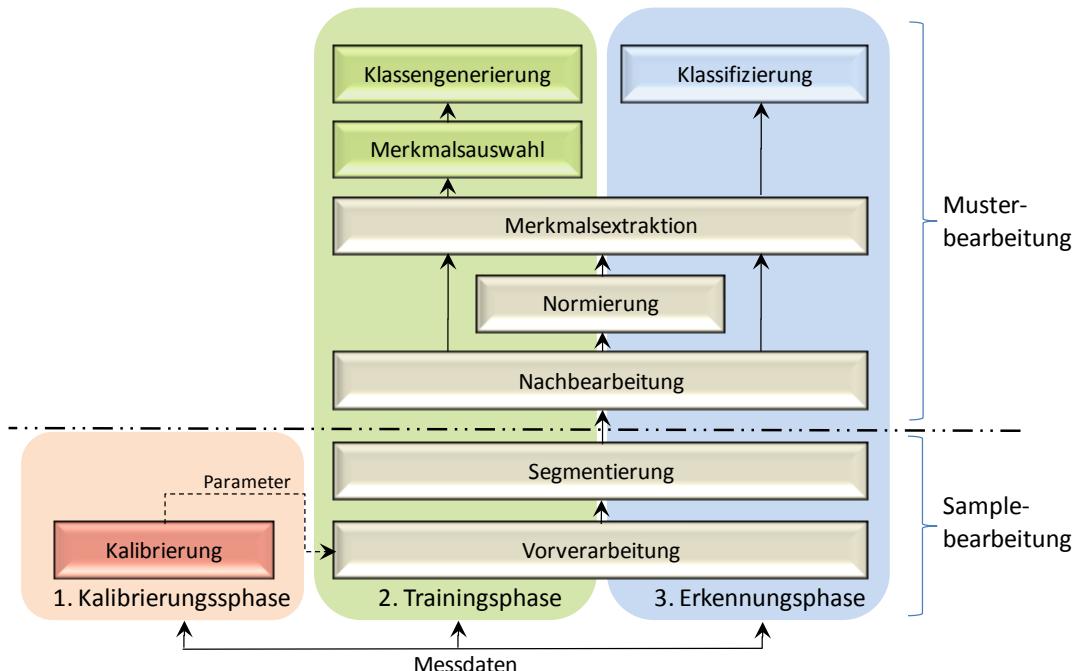


Abbildung 3.4: modifiziertes Mustererkennungsmodell [Dzi07]

#### 3.4.1 Modellkomponenten der lokalen Mustererkennung

Aufgrund technischer Vorteile ist es sinnvoll, die Komponenten des allgemeinen Mustererkennungsmodells zu verfeinern. Die Komponente Kalibrierung wird zum Mustererkennungssystem hinzugefügt. Die Segmentierung wird aus der Vorverarbeitung herausgelöst,

da sie in diesem Fall unabhängig von der eigentlichen Vorverarbeitung ist. Ebenso geschieht dies für die Normierung, da diese nicht für alle Merkmale durchgeführt werden muss. Die Merkmalsauswahl wird nur in der Trainingsphase durchgeführt. Die Klassifizierung stellt während der Trainings- und Erkennungsphase einen unterschiedlichen Prozess dar und wird deshalb getrennt betrachtet. Nachfolgend werden die einzelnen Modellkomponenten im Detail erläutert.

### **3.4.1.1 Kalibrierung**

Der verwendete Beschleunigungssensor unterliegt im Ruhezustand Spannungsschwankungen, unter anderem bedingt durch Alter, Temperatur, bauliche Schwankungen und Erdbeschleunigung. Die Kalibrierung dient dazu, einen geeigneten Wert für die Ruhelage eines Sensors zu finden, so dass die ermittelten Daten so fehlerfrei wie möglich sind.

Der Nullpunkt, der die Ruhelage des Sensors charakterisiert, wird als Offset bezeichnet. Dieser wird berechnet, in dem der Mittelwert über alle Beschleunigungsdaten gebildet wird, die der Sensor während einer gewissen Zeit in Ruheposition ausgibt. Durch Subtraktion dieses Wertes kann zwischen einer positiven und negativen Beschleunigung unterschieden werden. Des Weiteren kann mit Hilfe des Offsets ermittelt werden, ob ein Muster beginnt bzw. endet.

In bestimmten Situationen ist es nicht möglich, in Ruhelage einen eindeutigen Nullpunkt zu bestimmen. Durch Zittern können Bewegungen erzeugt werden, die ein Schwanken der Beschleunigungsdaten um den Nullpunkt herum darstellen. Die Ruhelage wird in dem Fall somit fälschlicherweise nicht mehr vom System erkannt. Um dies zu vermeiden, werden die Extremwerte der Schwankungen um den Offset gespeichert und für die Definition des Ruhepunktes mit verwendet.

Zwischen der Aufnahme zweier Messwerte muss immer die gleiche Zeit, das sogenannte Samplingintervall, vergehen. In dieser Zeit werden Operationen ausgeführt, um die Werte zu verarbeiten. Das Zeitfenster muss so eingestellt werden, dass nur eine maximal notwendige Zeitdauer zwischen den Aufnahmen gewartet wird.

### **3.4.1.2 Datenerhebung**

Der Beschleunigungssensor ist in der Lage, Beschleunigungen von drei Achsen zu messen. Ein gemessener Datensatz wird Sample genannt. Ein Sample entspricht einem Tripel mit den Beschleunigungswerten der Achsen mit einer Auflösung von 12 Bit.

### **3.4.1.3 Vorverarbeitung**

Jedes Sample wird direkt nach dessen Aufnahme bearbeitet. Zuerst wird eine Bitreduktion von 12 auf 8 Bit durchgeführt, um damit physikalisches Rauschen zu verringern und eine bessere Berechenbarkeit der Nullwerte zu erreichen. Von diesen Werten wird der während der Kalibrierung berechnete Offset subtrahiert. Die aufgenommenen Daten unterliegen Schwankungen, die geglättet werden müssen. Störungen im Signalverlauf werden dadurch entfernt. Der im Beschleunigungssensor eingebaute Tiefpassfilter ist voreingestellt und kann nicht auf verschiedene Situationen angepasst werden. Es wird deshalb zusätzlich ein Tiefpassfilter verwendet, der das Signal verschieden stark glätten kann. Der

sogenannte moving average Filter zur Glättung von Zeitreihen ersetzt jedes Element der Zeitreihe ab dem  $k$ -ten Element durch den Mittelwert der  $k$  linken Nachbarelementen, das Element selbst mitgezählt. Umso höher der Wert für  $k$  gewählt wird, desto mehr Beschleunigungsspitzen werden entfernt.

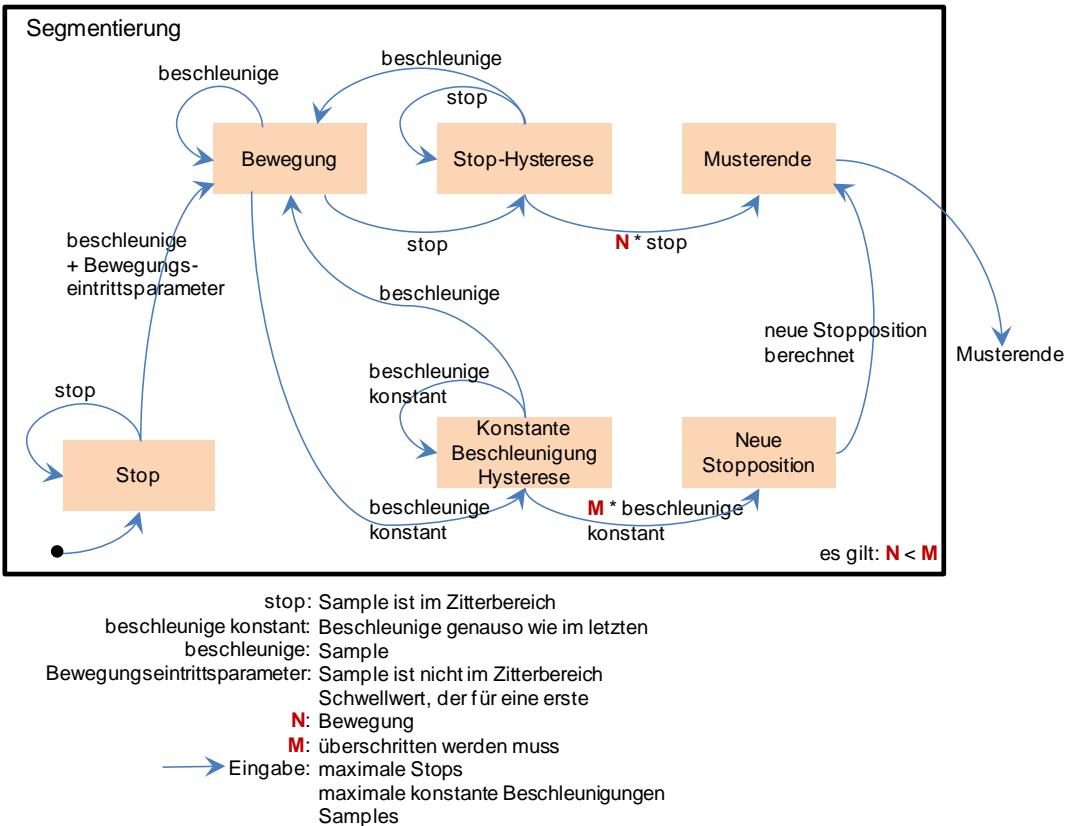
### 3.4.1.4 Segmentierung

Die Segmentierung stellt für jeden aufgenommenen Datenpunkt fest, ob es den Beginn, das Ende oder einen Teil eines laufenden Musters darstellt. Die zugehörige Darstellung ist in Abbildung 3.5 zu finden.

Der Messwert bestimmt dabei den Zustand, in dem sich der Sensorknoten befindet: Bewegungszustand oder Stoppzustand. Dieser kann bei jedem neuen Sample wechseln. Ein Muster ist definiert als eine Folge von Bewegungszuständen nach einem Stoppzustand und dem darauf folgenden Stopp. Die Werte für Musterkandidaten werden so lange gespeichert, bis kein Speicher mehr dafür vorhanden ist. Die Segmentierung endet aber erst, wenn ein Musterende gefunden ist. Ein Musteranfang ist dadurch charakterisiert, dass der Messwert außerhalb des Zitterbereichs und einem dazu akkumulierten Bewegungseintrittsparameter liegt. Darauffolgende Bewegungszustände müssen außerhalb des Zitterbereichs liegen. Eine Folge Stoppzuständen, welche Bewegungen innerhalb des Zitterbereichs entsprechen, stellt ein Musterende dar.

Eine Stopp-Hysterese sorgt dafür, dass ein kurzer Stopp von einem Musterende unterschieden werden kann. Erst wenn die Beschleunigungswerte für  $N$  Samples innerhalb des Zitterbereichs liegen, wird das als Musterende gewertet.

Die Konstante Beschleunigungs-Hysterese ist dafür notwendig, einen Stoppzustände bzw. Musterende auch bei veränderter Ruhelage zu erkennen. Verschiebt sich der Beschleunigungssensor, wirkt eine konstante Beschleunigung auf ihn und die Ruheposition und ein damit verbundenes Musterende wird nicht mehr erreicht. Bleibt die Beschleunigung für  $M$  Samples konstant, wird aus diesen Werten ein neuer Nullpunkt berechnet und dies als Musterende gewertet.



**Abbildung 3.5: Segmentierung mit Stopp- und Konstante Beschleunigungs-Hysterese**  
 [Dzi07]

### 3.4.1.5 Nachbearbeitung

Das aktuelle Muster wird verworfen, wenn während der Segmentierung kein Musterende bis zum Erreichen der Speichergrenze festgestellt werden konnte. Ist das Muster gültig, werden die Daten, die zur Erkennung eines Musterendes beitragen und somit keine relevante Bewegung darstellen, von den Musterdaten entfernt.

### 3.4.1.6 Normierung

Um eine Vergleichbarkeit von Mustern herzustellen, müssen diese normiert werden. Es existieren jedoch Merkmale, die etwa Informationen über Intensität und Zeit beinhalten, welche aus nicht normierten Daten gewonnen werden müssen. Diese Informationen müssen vor der Normierung extrahiert werden.

Die Normierung erfolgt einmal über die Zeit, die Werte werden linear auf die maximale Musterdauer interpoliert. Zum Anderen wird eine deformierende Wertnormierung ausgeführt, welche die Werte des Musters auf einen Bereich von 0 bis 255 abbildet.

### 3.4.1.7 Merkmalsextraktion

Merkmale werden aus dem gesamten Muster extrahiert. Es werden statistische Merkmale verwendet, die auf nicht normierten Daten basieren. Dazu gehören minimaler und maximaler Beschleunigungswert, Dauer, Mittelwert und Varianz der Musterdaten. Ein Verfahren, welches Merkmale aus normierten Daten extrahiert, ist die Berechnung eines Histogramms. Musterwerte werden nach dem Beschleunigungswert sortiert und die Werte in gleich große Klassen unterteilt. Für jede Klasse ergibt sich das Merkmal aus der Differenz

ihres größten und kleinsten Wertes. Die Klassengröße und somit Anzahl der Merkmale ist variabel.

Ähnlich der Musternormierung müssen auch Merkmale vergleichbar sein und somit normiert werden. Dies wird für jedes Merkmal anhand seiner minimalen und maximalen Werte durchgeführt, die während des Trainings klassenübergreifend erlernt wurden. Diese Werte stellen den Erwartungsraum des Merkmals für alle Klassen dar. Alle Merkmale der Trainingsdaten und die des zu klassifizierenden Musters werden anhand der jeweiligen minimalen und maximalen Werte auf einen Bereich von 0 bis 255 linear projiziert.

### 3.4.1.8 Merkmalsauswahl

Die Merkmalsauswahl wird aufgrund der Komplexität einer automatischen Auswertung nur manuell durchgeführt. Dazu werden die Abstände einzelner Merkmale zwischen verschiedenen Klassen betrachtet. Werden die Klassen durch hohe Abstände gut voneinander abgetrennt, werden die entsprechenden Merkmale für die Mustererkennung verwendet.

### 3.4.1.9 Klassengenerierung und Klassifizierung

Eine typische Klassifizierung würde ein zu erkennendes Muster mit allen Trainingsdaten vergleichen. Aufgrund der verfügbaren Speichergröße auf den Sensorknoten ist dies aber nicht möglich. Zur Lösung des Problems wird ein gewichteter überwachter k-means-Algorithmus [Kal01] verwendet.

Üblicherweise arbeitet ein k-means-Algorithmus nicht überwacht, so dass alle Trainingsdaten gespeichert werden müssen und dann  $k$  Klassen zugeordnet werden. Aufgrund des überwachten Verfahrens ist dies hier nicht nötig. Eine weitere Eigenschaft ist die Bildung von Klassenzentren anhand der Trainingsdaten. Dies führt hier zu einer weiteren Reduktion des Datenumfangs. Zur Klassengenerierung wird für jede Klasse aus den Merkmalsvektoren der zugehörigen Trainingsdaten der Mittelwert gebildet, die entstehenden Vektoren fungieren als Referenzvektoren. Da die minimalen und maximalen Werte für die Merkmalsnormierung feststehen, wird diese auf den Referenzvektoren angewandt. Für jede Klasse wird somit ein normierter Merkmalsvektor gespeichert, welcher den Mittelpunkt der Trainingsdaten darstellt.

Zur Klassifizierung muss ein Muster mit den Referenzvektoren aller Klassen verglichen werden. Die extrahierten Merkmale des Musters müssen dazu denen der Referenzvektoren entsprechen. Der Abstand  $d(xv, rv)$  dieses Merkmalsvektors  $xv$  zu allen Referenzvektoren  $rv$  wird mittels des Euklidischen Abstands berechnet.

$$d(xv, rv) = \|xv - rv\| = \sqrt{\sum_{i=1}^m (xv_i - rv_i)^2} \quad (2)$$

Zur Optimierung des Rechenaufwands wird auf das Ziehen der Wurzel verzichtet, da nur ein relativer Abstand interessant ist. Das Muster wird der Klasse mit dem minimalen Abstand zum zugehörigen Referenzvektor zugeordnet.

### 3.4.2 Modellphasen der lokalen Mustererkennung

Neben der Trainings- und Erkennungsphase wird eine Kalibrierungsphase hinzugefügt. Die Phasen werden in einem Automaten abgearbeitet (siehe Abbildung 3.6). Je nach Zustand des Automaten werden zugehörige Routinen aktiviert.

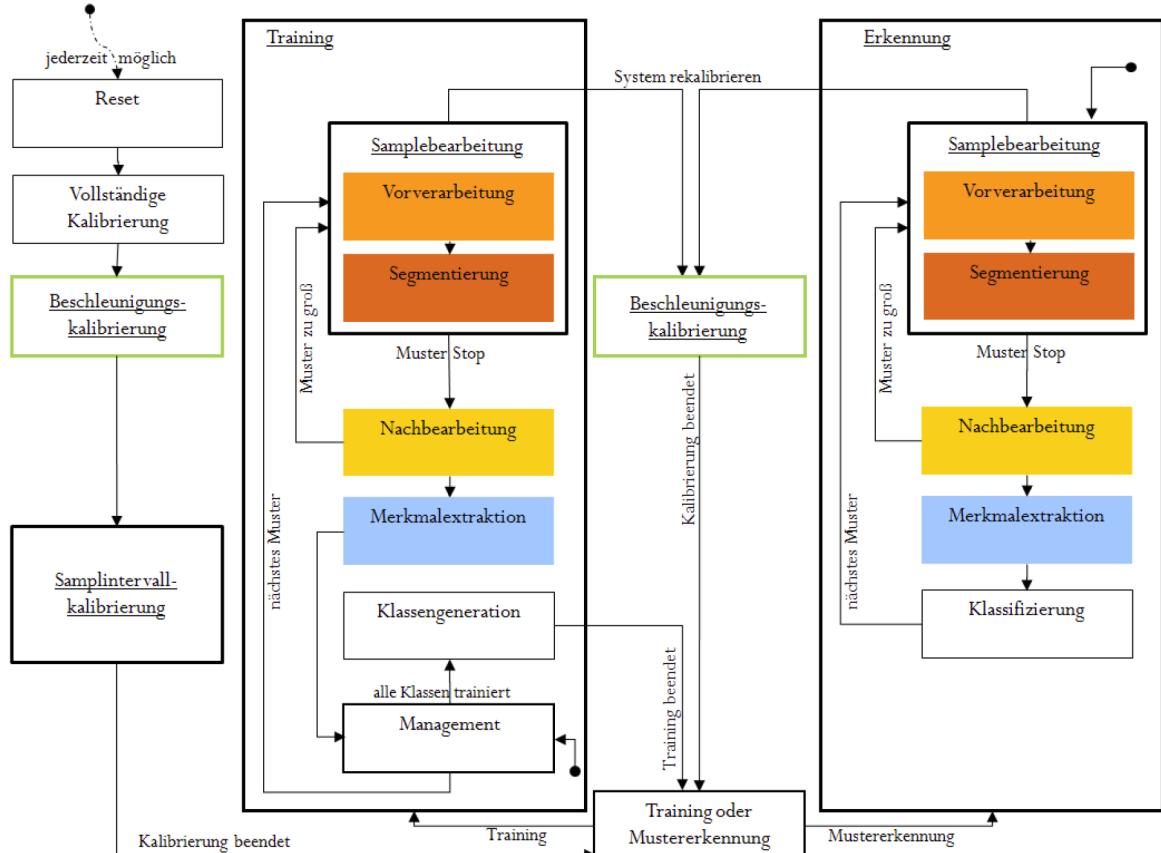


Abbildung 3.6: Automatendarstellung der lokalen Mustererkennung [Dzi07]

### 3.4.2.1 Kalibrierungsphase

Die aufgenommenen Daten sollen dem tatsächlichen Verlauf einer Bewegung entsprechen. Um Daten verwerten zu können, muss der Sensorknoten in Ruhelage kalibriert werden.

### 3.4.2.2 Trainingsphase

Daten vom Beschleunigungssensor werden aufgenommen und verarbeitet. Entstehende Merkmalsvektoren werden akkumuliert und daraus Referenzvektoren gebildet. Das Training wird überwacht durchgeführt, so dass nicht alle Trainingsdaten gespeichert werden müssen. Dadurch kann das Training lokal auf den Sensorknoten durchgeführt werden.

### 3.4.2.3 Erkennungsphase

Es werden dieselben Komponenten wie in der Trainingsphase bis auf die Merkmalsauswahl durchlaufen. Zur Erkennung unbekannter Muster werden die in der Trainingsphase erzeugten Referenzvektoren zum Vergleich herangezogen.

#### 3.4.2.4 Datenbearbeitung

Daten eines Musters werden zum Einen einzeln direkt nach der Datenaufnahme (Sampelbearbeitung), zum Anderen auch in ihrer Gesamtheit (Musterbearbeitung) behandelt. Während der Musterbearbeitung können keine neuen Daten erfasst werden, da die Rechenkapazität des Sensorknotens dies nicht zulässt.

### 3.5 Verteilte Mustererkennung

Durch die Verwendung von Daten mehrerer Sensorknoten zur Entscheidungsfindung soll eine höhere Erkennungsgenauigkeit erreicht werden. Zur Umsetzung der verteilten Mustererkennung wird das Omnibusmodell verwendet. Es werden Methoden aufgezeigt, die eine verteilte Entscheidungsfindung ermöglichen. Alle im Omnibusmodell aufgezeigten Schnittstellen stehen im verwendeten Mustererkennungssystem zur Verfügung, es wird aber nur von der Merkmalsfusion („Soft-decision“-Fusion) und Klassifikationsfusion („Hard-decision“-Fusion) Gebrauch gemacht. Die Fusion auf der Signalebene wird aufgrund der hohen technischen Anforderungen nicht betrachtet.

#### 3.5.1 Methoden

Daten werden mittels der komplementären Integration [Rus07] fusioniert. Dabei werden vergleichbare Sensordaten fusioniert, um eine größere Bandbreite an Informationen zu erreichen. Die Aussagekraft einzelner Sensordaten kann überprüft und relativiert werden. Durch die komplementäre Integration können Muster aus verschiedenartigen Musterteilen genauso wie aus identischen Teilen zusammengesetzt werden. Dadurch entstehende Muster werden in beiden Fällen gleich behandelt. Fusionierte Daten können auf verschiedene Weisen ausgewertet werden.

- *Klassifikationsfusion*  
Ergebnisse der Klassifikation mehrerer Sensorknoten werden gemittelt oder gewichtet analysiert. Gewichte können zusätzliche Aussagen über Qualität oder Zuverlässigkeit einzelner Sensorknoten in die Berechnung einbringen.
- *Merkmalsfusion*  
Die bei der Merkmalsextraktion erzeugten Merkmale mehrerer Sensorknoten werden zur Berechnung einer Klassifikation herangezogen. Aufgrund der höheren Anzahl an Daten lassen sich präzisere Informationen zur Entscheidungsfindung nutzen.
- *Kooperative Fusion*  
Klassifikationsfusion und Merkmalsfusion werden verknüpft. Der Ansatz dabei ist, dass wenn die Klassifikationsfusion Ergebnisse aus qualitativ schlechten Daten erhält, diese noch um Daten aus Merkmalsfusion ergänzt werden kann.

### 3.5.2 Phasen und Komponenten der verteilten Mustererkennung

Das verteilte Mustererkennungssystem wird auf das lokale System aufgesetzt, wie in Abbildung 3.7 zu sehen ist. Dazu werden neue Komponenten und Phasen hinzugefügt, welche eine Kommunikation zwischen den Sensorknoten und eine Auswertung fusionierter Daten ermöglichen.

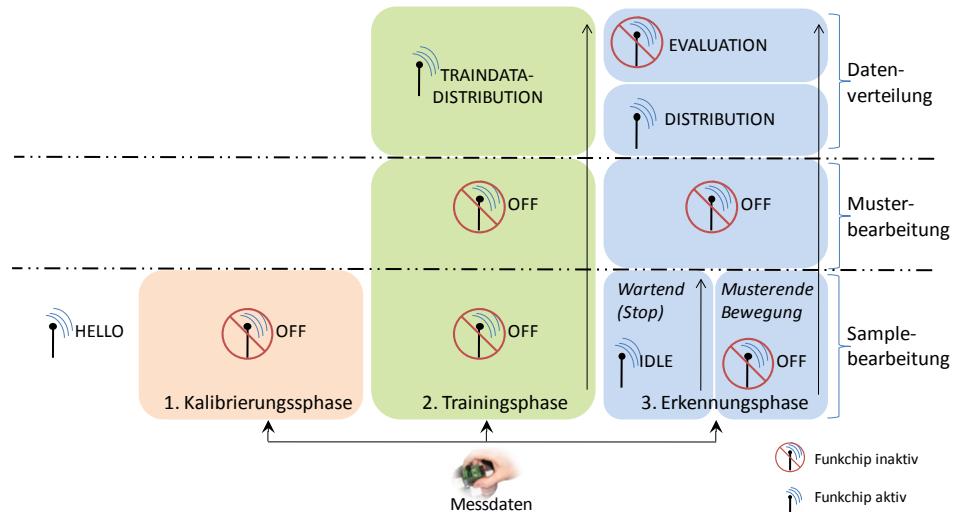


Abbildung 3.7: Lokales Mustererkennungssystem erweitert um verteilte Erkennung [Dzi07]

Die zusätzlichen Phasen und Komponenten, welche für die verteilte Mustererkennung eingeführt wurden, werden dem Automaten der lokalen Mustererkennung hinzugefügt (siehe Abbildung 3.8).

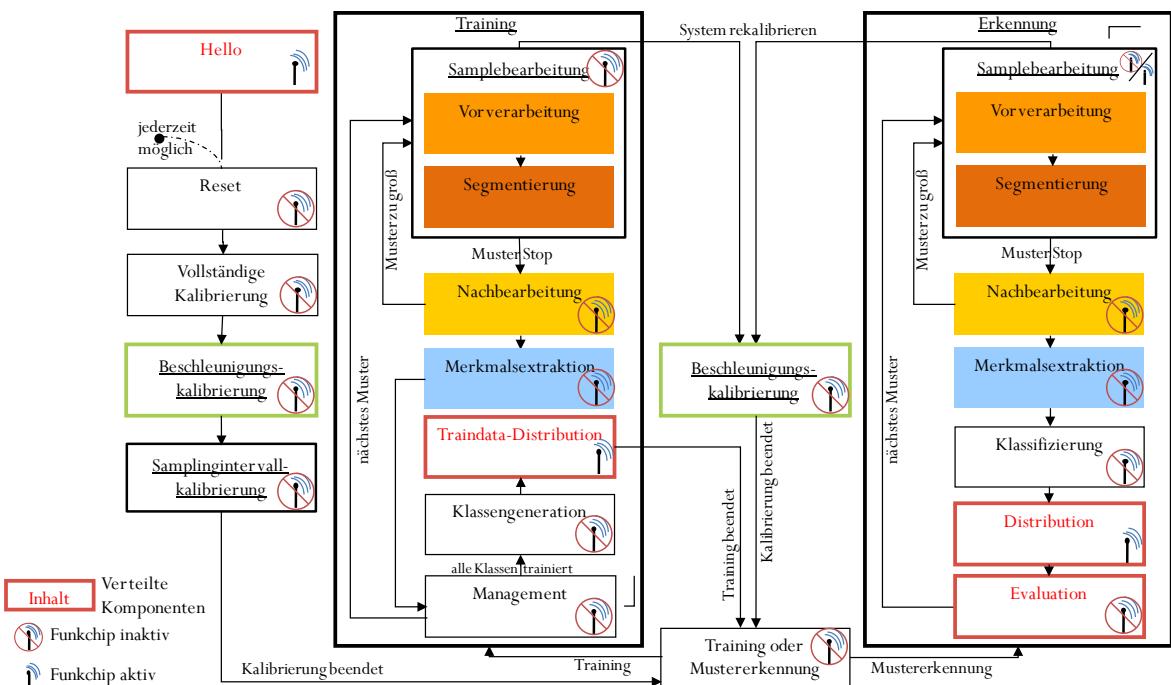


Abbildung 3.8: Um Zustände erweiterter Automat der verteilten Mustererkennung [Dzi07]

Während der „Hello“-Phase direkt nach dem Einschalten tauschen Sensorknoten ihre Identitäten aus, um sich bei anderen Knoten zu registrieren. Nur Daten von registrierten Knoten können später in die Klassifikation einfließen.

Im „Off“-Zustand können keine Nachrichten empfangen werden. Dieser wird dann eingenommen, wenn der Sensorknoten in einer Berechnungs- oder Kalibrierungsphase ist und nicht durch andere Ereignisse beeinflusst werden sollte.

Nachdem Sensorknoten ihre Klassifikationen erzeugt haben, werden die Ergebnisse und Merkmalsvektoren an andere Knoten im „Distribution“-Zustand verschickt. In variabel festlegbaren Zeitabständen wird überprüft, ob Daten von allen registrierten Knoten erhalten worden sind. Ist diese nicht der Fall, wird eine Anfrage an die entsprechenden Knoten gestellt. Erhaltene Daten werden in einer festgelegten Reihenfolge abgespeichert.

Falls die Methode der Merkmalsfusion verwendet wird, werden die nach dem Training generierten Referenzvektoren im „Traindata-Distribution“-Zustand verschickt. Erhaltene Vektoren werden wie im „Distribution“-Zustand in einer festgelegten Reihenfolge gespeichert.

Ausgetauschte Daten werden im „Evaluation“-Zustand ausgewertet. Zur Durchführung der Auswertung müssen Daten von allen registrierten Knoten vorliegen. Theoretisch würde die Auswertung auf einem Sensorknoten genügen, da nach der Fusion bei allen registrierten, am Ereignis beteiligten Sensorknoten im Netz dieselben Daten vorliegen sollten. Auf diese Auswahl wird verzichtet. Die in Abschnitt 3.5.1 vorgestellten Konzepte der Auswertung werden wie folgt konkretisiert:

- *Klassifikationsfusion*  
Klassifikationsergebnisse werden gleich gewichtet betrachtet. Gibt es mehr als 50% gleiche Stimmen, wird das zugehörige Ergebnis als Mehrheitsentscheidung gewählt.
- *Merkmalsfusion*  
Der Klassifikationsalgorithmus wird aus der lokalen Mustererkennung übernommen und auf die im „Traindata-Distribution“-Zustand fusionierten Referenzmerkmalsvektoren und im „Distribution“-Zustand fusionierten Merkmalsvektoren angewandt. Der kürzeste Euklidische Abstand liefert das Klassifikationsergebnis. Die Verteilung der Vektoren einzelner Knoten nach der Fusion ist aufgrund einer festgelegten Speicherreihenfolge, die sich nach der Knoten-ID richtet, immer gleich.
- *Kooperative Fusion*  
Sofern während der Klassifikationsfusion keine 50% Mehrheit an gleichen Stimmen erreicht wird, gilt das Ergebnis der Merkmalsfusion als verteilte Entscheidung.
- *Kooperative Fusion mit Vetorecht*  
Die Merkmalsfusion wird dann herangezogen, wenn auch nur zwei beliebige Klassifikationsergebnisse verschiedener Knoten ungleich sind.

## 4 Erweiterung der verteilten Ereigniserkennung

Eine Ereigniserkennung, welche anhand von Daten nur eines Sensorknotens ausgeführt wird, weist inhärente Nachteile gegenüber der Verwendung von Daten verschiedener Quellen auf.

[Rus07] betont die Unsicherheit bei der Interpretation eines Einzelsignals und weist auf eine höhere Glaubwürdigkeit und Wahrnehmungsfähigkeit mehrerer Signale hin. In [Gu05] wird die Auffassung vertreten, dass Beobachtungen der Umwelt eines einzelnen Knotens kein verlässlicher Hinweis auf Ereignisse in einem Netzwerk sind. Eigenschaften lassen sich nicht vollständig und robust erfassen. Kombiniert man Daten von mehreren Quellen, erhält man eine höhere Genauigkeit und kann genauere Rückschlüsse ziehen [Lli90] , [Kle93] .

Zu diesem Schluss kommt auch [Dzi07] , dessen verteiltes Ereigniserkennungssystem die Grundlage der hier vorgestellten Arbeit ist. Die lokale und verteilte Mustererkennung wurde in [Dzi07] unter Laborbedingungen entwickelt und getestet. Der Aufbau der Laborversuche war so konzipiert, dass verschiedene geometrische Figuren wie ein Kreis oder Dreieck nachgezeichnet werden sollten. Die Formen der Figuren waren dabei als Vorlage vorgegeben, so dass eine präzise Nachzeichnung möglich war. Die verteilte Mustererkennung wurde mit einer festgelegten Anzahl von drei Sensorknoten durchgeführt, wobei dieselbe Figur von den Knoten gleichzeitig nachgezeichnet wurde.

Die angenommenen Laborbedingungen sind mit Restriktionen und Ausnahmen verbunden, die eine generelle Einsatzfähigkeit des verteilten Mustererkennungssystems beschränken. Eine Abweichung von den Bedingungen in der Praxis bedeutet, dass die für die korrekte Funktionsweise des Systems einzuhaltenden Annahmen verletzt werden. Um eine praktische Anwendbarkeit des Systems zu ermöglichen, welches das Ziel dieser Arbeit darstellt, muss das bestehende Mustererkennungssystem erweitert und verändert werden. Auf die dazu nötigen Maßnahmen wird im späteren Verlauf des Kapitels eingegangen. Vorher werden die Grenzen des Systems aufgezeigt und die daraus resultierenden Anforderungen beschrieben.

### 4.1 Grenzen des bestehenden Systems

Aufgrund der Entwicklung und Anwendung unter Laborbedingungen ist das bestehende System verschiedenen Beschränkungen unterworfen. Diese sollen im Folgenden dargestellt werden.

#### Ortsabhängigkeit

Die Betrachtung von verteilten Ereignissen in [Dzi07] ist so konzipiert, dass ein- und dasselbe Ereignis gleichzeitig bei verschiedenen Sensorknoten auftritt. Die dabei entstehenden Daten sind also mit Ausnahme des Ereignistyps unabhängig voneinander. Demzufolge ist es nicht entscheidend, welcher der Sensorknoten die Auswertung vornimmt, da sich ein Ereignis im Netzwerk nicht aufgrund des Ortes seines Auftretens unterschiedlich auswirkt.

### **Feste Anzahl an Knoten**

Ein Hindernis für die Erkennung von Ereignissen ist die beschränkte, fest vorgegebene Anzahl an Knoten. In der Praxis lässt sich keine Menge von Knoten definieren, welche immer alle Ereignisse wahrnimmt. Für die Erkennung wird vorausgesetzt, dass die Daten aller registrierten Sensorknoten vorliegen müssen, damit eine Klassifikation durchgeführt wird.

### **Begrenzte Anzahl an Knoten**

Für die verteilte Erkennung mittels der Merkmalsfusion werden die Merkmals- und Referenzvektoren der definierten Knoten gespeichert. Aufgrund der zur Verfügung stehenden geringen Speicherkapazität kann dabei nur eine geringe Anzahl an Knoten für diese Methode der verteilten Erkennung verwendet werden.

## **4.2 Anforderungen**

Die Grenzen des bestehenden Systems gehen mit Anforderungen einher, welche vom weiterentwickelten System erfüllt werden müssen, damit dieses in der Praxis eingesetzt werden kann.

### **Ortsunabhängigkeit**

In der Praxis ist es wahrscheinlich, dass ein Ereignis an einem bestimmten Ort auftritt und von diesem aus von vielen Sensorknoten unterschiedlich wahrgenommen wird. Ein Beispiel hierfür ist der für diese Arbeit verwendete Anwendungsfall. Ein Ereignis kann an einem beliebigen Bauzaunelement ausgelöst werden und erstreckt sich über weitere Zaunelemente. Die Knoten an den Zaunelementen nehmen das Ereignis jedoch alle unterschiedlich wahr. Weiter entfernte Knoten werden zum Beispiel nur eine deutlich abgeschwächte Bewegung aufnehmen. Beim Auftreten eines Ereignisses ist der genaue Ort nicht bekannt, dies ist aber für die Erkennung entscheidend. Das bestehende Erkennungssystem darf nicht auf einen Erkennungsort beschränkt sein und muss so angepasst werden, dass es Ereignisse ortsunabhängig erkennen und zudem auch den zugehörigen Ereignisort eindeutig bestimmen kann.

### **Dynamische Anzahl an Knoten**

Die Menge an Sensorknoten, deren Werte während des Trainings von Ereignissen verwendet werden, darf nicht festgelegt sein, sondern soll sich für verschiedene Ereignisse ändern können. Zur verteilten Erkennung eines Ereignisses darf nicht gefordert sein, dass von bestimmten Knoten Daten erhalten werden müssen. Dies ist im Labor realistisch, in der Praxis kann aufgrund von Störeinflüssen nicht davon ausgegangen werden, dass diese Knoten immer Daten liefern. Die einzelnen Sensorknoten haben während des Trainings nur eine lokale Sicht auf ihre Daten, wodurch die Schwierigkeit einer Differenzierung zwischen den verschiedenen Ereignissen besteht. Da nicht davon ausgegangen werden kann, dass Ereignisse immer von allen beteiligten Knoten gleichmäßig erkannt werden, kann ein einzelner Knoten nicht für sich entscheiden, wann das Training einer Klasse beendet ist. Es muss eine Möglichkeit geschaffen werden, das Training zu koordinieren.

## **Unbegrenzte Anzahl an Knoten**

Für die verteilte Ereigniserkennung mittels der Merkmalsfusion sollen auch die Daten theoretisch unbegrenzt vieler Sensorknoten verwendet werden können. Da der Speicherplatz auf den Sensorknoten begrenzt ist, muss eine Entscheidung getroffen werden, welche Referenzmerkmale der Knoten für die Erkennung verwendet werden. Aufgrund der hohen Anzahl zugrunde liegender Merkmale und der damit verbundenen schwierigen manuellen Auswahl, muss eine automatische Auswahl getroffen werden.

## **4.3 Konzeptionelle Lösungsansätze**

Es wurde eine Reihe von Anforderungen beschrieben, welche Anpassungen des bestehenden Systems notwendig machen. Theoretische Grundlagen und Lösungsmöglichkeiten für konzeptionelle Methoden und Vorgehensweisen, welche dafür eingesetzt werden, sollen hier beschrieben werden.

### **4.3.1 Ortsunabhängigkeit**

Die verteilte Erkennung eines Ereignisses soll an einem beliebigen Ort des Netzwerks stattfinden können, ohne dass der Ort des Ereignisses bekannt ist. Dazu müssen Änderungen am Training und der Erkennung vorgenommen werden.

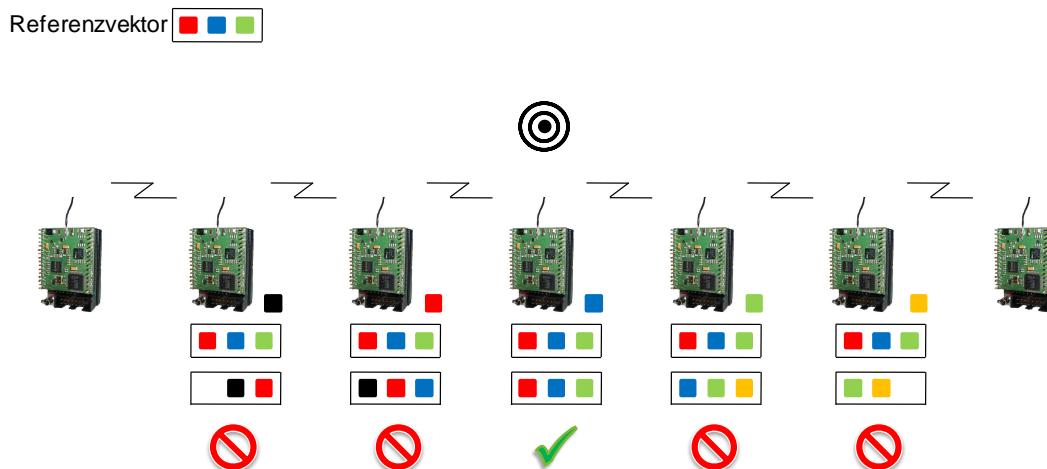
#### **4.3.1.1 Anpassung des Training und der Erkennung**

Während des Trainings wird jedes Ereignis an nur einem festen Ort ausgeführt. Der Sensorknoten, welcher sich am nächsten zu diesem Ort befindet, wird als zentraler Knoten definiert. Dieser Knoten liegt somit am nächsten zum Entstehungsort der Ereignisse. Alle anderen Sensorknoten, welche das Ereignis wahrnehmen, sollen nicht-zentrale Sensorknoten genannt werden. Die Daten der Sensorknoten, welche die Ereignisse wahrnehmen, werden auf der Merkmalsebene fusioniert, da ein Ereignis durch die Daten mehrerer Knoten definiert wird und die entstehenden Merkmalsvektoren aller Knoten somit genau dieses Ereignis beschreiben. Die Anordnung der Daten während der Fusion muss dabei eindeutig und wiederholbar sein. Jeder Sensorknoten extrahiert während der Musterbearbeitung die gleichen Merkmale aus den eigens aufgenommenen Rohdaten. Diese haben je nach Position des Knotens in Abhängigkeit vom Ort des Ereignisses unterschiedliche Ausprägungen. Die verwendeten Merkmale des Systems müssen daher für die einzelnen Knoten getrennt betrachtet werden und sollen im weiteren Verlauf der Arbeit für jeden Knoten als eigenständige Merkmale angesehen werden.

Da die Fusion der Daten auf der Merkmalsebene stattfindet, werden für die verteilte Ereigniserkennung eines unbekannten Musters, welche aus Effizienzgründen lokal auf den Sensorknoten durchgeführt wird, die Merkmalsvektoren der am Ereignis beteiligten Knoten verwendet. Die Schwierigkeit besteht darin, die Daten der verschiedenen Sensorknoten lokal zuzuordnen, da der Ort des Ereignisses nicht bekannt ist. Die Anordnung muss der des Trainings gleichen. Dazu muss ein Bezugspunkt angenommen werden, anhand dessen die Daten angeordnet werden. Dafür wird der zentrale Knoten gewählt. Jeder das Ereignis wahrnehmende Knoten nimmt an, dass er den zentralen Knoten dar-

stellt und sammelt Daten der umliegenden Knoten, welche in einer vorgegebenen Reihenfolge relativ zum Knoten angeordnet werden. Anhand der fusionierten Daten wird von den Sensorknoten eine Entscheidungsfindung mittels des bestehenden Minimum-Distanz-Klassifikators vorgenommen. Die Referenzvektoren sind dabei auf allen Sensorknoten gleich. Es sollte nur eine relevante Klassifikation geben, und zwar von dem Knoten, welcher tatsächlich den zentralen Knoten darstellt und sich somit am nächsten zum Entstehungsort des Ereignisses befindet. Somit entspricht nur dessen Anordnung der Daten der des Trainings. Alle anderen Knoten sollten kein korrektes Ereignis erkennen. Dementsprechend ist auch nur diese eine Klassifikation verwendbar. Die Klassifikationen der übrigen Knoten sollen verworfen werden. Das wird mit Hilfe einer Zurückweisungstechnik umgesetzt, welche nicht zuzuordnende Muster erkennt und die zugehörige Klassifikation für ungültig erklärt. Mit falschen Klassifizierungen des zentralen Knotens sollte ebenso verfahren werden.

In Abbildung 4.1 ist der Ablauf einer verteilten Erkennung dargestellt. Die Merkmalsvektoren sind vereinfacht als Quadrate dargestellt. Für die Anordnung der Sensorknoten soll angenommen werden, dass sie sich in einer Reihe befinden. Der als blaues Quadrat dargestellte Merkmalsvektor stellt den Merkmalsvektor des zentralen Knotens während des vorher durchgeföhrten Trainings und der Erkennung dar. Der durch das Training gebildete Referenzvektor entspricht den fusionierten Daten von drei Sensorknoten: linker Nachbar vom zentralen Knoten (rotes Quadrat), zentraler Knoten (blaues Quadrat), rechter Nachbar vom zentralen Knoten (grünes Quadrat). Jeder Sensorknoten nimmt an, dass er der zentrale Knoten ist und ordnet seinen Merkmalsvektor zwischen die Merkmalsvektoren der linken und rechten Nachbarknoten an. Die äußersten Sensorknoten nehmen kein Ereignis wahr und liefern somit keine Daten. Nur beim zentralen Knoten ergibt sich eine Übereinstimmung der Anordnung mit der des Trainings, wodurch das Ereignis erkannt werden sollte.



**Abbildung 4.1: Anpassung der verteilten Erkennung**

Zur Anzeige der Klassifikation und des Ortes des zu erkennenden Ereignisses wird ein zusätzliches System verwendet. Falls die Zurückweisung nicht korrekt arbeitet und auf

diesem System mehrere Klassifikationen präsent sind, werden diese mit Hilfe einer gewichteten Klassifikationsfusion zu einem Gesamtergebnis zusammengefasst.

#### **4.3.1.2 Struktur der Ereigniserkennung**

Wie im vorherigen Abschnitt geschildert, müssen die Daten mehrerer Sensorknoten während des Trainings und der Erkennung anhand einer eindeutigen Struktur angeordnet werden. Betrachtet man ein Sensornetz als eine Reihe nebeneinander angeordneter Knoten, ergibt sich für jeden Knoten genau ein linker und rechter Nachbarknoten. Zur Erstellung der Struktur soll es genügen, die jeweils am dichtesten beieinander liegenden Knoten als Nachbarknoten anzusehen. Spezialfälle wie sich schneidende Linien sollen hier nicht betrachtet werden. Diese Struktur trifft auch auf den Bauzaun-Anwendungsfall zu. Die einzelnen Bauzaunelemente sind in einer Reihe miteinander verbunden und haben jeweils ein benachbartes Zaunelement.

Zur Darstellung der Struktur wird noch eine Identifikation der Netzwerknoten benötigt. Es gibt für jeden Sensorknoten eine eindeutige Kennung, die auch bei Energieausfall oder Beschreiben des Speichers erhalten bleibt, da diese im nichtflüchtigen Speicher steht. Dabei handelt es sich um die sogenannte Konfigurations-ID. Das Netzwerk wird so aufgebaut, dass der linke Nachbarknoten eines jeden Knotens eine um 1 kleinere Konfigurations-ID hat, der rechte Nachbarknoten eine um 1 größere ID. Im Verlauf der Reihe steigt die Kennung der Sensorknoten also immer um 1. Wichtig ist, dass es keine Lücken bzw. fehlende Nummern gibt, da sonst auf dieser Struktur aufbauende Algorithmen fehlerhaft arbeiten.

Mit Hilfe dieser Kennungen können Daten von mehreren Sensorknoten an einem beliebigen Punkt des Netzwerks angeordnet werden. Sind mehrere Knoten gegeben, werden deren Kennungen aufsteigend sortiert. Die aufsteigende Ordnung der Knotenkennungen stellt somit die eindeutige Struktur dar. Die Daten werden entsprechend der entstehenden Reihenfolge der Knoten angeordnet. Dies ist bei der Trainings- und Erkennungsphase notwendig und soll später an diesen Punkten genauer beschrieben werden.

#### **4.3.1.3 Zurückweisungsmethoden**

In vielen Anwendungen ist es sinnvoll, eine Klassifikation mit einer Zurückweisungsmöglichkeit zu verbinden. Ist die Klassifikation nicht korrekt, kann das falsche Ergebnis zu hohen Kosten oder sogar gefährlichen Situationen führen [Hei04]. Dies trifft auch auf den Anwendungsfall dieser Arbeit zu. Falsche Ergebnisse zu liefern, bedeutet mehr Funkverkehr und Energieverbrauch der Sensorknoten. Zudem muss dann an anderer Stelle mit diesen Ergebnissen umgegangen werden. Ein falsches Ergebnis bedeutet auch gleichsam einen Falschalarm, welcher weitere unnötige Aktionen nach sich ziehen kann.

In Abschnitt 4.3.1.1 wurde beschrieben, dass die Erkennung eines Ereignisses von mehreren Sensorknoten ausgeführt wird. Die Klassifikation ist dann falsch und sollte nicht verwendet bzw. ausgegeben werden, wenn sie von einem nicht-zentralen Knoten errechnet wurde. Diese können niemals ein Ereignis korrekt auswerten, da die Anordnung der Daten für die Klassifikation nicht der des Trainings entspricht und dementsprechend kein trainiertes Ereignis vorliegen kann. Selbst wenn die Klassifikation mit dem Ereignistyp übereinstimmt, sollte diese als nicht korrekt eingestuft werden. Beim zentralen Knoten

muss mehr differenziert werden. Ereignisse, welche korrekt erkannt werden, entsprechen den trainierten Daten und sollten somit verwendet werden. Falsche Klassifikationen sollten verworfen werden, da die vom Muster bereitgestellte Klasseninformation nicht den im Training aufgenommenen Informationen entspricht. Dies gilt auch für die Klassifikation von nicht trainierten Ereignissen, welche niemals korrekt sein kann.

Die Anforderungen an die Zurückweisung lassen sich wie folgt zusammenfassen:

**Tabelle 4.1: Anforderungen an die Zurückweisung**

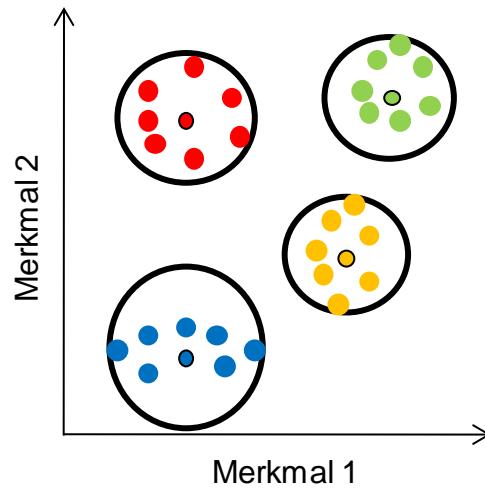
	Zentraler Knoten	Nicht-zentraler Knoten
Richtig erkanntes Ereignis	Nicht zurückweisen	Zurückweisen
Falsch erkanntes Ereignis	Zurückweisen	Zurückweisen

Es gibt eine Vielzahl von Arbeiten, in denen Zurückweisungsmöglichkeiten vorgestellt werden. Diese sind aber nicht uneingeschränkt verwendbar, da die Art der Zurückweisung auch mit den Anforderungen der Anwendung übereinstimmen muss [Mou06].

Eine Möglichkeit der Zurückweisung wird in [Bak96] beschrieben. Der dort verwendete Algorithmus arbeitet mit einer Repräsentation der Klassen durch die Karhunen-Loéve-Erweiterung [Fuk90]. Vorberechnungen wären auf dem hier verwendeten externen System noch möglich, die Zurückweisung auf den Sensorknoten ist aufgrund der hohen Rechen- und Speicheranforderung aber nicht durchführbar. Da die Kapazitäten der Sensor-knoten begrenzt sind, muss eine Möglichkeit gefunden werden, die Zurückweisung ohne großen zusätzlichen Rechenaufwand und mit möglichst wenig Speicherbedarf auszuführen.

### Distanzzurückweisung

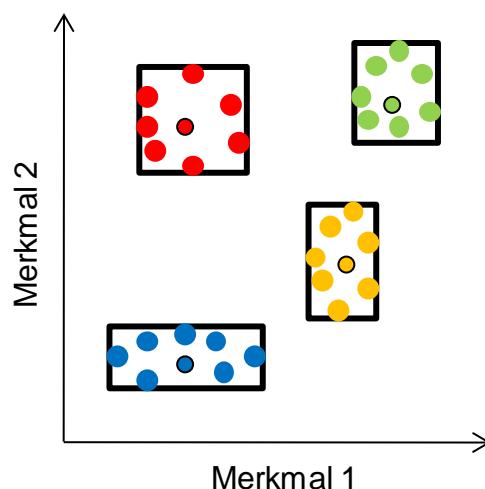
Eine Möglichkeit, besteht darin, den bestehenden Klassifikationsalgorithmus für die Zurückweisung zu benutzen, wie in [Dzi07] angedacht wird. Ein Muster wird der Klasse mit dem geringsten Abstand des Merkmalsvektors dieses Musters zu allen Referenzvektoren der Klassen zugeordnet. Ist dieser Abstand allerdings zu groß, dann ist die Zuordnung nicht eindeutig und sollte verworfen werden. Die Entscheidung, wann eine Zurückweisung erfolgen soll, lässt sich mit Hilfe eines festgelegten Zurückweisungsradius festlegen. Dieser sollte klassenspezifisch festgelegt werden, da die Verteilung der Daten innerhalb der Klassen unterschiedlich ist. In Bezug auf die Trainingsdaten ist es sinnvoll, für jede Klasse  $k$  den maximal vom Referenzvektor entfernten Trainingsvektor  $p_{k,max}$  in diesen Radius mit einzubeziehen, wie in Abbildung 4.2 dargestellt ist. Dieser gehört zu den trainierten Daten und sollte somit nicht zurückgewiesen werden. Alle anderen Merkmalsvektoren, die weiter entfernt als  $p_{k,max}$  liegen, beinhalten Informationen, die nicht mehr als charakteristisch für die Klasse angesehen werden können. Diese Form der Zurückweisung soll Distanzzurückweisung genannt werden.



**Abbildung 4.2: Klassen mit Zurückweisungsradius bei der Distanzzurückweisung**

### Merkmalszurückweisung

Eine andere Möglich bestehet darin, die minimalen und maximalen Werte, die für jedes verwendete Merkmal während des Trainings aufgetreten sind, als Zurückweisungsgrenzen zu benutzen. Dies geschieht wie bei der Distanzzurückweisung klassenspezifisch (siehe Abbildung 4.3). Soll ein neues Muster erkannt werden, wird nach der Klassifikation für alle Merkmale überprüft, ob diese innerhalb der Grenzen liegen. Dies ist mit einem leicht höheren Aufwand verbunden, da zwei Vergleichsoperationen für jedes Merkmal durchgeführt werden müssen. Die Distanzzurückweisung braucht insgesamt nur einen Vergleich. Zudem müssen die minimalen und maximalen Werte für jede Klasse auf den Sensorknoten gespeichert werden. Die Begrenzung der Merkmalswerte sorgt für eine genauere Eingrenzung der trainierten Daten. Aufgrund dessen muss das Training sorgfältig und intensiv durchgeführt werden, denn wenn nur ein Merkmal leicht differiert, wird das Muster schon zurückgewiesen. Diese Form der Zurückweisung soll Merkmalszurückweisung genannt werden.



**Abbildung 4.3: Klassen mit Merkmalswertbegrenzung bei der Merkmalszurückweisung**

### 4.3.2 Merkmalsauswahl

Die Erkennung eines Musters mittels des in [Dzi07] verwendeten Minimum-Distanz-Klassifikators benötigt das Vorhandensein von Referenzvektoren, welche mit Merkmalen des Musters verglichen werden. Bei dem Ansatz der erweiterten verteilten Mustererkennung werden auf jedem Knoten die extrahierten Merkmale aller anderen Knoten angeordnet und mit dem Referenzvektor verglichen. Dieser wird dementsprechend aus den Trainingsdaten aller an den Ereignissen teilnehmenden Knoten gewonnen. In der bisherigen verteilten Mustererkennung unter Nutzung der Methode der Merkmalsfusion war dies auf ein Maximum von drei Knoten beschränkt und somit im Rahmen der verfügbaren Speicher- und Rechenkapazität.

In der Praxis ist es möglich, dass mehr als drei Knoten an Ereignissen beteiligt sind. Der in dieser Arbeit verwendete Klassifikator benötigt für eine Klassifizierung  $K * N$  Operationen, wobei  $K$  die Anzahl der Klassen und  $N$  die Dimension des Merkmalsvektors ist. Der Speicherbedarf ist  $K * N$  für alle Referenzvektoren und  $N$  für den Merkmalsvektor des zu erkennenden Musters. Eine deutliche Vergrößerung von  $N$ , wie sie aufgrund der höheren Anzahl von Knoten zu erwarten ist, würde die Kapazitätsgrenzen überschreiten.

Neben dem Rechen- und Speicheraufwand gibt es einen weiteren Grund, warum die Dimension eines Merkmalsvektors nicht beliebig groß werden sollte. Dieser besagt, dass eine Erhöhung der Dimension eine Verringerung der Klassifikationsgüte verursachen kann [Dev82]. Eine minimale Fehlerrate  $E_{min}$  würde dann auftreten, wenn die Dichten aller Klassen vollständig bekannt wären. Diese Fehlerrate ist eine mit der Anzahl der Dimensionen monoton fallende Funktion. Ein hinzugefügtes Element kann Informationen nicht zerstören, der Informationsgewinn nimmt zu. In der Praxis sind die Dichten der Klassen nur selten vollständig bekannt. Anstelle dieses Wissens muss eine endliche Trainingsmenge verwendet werden [Hei04]. Aufgrund der durch die hohe Dimension verursachten großen Anzahl möglicher Kombinationen von Merkmalswerten können dadurch nicht alle Möglichkeiten abgedeckt werden. Es muss sich also auf eine repräsentative Trainingsmenge  $N_S$  beschränkt werden. Dadurch kommt es zu einer sogenannten Überanpassung. Der Klassifikator orientiert sich an den Daten, die nicht für die gesamte Datenmenge charakteristisch sind und passt sich zu sehr an das Rauschen in den Trainingsdaten an. Bei der Erkennung von neuen Daten werden dann eventuell schlechtere Ergebnisse geliefert, als mit einer größeren Menge spezifischer Daten erreicht worden wäre [Fer03].

Abbildung 4.4 zeigt eine Veränderung von Fehlerraten bezüglich der Anzahl der Dimensionen mit verschiedenen großen Trainingsmengen.

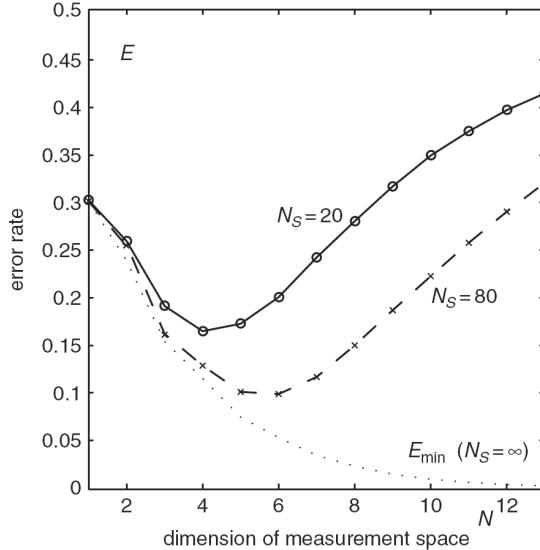


Abbildung 4.4: Fehlerraten verglichen mit der Dimension des Datenraums [Hei04]

Zur Lösung der geschilderten Probleme soll eine Merkmalsauswahl durchgeführt werden. Dabei wird eine Menge mit  $n$  gegebenen Merkmalen auf eine Untermenge mit  $n'$  Merkmalen reduziert. Die Untermenge sollte die geeigneten Merkmale enthalten, die eine möglichst hohe Differenzierbarkeit der verschiedenen Klassen ermöglichen. Zudem sollte die Anzahl mindestens so stark reduziert werden, dass eine rechen- und spechertechnisch lösbare Komplexität gegeben ist.

Zu beachten ist, dass die Merkmalsauswahl nicht für gegebene  $n$  Merkmale des Systems durchgeführt, sondern für  $n * m$  Merkmale bei  $m$  Knoten durchgeführt wird, da die Merkmale für jeden Knoten spezifisch aufgefasst werden.

Aufgrund der hohen Dimension ist keine manuelle Merkmalsauswahl, wie sie bisher vorgenommen wurde, möglich. Sie soll jetzt automatisch mit Hilfe ausgewählter, bestehender Verfahren durchgeführt werden. Wie in Abbildung 4.5 dargestellt ist, besteht die Merkmalsauswahl im Allgemeinen aus einem Mengensuchverfahren, welches Untermengen zur Findung einer bestmöglichen Menge auswählt. Zur Beurteilung dieser Untermengen wird weiterhin ein Maß zur Gütebewertung von Merkmalsmengen benötigt.

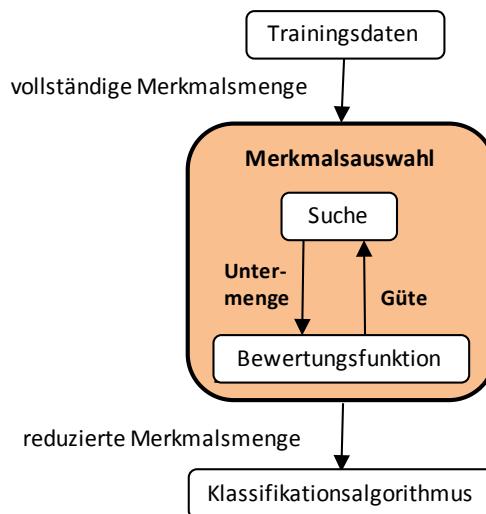


Abbildung 4.5: Ablauf der Merkmalsauswahl nach [Gut]

#### 4.3.2.1 Gütekriterien für Merkmale

Die Güte einer Merkmalsmenge ist durch die Fähigkeit dieser Menge, zwischen mehreren Klassen genau unterscheiden zu können, ausgezeichnet [Web02]. Das Gütekriterium hängt folglich mit dem Abstand der Merkmalsvektoren der Klassen zusammen. Hierbei kann allerdings nicht nur der Mittelwert betrachtet werden, da eine große Streuung der Daten innerhalb einer Klasse zu schlechten Ergebnissen führen kann, wie in Abbildung 4.6 dargestellt ist. Es sollten also Abstände der Verteilungsdichten betrachtet werden. Praktisch alle Gütekriterien basieren daher auf verallgemeinerten Abstandsmaßen [Nie83].

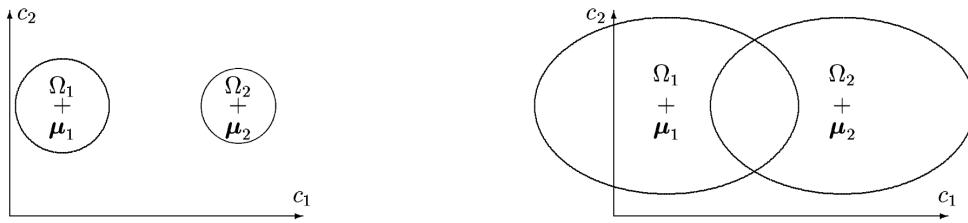


Abbildung 4.6: Verteilungsdichten von Klassen [Nie83]

Die Güte einer Merkmalsmenge kann grundlegend auf zwei Arten gemessen werden. Sogenannte Wrapper messen die Genauigkeit anhand des zu verwendenden Klassifikationsalgorithmus. Es wird untersucht, wie sich eine Merkmalsmenge auf den Klassifikator anhand einer separaten Trainings- und Testmenge auswirkt. Bei den sogenannten Filtern werden generelle Charakteristika der Daten genutzt, um eine Merkmalsmenge zu bewerten. Filter arbeiten unabhängig vom Klassifikationsalgorithmus. Wrapper liefern im Allgemeinen bessere Ergebnisse als Filter, da sie sich auf den im System verwendeten Klassifikationsalgorithmus beziehen. Für rechenintensive Algorithmen verursacht dies allerdings auch einen möglicherweise nicht realisierbaren Rechenaufwand.

#### Wrapper

Das Erreichen einer minimalen erwarteten Fehlerrate ist eines der Hauptziele bei dem Entwurf eines Erkennungssystems. Zur Untersuchung der Unterscheidbarkeit zwischen Klassen wird daher die Fehlerrate als eine der gebräuchlichsten Metriken herangezogen. Diese stellt das Verhältnis von falsch klassifizierten Mustern zu allen Mustern dar. Zur formalen Angabe einer Fehlerrate sollen einige Notationen nach [Web02] eingeführt werden. Seien die Trainingsdaten gegeben als  $Y = \{y_i, i = 1, \dots, n\}$ . Jedes Muster  $y_i$  besteht aus zwei Teilen,  $y_i = \{x_i^T, z_i^T\}$ .  $x$  stellt den Datenvektor dar,  $z$  entspricht der zugehörigen Klasse.  $w(z)$  gibt die Klasse zu  $x_i$  an. Mit Hilfe der gegebenen Entscheidungsregel des Klassifikators  $\eta(x; Y)$ , welche die Klasse liefert, zu der ein Muster  $x$  anhand  $Y$  zugewiesen wird, lässt sich eine Verlustfunktion  $Q(w(z), \eta(x; Y))$  angeben:

$$Q(w(z), \eta(x; Y)) = \begin{cases} 0 & \text{wenn } w(z) = \eta(x; Y) \\ 1 & \text{sonst} \end{cases} \quad (3)$$

Um die wahre Fehlerrate anzugeben, müsste ein unendlich großer Testdatensatz mit gleicher Verteilung wie die der Trainingsdaten zur Verfügung stehen. Diese Fehlerrate muss also geschätzt werden. Eine Möglichkeit dafür bietet die Angabe der scheinbaren Fehler-

rate. Als Testdaten werden hierbei die Trainingsdaten verwendet. Die scheinbare Fehlerrate lässt sich als das Mittel über die Anwendung der Verlustfunktion auf alle Trainingsdaten darstellen:

$$e_A = \frac{1}{n} \sum_{i=1}^n Q(w(z_i), \eta(x_i; Y)) \quad (4)$$

Da die für Ermittlung des Fehlers verwendeten Trainingsdaten schon vorher für den Entwurf des Klassifikators verwendet wurden, ist diese Fehlerrate zu optimistisch. Es entsteht ein Schätzfehler. Mit Hilfe des Verfahrens der Kreuzvalidierung kann ein annähernd verzerrungsfreier Schätzwert, also mit nur einer geringen Abweichung zur wahren Fehlerrate, ermittelt werden [Web02].

Die Grundidee der Kreuzvalidierung besteht darin, dass die vorhandenen Daten in eine Trainings- und eine Testmenge aufgespalten werden. Der Klassifikator wird anhand der Trainingsmenge erstellt. Die Fehlerrate wird anhand der unabhängigen Testmenge berechnet, welche nicht für das Training verwendet wurde. Es existieren verschiedene Methoden der Kreuzvalidierung, welche in [Web02] umfassend beschrieben werden. Bei der *Holdout*-Methode, welche die einfachste Möglichkeit darstellt, werden die Daten in zwei Teilmengen aufgespalten. Dies ist sehr ineffizient und die Fehlerrate hängt stark von der Wahl der beiden Mengen ab. Ein berechnungsaufwändigeres Verfahren ist, die Trainingsdaten in  $k$  Teilmengen aufzuspalten und in  $k$  Durchläufen jeweils mit einer dieser Teilmengen zu testen und die restlichen  $k - 1$  Teilmengen als Trainingsmenge zu benutzen. Mit einem noch größeren Berechnungsaufwand ist die sogenannte *leave-one-out* Kreuzvalidierung verbunden. Hierbei werden jeweils  $n - 1$  Muster als Eingabe des Klassifikators und das verbleibende Muster als Testdatum verwendet wird. Es werden demzufolge  $n$  Klassifikationen ausgeführt (siehe Abbildung 4.7).

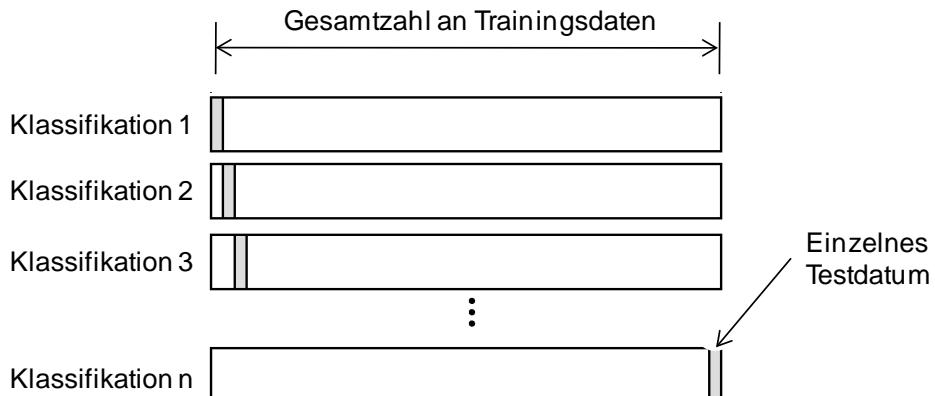


Abbildung 4.7: leave-one-out Kreuzvalidierung nach [Gut2]

Sei  $Y_j$  die Trainingsmenge ohne das Muster  $x_j$ , dann definiert sich der Kreuzvalidierungsfehler  $e_{cv}$  als durchschnittlicher Fehler aller  $n$  Klassifikationen wie folgt:

$$e_{cv} = \frac{1}{n} \sum_{j=1}^n Q(w(z_j), \eta(x_j; Y_j)) \quad (5)$$

Zur Ausführung des Verfahrens müssen  $n$  Klassifikatoren erstellt werden, was für große  $n$  zu einem hohen Rechenaufwand führt. Trotz der hohen Varianz der Entscheidungsregeln arbeitet diese Methode annähernd verzerrungsfrei. Aufgrund dieser Eigenschaft soll es auch als ein Gütekriterium in dieser Arbeit verwendet werden. Im Folgenden wird es nur noch abgekürzt als Kreuzvalidierung bezeichnet.

## Filter

Es gibt verschiedene Ansätze, Charakteristika von Daten als Maßzahl ihrer Güte anzugeben. Dazu gehören Korrelations- und informationstheoretische Messungen, wie etwa Klassenkonsistenz-Methoden [Das00] und korrelations-basierte Methoden [Yu03] [Hal97], auf die hier nicht näher eingegangen werden soll. Andere gebräuchliche Ansätze untersuchen die Trennbarkeit von Klassen. Dazu gehören probabilistische Distanzmethoden, welche Abstände von Verteilungsdichten berechnen. Beispiele für solche Methoden sind die Divergenz, der Chernoff-Abstand und der Bhattacharyya-Abstand. Eine komplettere Liste findet sich in [Dev82], [Che76]. Der Nachteil dieser Methoden ist, dass die zugehörigen Formeln in Integralform gegeben sind und für deren Berechnung eine Abschätzung der Fehlerwahrscheinlichkeit und eine geschlossene Form des Integrals gegeben sein müssen. Dies schränkt die praktische Nutzbarkeit ein, so dass diese Methoden hier vernachlässigt werden sollen.

Ein einfacheres Kriterium für Trennbarkeit bezieht sich auf die Verteilung der Merkmalsvektoren im Raum. Ausführlich wird dieses Verfahren in [Dev82] und [Hei04] diskutiert. Zur Beschreibung der Verteilung sollen die folgenden Notationen nach [Hei04] eingeführt werden. Gegeben sei eine Trainingsmenge  $T_S$  mit  $N_S$  Mustern und  $K$  Klassen. Eine Klasse  $w_k$  beinhaltet  $N_k$  Muster, welche mit  $z_{k,i}$  bezeichnet werden,  $i = 1, \dots, N_k$ . Muster in  $T_S$  werden ohne Klassenzugehörigkeitsangabe mit  $z_n$  bezeichnet. Der Mittelpunkt einer Klasse ist gegeben durch:

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N_k} z_{k,n} \quad (6)$$

Der Mittelpunkt der gesamten Trainingsdaten ist definiert als:

$$\mu = \frac{1}{N_S} \sum_{n=1}^{N_S} z_n \quad (7)$$

Die mittlere Streuung der Merkmalsvektoren innerhalb aller Klassen lässt sich in der so genannten Inner-Klassen-Matrix darstellen. Diese stellt das Rauschen der Daten dar.

$$S_w = \frac{1}{N_S} \sum_{k=1}^K \sum_{n=1}^{N_k} (z_{k,n} - \mu_k)(z_{k,n} - \mu_k)^T \quad (8)$$

Im Folgenden wird die Funktion  $trace(M)$  verwendet, welche die Summe der Diagonalelemente einer Matrix  $M$  darstellt.  $Trace(S_w)$  ist folglich ein Maß für die Varianz der Merkmale, gemittelt über alle Klassen.

Im Gegensatz dazu beschreibt die Zwischen-Klassen-Matrix die Streuung der Klassenmittelpunkte um den Mittelpunkt der gesamten Daten:

$$S_b = \frac{1}{N_S} \sum_{k=1}^K N_k (\mu_k - \mu)(\mu_k - \mu)^T \quad (9)$$

$trace(S_b)$  ist ein Maß für die durchschnittliche Distanz der Klassenmittelpunkte zum globalen Mittelpunkt.

Zur Veranschaulichung der Streuung sollen die Matrizen hier durch Ellipsen dargestellt werden. Jede Streuungsmatrix  $S$  entspricht der Ellipse  $zS^{-1}z^T = 1$ , die als grobe Kontur der zugehörigen Datenpunkte zu sehen ist und ein Gefühl für die Verteilung der Daten geben soll. In Abbildung 4.8 sind Daten von vier Klassen mit den zugehörigen Streuungsmatrizen und der Zwischen-Klassen-Matrix dargestellt.

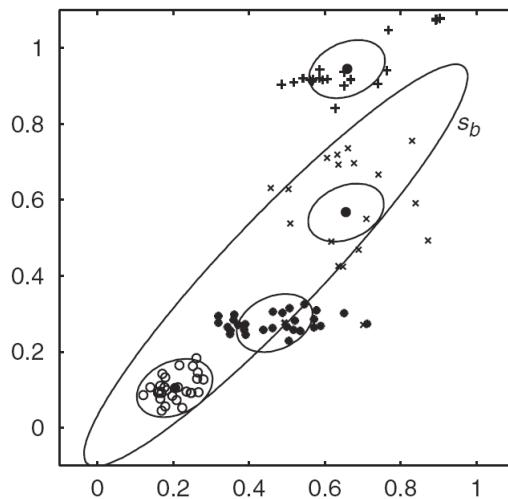


Abbildung 4.8: Streuung der Daten, dargestellt durch Ellipsen [Hei04]

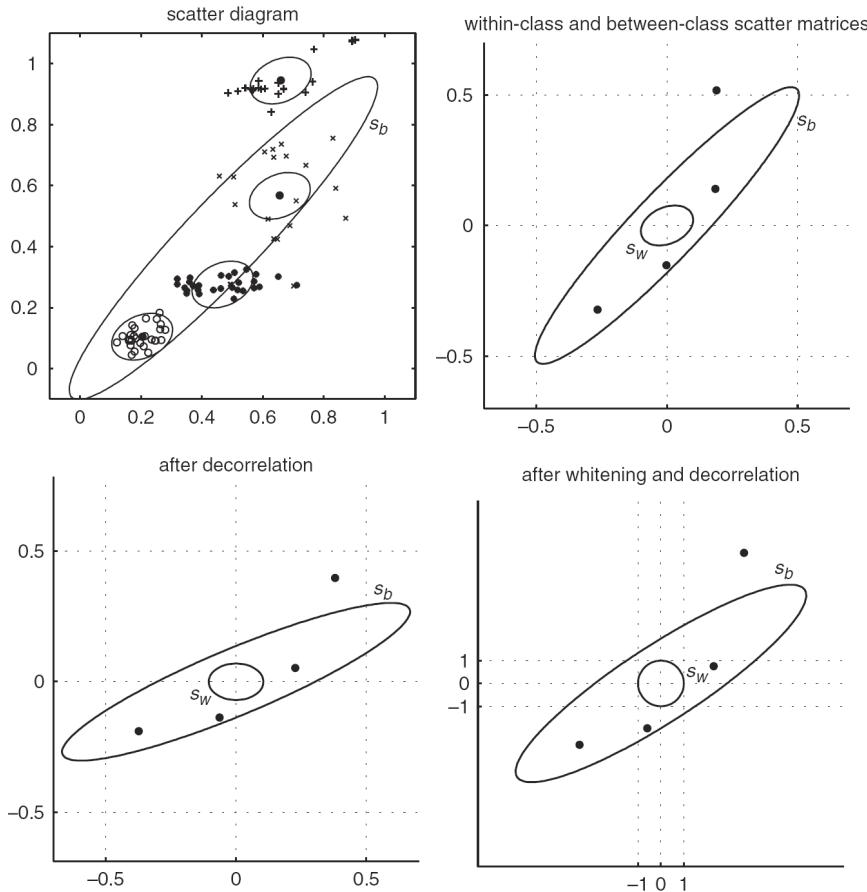
Ein Gütekriterium, welches die Trennbarkeit der Klassen ausdrückt, ist das Verhältnis aus der Distanz innerhalb der Klassen und der Distanz zwischen den Klassen

$$\frac{J_{Inter}}{J_{Intra}} = \frac{trace(S_b)}{trace(S_w)} \quad (10)$$

Liegen die Klassenmittelpunkte weit voneinander entfernt und sind die Daten innerhalb der Klassen nahe um ihren Mittelpunkt verteilt, dann sind die Klassen gut separierbar und das Maß nimmt große Werte an. Der Wert ist kleiner, wenn viel Rauschen der Daten vorliegt und die Klassenmittelpunkte nah beieinander liegen.

Dieses Maß beachtet zwar die Schwankungen der Daten, ignoriert aber deren Effekt auf die Trennbarkeit, welche durch die Korrelation zwischen den Komponenten der Merkmalsvektoren verursacht wird [Dev82]. Bezogen auf die Ellipsendarstellung der Matrizen,

wäre ein hohes Rauschen der Daten, dargestellt durch eine große Ellipse der Inner-Klassen-Matrix, dann harmlos, wenn sie nur eine geringe Schnittmenge mit der Ellipse der Zwischen-Klassen-Matrix hat [Hei04]. Nach [Hei04] kann dieser Defekt des oben beschriebenen Maßes dadurch korrigiert werden, dass die Merkmalsvektoren einer Transformation  $A$  ausgesetzt werden, so dass die Inner-Klassen-Matrix die Identitätsmatrix darstellt, d.h.  $AS_wA^T = I$ .  $A$  kann durch Faktorisierung gefunden werden, wobei gilt  $S_w = V\Lambda V^T$ . Daraus folgt  $A = \Lambda^{-1/2}V^T$ . Eine Darstellung des Prozesses ist in Abbildung 4.9 gegeben.  $\Lambda$  stellt eine Diagonalmatrix der Eigenwerte von  $S_w$  dar,  $V$  ist eine unitäre Matrix mit den dazu gehörigen Eigenvektoren.  $V^T$  dekorreliert das Rauschen. Veranschaulicht durch die Ellipsendarstellung, führt es eine Rotation aus, welche die Achsen von  $S_w$  anpasst.  $\Lambda^{-1/2}$  stellt eine Skalierung der Achsen dar. Die normalisierte Inner-Klassen-Matrix kann als ein Kreis mit Einheitsradius angesehen werden.



**Abbildung 4.9: Inner-Klassen und Zwischen-Klassen Abstand [Hei04]**

Im transformierten Raum erhält man das Kriterium:

$$\begin{aligned}
 J_{Inter/Intra} &= \text{trace}(\Lambda^{-1/2}V^T S_b V \Lambda^{-1/2}) \\
 &= \text{trace}(V \Lambda^{-1} V^T S_b) \\
 &= \text{trace}(S_w^{-1} S_b)
 \end{aligned} \tag{11}$$

Dieses Maß wird häufig in der Praxis benutzt und soll auch in dieser Arbeit als Gütekriterium eingesetzt werden.

### 4.3.2.2 Mengensuchverfahren

Möchte man eine Merkmalsmenge mit  $n$  Merkmalen auf eine Untermenge mit  $n'$  Merkmalen reduzieren, könnte man alle möglichen Merkmalskombinationen mit Hilfe eines Gütemaßes bewerten, um die optimale Untermenge zu finden. Die Anzahl dieser Kombinationen ist:

$$\binom{n}{n'} = \frac{n!}{n'! (n - n')!} \quad (12)$$

Selbst bei moderaten Werten von  $n$  und  $n'$  ergibt sich dabei ein hoher Rechenaufwand. Weiterhin ist  $n'$  nicht immer vorher bekannt, so dass der Aufwand zusätzlich steigen kann. Aufgrund der hohen Dimension des Merkmalsvektors, der in dieser Arbeit vorliegt, ist diese Möglichkeit daher auszuschließen. Die Lösung des Problems besteht darin, die möglichen Untermengen so zu durchsuchen, dass der Rechenaufwand verringert wird. Im Allgemeinen gibt es zwei Strategien, wie Merkmalsmengen durchsucht werden können: suboptimal und optimal [Web02]. Zusätzlich muss ein Abbruchkriterium festgelegt werden, welches das Ende der Suche bestimmt.

### Optimale Suchverfahren

Ein Ansatz, um eine optimale Merkmalsmenge zu finden, ist die des Dynamischen Programmierens [The03]. Eine effizientere Möglichkeit bietet der in [Nar77] beschriebene *branch&bound*-Algorithmus. Dieser setzt voraus, dass sich das verwendete Gütekriterium monoton verhält, d.h. die Güte einer Merkmalsmenge ist immer kleiner als die Güte einer größeren Menge, welche die Merkmalsmenge enthält. Man kann sich den *branch&bound*-Algorithmus wie eine Tiefensuche in einem Baum vorstellen. Die Knoten repräsentieren alle Permutationen von Merkmalsmengen, wobei die Wurzel die vollständige Merkmalsmenge und die Blätter die Mengen mit der Dimension, auf die reduziert werden soll, darstellt. Der Baum wird einmal bis zum rechten Blatt durchlaufen und die Güte dieser Merkmalsmenge wird als Schwellwert verwendet. Die nachfolgende Tiefensuche muss nur Knoten mit Güte höher als dieser Schwellwert weiter verfolgen, da aufgrund des Monotoniekriteriums keine höhere Güte in diesem Zweig erreicht werden kann. Der Schwellwert wird nur in Blättern und bei dortiger höherer Güte aktualisiert. Nach Beendigung der Tiefensuche ist die Merkmalsmenge des Blattes der letzten Aktualisierung die optimale Merkmalsmenge. In Abbildung 4.10 ist dieses Vorgehen anhand der Auswahl dreier Merkmale aus fünf dargestellt.

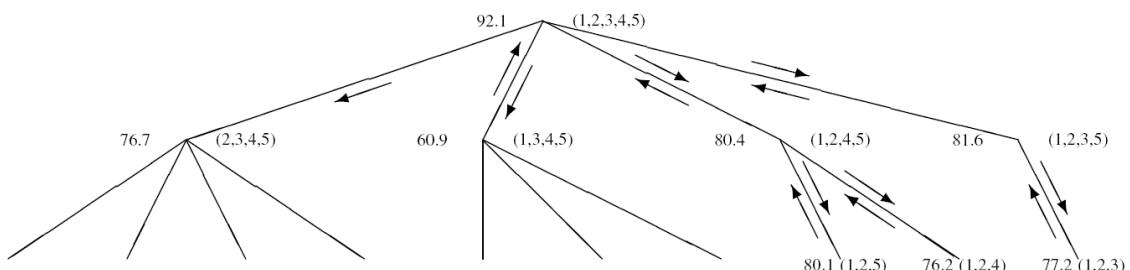


Abbildung 4.10: Baumsicht des branch&bound-Algorithmus mit Gütbewertung [Web02]

Da nicht alle Zweige des Baums und dementsprechend nicht alle Teilmengen untersucht werden müssen, ergibt sich hier ein geringerer Rechenaufwand als bei der erschöpfenden Suche. Bei einer erwarteten hohen Merkmalsdimension wie in dieser Arbeit ist dieser aber trotzdem unzumutbar hoch. Des Weiteren steht hier die zu findende Anzahl von Merkmalen nicht fest (siehe Abbruchkriterium). Ein weiterer Nachteil ist die notwendige Verwendung eines monotonen Gütekriteriums.  $J_{Inter/Intra}$  gehört dazu, Fehlerraten wie  $e_{cv}$  im Allgemeinen aber nicht. Deshalb soll diese Methode in der vorliegenden Arbeit nicht verwendet werden.

### **Suboptimale Suchverfahren**

Suboptimale Suchverfahren betrachten nicht alle möglichen Untermengen, so dass unter Umständen nicht die bestmögliche Menge gefunden wird. Es wird stattdessen nur eine Menge von potentiell brauchbaren Merkmalskombinationen betrachtet.

Die sequentielle Vorwärtsauswahl [Whi71] ist ein *greedy*-Verfahren, welches eine Merkmalsmenge sukzessive aufbaut. Sie beginnt mit einer leeren Merkmalsmenge und fügt in jedem Schritt ein neues Merkmal hinzu, welches, zusammen mit bestehenden Merkmalen evaluiert, die höchste Güte erreicht. Dies wird solange fortgeführt, bis ein gewähltes Abbruchkriterium erreicht wird. Die umgekehrte Variante davon ist die sequentielle Rückwärtsauswahl, welche mit allen Merkmalen beginnt und diese Menge dann sukzessive verkleinert. Der Nachteil dieser Methoden ist, dass ein einmal gewähltes Merkmal (Vorwärtsauswahl) bzw. verworfenes Merkmal (Rückwärtsauswahl) nicht mehr verworfen bzw. wieder hinzugenommen werden kann. Da Merkmale zusammen evaluiert werden, kann es vorkommen, dass ein gewähltes Merkmal zusammen mit später gewählten Merkmalen eine schlechtere Güte ergibt, als es ein anderes Merkmal täte.

In [Ste76] wird ein Algorithmus beschrieben, welcher ein Zurückgehen erlaubt und somit dieses Problem behebt. Dieser wird *plus-l-take-away-r* genannt. In jedem Schritt werden  $l$  Merkmale mittels der Vorwärtssuche hinzugenommen und danach die schlechtesten  $r$  Merkmale wieder entfernt. Eine Menge von Merkmalen einer Stufe des Algorithmus muss also nicht notwendigerweise Teilmenge der nächsten Stufe sein. Ein Problem des *plus-l-take-away-r* Algorithmus ist, dass gute Werte für  $l$  und  $r$  geschätzt werden müssen.

Eine Weiterentwicklung des Algorithmus wird in [Pud94] beschrieben. Die sogenannte gleitende Suche erlaubt eine höhere Flexibilität bei der Hinzunahme und Wegnahme von Merkmalen, da die Anzahl dieser Teilschritte nicht mehr fest vorgegeben ist. Die anfängliche Merkmalsmenge wird mit zwei Merkmalen durch zweifache Anwendung der Vorwärtsauswahl initialisiert. Jeder Schritt des Algorithmus besteht aus zwei Teilschritten:

- *Teilschritt 1 (Aufnahme)*

Das Merkmal wird hinzugenommen, welches zusammen mit bestehenden Merkmalen die höchste Güte erreicht. Entspricht einfacher Anwendung der Vorwärtsauswahl.

- *Teilschritt 2 (Bedingter Ausschluss)*

Das Merkmal wird gesucht, dessen Wegnahme die Güte der entstehenden Merkmalsmenge  $M$  am wenigsten verringert. Ist diese Güte höher als die Güte der bisherigen Merkmalsmenge mit Dimension von  $M$ , so wird das Merkmal entfernt und sofern die Dimension von  $M$  größer als 2 ist, mit Teilschritt 2 fortgefahrene. Ist die Güte jedoch kleiner oder gleich der bisherigen Güte, so wird das Element nicht entfernt und mit Teilschritt 1 weitergemacht.

Nach jedem Vorwärtsschritt wird also eine Menge von Rückwärtsschritten ausgeführt, solange die entstehenden Teilmengen eine bessere Güte als die vorher berechneten Mengen der jeweiligen Stufe aufweisen. Mit diesem Verfahren können nahezu optimale Merkmalsmengen gefunden werden, ohne den Rechenaufwand eines optimalen Suchverfahrens durchführen zu müssen. Daher soll es in dieser Arbeit verwendet werden.

### **Abbruchkriterium**

Es muss ein Kriterium definiert werden, an dem das Suchverfahren abgebrochen werden soll. Sinnvolle Möglichkeiten hierfür sind

- Wenn eine vorgegebene Anzahl von  $n'$  Merkmalen gefunden ist
- Wenn die beste Veränderung die Güte der Merkmalsmenge schlechter macht oder keine Verbesserung eintritt
- Wenn eine maximal erlaubte Anzahl an Merkmalen gefunden ist

Eine feste Anzahl zu findender Merkmale festzulegen, ist meist nicht möglich, da die optimale Anzahl nur mit Hilfe umfangreicher Tests bestimmt werden könnte. Daher soll diese Möglichkeit für diese Arbeit ausgeschlossen werden. In manchen Systemen ist aufgrund von Beschränkungen notwendig, eine Obergrenze für die Größe der Merkmalsmenge zu setzen. Aufgrund der begrenzten Speicher- und Rechenkapazität der in der vorliegenden Arbeit verwendeten Sensorknoten wird eine maximale zulässige Anzahl von 10 Merkmalen festgelegt. Es ist möglich, dass die Merkmalsmenge mit steigender Dimension auch eine geringere Güte aufweisen kann oder die Güte nicht mehr steigt. Die Merkmale, bei denen keine Verbesserung eingetreten ist, sollten für die Auswahl nicht berücksichtigt werden. Es ist nicht sinnvoll, die Suche beim ersten Auftreten dieses Verhaltens abzubrechen, da durch bestimmte Suchverfahren wie etwa die gleitende Suche im weiteren Verlauf des Algorithmus noch Verbesserungen eintreten können, da auch bereits ausgewählte Merkmale wieder verworfen werden. In dieser Arbeit sollen Merkmale bis zur maximal zulässigen Zahl berechnet und dann überprüft werden, ab welcher Stelle eine Verschlechterung oder gleichbleibende Güte auftritt. Zusätzliche Merkmale bedeuten für das verteilte

Erkennungssystem einen höheren Berechnungsaufwand und unter Umständen mehr Funkverkehr. Wenn keine Verbesserung der Güte auftritt, muss dieser zusätzliche Aufwand nicht betrieben werden.

### 4.3.3 Systemarchitektur und Anpassung des Mustererkennungsmodells

Die Anforderungen, welche die Erweiterung der verteilten Mustererkennung mit sich bringt, können nur teilweise von den Sensorknoten allein erfüllt werden.

Eine ortsunabhängige Erkennung zu erreichen wäre damit möglich, einen Knoten für das Training als zentralen Knoten zu definieren. Dadurch wären alle Knoten des Sensornetzes in der Lage, die Anordnung der Referenzvektoren der Knoten, welche die Ereignisse während des Trainings wahrgenommen haben, selbst vorzunehmen. Ein Problem stellt die Verwendung der in Abschnitt 4.3.1.3 beschriebenen Zurückweisungsmethoden dar. Für die Distanzzurückweisung müssten die Merkmalsvektoren aller Trainingsdaten auf den Sensorknoten gespeichert werden, um daraus den am weitesten vom Referenzvektor entfernten Vektor zu bestimmen. Dies ist speichertechnisch für kleine Trainingsmengen möglich, für größere jedoch nicht umsetzbar.

Das Training könnte von den Sensorknoten koordiniert werden, wenn die Sensorknoten während des Trainings Meldungen über wahrgenommene Ereignisse versenden. Sensorknoten, die kein Ereignis wahrnehmen, könnten dann auch zum Training des nächsten Musters oder der nächsten Klasse übergehen.

Der Austausch von Referenzvektoren mit allen Merkmalen nach Abschluss des Trainings zwischen den Sensorknoten bedeutet, dass eine eventuell große Menge an Referenzvektoren für die Erkennung verwendet werden muss. Dies ist, wie schon in Abschnitt 4.1 und 4.3.2 beschrieben, eine Anforderung an die Rechen- und Speicherkapazitäten der Sensorknoten, die nicht erfüllt werden kann. Die Menge an Werten muss mit Hilfe einer Merkmalsauswahl verkleinert werden. Die Berechnung kann aufgrund des damit verbundenen hohen Speicher- und Rechenaufwands nicht auf den Sensorknoten ausgeführt werden.

Zur Lösung der geschilderten Probleme wird ein zusätzliches externes System mit einer höheren Speicher- und Rechenkapazität im Vergleich zu den Sensorknoten eingesetzt. Angedacht ist, dass dieses System nur für das Training und zur Anzeige von Klassifikationsergebnissen verwendet werden soll. Die Erkennung unbekannter Ereignisse soll, wie auch in [Dzi07] umgesetzt, im Sensor Netz von den Sensorknoten ohne Zuhilfenahme externer Hardware oder Server-Sensorknoten durchgeführt werden, wodurch sich dieser Ansatz von vergleichbaren Arbeiten abhebt.

Wie in den Abschnitten 4.3.1.1 und 4.3.1.3 bereits erwähnt wurde, sollte nur von dem Knoten, an dem das Ereignis auftritt, eine korrekte Klassifikation erfolgen. Die Klassifikationen der übrigen Knoten sollten zurückgewiesen werden. Da dieses Verhalten zum Zeitpunkt der Entwicklung des Systems nicht verifiziert werden kann, muss eine Möglichkeit gefunden werden, mit mehreren Klassifikationen umzugehen. Dies soll vorerst auf dem externen System durch Fusion der Klassifikationen zu einem Gesamtergebnis geschehen. Sollte es sich als notwendig erweisen, dieses Verfahren auch in Zukunft anzuwenden, müsste es auf das Sensorknotensystem übertragen werden. Die Ergebnisse der Klassifikationsfusion geben weiterhin Aufschluss darüber, welche Güte der Klassifikation im Sen-

sornetz durch die Zurückweisungsmethoden erzielt wird. Die Ergebnisse können evaluiert und zur weiteren Verbesserung der Erkennung eingesetzt werden.

Das vorhandene System wird in zwei kooperierende Teilsysteme zerlegt, wie in Abbildung 4.11 dargestellt ist. Das bisher verwendete Modell der verteilten Mustererkennung wird aufgespalten. Das Training und die Erkennung werden teilweise auf dem externen System und teilweise auf den Sensorknoten ausgeführt. Zusätzlich werden neue Komponenten hinzugefügt und bestehende Komponenten in ihrer Funktionsweise verändert. Im Folgenden sollen die Modellphasen der Teilsysteme zusammen mit der Kommunikation untereinander und in Bezug auf die Lösung der beschriebenen Probleme geschildert werden. Auf die einzelnen Komponenten der Phasen wird für die Teilsysteme in den Abschnitten 4.4 und 4.5 detailliert eingegangen.

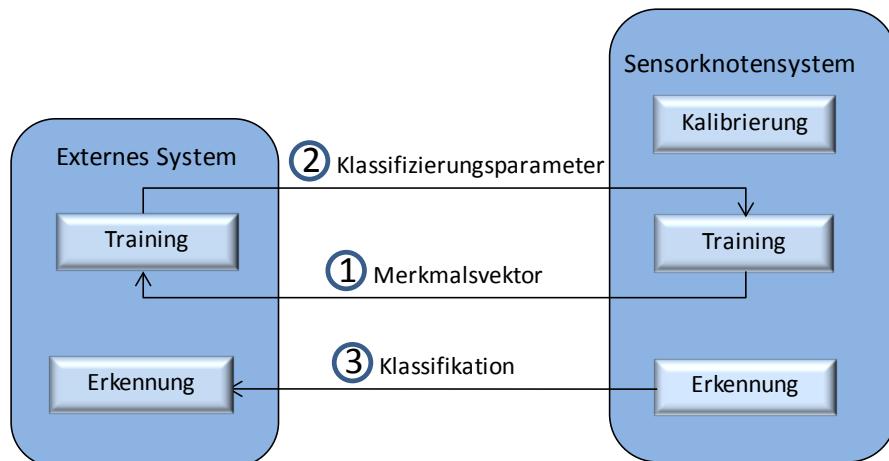


Abbildung 4.11: Systemarchitektur

#### 4.3.3.1 Modellphasen des externen Systems

##### Trainingsphase

Das Ziel des Trainings auf dem externen System ist es, Parameter, die für eine verteilte Ereigniserkennung auf den Sensorknoten benötigt werden, zu berechnen. Dazu werden die Merkmalsvektoren aller Sensorknoten, die an den trainierten Ereignissen beteiligt sind, gespeichert. Dadurch ergibt sich eine globale Sicht auf alle Daten. Die Sensorknoten müssen nicht mehr zwischen einzelnen Ereignissen während des Trainings unterscheiden. Die Steuerung des Trainings wird somit vom externen System übernommen. Da ein Ereignis durch Daten mehrerer Sensorknoten beschrieben wird, werden die Merkmalsvektoren nach dem Training aller Klassen fusioniert. Mit Hilfe der fusionierten Merkmalsvektoren werden die Merkmale ausgewählt, welche die verschiedenen Klassen am deutlichsten differenzieren. Die Referenzvektoren werden unter Verwendung dieser Merkmale erzeugt, wodurch sich eine Verringerung der Dimension ergibt. Die Zurückweisungsradien für die Distanzzurückweisung sowie die minimalen und maximalen Merkmalswerte für die Merkmalszurückweisung werden anhand der fusionierten Merkmalsvektoren und der ausgewählten Merkmale berechnet. Weiterhin werden die Werte für die Merkmalsnormierung während der Erkennung von den Sensorknoten benötigt. Diese werden auf dem externen System für alle zu verwendenden Merkmale berechnet. Des Weiteren ist ein Wissen über die Sensorknoten, deren Daten für das Training verwendet wurden, für die Erkennung

erforderlich. Die berechneten Parameter werden abschließend an alle Sensorknoten per Funk verschickt.

### **Erkennungsphase**

Die Erkennungsphase dient auf dem externen System dazu, Klassifikationsergebnisse mehrerer Knoten für ein unbekanntes Muster zu sammeln und danach zu einem Gesamtergebnis zu fusionieren. Die entstehende Klassifikation stellt das erwartete Ereignis dar. Ort und Typ des Ereignisses werden weiterhin auf einer Umgebungskarte dargestellt.

#### **4.3.3.2 Modellphasen des Sensorknotensystems**

##### **Kalibrierungsphase**

Die Kalibrierungsphase wird vor Beginn des Trainings ohne Änderungen durchgeführt.

##### **Trainingsphase**

Merkmalsvektoren werden aus aufgenommenen Beschleunigungsdaten berechnet. Diese werden nicht auf den Sensorknoten gespeichert, sondern an das externe System übermittelt. Diese sind dabei nicht merkmalsnormiert, da für eine Normierung zuerst das Training komplett abgeschlossen sein muss. Die Sensorknoten benötigen kein Wissen über aktuell zu trainierende Klassen, da dies vom externen System übernommen wird. Sie können somit aber nicht selbst über das Ende des Trainings entscheiden. Dieses wird durch die Verteilung von Klassifikationsparametern durch das externe System signalisiert. Die empfangenen Parameter werden gespeichert.

##### **Erkennungsphase**

Bei der Wahrnehmung eines Ereignisses wird eine verteilte Erkennung vorgenommen. Die eigenen Merkmalsvektoren werden versendet und die Daten der anderen Knoten anhand der Klassifikationsparameter fusioniert. Die Parameter dienen auch zur anschließenden Klassifikation des fusionierten Merkmalsvektors. Falls keine Zurückweisung der Klassifikation erfolgt, wird diese zusammen mit der minimalen Distanz an das externe System übermittelt.

## **4.4 Externes System**

Die Anforderungen an die erweiterte verteilte Ereigniserkennung machen eine Verwendung eines zusätzlichen externen Systems notwendig. Dieses übernimmt die Steuerung des Trainings, die Verarbeitung der Trainingsdaten der Sensorknoten sowie die anschließende Verteilung der Ergebnisse. Weiterhin fasst es Klassifikationsergebnisse mehrerer Knoten zu einem Ergebnis zusammen.

Das externe System muss so viel Speicher besitzen, um die Merkmalsvektoren aller Sensorknoten während des Trainings speichern zu können. Die rechenintensivste Funktion des externen Systems stellt die Merkmalsauswahl dar. Da während des Trainings die Dauer der Berechnung keine große Rolle spielt, muss mindestens garantiert sein, dass die Berechnungen in einer annehmbaren Zeitdauer durchgeführt werden können.

Die mit der Nutzung des externen Systems verbundene Aufspaltung des bisher verwendeten Modells der verteilten Mustererkennung führt dazu, dass einige Komponenten, wel-

che vorher auf den Sensorknoten ausgeführt wurden, auf das externe System verlagert werden. Weiterhin müssen neue Komponenten hinzugefügt werden, um den Anforderungen der erweiterten verteilten Ereigniserkennung zu genügen.

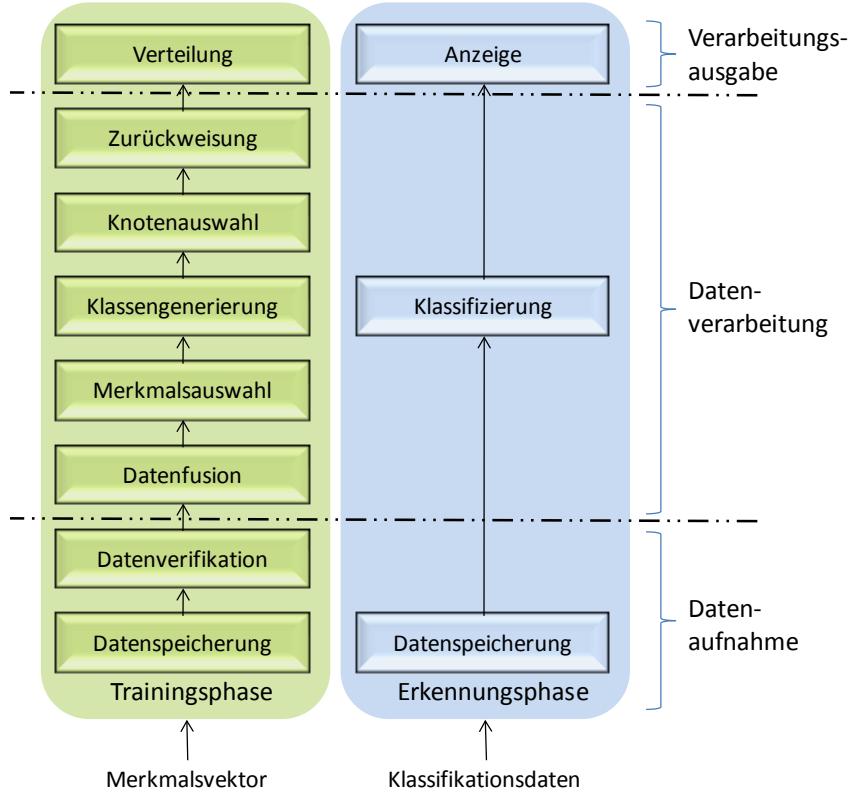


Abbildung 4.12: Verteilte Mustererkennung auf dem externen System

Die Komponenten Datenspeicherung und Datenverifikation werden hinzugefügt. Hiermit werden alle Trainingsdaten gespeichert. Die Datenfusion dient der Merkmalsfusion der Daten aller Sensorknoten. Die Merkmalsauswahl wird automatisch durchgeführt und deren Ergebnis in die Klassengenerierung einbezogen. Dabei kann es auch dazu kommen, dass alle Merkmale einiger Knoten nicht ausgewählt werden. Diese Knoten werden in der Komponente Knotenauswahl herausgefiltert. Die Zurückweisungskomponente berechnet die Zurückweisungsparameter verschiedener Zurückweisungsmethoden. Durch die bisherige verwendete Verteilungskomponente des Trainings wurden Referenzvektoren einzelner Sensorknoten verteilt. In diesem Fall werden die zu verwendenden Merkmale einzelner Knoten, die dazugehörigen Referenzdaten und Normierungsinformationen, die Informationen für Distanz- und Merkmalszurückweisung sowie die relevante Knotenmenge an alle Sensorknoten versendet. Die Datenspeicherung während der Erkennung dient dem Speichern von Klassifikationen, die von den Sensorknoten an das externe System übermittelt werden. Liegen mehrere Klassifikationen zu einem Ereignis vor, muss eine Gesamtklassifikation durch eine Klassifikationsfusion errechnet werden. Diese wird dann durch die Komponente Anzeige dargestellt.

Das externe System selbst verfügt nicht über die Möglichkeit, Funkübertragungen zu realisieren. Die Datenaufnahme von und Datenverteilung an andere Sensorknoten wird über eine serielle Verbindung zwischen dem externen Systems und einem Knoten erreicht,

wodurch Daten zeichenweise übertragen werden können. Dieser Knoten kann somit eigene Daten oder empfangene Daten anderer Knoten an das System übertragen. Bei Empfang von Daten auf der seriellen Schnittstelle können diese an die Knoten im Sensornetzwerk weitergeleitet werden. Neben der Datenübertragung werden über die serielle Verbindung auch Meldungen über aktuelle Vorgänge gemacht, die dem externen System oder den Sensorknoten explizit bekannt gemacht werden müssen. Dazu gehören etwa die abgeschlossene Datenaufnahme eines Ereignisses im Netzwerk während des Trainings oder der Signalisierung des Trainingsendes durch das externe System. Der Knoten soll zukünftig externer Sensorknoten genannt werden.

Es gibt zwei Kontexte, in denen der externe Knoten im Gesamtsystem integriert ist:

- a. Ein Sensorknoten, der keine Ereignisdaten aufnehmen und keine Erkennung ausführen kann und somit nicht in das Erkennungsnetzwerk integriert ist. Er kann aber Funkdaten anderer Sensorknoten empfangen und diese an das externe System übertragen. Weiterhin kann er Daten oder Meldungen auf der seriellen Schnittstelle empfangen und diese an die anderen Sensorknoten weiterleiten.
- b. Ein Sensorknoten im Erkennungsnetzwerk, der zusätzlich das Verteilen und Empfangen von Daten oder Meldungen über die serielle Verbindung beherrscht.

Bei Möglichkeit a müssen einige Funktionen des normalen Knotensystems abgeschaltet werden und kleinere Anpassungen gemacht werden. In Möglichkeit b können die Zusatzfunktionen ohne weitere Veränderung der eigentlichen Funktionen eingebaut werden. Dafür ist dieser Knoten mit zusätzlichen Aufgaben belastet, wodurch die eigentlichen Erkennungsfunktionen beeinträchtigt werden können. Möglichkeit a ist nicht an einen Standort gebunden und kann die Konzentration aufgrund der fehlenden Erkennungsfunktion ganz auf die Kommunikation mit dem externen System und den anderen Knoten lenken. Diese ist daher b vorzuziehen. Im Betrieb des Systems wird nie zwischen den Kontexten gewechselt. Während des Trainings ist es entscheidend, dass sich der externe Knoten im Funkbereich der Knoten befindet, welche die Ereignisse wahrnehmen, da dieser dessen Daten empfangen muss. Während der Erkennung reicht es, in der Funkreichweite irgendeines Sensorknotens zu sein, um Klassifikationsdaten zu empfangen. Gründe hierfür werden in Abschnitt 4.5 beschrieben.

Das allgemeine Paketformat, welche für die Übertragung zwischen externen System und externen Sensorknoten und umgekehrt verwendet wird, sieht wie folgt aus:

Nachrichtentyp	Knoten-ID	Klasse	Daten
----------------	-----------	--------	-------

**Abbildung 4.13: Allgemeines Paketformat zur Kommunikation zwischen externem System und externem Sensorknoten**

Der Nachrichtentyp bezeichnet die Funktion des Datenpakets. Der Typ kann entweder die Art der übertragenen Daten bestimmen oder eine bestimmte Meldung über aktuelle Vorgänge darstellen. Die Knoten-ID bestimmt die Kennung des Sensorknotens des Netzwerks, von dem zu übertragende Daten stammen. Sind Daten klassenspezifisch, wie etwa Referenzvektoren, wird die zugehörige Klasse im Klassenfeld angegeben. Die Daten sind

je nach Nachrichtentyp unterschiedlich. Nicht für alle Nachrichtentypen werden alle Felder belegt. Handelt es sich bei dem Paket um eine Meldung, wird nur das Nachrichtentyp-Feld belegt. Auf die spezifischen Paketinhalte wird an den entsprechenden Stellen genauer eingegangen.

Im weiteren Verlauf des Kapitels wird auf die Funktionsweise der einzelnen Komponenten und Phasen des externen Systems und im Zuge dessen auch auf die Kommunikation mit den Sensorknoten eingegangen.

#### **4.4.1 Datenspeicherung – Training**

Die Anzahl an Trainingsdaten kann dynamisch für jedes neue Training festgelegt werden. Zu Beginn eines Trainings muss der Benutzer folgende Angaben machen:

- Anzahl zu trainierender Klassen
- Anzahl Trainingsmuster pro Klasse
- Knoten-ID des zentralen Knotens
- Relevanz-Anforderung

Die Knoten-ID des zentralen Knotens wird nicht für die Datenspeicherung verwendet, aber für spätere Berechnungen benötigt. Auf die Relevanz-Anforderung wird in der Komponente Datenverifikation eingegangen.

Merkmalsvektoren der an einem Ereignis teilnehmenden Sensorknoten werden mittels des externen Knotens empfangen und über die serielle Schnittstelle zum externen System übertragen. Das zugehörige Paket wird durch den Nachrichtentyp 0 identifiziert. Der Merkmalsvektor wird anhand der Knoten-ID, der aktuell trainierten Klasse und der aktuellen Trainingsmusterzahl gespeichert.

Der externe Knoten erkennt, wenn alle Sensorknoten die Aufnahme und das Versenden von Merkmalsvektoren eines Ereignisses beendet haben. Wie dies geschieht, wird in Abschnitt 4.5.1 näher erläutert. Damit auch das externe System das Trainingsende eines Ereignisses erkennt, wird durch den externen Knoten eine Meldung gemacht.

Daraufhin wird die aktuelle Trainingsmusterzahl auf dem externen System um 1 inkrementiert. Erreicht diese einen Wert größer oder gleich der vorher definierten Anzahl an Trainingsmustern pro Klasse, wird eine Datenverifikation durchgeführt. Schlägt diese fehl, muss das Training der aktuellen Klasse fortgesetzt werden. Gibt es keinen Fehler, wird die aktuelle Klasse um 1 inkrementiert, die Trainingsmusterzahl auf 0 zurückgesetzt und mit dem Training der nächsten Klasse begonnen.

#### **4.4.2 Datenverifikation**

Ein Ereignis wird von mehreren Sensorknoten wahrgenommen. Dabei kann es dazu kommen, dass im Verlauf des Trainings einer Klasse nicht alle dieser Knoten regelmäßige Daten aufnehmen und senden. Bezogen auf den Anwendungsfall, ist das einfach nachvollziehbar. Der zentrale Knoten, an dem das Ereignis ausgelöst wird, sollte immer das Ereignis registrieren, da das zugehörige Bauzaunelement eine starke Bewegung aufweist.

Es wird vermutet, dass die Knoten an den Zaunelementen links und rechts davon mit zunehmender Entfernung das Ereignis unregelmäßiger wahrnehmen werden, da die Schwingung aufgrund physikalischer Bedingungen abnimmt.

Nachdem die vordefinierte Anzahl  $t$  an Trainingsmustern einer Klasse  $k$  erreicht ist, wird eine Überprüfung der Daten vorgenommen, welche von den Sensorknoten geliefert wurden.  $x$  stellt die Relevanz-Anforderung dar und entspricht der prozentual minimal zu liefernden Menge an Trainingsdaten eines Knotens, welche dieser für eine noch zu definierende Menge an Klassen liefern muss, damit dessen Daten als relevant angesehen werden. Nur die Daten der Knoten, welche die Relevanz-Bedingung erfüllen, sollen als Trainingsdaten der Klasse  $k$  verwendet werden. Dieser Wert muss zu Beginn des Trainings angegeben werden. Es gibt verschiedene Möglichkeiten,  $x$  auf die Daten der Knoten für verschiedene Klassen anzuwenden, die nachfolgend diskutiert werden sollen:

- a. Nur die Daten der Knoten werden verwendet, welche  $x\%$  von  $t$  Trainingsmustern für die aktuelle und jede bisher trainierte Klasse geliefert haben. Die Knoten, die Daten liefern, sind somit für alle Klassen gleich.
- b. Nur die Daten der Knoten werden verwendet, welche  $x\%$  von  $t$  Trainingsmustern für die aktuelle Klasse geliefert haben. Die zu verwendenden Daten der Knoten werden klassenspezifisch bestimmt.

Durch das System soll eine möglichst hohe Dynamik in der Verwendung der Daten von Knoten erreicht werden. Eine Anforderung an das zu entwickelnde System ist, dass sich die Menge an Sensorknoten, deren Werte während des Trainings von Ereignissen verwendet werden, sich für verschiedene Ereignisse ändern darf. Möglichkeit a ist dafür weniger geeignet, da dabei alle Knoten, deren Daten verwendet werden, eine ausreichende Anzahl an Merkmalsvektoren für alle Klassen liefern müssen. Wenn in der Menge der Ereignisse ein Ereignis  $E$  auftritt, welches nur von sehr wenigen Sensorknoten wahrgenommen wird, dann hat das Auswirkungen auf die Daten von anderen Sensorknoten, welche auch andere Ereignisse wahrgenommen haben. Diese werden nicht verwendet, weil diese Sensorknoten keine Daten für das Ereignis  $E$  liefern. Unter Umständen werden also nach dem Training aller Ereignisse nur Daten von wenigen Sensorknoten verwendet. Möglichkeit b bietet dagegen eine hohe Dynamik, da ein Knoten, dessen Daten verwendet werden, nur Daten für mindestens eine Klasse liefern muss. Wenn aber wirklich nur Daten für eine Klasse verwendet werden, stellt dies aber auch gleichzeitig eine schlechte Differenzierbarkeit der anderen Klassen dar. Jeder Knoten, dessen Daten verwendet werden, muss alle trainierten Klassen unterscheiden können. Wenn aber nur Daten für eine Klasse vorliegen, dann sind die restlichen Klassen von dem Sensorknoten nicht mehr unterscheidbar. Deshalb muss eine Zusatzbedingung hinzugefügt werden. Um eine Differenzierbarkeit der Klassen zu gewährleisten, dürfen von einem Knoten höchstens für eine Klasse nicht ausreichend Daten vorliegen. Da Daten für die anderen Klassen vorliegen, sind somit alle Klassen voneinander unterscheidbar.

Formal ergibt sich also die Möglichkeit c, welche in dieser Arbeit verwendet werden soll:

- c. Nur die Daten der Knoten werden verwendet, welche  $x\%$  von  $t$  Trainingsmustern für  $k - 1$  Klassen bei  $k$  zu trainierenden Klassen geliefert haben. Die zu verwendenden Daten der Knoten werden klassenspezifisch bestimmt.

Abbildung 4.14 zeigt ein Beispiel für die gelieferten Daten mehrerer Knoten für eine Klasse. Knoten 12 stellt den zentralen Knoten dar. Die Daten aller Knoten oberhalb der Relevanz-Anforderung (hier: 85%) werden als gültig angesehen (Knoten 10, 11, 12, 13, 14).

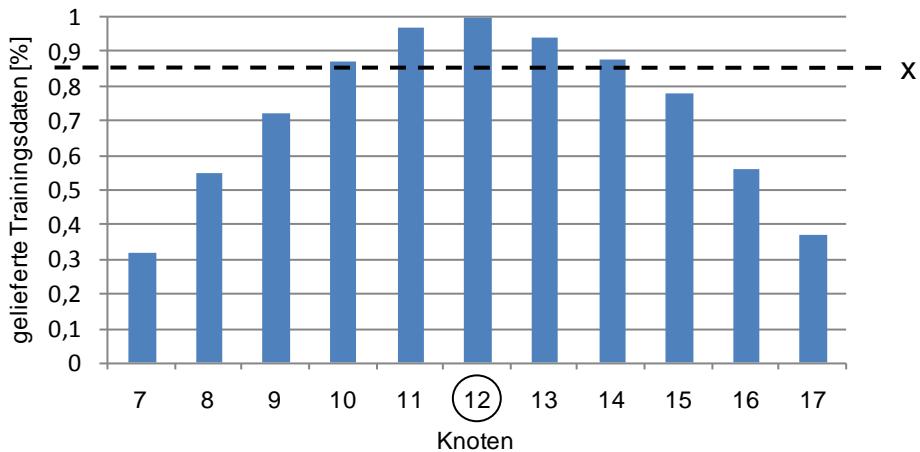


Abbildung 4.14: Trainingsdatenverteilung einer Klasse mit Relevanz-Anforderung

Die ursprüngliche Anforderung war, insgesamt  $t$  Muster für eine Klasse zu trainieren. Wenn der Wert von  $x$  unter 100% liegt und es Knoten gibt, die weniger als 100% und mehr als  $x\%$  Daten geliefert haben, schlägt die Datenverifikation fehl, da nicht von allen ausgewählten Knoten der Klasse genügend Daten geliefert wurden. Das Training muss also fortgesetzt werden. Nach jedem weiteren Trainingsschritt wird eine erneute Verifikation durchgeführt, diesmal aber nur für die bereits ausgewählten Knoten. Es wird überprüft, ob von diesen Knoten  $t$  Muster geliefert wurden. Dabei reicht es nicht, dass jeder Knoten  $t$  Merkmalsvektoren liefert, sondern es müssen von allen Knoten Merkmalsvektoren für  $t$  verschiedene Ereignisse im Verlauf des Trainings der Klasse geliefert worden sein. Nur dann sind genau  $t$  Muster von den ausgewählten Knoten trainiert worden.

Der Wert für  $x$  sollte nicht zu klein gewählt werden, da man sonst Knoten auswählt, die unregelmäßig Daten liefern. Dies kann die Erkennungsrate verschlechtern. Außerdem können dadurch unter Umständen sehr viele zusätzliche Trainingsschritte notwendig sein. Ein zu hoher Wert für  $x$  lässt viele aussagekräftige Daten außen vor.

#### 4.4.3 Datenfusion

Die Datenfusion stellt den ersten Verarbeitungsschritt der aufgenommenen Daten dar und wird dann ausgeführt, wenn das Training für alle Klassen abgeschlossen ist.

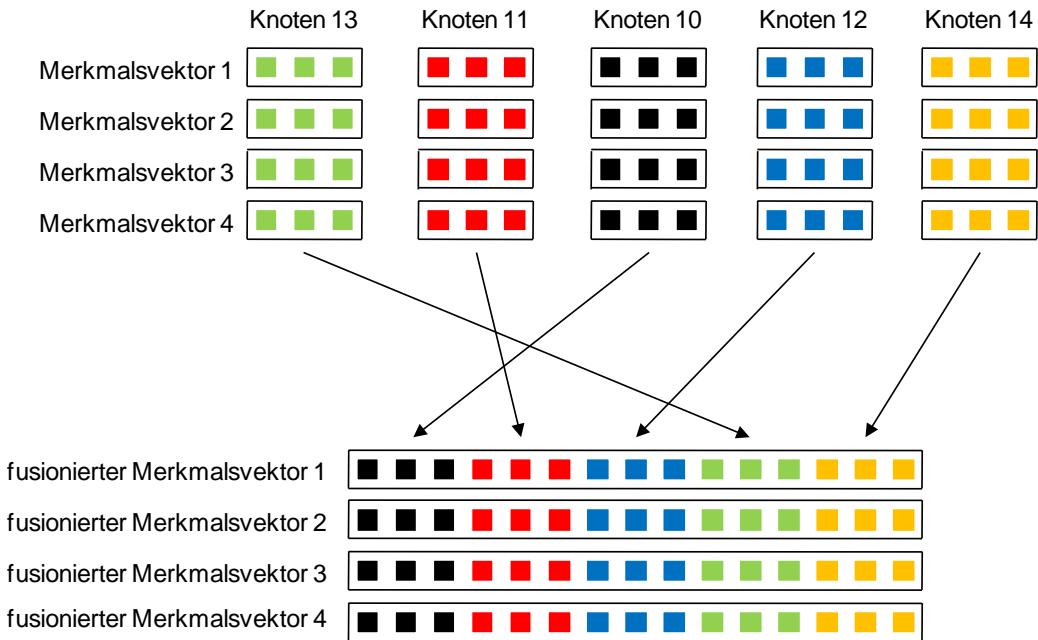
Da ein Ereignis durch die Daten mehrerer Sensorknoten definiert wird, sollen die gespeicherten Merkmalsvektoren fusioniert werden. Im vorherigen Schritt der Datenverifikation wurden relevante Knoten klassenspezifisch ausgewählt und eine vorgegebene Anzahl an Trainingsmustern für jede Klasse trainiert. Die Merkmalsvektoren müssen während der

Fusion anhand einer eindeutigen Struktur angeordnet werden. Dazu wird die in Abschnitt 4.3.1.2 eingeführte aufsteigende Ordnung der Knotenkennungen verwendet.

Es kann vorkommen, dass für verschiedene Klassen die Daten unterschiedlicher Sensor-knoten durch die Relevanz-Anforderung als relevant ausgewählt wurden. Wenn die Daten eines Knotens für eine Klasse  $k_1$  nicht ausgewählt wurden, heißt das, dass nicht genü-gend Daten geliefert wurden. Für eine andere Klasse  $k_2$  hat dieser Knoten aber genü-gend Daten geliefert. Um eine Vergleichbarkeit der beiden Klassen herzustellen, müssen Werte für die Daten des Knotens für Klasse  $k_1$  angenommen werden. Nicht genügend Daten geliefert soll in diesem Fall bedeuten, dass von dem Knoten keine Daten geliefert wurden. Selbst wenn der Knoten wenige Merkmalsvektoren geliefert hat, sollen diese nicht betrachtet werden, da deren Aussagekräftigkeit nicht bestimmt werden kann. Die anzunehmenden Merkmalswerte für die Daten dieses Knotens sollen daher einem Ruhe-zustand des Sensorknotens entsprechen. Die Werte hängen dabei von den verwendeten Merkmalen des Sensorknotensystems ab und sollen daher in dem dazugehörigen Ab-schnitt 5.3 besprochen werden.

Für die relevanten Knoten jeder Klasse wurde eine vordefinierte Anzahl an Mustern tra-i-niert. Die Merkmalsvektoren dieser Knoten liegen auf dem externen System getrennt vor und können für jedes Muster anhand der aufsteigenden Ordnung der Knotenkennungen zusammengefügt werden. Für Knoten, die für eine Klasse nicht relevant sind, dafür aber für eine beliebige andere Klasse, werden die vom Benutzer definierten Werte für alle Mu-ster angenommen und die entstehenden Merkmalsvektoren entsprechend der Ordnung der Knotenkennungen zu den Merkmalsvektoren der relevanten Knoten hinzugefügt. Die fu-sionierten Merkmalsvektoren müssen dann noch anhand der minimalen und maximalen Merkmalswerte über alle Klassen merkmalsnormiert werden, da von den Sensorknoten während des Trainings nur nicht normierte Merkmalsvektoren geliefert werden. Die vom Benutzer angegebenen Daten sollten dabei nicht in Bestimmung der minimalen und ma-ximalen Werte einfließen, da es sich bei den Daten um keine echten Bewegungsdaten handelt und die Merkmalsnormierung der tatsächlich gelieferten Merkmalsvektoren sonst verfälscht wird.

Abbildung 4.15 zeigt eine Fusion von vier Merkmalsvektoren einer Klasse von fünf Sen-sorknoten. Jeder Merkmalsvektor besteht aus drei Merkmalen. Die Merkmalsvektoren werden anhand der Kennungen der Knoten in aufsteigender Ordnung zusammengefügt.



**Abbildung 4.15: Fusion von Merkmalsvektoren einer Klasse**

#### 4.4.4 Merkmalsauswahl

Die Merkmalsauswahl wird anhand der fusionierten Merkmalsvektoren vorgenommen. Sind  $m$  relevante Knoten für alle Klassen ausgewählt worden und wurden pro Knoten  $n'$  Merkmale verwendet, so müssen nun  $n' * m$  Merkmale mittels der Merkmalsauswahl reduziert werden. Die Anordnung der Merkmale der Knoten wird anhand der Ordnung der aufsteigenden Knotenkennungen vorgenommen und entspricht somit der Anordnung der fusionierten Merkmalsvektoren. Die hier verwendeten Verfahren zur Merkmalsauswahl wurden in Abschnitt 4.3.2 ausführlich beschrieben.

Der Benutzer muss vor dem Beginn der Merkmalsauswahl folgende Angaben machen:

- Maximal erlaubte Anzahl an Merkmalen
- Zu verwendendes Gütekriterium

Die maximal erlaubte Anzahl an Merkmalen ist systembedingt und legt fest, wie viel Merkmale die Sensorknoten verarbeiten können. Dieser Wert ist auf dem Sensorknotensystem definiert und sollte auch hier verwendet werden. Für die Wahl des Gütekriteriums gibt es zwei Möglichkeiten: Kreuzvalidierung und Inter/Intra-Klassenabstand. Nachdem die Werte gewählt sind, wird die Merkmalsauswahl anhand der gleitenden Suche und dem gewählten Gütekriterium ausgeführt.

Die gleitende Suche hat bezogen auf die Topologie der Sensorknoten einen Nachteil. Diese beginnt, neu hinzuzunehmende oder zu verwerfende Merkmale gemäß der durch die Datenstruktur induzierten Reihenfolge zu untersuchen. Die Reihenfolge entspricht dabei der aufsteigenden Ordnung der Knoten. Die Merkmale des zentralen Knotens sollten aber zuerst untersucht werden, da dort Ereignisse ausgelöst werden und diese dort am wesentlichsten wahrgenommen werden. Bei weiter vom Zentrum entfernten Knoten ist auch die Wahrscheinlichkeit hoch, dass Ereignisse nur unregelmäßig wahrgenommen

werden, nahe des Zentrums werden mit hoher Wahrscheinlichkeit fast immer Daten geliefert, was auch eine höhere Aussagekraft bedeutet. Die gleitende Suche verhält sich so, dass bei gleicher erreichter Güte durch zwei Merkmale das zuerst betrachtete Merkmal ausgewählt werden würde. Dadurch ergibt sich bei gleicher Güte eine implizite höhere Gewichtung von Merkmalen, die vor anderen Merkmalen bewertet werden. Da die Merkmale des zentralen Knotens aufgrund der topologischen Ordnung später betrachtet werden als die Merkmale der linken Nachbarknoten, würden somit Merkmale der weiter vom Entstehungsort des Ereignisses entfernten Sensorknoten bei gleicher Güte den Merkmalen des zentralen Knotens vorgezogen werden.

Um diesen Defekt zu beheben, wird die gleitende Suche so modifiziert, dass die Suche nach einem hinzuzunehmenden bzw. zu verworfenden Merkmal zuerst für alle Merkmale des zentralen Knotens durchgeführt wird, dann für den linken und rechten Nachbarknoten und immer so weiter nach außen gehend. Dadurch ergibt sich eine implizit abnehmende Gewichtung von Merkmalen vom zentralen Knoten zu Merkmalen beiderseitig weiter entfernter Knoten. Die dazu benötigte Knotenkennung des zentralen Knotens wird zu Beginn des Trainings vom Benutzer angegeben.

Neben der Einbeziehung der Topologie des Sensornetzes ist es auch möglich, Energieersparnisse mit Hilfe der Merkmalsauswahl zu erreichen.

Eine Möglichkeit besteht darin, die Merkmalsauswahl an spätere Merkmalsberechnungen der Knoten anzupassen. Theoretisch müssten von jedem Knoten nur die Merkmale während der Erkennungsphase berechnet werden, die auch von anderen Knoten für die Erkennung verwendet werden. Für die Merkmalsauswahl bedeutet das, dass für ein neu hinzuzunehmendes bzw. zu verworfendes Merkmal zusätzlich überprüft werden sollte, ob durch dieses eine Einsparung der Merkmalsberechnung erreicht werden kann. Die Entscheidung kann auf bestimmten Regeln basieren, wie etwa dass von zwei Merkmalen und annähernd gleicher erreichter Güte eher das Merkmal genommen wird, wodurch eine Ersparnis der Berechnung erzielt wird. Durch diese Anpassung der Merkmalsauswahl werden Merkmalsberechnungen und die Anzahl der Daten, die per Funk verschickt werden, verringert. Wenn jedoch Ereignisse in der Praxis nur selten auftreten, wie bei der Überwachung eines Bauzauns, sind die Effekte auf lange Sicht gesehen nur wenig effizient. Zudem kann sich die Erkennungsrate durch die Beeinflussung der Merkmalsauswahl und damit verbundene eventuelle Auswahl einer weniger guten Merkmalsmenge verschlechtern. Daher soll hier auf die Implementierung dieser Anpassung verzichtet werden.

Eine weitere Möglichkeit ist, eher Merkmale von solchen Knoten auszuwählen, von denen schon Merkmale ausgewählt wurden. Dadurch verringert sich die Anzahl der Knoten, deren Daten für eine Erkennung verwendet werden. Die Ereigniserkennung ist somit weniger anfällig für Paketverluste. Zudem müssten abhängig vom Verteilungsverfahren der Daten eventuell weniger Pakete verschickt werden. Da die erwartete Menge an Knoten aufgrund der Merkmalsbeschränkungen der Sensorknoten als relativ gering anzunehmen ist, würde sich mit der beschriebenen Maßnahme in diesem Fall keine deutliche Verbesserung erzielen lassen. Wie bei der vorher beschriebenen Möglichkeit kann auch hier die Erkennungsrate sinken. Eine Umsetzung wird daher nicht vorgenommen.

Als Ergebnis der Merkmalsauswahl wird eine Bitmaske der Länge  $n' * m$  geliefert, die für jedes Merkmal angibt, ob es ausgewählt wurde oder nicht. Für die Bitmaske gilt die Anordnung der Merkmale der Knoten nach der Ordnung der aufsteigenden Kennungen.

#### 4.4.5 Klassengenerierung

Die Referenzvektoren für alle Klassen werden anhand der fusionierten Merkmalsvektoren durch Mittelwertbildung ermittelt. Die tatsächlich zu verwendenden Referenzwerte werden durch die ausgewählten Merkmale der Merkmalsauswahl bestimmt, indem die Bitmaske auf die Referenzvektoren angewandt wird.

Abbildung 4.16 zeigt einen Referenzvektor einer Klasse, welcher anhand von fusionierten Merkmalsvektoren von fünf Sensorknoten mit jeweils drei Merkmalen erstellt wurde. Vier ausgewählte Merkmale lassen in diesem Beispiel einen um 73% dimensionsreduzierten Referenzvektor entstehen.

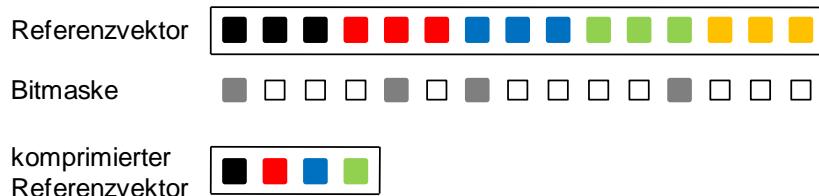


Abbildung 4.16: Generierung eines komprimierten Referenzvektors einer Klasse

#### 4.4.6 Knotenauswahl

Durch die Merkmalsauswahl werden von einigen Knoten unter Umständen keine Merkmale ausgewählt. Dies ist beispielsweise garantiert der Fall, wenn weniger Merkmale ausgewählt wurden als relevante Knoten existieren. Diese Daten werden niemals verwendet. Die Sensorknoten benötigen aber nur Informationen über die Knoten, deren Daten auch während des Trainings verwendet wurden. Die Menge an Knoten, welche ursprünglich am Training der Ereignisse beteiligt war, wird somit auf eine kleinere Menge reduziert. Für jeden relevanten Knoten wird anhand der Bitmaske überprüft, ob mindestens ein Merkmal von diesem Knoten ausgewählt wurde. Ist dies nicht der Fall, so wird der Knoten aus der Knotenliste entfernt. Im Beispiel in Abbildung 4.17 werden nur von vier der fünf Sensor-knoten Merkmale ausgewählt. Knoten 14 wird somit entfernt.

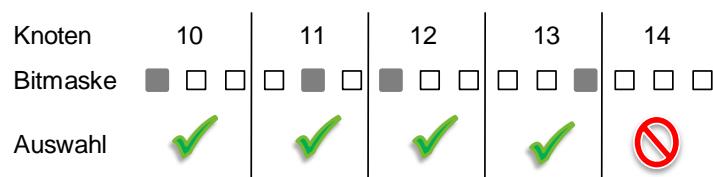


Abbildung 4.17 Knotenauswahl

Wie schon in Abschnitt 4.4.4 erwähnt, kann man die Knotenauswahl in die Merkmalsauswahl einfließen lassen. Weniger Knoten bedeuten eine geringere Wahrscheinlichkeit von Übertragungsfehlern, aber durch die Beeinflussung der Merkmalsauswahl auch eine mögliche Senkung der Erkennungsrate. Es muss untersucht werden, wie sich Übertragungsfehler auf die Erkennungsrate auswirken, um eine optimale Knotenzahl bestimmen zu

können. Dies lässt sich aber nur mit Hilfe von vielen Experimenten ermitteln. Aufgrund dessen und der im genannten Abschnitt angeführten Gründe soll das nur eine theoretische Betrachtung darstellen.

Ein weiteres Problem stellt die systembedingte maximale Anzahl an Knoten dar, welche von den Sensorknoten verwaltet werden können. Für das Sensorknotensystem ist diese Anzahl aufgrund des zur Verfügung stehenden Speichers auf 10 Knoten beschränkt. Dieses Problem könnte man auch durch die Merkmalsauswahl lösen, indem ab dem Zeitpunkt, an dem diese Anzahl erreicht ist, nur noch Merkmale von Knoten ausgewählt werden, von denen schon Merkmale verwendet werden. Dadurch erhöht sich die Knotenzahl nicht. In dieser Arbeit ist der Speicherbedarf auf den Sensorknoten für das statische Vorhalten von Speicherplatz für eine Menge von Knoten relativ gering, da die Merkmalsvektoren dieser Knoten nicht vollständig gespeichert werden müssen. Somit wird dieser Wert hoch genug gesetzt, so dass die Grenze für den Anwendungsfall dieser Arbeit nicht erreicht wird.

Die Knotenauswahl wird in diesem Fall somit als getrennte Komponente betrachtet, da sie unabhängig von der Merkmalsauswahl arbeitet.

#### **4.4.7 Zurückweisung**

Die Möglichkeiten einer Zurückweisung wurden in Abschnitt 4.3.1.3 besprochen. Da durch die theoretische Betrachtung der Distanz- und Merkmalszurückweisung nicht festgestellt werden kann, welche davon am geeignetsten ist, sollen beide Zurückweisungsarten von den Sensorknoten getrennt durchgeführt werden und zur Auswertung in dieser Arbeit verwendet werden. In der praktischen Anwendung soll das System auf eine Methode beschränkt sein.

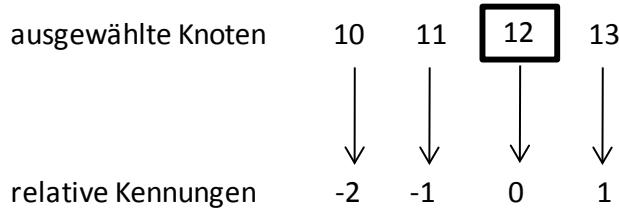
Die Zurückweisungsradien für die Distanzzurückweisung werden anhand der fusionierten Merkmalsvektoren berechnet. Dasselbe gilt für die Merkmalszurückweisung. Hierbei müssen die minimalen und maximalen Werte für die Merkmalsgrenzen aus diesen Daten gewonnen werden. In beiden Fällen werden wie schon bei der Klassengenerierung nur Werte von ausgewählten Merkmalen verwendet.

#### **4.4.8 Verteilung**

Die Ergebnisse der Komponenten Merkmalsauswahl, Klassengenerierung, Knotenauswahl und Verteilung müssen nun an die Sensorknoten übertragen werden. Dazu werden diese zuerst über die serielle Verbindung an den externen Knoten übermittelt.

Die anhand der Knotenauswahl gewählten Knoten sollen übertragen werden. Dazu werden diese anhand der festgelegten aufsteigenden Ordnung der Knotenkennungen sortiert. Die entstehende Folge von Kennungen kann aber so nicht übertragen werden, da die Sensorknoten des Netzwerks kein Wissen über die Kennung des zentralen Knotens haben. Dies wird aber als Bezugspunkt für die Erkennung benötigt. Daher wird mit relativen Kennungen gearbeitet. Die Knoten-ID des zentralen Knotens wird von jeder Kennung in der sortierten Folge abgezogen. Die 0 stellt somit den zentralen Knoten dar. In der Menge links davon liegen absteigende negative relative Kennungen vorliegen vor, rechts der 0

steigen die relativen Kennungen an, wie in Abbildung 4.18 mit Knoten 12 als zentralem Knoten zu sehen ist. Diese relativen Kennungen entsprechen der Nachbarschaftsrelation. Der externe Knoten und der Sensorknoten halten einen statischen Speicherplatz vor, um die relativen Kennungen zu speichern. Bei Erhalt dieser Kennungen über die serielle Schnittstelle muss vom externen Knoten neben den diesen somit zusätzlich auch die Anzahl gespeichert werden, da diese sich aus dem statischen Speicher nicht erfassen lässt.



**Abbildung 4.18: Erstellung relativer Kennungen**

Die Bitmaske ist eine Darstellung der Auswahl von Merkmalen der relevanten Knoten vor der Knotenauswahl. Da nur die Kennungen der Knoten übertragen werden, von denen auch Merkmale ausgewählt wurden, genügt es, nur die Auswahl der Merkmale dieser Knoten darzustellen. Verworfene Knoten während der Knotenauswahl liefern kein zu verwendendes Merkmal, somit wird die Information in der Bitmaske nicht benötigt. Ausgehend von der vollständigen Bitmaske wird eine komprimierte Bitmaske erstellt, welche nur eine Darstellung von Merkmalen solcher Knoten ist, die auch mindestens ein Merkmal liefern.

Die durch die Komponente Klassengenerierung erzeugten komprimierten Referenzvektoren werden klassenweise übertragen. Da die Sensorknoten und der externe Knoten kein Wissen über die Anzahl der trainierten Klassen hat, da die Steuerung des Trainings vom externen System übernommen wird, wird dies für den externen Knoten anhand der Klasseninformationen, die im Paket über die serielle Verbindung enthalten sind, extrahiert. Im externen System beginnen die Nummern, wodurch die Klassen identifiziert werden, bei 0. Für jede weitere Klasse wird dieser Wert um 1 erhöht. Die Anzahl der Klassen wird durch den externen Knoten dadurch identifiziert, indem er für jedes erhaltene Referenzvektorpaar das Maximum aus bisher gespeicherter Anzahl und aktueller Klasseninformation plus 1 des Pakets bildet.

Die Zurückweisungsradien der Distanzzurückweisung werden für alle Klassen in einem Paket übertragen, da sonst in einem Paket nur ein Wert übertragen werden würde. Das würde eine Verschwendug der Ressourcen darstellen. Die Radien werden nach Klassennummer aufsteigend sortiert, um sie bei Erhalt den Klassen zuordnen zu können. Die minimalen und maximalen Werte der Merkmale der Merkmalszurückweisung werden getrennt und für jede Klasse einzeln übermittelt.

Da die Sensorknoten über keine Merkmalsnormierungsinformationen verfügen, müssen auch diese übertragen werden. Diese wurden schon bei der Normierung der Merkmalsvektoren während der Datenfusion verwendet und können auf gleiche Weise wieder ermittelt werden. Hier genügen aber die Informationen für die ausgewählten Merkmale. Da die

Informationen über alle Klassen gebildet werden, müssen hier nur die minimalen und maximalen Werte getrennt transferiert werden.

Nachdem der externe Knoten alle Daten empfangen und gespeichert hat, werden diese mittels Broadcast in einer sogenannten EXTERNALTRAINRESULTS-Nachricht an die Sensorknoten im Netzwerk übermittelt. Der Dateninhalt dieser Nachricht ähnelt stark dem Paketformat für die serielle Datenübertragung, enthält aber zusätzliche Informationen. Wie in Abbildung 4.19 zu sehen ist, wird der Typ der Daten, die Klasse und die Daten selbst übertragen. Da die Anzahl der Daten statisch festgelegt ist, müssen die vom externen Knoten ermittelte Klassenanzahl und Knotenanzahl extra übertragen werden, da diese sich in diesem Fall nicht extrahieren lassen. Es muss keine Information über die Anzahl der ausgewählten Merkmale gegeben werden, da diese sich aus der Bitmaske ermitteln lässt.

Datentyp	Klasse	Klassenanzahl	Knotenanzahl	Daten
----------	--------	---------------	--------------	-------

Abbildung 4.19 Datenpaket der EXTERNALTRAINRESULTS-Nachricht

Die Daten werden wie bei der seriellen Übertragung auf die Nachrichten verteilt. Der Datentyp bestimmt die Art der Daten. Jede Nachricht wird vom externen Knoten mittels Vorwärtsfehlerkorrektur zweimal übertragen, um eventuelle Paketverluste zu handhaben. Paketverluste könnten auch durch Bestätigungen erkannt werden, wenn sich aber sehr viele Sensorknoten im Bereich des externen Knotens befinden und diese die Nachrichten annähernd gleichzeitig empfangen, müssen alle eine Bestätigung senden. Da die Daten von den Sensorknoten weitergeleitet werden, worauf bei der Beschreibung des Sensor-knotensystems genauer eingegangen wird, ist es nicht entscheidend, dass alle Sensor-knoten in Funkreichweite die Nachrichten empfangen. Daher reicht es aus, mit Vorwärts-fehlerkorrektur zu arbeiten.

Nachdem alle Daten verschickt sind, wird darüber eine Meldung an das externe System gemacht, woraufhin dieser zur Erkennungsphase wechselt.

#### 4.4.9 Datenspeicherung – Erkennung

Die Erkennung eines unbekannten Ereignisses wird von mehreren Sensorknoten durchgeführt. Aufgrund der angewandten Zurückweisung sollte aber pro Ereignis nur eine Klassifikation gültig sein und versendet werden. Wenn die Zurückweisung versagt, kann es zu mehreren gültigen Klassifikationen kommen. Bevor eine Gesamtklassifikation berechnet werden kann, müssen die eintreffenden Klassifikationen, die zu einem Ereignis gehören, auf dem externen System gespeichert werden.

Während des Trainings war es für den externen Knoten möglich, das Ende einer Daten-aufnahme und Versenden derer Merkmalsvektoren aller Sensorknoten im Ausbreitungs-bereich des Ereignisses festzustellen. Da sich der externe Knoten nun nicht mehr unbedingt im Funkbereich der Knoten befindet, welche ein unbekanntes Ereignis wahrnehmen, da dies an einem beliebigen, weiter entfernten Ort stattfinden kann, hat dieser nicht mehr genügend Informationen, um solch eine Feststellung zu treffen. Es muss also eine andere

Möglichkeit gefunden werden, die Klassifikationen mehrerer Knoten einem Ereignis zuzuordnen. Diese wird auf dem externen System durchgeführt. Wird eine Klassifikationsübertragung vom externen Knoten empfangen, so wird ein Zeitgeber gestartet, sofern keiner aktiv ist. Die aktuelle Ereignisnummer wird um 1 erhöht. Alle Klassifikationen, die in dem darauffolgenden definierten Zeitfenster eingehen, werden der aktuellen Ereignisnummer zugeordnet.

Die Klassifikationsergebnisse der Sensorknoten werden vom externen Knoten empfangen und über die serielle Schnittstelle an das externe System geleitet. Die Klassifikationen werden anhand der Knoten-ID und der aktuellen Ereignisnummer gespeichert. Für eine spätere Bewertung mehrerer Ergebnisse reicht die Klasseninformation nicht aus. Die Sensorknoten senden daher zusätzlich die minimale Distanz, anhand derer die Klasse ausgewählt wurde. Diese wird zusätzlich zu der Klassifikation auf dem externen System gespeichert.

#### **4.4.10 Klassifizierung**

Wenn das Ende des Zeitfensters der Datenaufnahme erreicht ist, müssen die gespeicherten Klassifikationsergebnisse zu einem Gesamtergebnis zusammengeführt werden. Dazu wird eine gewichtete Klassifikationsfusion ausgeführt. Die Distanzen, welche von Sensorknoten während des Zeitfensters geliefert wurden und dort zu den vorliegenden Klassifikationen geführt haben, werden hierzu verwendet. Von allen Distanzen wird die minimale Distanz bestimmt. Der zugehörige Knoten sollte dem Knoten entsprechen, an dem das Ereignis ausgelöst wurde, da dessen Anordnung der Daten der Sensorknoten der des Trainings entspricht und somit den besten Distanzwert erzielen sollte. Die Klassifikation dieses Knotens stellt die fusionierte Klassifikation dar.

#### **4.4.11 Anzeige**

Auf dem externen System gibt es die Möglichkeit, eine Karte der Umgebung zu laden, wo die Sensorknoten ausgebracht wurden. Die Position der Sensorknoten auf dieser Karte können auf verschiedene Arten ermittelt werden:

- a. Durch Lokalisierungsmethoden
- b. Durch GPS
- c. Durch manuelle Eintragung.

Lokalisierung ist nicht Thema der Arbeit, aber trotzdem für das System sinnvoll. In dieser Arbeit soll darauf verzichtet werden. Es gibt aber die Möglichkeit c, die Knoten per Hand auf der Karte einzutragen.

Nach der Klassifizierung sind zwei Informationen bekannt: die Kennung des Sensorknotens, an dem das Ereignis ausgelöst wurde und die Art des Ereignisses. Der Ort des Ereignisses wird anhand der manuell eingetragenen Knoten angezeigt und die Art des Ereignisses durch eine der Klassifikation entsprechenden Farbgebung kenntlich gemacht.

## 4.5 Anpassung des Sensorknotensystems

Die Verwendung eines externen Systems wirkt sich auf die Arbeitsweise des bestehenden Sensorknotensystems aus. Da die Steuerung des Trainings und Auswertung der Trainingsdaten vom externen System übernommen wird, sind die Sensorknoten während dieser Zeit nur für die Versendung der aus den Rohdaten der Muster extrahierten Merkmalsvektoren an dieses System zuständig. Werden Klassifikationsparameter als Ergebnis des Trainings vom externen System empfangen, werden diese für die daraufhin beginnende verteilte Erkennung verwendet. Die dafür verwendete Merkmalsfusion wird aus Effizienzgründen lokal und nicht auf dem externen System durchgeführt. Bei einer nicht eindeutigen Klassifikation werden die Ergebnisse verworfen, ansonsten an das externe System übermittelt.

Da mit der Einführung des externen Systems eine Aufspaltung des Modells der verteilten Mustererkennung verbunden ist, werden einige Komponenten nicht mehr auf dem Sensorknotensystem verwendet. Zusätzlich benötigte Funktionen bedingen das Hinzufügen neuer Komponenten.

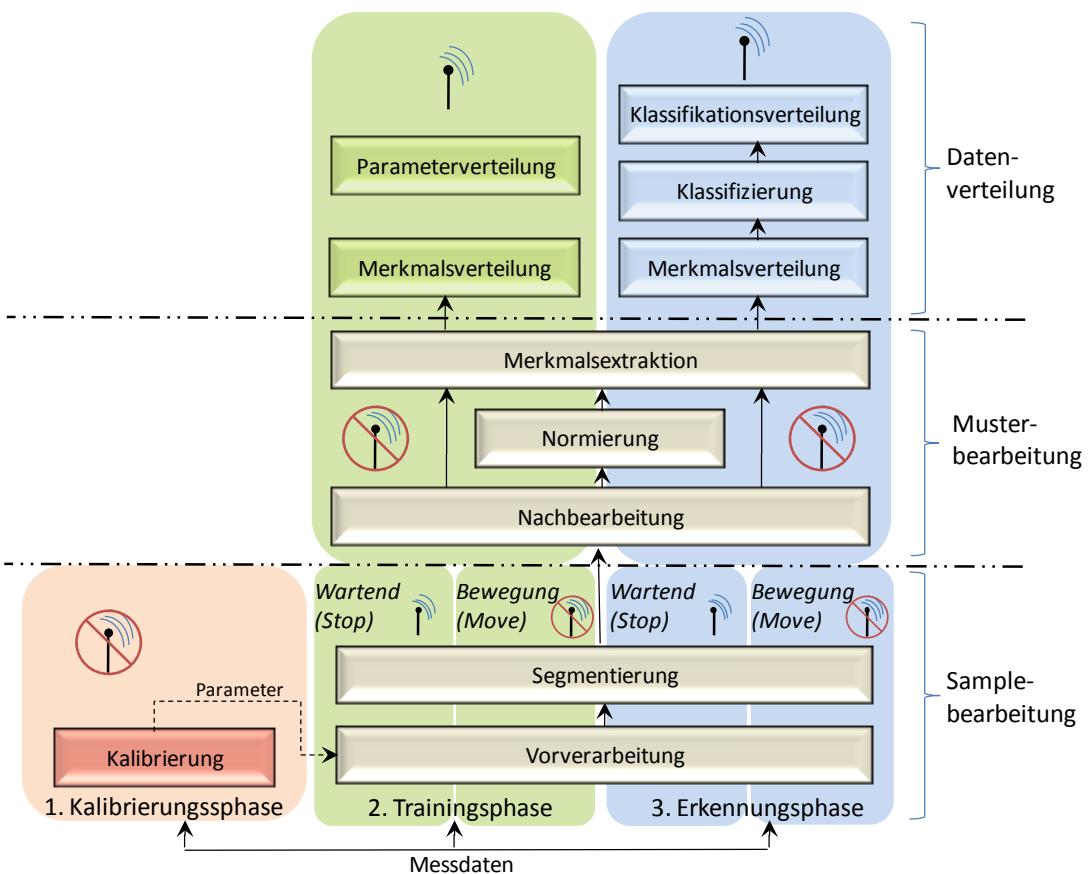


Abbildung 4.20: Verteilte Mustererkennung auf dem Sensorknotensystem

Die bisher verwendete Hello-Phase, welche vor der Kalibrierungsphase angesiedelt war, wird nicht mehr benötigt, da die Knoten, deren Daten verwendet werden, von dem externen System bestimmt werden. Die Kalibrierung der Sensorknoten muss weiterhin ausgeführt werden. Die Komponenten, die während der Samplebearbeitung und Musterbearbei-

tung ausgeführt werden, bleiben bestehen, da die Fusion der Daten während des Trainings und der Erkennung auf den Knoten auf der Merkmalsebene geschieht. Die Sensorknoten müssen dazu wie bisher Merkmale aus den Rohdaten extrahieren.

Der berechnete nicht normierte Merkmalsvektor wird während des Trainings durch die Komponente Merkmalsverteilung an das externe System versendet. Die Merkmalsauswahl und Klassengenerierung wird vom externen System übernommen und hier nicht mehr benötigt. Die bisher danach folgende Traindistribution-Komponente, in der die berechneten Referenzvektoren an die anderen Sensorknoten übermittelt wurden, wird durch die Komponente Parameterverteilung ersetzt. Dabei werden die vom externen System empfangenen Parameter, welche für die verteilte Erkennung verwendet werden, gespeichert und an andere Knoten weitergeleitet.

Die Merkmalsverteilung während der Erkennungsphase entspricht in etwa der bisherigen Distribution-Komponente. Merkmalsvektoren werden an andere Sensorknoten verteilt und empfangene Daten anhand zu verwendender Merkmale und relativer Knotenkennungen in einer bestimmten Reihenfolge angeordnet. Im Gegensatz zur bisherigen Verteilung müssen nicht Daten von allen Sensorknoten empfangen werden, von denen Daten verwendet werden würden. Die Merkmalsnormierung wird an diesem Punkt durchgeführt, da die vorliegenden Normierungsinformationen anhand fusionierter Daten während des Trainings erstellt wurden und somit nur nach einer Fusion angewandt werden können. Anhand der fusionierten Daten und der Referenzvektoren wird eine Klassifizierung unter Verwendung des Minimum-Distanz-Klassifikators durchgeführt. Das Ergebnis wird durch eine Zurückweisungstechnik überprüft und bei Nichteindeutigkeit verworfen. Nicht verworfene Klassifikationen werden zusammen mit der minimalen Distanz, die zur Klassifizierung geführt hat, an das externe System versendet.

Der Funkchip muss ausgeschaltet sein, wenn der Sensorknoten neben momentan auszuführenden Berechnungen keine eintreffenden Nachrichten verarbeiten kann. Dies ist während der Kalibrierungsphase, der Samplebearbeitung einer Bewegung und der gesamten Musterbearbeitung der Fall. Wartet der Sensorknoten auf ein neues Ereignis und nimmt keine Bewegung wahr, so kann er auf eintreffende Nachrichten reagieren. Wie die Reaktion ausfällt, hängt vom Nachrichtentyp ab und wird in den entsprechenden folgenden Abschnitten beschrieben. Wenn Daten während der Trainings- und Erkennungsphase verteilt werden sollen, muss der Funkchip natürlich aktiviert sein.

Im Folgenden werden die aufgrund der Erweiterung der verteilten Mustererkennung neu hinzugefügten oder in ihrer Funktionsweise veränderten Komponenten und Phasen des Sensorknotensystems beschrieben. Es wird auch auf die Kommunikation zwischen den Knoten und mit dem externen System eingegangen.

#### 4.5.1 Merkmalsverteilung - Training

Berechnete Merkmalsvektoren werden während des Trainings nicht dauerhaft auf dem Sensorknoten gespeichert oder verarbeitet, sondern müssen an das externe System gesendet werden. Der externe Knoten, welcher über eine serielle Verbindung an das externe System angeschlossen ist, empfängt die Merkmalsvektoren und übermittelt sie an das externe System.

Es soll angenommen werden, dass sich ein beliebiges Ereignis im untersuchten Anwendungsfall dieser Arbeit nur maximal soweit ausbreitet bzw. von den Sensorknoten wahrgenommen wird, wie auch die Funkreichweite der Sensorknoten ist. Befindet sich der externe Knoten während des Trainings, wie in Abschnitt 4.4 gefordert, im Funkbereich dieser Knoten, so kann hier mit einer 1-hop Nachrichtenübertragung gearbeitet werden.

Hat ein Sensorknoten ein Ereignis wahrgenommen und die Berechnung des Merkmalsvektors abgeschlossen, wird dieser mittels einer PATTERN-Nachricht übermittelt. Der Dateninhalt besteht aus dem Typ der Nachricht und dem Merkmalsvektor. Die Nachricht kann entweder eine DISTRIBUTE-Nachricht darstellen, was bedeutet, dass der Merkmalsvektor direkt nach seiner Berechnung zum ersten Mal verteilt wird, oder eine RESPONSE-Nachricht sein, was die Reaktion auf den Erhalt eines Merkmalsvektors eines anderen Knotens darstellt.

Nachrichtentyp	Merkmalsvektor
----------------	----------------

**Abbildung 4.21: Datenpaket der PATTERN-Nachricht**

Theoretisch würde es genügen, wenn jeder Sensorknoten eine PATTERN-Nachricht verschickt und diese nur vom externen Knoten verarbeitet wird und von den anderen Knoten des Netzwerks ignoriert wird. Dadurch ist die Gefahr eines Paketverlustes sehr hoch, was sich während der Versuche als nicht praktikabel erwies. Es gibt verschiedene Möglichkeiten, damit umzugehen. Zwei davon sind:

- a. Der externe Knoten sendet bei Erhalt einer PATTERN-Nachricht eine Bestätigung an Senderknoten. Bei Nichterhalt einer Bestätigung wird die Nachricht nach einer gewissen Wartezeit nochmals gesendet.
- b. Die Nachricht wird zu gewissen Zeitpunkten mehrmals gesendet.

Die Untersuchung und Implementierung eines Protokolls, welches eine hohe Wahrscheinlichkeit einer fehlerfreien Datenübertragung garantiert, steht nicht im Mittelpunkt dieser Arbeit. Es soll eine Methode verwendet werden, die einfach umzusetzen ist, sich gut in den Verlauf der erweiterten verteilten Ereigniserkennung einfügt und zudem auch ein möglichst hohes Maß an Übertragungssicherheit bietet.

Möglichkeit a kann mit langen Verzögerungen einhergehen, wenn Bestätigungen verloren gehen und erneute Sendungen vorgenommen werden müssen. Möglichkeit b kann relativ einfach in den Ablauf der Merkmalsverteilung eingebaut werden und benötigt kein zusätzliche Verwaltung einer Bestätigungsroutine. Bei beiden Möglichkeiten ist keine sichere Garantie der erfolgreichen Übertragung gegeben, da die Anzahl der Sendungen aus Zeit- und Energiegründen nicht beliebig groß gewählt werden kann. Aufgrund der einfacheren Handhabbarkeit wird Möglichkeit b gewählt. Der Ablauf wird in Abbildung 4.22 illustriert. Merkmalsvektoren werden dabei nach Abschluss derer Berechnung zweifach mit einer Wartezeit von 1 Sekunde zwischen den Übermittlungen als DISTRIBUTE-Nachricht versendet (Abbildung 4.22 (a), einfache Übermittlung dargestellt). Wird von einem anderen Sensorknoten danach eine DISTRIBUTE-Nachricht empfangen, so wird der eigene Merkmalsvektor zweimal als RESPONSE-Nachricht versendet (Abbildung 4.22 (b)). So-

lange andere Knoten also ihre Merkmalsvektoren zum ersten Mal verschicken, wird der eigene Merkmalsvektor noch einmal übermittelt (Abbildung 4.22 (c)). Dabei entstehen keine zusätzlichen Wartezeiten, aber der Datenoverhead erhöht sich. Dieser zusätzliche Kommunikationsaufwand verbessert die Übertragungsfehlerrate.

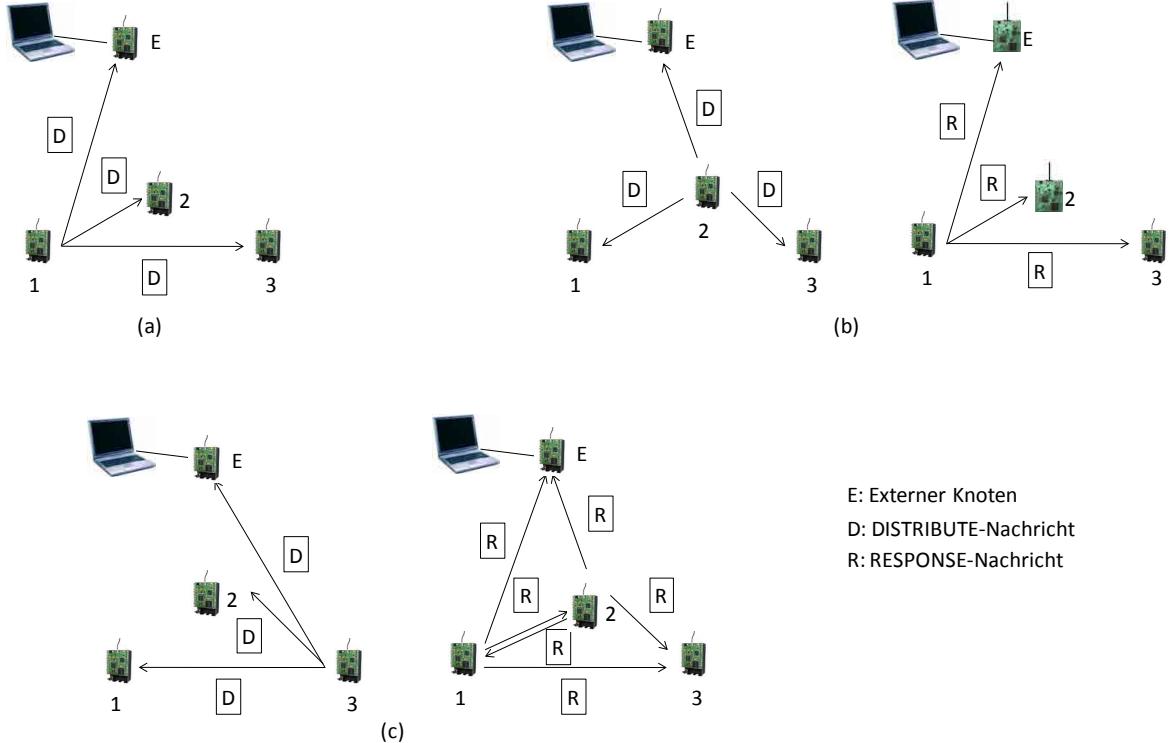


Abbildung 4.22 Verteilung von Merkmalsvektoren

Da von jedem Knoten zwei DISTRIBUTE-Nachrichten verschickt werden, aber nur auf eine davon reagiert werden soll, werden die Knoten-IDs der Sender in einem Ringspeicher vorgehalten. Dieser sollte möglichst so groß sein, um mindestens die Kennungen aller ein Ereignis wahrnehmenden Knoten aufzunehmen. In dem Ringspeicher sind 20 Einträge möglich. Sofern doch einmal ein Überlauf auftritt, werden die ältesten gespeicherten Kennungen zuerst gelöscht. Weiterhin wird der Ringspeicher aus Effizienzgründen vom letzten Eintrag rückwärts durchsucht. Wenn die Kennung des Senders in dem Ringspeicher vorliegt, muss der eigene Merkmalsvektor nicht versendet werden, ansonsten wird dies getan und die Kennung in den Speicher eingefügt.

Aufgrund der räumlichen Ausbreitung sowie des damit verbundenen unterschiedlichen örtlichen Auftretens eines Ereignisses nehmen die Sensorknoten dieses zu verschiedenen Zeitpunkten und verschieden lang wahr. Der Versand der berechneten Merkmalsvektoren der Knoten geschieht somit auch zu verschiedenen Zeiten. Der externe Knoten und die Sensorknoten müssen feststellen, wann alle Sensorknoten diesen Versand beendet haben. Dies ist dabei aus unterschiedlichen Gründen notwendig. Die Sensorknoten sollten erst danach wieder neue Sensordaten aufnehmen können, da mindestens ein Knoten vorher mit Berechnungen beschäftigt ist und somit die Aufnahme eines neuen Ereignisses nicht vorgenommen werden kann. Alle Sensorknoten verbleiben nach dem Versand ihrer Merkmalsvektoren in einem Wartezustand und lauschen nur auf

eintreffende PATTERN-Nachrichten. Der externe Knoten muss dem externen System signalisieren, dass die Daten eines Ereignisses vollständig sind. Damit werden die Ereignisse auf dem externen System automatisch voneinander abgegrenzt.

Zur Umsetzung wird auf den Sensorknoten und auf dem externen System ein Zeitgeber verwendet. Dieser wird nach jedem Empfang einer DISTRIBUTE- oder RESPONSE-Nachricht neu gestartet. Wenn innerhalb des Zeitfensters  $t_p$  keine neuen Nachrichten eintreffen, wird davon ausgegangen, dass alle Sensorknoten ihre Merkmalsvektoren übermittelt haben. Dazu muss das Fenster aber genügend groß gewählt werden, um mindestens die maximale Aufnahmedauer eines Ereignisses plus Puffer abzuwarten.  $t_p$  wurde in Versuchen empirisch ermittelt und auf 14 Sekunden festgelegt. Da alle Nachrichten per Broadcast verschickt werden, sollten die Sensorknoten und der externe Knoten das Ende des Zeitfensters etwa zeitgleich registrieren, so dass das ganze System zur etwa gleichen Zeit für die Aufnahme eines neuen Ereignisses bereit ist.

Sensorknoten, welche ein Ereignis nicht wahrnehmen, sollten nach Erhalt einer PATTERN-Nachricht keine Sensordaten mehr aufnehmen, da dies bedeutet, dass ein Ereignis vor einer gewissen Zeit ausgelöst wurde und noch immer keine Bewegung festgestellt wurde. Die Zeit bis zum Erhalt einer PATTERN-Nachricht stellt somit eine Begrenzung der Möglichkeit der Aufnahme eines Ereignisses dar. Diese Zeit wird mit einem Puffer reguliert, der zwischen den Abschluß der Merkmalsberechnung und das Versenden gelegt wird. Diese Sensorknoten warten dann auf das Ende der Merkmalsübermittlungen. Der einzige Unterschied zu den Knoten, welche das Ereignis wahrgenommen haben, ist, dass sie zwar Daten empfangen und den Zeitgeber aktualisieren, aber keine Merkmalsvektoren verschicken.

#### 4.5.2 Parameterverteilung

Das externe System verteilt mit Hilfe des externen Knotens nach Abschluss des Trainings die Klassifikationsparameter an die Knoten im Sensorsnetz. Dabei werden wie in Abschnitt 4.4.8 beschrieben, EXTERNALTRAINRESULTS-Nachrichten verschickt.

Erhält ein Sensorknoten eine EXTERNALTRAINRESULTS-Nachricht, so signalisiert dies das Ende des Trainings. Die Aufnahme von Sensordaten wird unterbrochen und auf weitere Nachrichten gewartet. Die Daten und der Datentyp jeder erhaltenen EXTERNALTRAINRESULTS-Nachricht werden gespeichert.

Da nicht alle Knoten des gesamten Netzwerks im Funkbereich des externen Knotens liegen, muss hier eine multi-hop Übertragung stattfinden, um alle Sensorknoten zu erreichen. Dies wird durch Fluten der Klassifikationsparameter durch das Sensorsnetz erreicht. Nachdem ein Sensorknoten alle Daten erhalten hat, werden diese per Broadcast weitergeleitet. Da es eine festgelegte Anzahl unterschiedlicher Datentypen gibt, lässt sich ermitteln, wann alle Parameter vom externen System erhalten wurden. Dies wird nach Erhalt jeder Nachricht geprüft. Sind die Daten vollständig, geht der Knoten in einen Sendezustand. Ab diesem Zeitpunkt werden die gespeicherten Klassifikationsparameter einmal weitergeleitet. Da dies von allen Sensorknoten getan wird, die alle Daten erhalten haben, erhält der Sensorknoten unter Umständen sehr viele Nachrichten, deren Daten er schon

verarbeitet hat. Um während des Sendens nicht gestört zu werden, werden während der gesamten Sendezeit alle eintreffenden Nachrichten ignoriert.

Wurden alle Nachrichten verschickt, geht der Sensorknoten in einen Wartezustand. Dieser Zustand bedeutet, dass auf das Ende des Versands der Klassifikationsparameter von allen Sensorknoten gewartet wird. Hier wird ähnlich wie bei der Merkmalsverteilung mit einem Zeitgeber gearbeitet. Trifft eine EXTERNALTRAINRESULTS-Nachricht ein, so wird der Zeitgeber neu gestartet. Treffen in dem folgenden Zeitfenster keine neuen Nachrichten ein, so kann angenommen werden, dass der Austausch der Parameter abgeschlossen ist. Alle Sensorknoten im Netzwerk sollten etwa zu selben Zeit das Ende des Zeitfensters registrieren. Danach startet die Erkennungsphase. Um die Trainingsphase von der Erkennungsphase abzugrenzen, wird ein Flag benutzt, welches nach Ablauf des Zeitgebers gesetzt wird.

Die Weiterleitung der Klassifikationsparameter durch alle Sensorknoten kostet zwar mehr Energie, als wenn dies nur durch bestimmte einzelne Knoten getan würde, hat aber verschiedene Vorteile. Es können Paketverluste ausgeglichen werden, da ein Knoten dieselben Daten von mehreren Knoten empfängt. Auch ist die Wahrscheinlichkeit höher, dass mehrere hops entfernte Knoten erreicht werden, wenn die Daten von vielen Knoten weitergeleitet werden. Gleichzeitig tritt kein Overhead durch ein Routing-Protokoll auf.

#### 4.5.3 Merkmalsverteilung – Erkennung

Die Merkmalsverteilung während des Trainings diente dazu, berechnete Merkmalsvektoren der Sensorknoten an das externe System zu senden. Im Fall der Erkennung soll eine Merkmalsfusion auf den Sensorknoten vorgenommen werden. Dazu müssen die Merkmalsvektoren zwischen den Knoten ausgetauscht werden. Auch hier soll wieder angenommen werden, dass eine 1-hop Übertragung für das Erreichen aller an einem Ereignis beteiligten Sensorknoten ausreichend ist.

Wie schon bei der Merkmalsverteilung während des Trainings werden PATTERN-Nachrichten verwendet, die im Datenpaket den Nachrichtentyp und den Merkmalsvektor enthalten. Durch den Nachrichtentyp kann zwischen DISTRIBUTE- und RESPONSE-Nachrichten unterschieden werden. Der Verlauf der Verteilung ist identisch, die Behandlung der Nachrichten unterscheidet sich jedoch stark von denen des Trainings.

Die Bedingung zur Ausführung der Erkennung eines Ereignisses ist, dass ein Sensorknoten ein zu erkennendes Ereignis selbst wahrnimmt. Jeder Knoten nimmt für die Erkennung an, dass er sich im Zentrum des Ereignisses befindet. Ein Knoten im Zentrum des Ereignisses sollte dieses aufgrund der hohen Intensität immer und aufgrund der räumlichen Lage auch mit geringer zeitlicher Verzögerung wahrnehmen. Erhält ein Sensorknoten  $N_1$  jedoch eine PATTERN-Nachricht eines anderen Knotens  $N_2$ , so bedeutet dies, dass Knoten  $N_2$  ein Ereignis wahrgenommen hat, Knoten  $N_1$  aber nicht oder noch nicht. Die Annahme für die Lage im Zentrum des Ereignisses ist somit verletzt und der Knoten kann das Ereignis nicht korrekt erkennen. Die Erkennung darf demnach nicht ausgeführt werden. Durch die Bedingung werden die Sensorknoten von der Erkennung ausgeschlossen, die zwar in Funkreichweite der Knoten liegen, die das Ereignis wahrnehmen, aber selbst keine Beschleunigung wahrnehmen oder mit so einer hohen zeitlichen Verzöge-

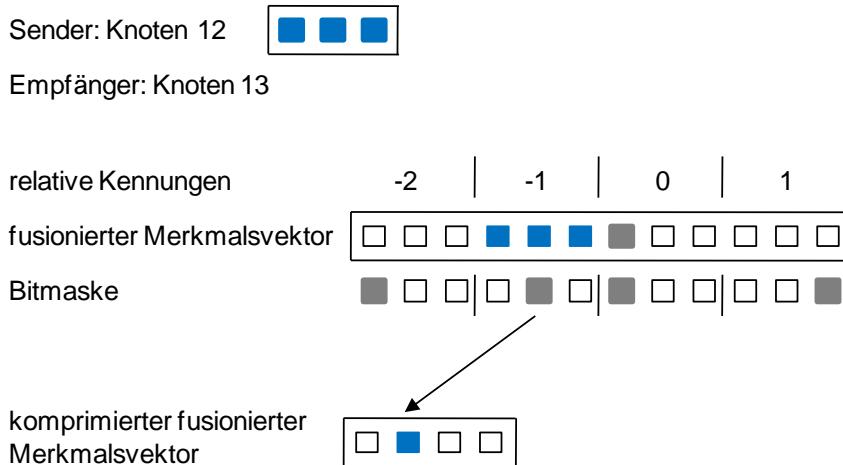
rung, dass die Wahrnehmung nicht mehr dem ursprünglichen Ereignis zugeordnet werden kann.

Zur Unterscheidung, ob ein Sensorknoten eine Erkennung durchführen darf oder nicht, gibt es zwei Zustände. Ein Wartezustand wird dann angenommen, wenn keine Erkennung durchgeführt werden darf. Es werden dann keine Sensordaten mehr aufgenommen, sondern nur auf das Ende der Merkmalsverteilung durch Zeitgeberaktualisierungen bei Erhalt von PATTERN-Nachrichten gewartet. Da keine Erkennung vorgenommen wird, müssen erhaltene Merkmalsvektoren auch nicht ausgewertet werden. Der andere Zustand ist der Erkennungszustand und wird dann angenommen, wenn die Berechnung der Merkmalsvektoren abgeschlossen ist.

Nach der Merkmalsvektorberechnung wird der Merkmalsvektor zweifach als DISTRIBU-TE-Nachricht versendet. Ein Zeitgeber wird gestartet und auf weitere PATTERN-Nachrichten gewartet.

Merkmalsvektoren des eigenen Knotens und anderer Sensorknoten werden anhand der gespeicherten relativen Knotenkennungen und ausgewählter Merkmale fusioniert. Jeder Sensorknoten soll die Merkmalsvektoren relativ zu der eigenen Kennung identisch zum Training anordnen. Die Anordnung des Trainings ist durch die gespeicherten relativen Kennungen definiert. Diese wurden anhand eines Bezugspunktes berechnet, der Kennung des zentralen Knotens im Training. Wenn jetzt jeder Sensorknoten annimmt, dass er der zentrale Knoten ist und die Berechnung identisch zu der des Trainings vorgenommen wird, erhält man relative Kennungen, die mit denen des Trainings vergleichbar sind. Während des Trainings wurde die Kennung des zentralen Knotens von den ausgewählten Knoten subtrahiert. Wird jetzt ein Merkmalsvektor empfangen, wird die eigene Knotenkennung von der Knotenkennung des Senders abgezogen. Die gespeicherte Knotenliste wird daraufhin durchlaufen und die berechnete relative Kennung mit den gespeicherten verglichen. Ergibt sich eine Übereinstimmung, so wird der Merkmalsvektor anhand der Übereinstimmungsposition angeordnet, wie in Abbildung 4.23 nachvollziehbar ist. Die Kennung des eigenen Knotens ist 13, und von Knoten 12 wird ein Merkmalsvektor empfangen. Die relative Kennung ist somit -1 und an Position 2 zu finden.

Neben der Anordnung der Merkmalsvektoren anhand der relativen Kennungen muss auch eine Auswahl von Werten anhand der während des Trainings ausgewählten Merkmale durchgeführt werden. Der entstehende komprimierte fusionierte Merkmalsvektor enthält nur so viele Werte, wie auch Merkmale insgesamt ausgewählt wurden. Ein empfangener Merkmalsvektor wird entsprechend der gefundenen Position in der Knotenliste verarbeitet. Für die Auswahl der Werte des Merkmalsvektors wird der Bereich der Bitmaske entsprechend der Position nach ausgewählten Merkmalen durchsucht. Die Anordnung der Werte im komprimierten fusionierten Merkmalsvektor ergibt sich aus der Verteilung der ausgewählten Merkmale der Bitmaske. In Abbildung 4.23 ist zu sehen, dass ein Wert ausgewählt und an zweiter Stelle angeordnet wird, da ein Merkmal in dem untersuchten Bereich der Bitmaske insgesamt an zweiter Stelle von vier zu finden ist.



**Abbildung 4.23:** Verteilung eines Merkmalsvektors

Auf dem Sensorknoten wird nur der komprimierte fusionierte Merkmalsvektor gespeichert, da dort alle für die Klassifikation relevanten Daten zu finden sind. Empfangene Merkmalsvektoren werden nach Auswahl und Anordnung der enthaltenen Werte verworfen. Aufgrund des möglichen mehrmaligen Empfangens eines Merkmalsvektors vom gleichen Knoten muss wie bei der Merkmalsverteilung beim Training eine Möglichkeit eingebaut werden, um eine mehrfache Behandlung zu vermeiden. Auch hier kann dazu der schon in Abschnitt 4.5.1 beschriebene Ringspeicher verwendet werden.

Weiterhin zu beachten ist, dass nicht nur Merkmalsvektoren von anderen Sensorknoten, sondern auch der eigene Merkmalsvektor behandelt werden muss. Dazu wird dieser, bevor er verschickt wird, einmal ausgewertet.

Der Empfang eines **DISTRIBUTE**-Pakets bedeutet, dass ein Sensorknoten mit der Berechnung eines Merkmalsvektors fertig ist und diesen an andere Sensorknoten verteilen möchte. Bei der Merkmalsverteilung während des Trainings wurde auf eine **DISTRIBUTE**-Nachricht mit zwei **RESPONSE**-Nachrichten geantwortet, um eine Mehrfachsendung des eigenen Merkmalsvektors in den Verlauf der Verteilung einzubauen. Im Fall der Erkennung soll dies auch so durchgeführt werden, allerdings aus einem anderen Grund. Verschickt ein Sensorknoten  $N_1$  seinen Merkmalsvektor, kann ein anderer Knoten  $N_2$  noch mit der Berechnung der eigenen Merkmale beschäftigt sein. Der parallele Empfang von Paketen jeglicher Art ist dabei nicht möglich. Dieser Knoten empfängt also den Merkmalsvektor nicht. Der Merkmalsvektor muss also zu einem späteren Zeitpunkt noch einmal verschickt werden, wenn Knoten  $N_2$  Pakete empfangen kann. Das ist genau dann, wenn dieser seinen eigenen Merkmalsvektor übermittelt hat. Empfängt Knoten  $N_1$  also eine **DISTRIBUTE**-Nachricht, muss eine Antwort mit dem eigenen Merkmalsvektor an Knoten  $N_2$  geschickt werden, damit dieser auch den Merkmalsvektor von Sensorknoten  $N_1$  auswerten kann. Allgemein gesagt, erhält ein Sensorknoten eine **DISTRIBUTE**-Nachricht und hat selbst schon den eigenen Merkmalsvektor verschickt, so sendet er als Antwort darauf den Merkmalsvektor noch einmal an den Sender der **DISTRIBUTE**-Nachricht. Diese **RESPONSE**-Nachricht wird zweifach gesendet, um Paketverlusten vorzubeugen. Durch dieses Verfahren ist sichergestellt, dass alle Sensorknoten im Netzwerk, welche ein

Ereignis wahrnehmen, die Daten aller anderen Knoten unabhängig von dem Zeitpunkt der Beendigung der Merkmalsberechnungen empfangen können.

Eine Möglichkeit der Einsparung von Funkübertragungen lässt sich dadurch erreichen, in dem die Behandlung empfangener Merkmalsvektoren betrachtet wird. Diese werden nur dann verwendet, wenn die relative Knotenkennung, welche sich aus Sender- und Empfängererkennung berechnet, in der Liste der gespeicherten relativen Knotenkennungen zu finden ist. RESPONSE-Nachrichten müssen genau dann nicht gesendet werden, wenn der Merkmalsvektor sowieso nicht vom Empfänger betrachtet wird. Die Berechnung der relativen Knotenkennung kann vor dem Senden der Antworten auf dem Sensorknoten, welcher eine DISTRIBUTE-Nachricht empfangen hat, simuliert werden. Dazu wird die Berechnung umgekehrt, die Kennung des Senders der DISTRIBUTE-Nachricht wird von der eigenen Kennung abgezogen. Ist die entstehende relative Kennung in der Liste der gespeicherten relativen Kennungen zu finden, dann muss der Merkmalsvektor an den Sender der DISTRIBUTE-Nachricht übermittelt werden, da dieser die Daten für die Klassifikation verwendet.

Durch die Verwendung von DISTRIBUTE- und RESPONSE-Nachrichten wäre es theoretisch möglich, dass die Sensorknoten nur die DISTRIBUTE-Nachrichten oder die RESPONSE-Nachrichten, welche als Antwort auf die eigene Merkmalsverteilung verschickt wird, verarbeiten. Eine RESPONSE-Nachricht müsste dazu noch eine Information über den Zielknoten enthalten. Dabei kann es trotz doppelter Übermittlung dazu kommen, dass manche Sensorknoten aufgrund von Paketverlusten nicht alle möglichen Merkmalsvektoren bekommen. Um dem Nichtempfang aller möglichen Merkmalsvektoren vorzubeugen, werden alle PATTERN-Nachrichten beim Empfang betrachtet. Durch die Verwendung des Ringspeichers können Duplikate erkannt und somit mit wenig Aufwand entschieden werden, ob der erhaltene Merkmalsvektor weiter verarbeitet werden soll. Ohne dieses Vorgehen würde ein Sensorknoten einen Merkmalsvektor maximal zweimal erhalten. Wenn auch Antworten auf DISTRIBUTE-Nachrichten von anderen Sensorknoten betrachtet werden, kann diese Zahl deutlich ansteigen, ohne zusätzliche Funkübertragungen realisieren zu müssen. Durch die erhöhte Anzahl betrachteter Merkmalsvektoren steigt die Wahrscheinlichkeit, alle möglichen Merkmalsvektoren der anderen Sensorknoten zu empfangen.

Bei der bisherigen verteilten Mustererkennung mussten auf den Sensorknoten die Merkmalsvektoren aller registrierten Knoten vorliegen. Diese Bedingung ist hier nicht möglich, da je nach Lage der wahrnehmenden Knoten unterschiedlich viele relevante Merkmalsvektoren erhalten werden. Im komprimierten fusionierten Merkmalsvektor können somit fehlende Werte entstehen. Das kann auch auf dem zentralen Knoten passieren, da ein Knoten  $N$  während des Trainings für maximal eine Klasse nicht ausreichend Daten liefern darf und dessen Daten der anderen Klassen trotzdem als relevant betrachtet werden. Wenn dieser Knoten  $N$  während der Erkennungsphase für diese Klasse keine Daten liefert, entstehen somit fehlende Werte auf dem zentralen Knoten, sofern von dem Knoten  $N$  Merkmale ausgewählt wurden. Dem komprimierten fusionierten Merkmalsvektor müssen zum Umgang mit den fehlenden Werten vor der Erkennung eines unbekannten Ereignisses Werte zugewiesen werden, welche einem Ruhezustand des Sensorknotens entsprechen.

chen, da in dem Fall angenommen werden kann, dass die fehlenden Werte durch fehlende Wahrnehmung des Ereignisses der entsprechenden Sensorknoten entstanden sind. Die Werte werden auch während der Datenfusion des Trainings für die Knoten verwendet, welche keine oder nicht genügend Daten liefern. Aufgrund der Merkmalsabhängigkeit sollen sie Abschnitt 5.3 besprochen werden.

Der Abschluss der Merkmalsverteilung wird wie während des Trainings registriert. Sensorknoten aktualisieren bei Erhalt einer PATTERN-Nachricht ihre Zeitgeber. Der Ablauf der Zeitgeber sollte damit bei allen Knoten etwa zeitgleich erfolgen. Knoten, die ein Ereignis wahrgenommen haben, beginnen mit der Merkmalsnormierung. Die anderen Sensorknoten kehren zur Sensordatenaufnahme zurück.

Die Merkmalsnormierung wird anhand der gespeicherten Normierungsinformationen an dem komprimierten fusionierten Merkmalsvektor durchgeführt, da nur für die während des Trainings ausgewählten Merkmale minimale und maximale Werte vorliegen.

Nach der Merkmalsnormierung wird die Klassifizierung ausgeführt.

#### 4.5.4 Klassifizierung

Die Klassifizierung erstellt aus dem fusionierten Merkmalsvektor, welcher als Ergebnis der Merkmalsverteilung und -normierung entstanden ist, eine Klassifikation, die als gültig angesehen oder verworfen wird.

Dafür werden zwei Phasen ausgeführt. In der ersten Phase wird mit Hilfe des bestehenden Minimum-Distanz-Klassifikators die minimale Distanz über alle Klassen vom fusionierten Merkmalsvektor zu allen gespeicherten Referenzvektoren bestimmt. In der zweiten Phase wird je nach verwendeter Zurückweisungstechnik eine Überprüfung der vorliegenden Daten vorgenommen. Die Distanzzurückweisung stellt fest, ob die Distanz innerhalb der Grenze der zugehörigen Klasse liegt. Die Merkmalszurückweisung verifiziert, ob die Werte des fusionierten Merkmalsvektors innerhalb der erlaubten Merkmalsgrenzen liegen. Werden die Daten zurückgewiesen, dann wird die Klassifikation verworfen. Im anderen Fall werden die Klassifikationsergebnisse an das externe System verschickt, welches entweder einen Alarm auslöst oder die Daten zur Weiterverarbeitung nutzt.

Damit ist die Klassifikation eines Ereignisses innerhalb des Sensornetzes ohne Zuhilfenahme des externen Systems abgeschlossen.

#### 4.5.5 Klassifikationsverteilung

Gültige Klassifikationen werden zusammen mit der minimalen Distanz an das externe System übermittelt. Die minimale Distanz wird für eine eventuelle Klassifikationsfusion auf dem externen System benötigt, sofern für ein Ereignis Klassifikationen von mehreren Sensorknoten eintreffen. Der externe Knoten liegt nicht im Funkbereich aller Knoten des Netzwerks. Da aber alle Knoten eine Klassifizierung bei Wahrnehmung eines Ereignisses durchführen können, muss hier eine multi-hop Übertragung umgesetzt werden. Dabei genügt es, dass der externe Knoten, welcher die Klassifikationsdaten per Funk empfängt und an das externe System weiterleitet, im Funkbereich eines Sensorknotens liegt, wie in Abschnitt 4.4 erwähnt. Zur Umsetzung der multi-hop Übertragung wird das Sensornetz mit den Klassifikationsdaten geflutet.

Die Daten werden mit Hilfe einer RECOGNITIONRESULTS-Nachricht übertragen. Wie in Abbildung 4.24 zu sehen ist, enthält diese neben den Klassifikationsdaten zusätzlich Informationen, die für die Weiterleitung der Nachricht im Netzwerk erforderlich sind.

Knoten-ID	Sendezeit	Klassifikation	Distanz
-----------	-----------	----------------	---------

Abbildung 4.24 Datenpaket der RECOGNITIONRESULTS-Nachricht

Empfängt ein Sensorknoten eine RECOGNITIONRESULTS-Nachricht, so wird diese weiter versendet. Um Endlosschleifen zu vermeiden, die dabei entstehen würden, soll jede Nachricht nur einmal weitergeleitet werden. Dazu wird der schon während der Merkmalsverteilung des Trainings und der Erkennung eingesetzte Ringspeicher verwendet. Da von einem Sensorknoten nicht entschieden werden kann, wann die Klassifikationsdaten eines Ereignisses vollständig übermittelt worden sind, da das Ereignis an einer beliebigen Stelle im Netzwerk auftreten kann, reicht es nicht aus, nur eine Knotenkennung im Ringspeicher vorzuhalten, da dieser nicht zu einem bestimmten Zeitpunkt gelöscht werden kann. Zur Unterscheidung zweier verschiedener RECOGNITIONRESULTS-Nachrichten eines Sensorknotens benötigt man somit eine zusätzliche Information. Diese stellt die anfängliche Sendezeit der Nachricht dar. Des Weiteren muss die Knotenkennung des ursprünglichen Senders der Nachricht in den Daten mitgeliefert, da diese bei einer Weiterleitung sonst verloren geht. Mit Hilfe der Knoten-ID und der Sendezeit im Datenteil der Nachricht lassen sich mehrfach empfangene Nachrichten durch Überprüfung des Ringspeichers identifizieren.

Die Weiterleitung der Nachrichten durch alle Pakete hat den Vorteil der hohen Wahrscheinlichkeit, dass diese auch bei dem externen Knoten ankommen. Wurde eine RECOGNITIONRESULTS-Nachricht erst einmal gesendet und von mehreren Knoten empfangen und weitergeleitet, ist diese hohe Wahrscheinlichkeit gegeben. Der kritische Punkt ist das Versenden der Nachricht vom ursprünglichen Knoten aus. Zur Sicherheit wird hier eine Nachricht zweifach versendet. Werden diese beiden Nachrichten durch eine Störung oder andere Ursachen von keinem anderen Sensorknoten empfangen, so gelangen die Klassifikationsdaten nicht zum externen Knoten. Deshalb wird zur weiteren Sicherung nach einigen Sekunden überprüft, ob die eigens versendete Nachricht als weitergeleitete Nachricht eines anderen Knoten in dieser Zeit empfangen wurde. Ist dies nicht der Fall, so wird die Nachricht noch einmal verschickt.

## 5 Anwendungsspezifische Anpassung und Erweiterung

Für das Sensorknotensystem lassen sich keine allgemeinen Parametrisierungen definieren, welche für alle möglichen Anwendungsfälle passend sind. Für jeden Fall muss versucht werden, die Werte zu finden, mit denen eine bestmögliche Erkennungsrate erzielt werden kann. Der Charakter mancher Anwendungsfälle bietet die Möglichkeit, Erweiterungen am System vorzunehmen, um durch die erhöhte Anpassung Verbesserungen zu erreichen.

Im weiteren Verlauf des Kapitels soll zuerst der Anwendungsfall beschrieben werden, auf den das in der vorliegenden Arbeit entwickelte System angewandt werden soll. Weiterhin soll dieser mit dem grundlegenden Anwendungsfall verglichen werden. Aufgrund festgestellter Unterschiede und Beobachtungen werden dann Anpassungen und Erweiterungen an bestehenden Komponenten des Sensorknotensystems vorgenommen.

### 5.1 Beschreibung des Anwendungsfalls

Der Anwendungsfall, der in dieser Arbeit betrachtet werden soll, ist die Erkennung sicherheitsrelevanter Ereignisse, welche an einem Bauzaun verursacht werden.

Ein Bauzaun besteht aus mehreren Bauzaunelementen. Jedes Element hat eine Abmessung von etwa 3,50 Meter x 2,00 Meter. An den beiden Seiten jedes Zaunelements befindet sich ein vertikales Rohr mit einem Hohlraum, an dem die Maschen befestigt sind. An jedem Element soll ein Sensorknoten angebracht werden. Dies kann an verschiedenen Stellen geschehen. Eine Anforderung dabei ist, dass die Knoten ihre Ausrichtung am Zaun im Verlauf der Zeit nicht ändern. Dies würde ständige Neukalibrierungen nach sich ziehen. Des Weiteren sollten die Knoten so montiert sein, wie sie vermutlich in der Realität befestigt würden. Nach [Wit07] eignet sich eine zum Schutz vor Umwelteinflüssen eine Anbringung im Hohlraum eines der beiden Rohre eines Zaunelements. Alle Sensorknoten werden aufgrund der einfacheren Befestigungsmöglichkeit nicht im Hohlraum, sondern außen am linken Rohr eines jeden Zaunelements auf die gleiche wie hier beschriebene Weise angebracht. Jeder Sensorknoten wird in eine Überputzdose eingebettet und diese mit Hilfe zweier Kabelbinder am Rohr festgemacht, so dass die Unterseite des Sensorknotens parallel zum Boden und die Vorderseite parallel zur Zaunfläche ausgerichtet sind und die Bedienelemente zur Öffnung der Dose vom Zaun weg zeigen. Die Dosenoberseite ist etwa 25 cm vom oberen Ende des Rohres entfernt. Die Befestigung eines Sensorknotens ist in Abbildung 5.1 dargestellt.



**Abbildung 5.1: Befestigung eines Sensorknotens am Bauzaun**

Die einzelnen Zaunelemente sind durch Koppelemente miteinander verbunden. Die Rohre der Zaunelemente sind in Betonsockel eingefasst. Diese typische Bauzaunkonstruktion wurde von jeweiligen Bauunternehmen montiert. Die Konstruktion erlaubt aufgrund des Spielraums in den Betonsockeln eine Schwankung des gesamten Elements senkrecht zur Laufrichtung des Bauzauns, wie in Abbildung 5.2 skizziert ist.



**Abbildung 5.2: Schwankung der Zaunelemente**

Am obersten Punkt des Zaunelements ist die Auslenkung maximal, am untersten Punkt gleich 0. Entsteht an einem Element eine Bewegung, so überträgt sich diese aufgrund der Verbindung auf benachbarte Elemente. Je weiter ein Zaunelement vom ursprünglichen Element entfernt ist, umso gedämpfter tritt die Bewegung dort auf.

Das bisherige Sensorknotensystem wurde zur Bewegungserkennung geometrischer Figuren verwendet. Die Rückseite der Sensorknoten wurde dabei parallel zum Boden ausgerichtet, so dass die Vorderseite nach oben zeigt. Die Figuren wurden auf einer glatten Oberfläche nachgezeichnet. Die Beschleunigungsdaten der X- und Y-Achse wurden für die Ereignisauswertung verwendet.

## 5.2 Anpassungen

Aufgrund der veränderten Lage und anderer Bewegungsformen der Sensorknoten im Gegensatz zum bisherigen Anwendungsfall ist eine Reihe von Anpassungen notwendig. Diese sollen im Folgenden beschrieben werden.

### 5.2.1 Beschleunigungsachsen

Der Beschleunigungssensor kann Beschleunigungsdaten der X-, Y- und Z-Achse messen. Die Lage der Achsen ist abhängig von der Ausrichtung des Sensorknotens. Dies ist Abbildung 5.3 in dargestellt.

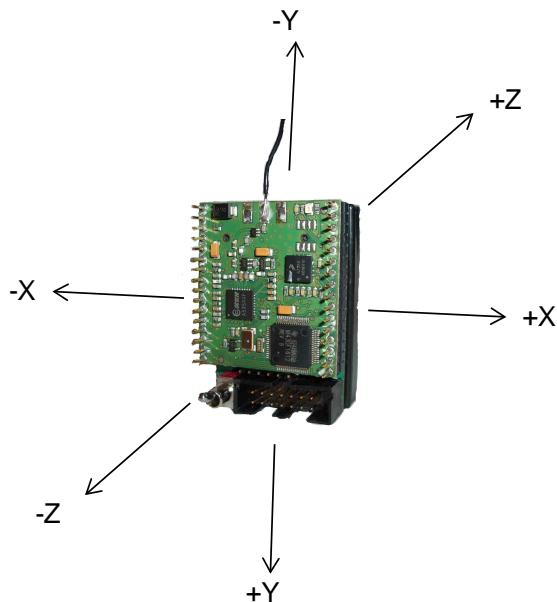
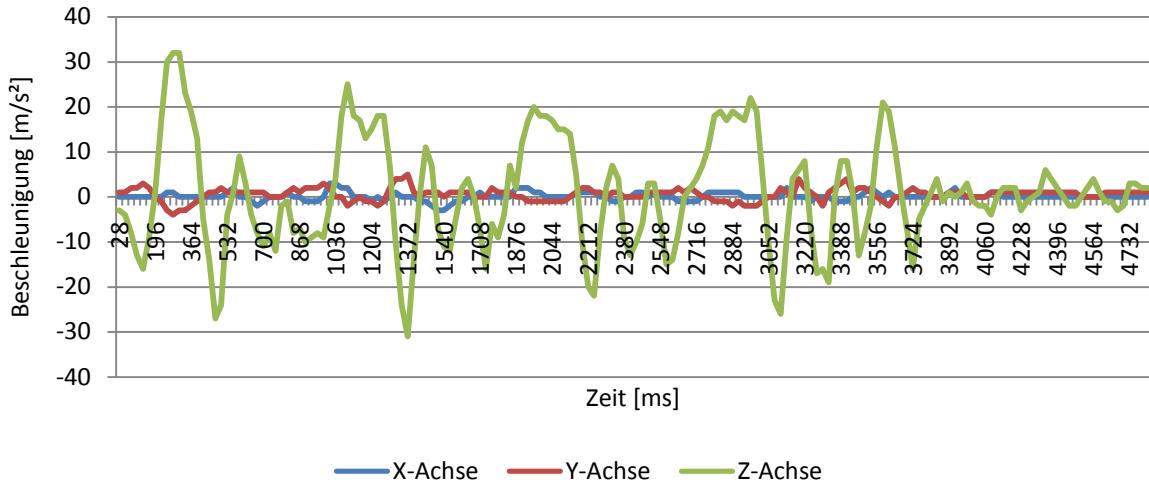


Abbildung 5.3 Beschleunigungsachsen eines Sensorknotens

Betrachtet man die Lageveränderung eines Sensorknotens bei der Bewegung des Bauzauns, so bewegt sich der Knoten hauptsächlich auf der positiven und negativen Z-Achse, geringfügig auf der Y-Achse und sollte keine Beschleunigung auf der X-Achse erfahren. Wenn keine oder nur geringe Beschleunigungen auf der X- und Y-Achse gemessen werden, dann ist eine Verwendung dieser Werte nutzlos und nicht aussagekräftig. Es reicht aus, die Daten, welche aufgrund der Beschleunigung auf der Z-Achse gemessen werden, zu verwenden.

Um die Beobachtung zu verifizieren, wird ein Vergleich der Beschleunigung auf den Achsen in der Situation des Anwendungsfalles durchgeführt.

Abbildung 5.4 zeigt vorverarbeitete, nicht normierte Beschleunigungswerte für die drei Achsen, die während der Bewegung des Bauzauns aufgetreten sind. Das dem die Daten liefernden Sensorknoten zugehörige Bauzaunelement wurde dabei mehrfach abwechselnd vor und zurück bewegt. Während bei den Daten der Z-Achse hohe Ausschläge in positiver und negativer Richtung und damit hohe Beschleunigungen zu verzeichnen sind, bewegen sich die Werte der X- und Y-Achse um den Nullpunkt, was nur geringen, kaum messbaren Beschleunigungen entspricht.



**Abbildung 5.4: Beschleunigungsdaten der X-, Y- und Z-Achse**

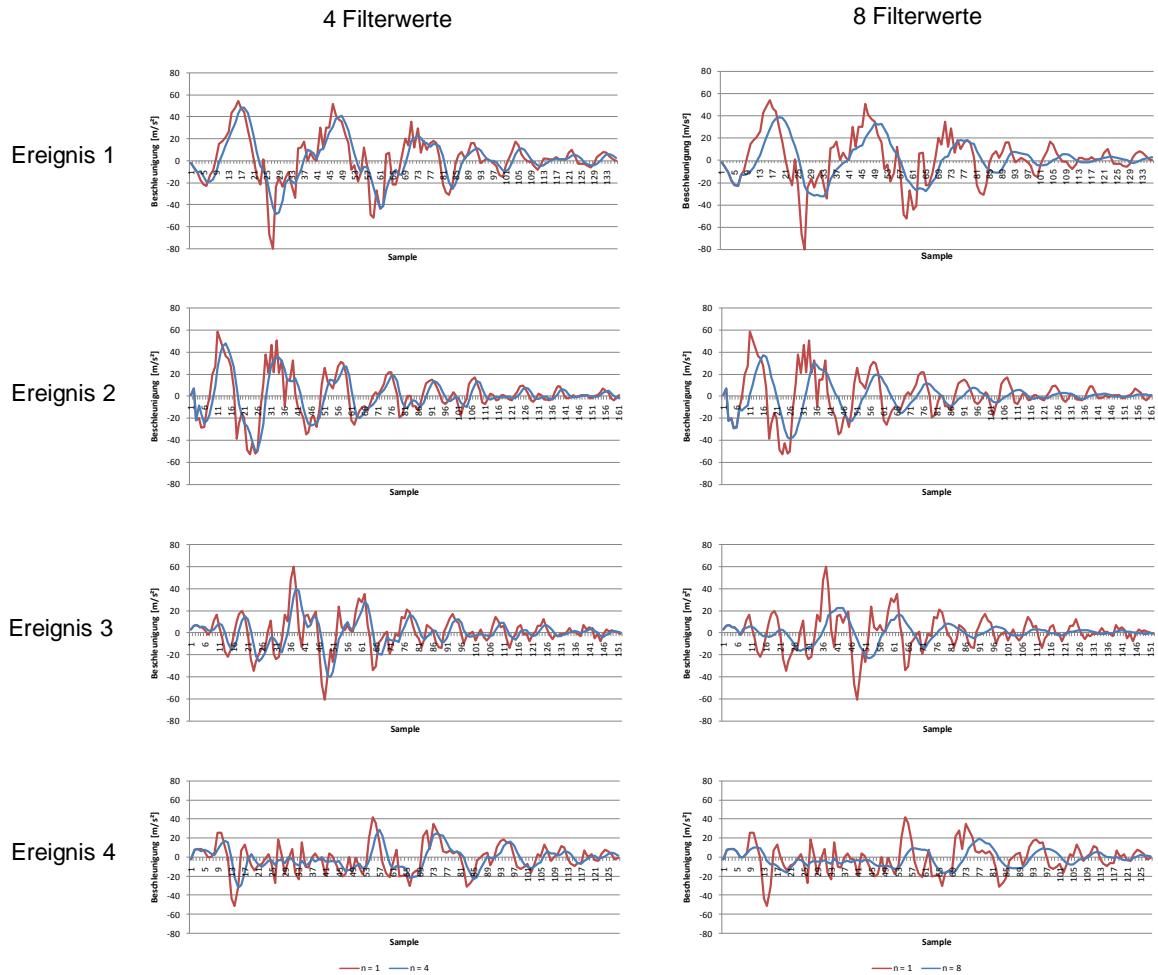
Wie gezeigt, misst der Beschleunigungssensor nur auf der Z-Achse hohe Beschleunigungswerte. Daher sollen auch nur diese Daten für Merkmalsberechnungen verwendet werden. Im Vergleich zu dem dieser Arbeit zugrunde liegenden System stehen weniger Daten zur Verfügung, da nur Beschleunigungsdaten von einer Achse verwendet werden. Dies stellt eine schwierigere Voraussetzung dar und lässt eine schlechtere Erkennungsrate erwarten.

### 5.2.2 Filtereinstellung

Neben dem eingebauten Hardwaretiefpassfilter des Beschleunigungssensors wird der moving average Tiefpassfilter während der Vorverarbeitung eingesetzt, um die aufgenommenen Zeitreihen zu glätten. Die Stärke der Glättung kann durch die Anzahl der Elemente variiert werden, die für die Neuberechnung eines Wertes verwendet werden. Es ist sinnvoll, die Anzahl für jeden Anwendungsfall neu festzulegen und zu untersuchen, da die Beschleunigungsdaten je nach Bewegungsform der Sensorknoten unterschiedlich stark geglättet werden müssen.

Für die Bestimmung eines möglichst guten Filterwerts muss zuerst die Auswirkung dieses Wertes betrachtet werden. Ein zu niedriger Wert glättet eine Kurve kaum, so dass danach immer noch starkes Rauschen vorherrscht. Ein zu hoher Wert kann die Kurve so stark glätten und somit dazu führen, dass sich Kurven von Ereignissen unterschiedlicher Klassen weniger gut voneinander unterscheiden. Es muss ein Wert dazwischen gefunden werden, so dass genügend Rauschen entfernt und die Unterscheidbarkeit der verschiedenen Klassen gewahrt bleibt.

Dafür werden Beschleunigungsdaten von vier Klassen mit unterschiedlichen Filtereinstellungen miteinander verglichen. In den Diagrammen, die in Abbildung 5.5 zu sehen sind, steht die blaue Kurve für ungefilterte Beschleunigungsdaten. Die rote Kurve in den Diagrammen auf der linken Seite steht für eine Filterung mit 4 Werten, auf der rechten Seite für 8 verwendete Werte.



**Abbildung 5.5: Verschiedene Filterung von Beschleunigungsdaten**

Wie man erkennen kann, werden durch einen Filter mit 4 Werten einige Beschleunigungsspitzen entfernt, aber trotzdem bleibt der ursprüngliche, für jede Klasse charakteristische Kurvenlauf gut erhalten. Die Filterung mit 8 Werten glättet zu viele Beschleunigungsspitzen, so dass sich die Kurven der einzelnen Klassen zu sehr aneinander annähern und weniger gut unterscheidbar sind. Eine Untersuchung der Filterung mit Werten zwischen 4 und 8 zeigte ebenso eine zu hohe Glättung der Spitzen, wie sie auch für 8 Werte zu beobachten ist. Bei Werten unter 4 zeigte sich Anwendung des Filters ein zu starkes Rauschen der Daten.

Als Folgerung daraus sollen für den vorliegenden Anwendungsfall 4 Filterelemente für den moving average Filter verwendet werden.

### 5.2.3 Segmentierung

Die bestehende Segmentierung stellt unter anderem fest, ob ein Musterende bis zum Erreichen der Speichergrenze für Rohdaten aufgetreten ist. Ist dies nicht der Fall, so wird das Muster als zu lang bewertet und verworfen. Für das Nachzeichnen von geometrischen Figuren ist das Verfahren anwendbar, da die Zeitdauer vom Beginn bis zum Ende der Bewegung relativ genau abgeschätzt werden kann.

Im jetzigen Anwendungsfall hört die Bewegung von Bauzaunelementen nach dem Auftreten eines Ereignisses nicht abrupt auf, diese schwingen noch eine gewisse Zeitdauer

nach. Man kann zwar abschätzen, wie lange die zu trainierenden Ereignisse ungefähr dauern, aber für das Nachschwingen ist das schwer möglich. Es ist natürlich möglich, die Zeitdauer der Ereignisse mit einem Puffer zu versehen, so dass der Bauzaun wieder in Ruhelage gelangen kann, jedoch sollten die Beschleunigungsdaten, die während des Nachschwingens auftreten, möglichst nicht verwendet werden, da diese nur eine geringe Aussagekraft über ein Ereignis haben.

Um nur die aussagekräftigen Daten aufzunehmen, muss vor dem Training die maximale Dauer über alle Ereignisse ermittelt werden. Nur über diese Dauer sollen Beschleunigungsdaten höchstens gespeichert werden. Das Samplingintervall wird an diese Dauer und die maximale Anzahl speicherbarer Rohdaten angepasst. Wenn von einem Muster kein Ende bis Erreichen der maximalen Dauer registriert wurde, dann wird die Datenaufnahme abgebrochen, das Muster aber trotzdem für gültig erklärt. Die Beschleunigungsdaten, die danach auftreten, sollen nicht betrachtet werden. Der Bauzaun muss allerdings wieder in Ruhelage sein, wenn das nächste Ereignis wahrgenommen werden soll. Durch die Wartezeiten während der Merkmalsverteilung soll dies als gegeben angenommen werden.

Durch das beschriebene Verfahren ist es nicht möglich, die Bewegungsdauer insgesamt zu messen, die ein Ereignis verursacht hat. Diese kann also nicht als Merkmal verwendet werden. Sollte dies erforderlich sein, können auch nach dem Erreichen der maximalen Dauer weiter Daten aufgenommen werden, ohne diese dauerhaft zu speichern, bis das Ende der Bewegung wahrgenommen wird. Die Menge der vom Beginn der Musters bis dahin aufgenommenen Daten entspricht dann der gesamten Bewegungsdauer. Der Nachteil hierbei ist, dass bis zum Musterende gewartet werden muss, bis mit der Musterbearbeitung fortgefahren werden kann.

## 5.2.4 Sonstige Anpassungen

Die Sensitivität des Beschleunigungssensors muss so eingestellt werden, dass Ereignisse ausreichend lang, aber Nachschwingen nur kurz wahrgenommen wird, so dass die Wahrnehmung kurzer Ereignisse nicht zu lang nach Ende ihres Auftretens andauert. Ein Wert von 2g erweist sich hier als passend, wie durch Abbildung 5.6 gezeigt wird. Es wurde ein Schütteln mit mehrfacher Vor- und Zurückbewegung des Bauzauns durchgeführt. Wie man sieht, übertragen sich die Auslenkungen des Bauzauns auf die Beschleunigungsdaten. Weiterhin ist eine gute Unterscheidbarkeit zwischen dem Schütteln und dem Ausschwingen des Bauzauns (etwa ab Sample 110) zu beobachten.

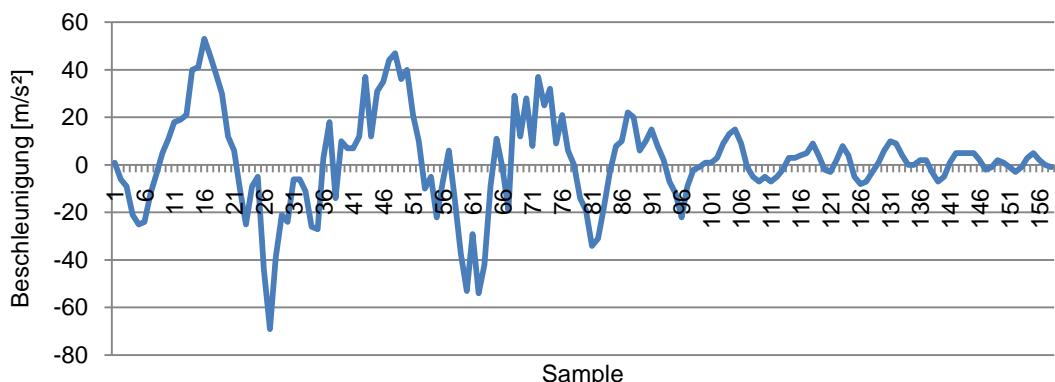


Abbildung 5.6: Verhalten der Beschleunigungsdaten bei 2g für Schütteln des Bauzauns

Der Bewegungseintrittsparameter [Dzi07] wird auf 3 gesetzt. Dieser Wert stellt sicher, dass kleine Erschütterungen am Bauzaun, wie etwa durch Windstöße verursacht, nicht registriert werden.

Ein Musterende wird dann erkannt, wenn eine Anzahl von Beschleunigungswerten innerhalb des Zitterbereichs liegt. Dieser Wert wird auf 15 gesetzt, damit Ereignisse nicht zu früh als beendet erkannt werden. Die Menge dieser Werte wird später wieder gelöscht. Ausnahme ist, wenn kein Ende bis zur maximalen Dauer der Ereignisse erkannt wird. Dann werden nur so viele Daten gelöscht, wie am Ende im Zitterbereich lagen.

Es ist von Vorteil, so viele Rohdaten wie möglich speichern zu können. Bei Erhöhung der Anzahl kann bei gleicher Samplingfrequenz längere Ereignisse aufgenommen oder bei gleicher Ereignislänge die Samplingfrequenz erhöht werden. Diese Anzahl ist aber durch den verfügbaren Speicher begrenzt. Im bisherigen System wurde mit 200 Werten gearbeitet. Ein Wert von 250 soll hier verwendet werden, da dieser im Bereich der Speichermöglichkeiten liegt und im Verlauf mehrerer Vorversuche dabei keine Probleme auftraten. Ab einem Wert von 300 traten bei Vorversuchen Überläufe auf, so dass eine korrekte Funktionsweise des Systems nicht mehr gegeben war.

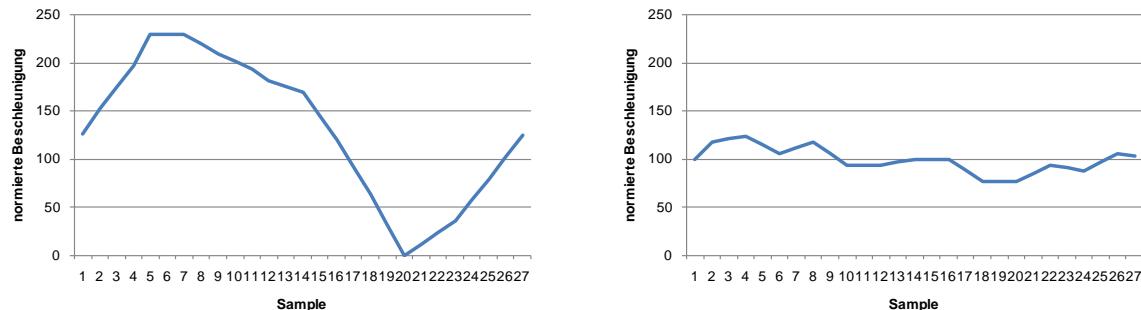
### 5.3 Erweiterungen

Auf jedem Sensorknoten werden 16 Merkmale berechnet. Diese ergaben sich bisher aus der Berechnung von Histogrammen für die normierten Rohdaten der X- und Y-Achse, welche in je 8 Histogrammklassen aufgeteilt wurden. Das Histogramm-Merkmal soll auch weiterhin verwendet werden, da damit gute Ergebnisse erzielt wurden. Da im angepassten System nur die Z-Achse verwendet werden soll und sich die Anzahl von 8 Klassen als guter Wert erwiesen hat, sind somit nur 8 Merkmale belegt und noch weitere 8 möglich. Dazu soll ein weiteres Merkmal eingebaut werden.

In den meisten Büchern zur Mustererkennung werden rechenintensive Verfahren besprochen [Nie83] [Web02] [The03] [Fuk90]. Dazu gehören die Hauptkomponentenanalyse (PCA), Fourier- oder Wavelet-Transformationen. Diese sind aber aufgrund der Speicher- und Rechenanforderungen auf den Sensorknoten nicht einsetzbar. Es muss also eine einfachere Möglichkeit gefunden werden, welche aber trotzdem Merkmale liefert, die eine gute Unterscheidbarkeit der Klassen gewährleisten.

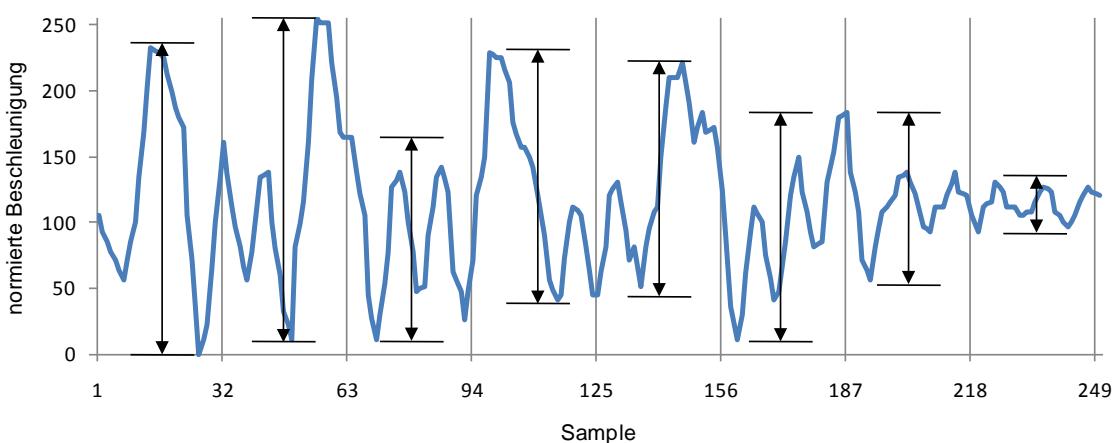
Die Rohdaten, welche aufgenommen werden, ergeben nach der Normierung eine feste Anzahl von Werten über die Zeit. Merkmale ergeben sich durch eine genaue Beschreibung dieser Kurve. Die Möglichkeit, die Kurve durch eine Auswahl an Stützstellen insgesamt zu interpolieren, ist auszuschließen, da selbst für wenige gegebene Stellen die Berechnung des Polynoms sehr aufwändig ist. Ein weiterer Ansatz ist, die Kurve stückweise zu interpolieren. Da 8 Merkmale zur Verfügung stehen, könnten die Daten in 4 gleich große Teile aufgespalten werden und linear interpoliert werden. Eine lineare Interpolation über viele Daten beschreibt das Teilstück aber nur ungenügend. Deshalb sollen die Daten in 8 Intervalle aufgespalten werden, was auch der möglichen Anzahl weiterer Merkmale entspricht. Da nur 8 Merkmale zur Verfügung stehen, soll nur noch der Anstieg betrachtet

werden. Wie in Abbildung 5.7 an zwei Teilstücken dargestellt, kann hier das Problem auftreten, dass Intensitätsänderungen im Verlauf der Kurven nicht berücksichtigt werden. Daten, die eine hohe Varianz aufweisen ergeben denselben Anstieg wie Daten mit geringer Varianz.



**Abbildung 5.7: Kurvenverläufe mit gleichem Anstieg**

Deshalb sollen nicht die Anstiege, sondern die Intensitätsänderungen innerhalb der 8 Intervalle als Merkmale dienen. Die Intensitätsänderung in einem Intervall ist dabei definiert als die Differenz zwischen dem maximalen und minimalen Wert des Intervalls. Grafisch ist dies in illustriert.



**Abbildung 5.8: Intensitätsänderungen in definierten Zeitintervallen**

Dieses Merkmal sagt zwar nichts über den Kurvenverlauf aus, ist für den Anwendungsfall sinnvoll, da die Stärke der Bewegung eines Zaunelements um die Ruhelage auf die Intensitäten der Kurve abgebildet wird. Eine geringe Bewegung in einem Zeitintervall bedeutet auch eine geringe Intensitätsänderung der entsprechenden Daten, eine starke Auslenkung des Zaunelements entspricht einer hohen Intensitätsänderung. Somit soll dieses Merkmal zusätzlich zum Histogramm-Merkmal verwendet werden.

Während der Datenfusion auf dem externen System und der Merkmalsverteilung während der Erkennung auf den Sensorknoten müssen unter Umständen für Merkmale Werte angenommen werden, die einer Ruhelage des Sensorknotens entsprechen (vgl. Abschnitte 4.4.3 und 4.5.4). Eine Ruhelage des Sensorknotens wird in der Praxis keine Werte liefern, soll aber hier theoretisch betrachtet werden. Die Daten können als eine Reihe von 0-

Werten angesehen werden. Zeit- und wertnormalisiert bleiben diese Werte gleich. Damit ergibt sich für das Histogramm-Merkmal und die Intensitätsänderung für alle Intervalle ein Wert von 0.

## 6 Auswertung

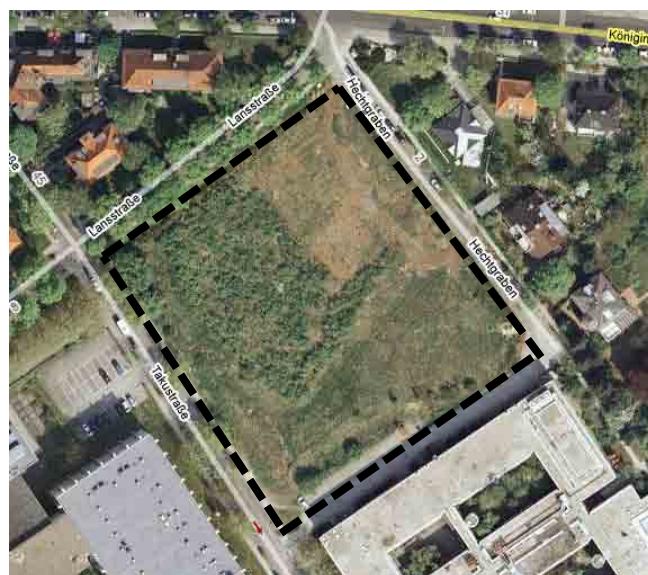
In diesem Kapitel werden Ergebnisse analysiert, welche durch die erweiterte verteilte Ereigniserkennung anhand eines großangelegten Feldversuchs erzielt wurden. Der Ablauf des Versuchs wird beschrieben. Für die Analyse werden Kennwerte verwendet, die vorgestellt werden. Die Ergebnisse werden mit dem dieser Arbeit zugrunde liegenden und einem verwandten Erkennungssystem verglichen. Verschiedene, vom externen System berechnete Klassifikationsparameter, werden bewertet. Zudem soll der Kommunikationsaufwand untersucht werden, der aufgrund der verteilten Ereigniserkennung auftritt.

### 6.1 Versuchsbeschreibung

Die Sensorknoten werden für den Versuch an einem Bauzaun angebracht. Für das Training und die Erkennung sollen mehrere Ereignisse betrachtet werden. Aus diesen Ereignissen ergeben sich weiterhin Systemparameter, die spezifiziert werden müssen.

#### 6.1.1 Aufbau des Bauzauns und Anbringung der Sensorknoten

Der Bauzaun, an dem der Versuch durchgeführt werden soll, dient zur Sicherung einer Hotelbaustelle. Es handelt sich dabei um ein etwa quadratisches Gelände. An jeder Seite befinden sich etwa 25-30 Bauzaunelemente, so dass eine Gesamtzahl von ca. 110 Elementen erreicht wird. Die einzelnen Elemente sind durch Kopplungen miteinander verbunden, so dass sich Bewegungen übertragen. Das Baustellengebiet ist in Abbildung 6.1 als Luftbild dargestellt. Der Verlauf des Bauzauns um die Baustelle ist als gestrichelte Linie eingezeichnet.



© 2008 Google

**Abbildung 6.1: Baustellengebiet mit Bauzaunverlauf**

An den Bauzaun werden 100 Sensorknoten, wie in Abschnitt 5.1 beschrieben, angebracht. Es werden zwar nur an einigen Stellen Ereignisse getestet, so dass eine solch hohe

Zahl theoretisch nicht nötig wäre, praktisch soll aber die Funktionsweise des Systems bei einer hohen Anzahl existierender Sensorknoten überprüft werden. Einige wenige Bauzaunelemente bleiben unbesetzt, diese Zahl ist aber vernachlässigbar, da dort keine Ereignisse ausgeführt werden sollen. Abbildung 6.2 zeigt den Aufbau des Bauzauns an allen vier Seiten des Geländes sowie einen an einem Zaunelement befestigten Sensor-knoten.



Abbildung 6.2: Aufbau des Bauzauns und Anbringung der Sensorknoten

### 6.1.2 Ereignisse

Es sollen vier Ereignisse trainiert und getestet werden. Diese werden so gewählt, dass sie sich in ihrem Ablauf und ihrer Auswirkung auf die Bewegung des Bauzauns unterscheiden, aber auch einen realistischen Charakter besitzen. Diese sind in Abbildung 6.21 dargestellt. Es handelt sich dabei um die Folgenden:

- *Langes Schütteln*

Ein Zaunelement wird jeweils viermal an dessen mittleren Punkt nach vorn und hinten bewegt, so dass sich eine Auslenkung des Zaunelements von ca. 10 cm ergibt. Nach Abschluss der Bewegung wird der Bauzaun losgelassen, so dass die Schwingung ausklingen kann.

- *Treten*

In der Mitte eines Zaunelements wird ein kräftiger Tritt gegen das Maschennetz ausgeführt, so dass eine möglichst hohe Auslenkung des Zaunelements erreicht wird.

- *Lehnen*

Mit dem Rücken zum Zaunelement lässt man sich in dessen Mitte dagegen fallen. Ein einfaches Anlehnen würde eine zu geringe Bewegung auslösen, so dass der Vorgang mit ein wenig Nachdruck durchgeführt werden muss.

- *Klettern*

Das Klettern wird an zwei Zaunelementen ausgeführt. Die Position beim Hochklettern ist zwischen den beiden Elementen. Die Hände werden auf diese aufgelegt. Der rechte Fuß wird in etwa 1/3 der Höhe des rechten Zaunelements zwischen den Maschen aufgesetzt, das selbe wird mit dem linken Fuß in etwa 2/3 der Höhe des linken Zaunelements getan. Danach wird das rechte Bein über die Zaunelemente geführt, so dass man etwa zwischen den beiden Elementen zum Sitzen kommt. Ein Bein befindet sich dann auf jeder Seite des Bauzauns. Aus rechtlichen Gründen wird kein Herüberklettern auf die andere Bauzaunseite durchgeführt. Das rechte Bein wird wieder zurückgeführt und zwischen den Maschen des rechten Zaunelements aufgesetzt. Von dieser Position aus wird ein Sprung auf den Boden ausgeführt.



(a)



(b)



(c)



(d)

**Abbildung 6.3: Ereignisse (a) langes Schütteln (b) Treten (c) Lehnen (d) Klettern**

### 6.1.3 Training

Jedes Ereignis wird fünfzehnmal trainiert. Während des Trainings der einzelnen Ereignisse sollen diese immer möglichst gleich ausgeführt werden. Die Gefahr des Trainings einer einzelnen Person ist, dass die entstehenden Trainingswerte zu angepasst sind. Deshalb werden die einzelnen Ereignisse jeweils zur Hälfte von zwei Personen trainiert. Es sollen keine Spezialfälle betrachtet werden, um eine möglichst hohe Einheitlichkeit für die Auswertung zu erreichen. Da durch die Gleichförmigkeit des Trainings keine großen Varianzen zu erwarten sind, soll die festgelegte Musteranzahl zur Charakterisierung der Klassen als genügend groß angenommen werden. Die Relevanz-Anforderung, welche in Abschnitt 4.4.2 beschrieben wird, wird mit 85% festgelegt. Dies erlaubt bei 15 Trainingsdurchläufen pro Klasse 2 fehlende Merkmalsvektoren pro Knoten. In dieser Arbeit wurden keine statistischen Erhebungen über eine optimale Relevanz-Anforderung gemacht. Es wird vermutet, dass durch eine niedrigere Relevanz-Anforderung Daten von Knoten für ein Ereignis verwendet werden würden, die dieses Ereignis nur unregelmäßig erkennen. Zudem kann dadurch der Trainingsaufwand deutlich ansteigen.

Das Training wird an einem Teil des Bauzauns durchgeführt, der eine geradlinige, ebene Aneinanderreihung von Bauzaunelementen aufweist. Die einzelnen Elemente sind so miteinander verbunden, dass eine möglichst gleichmäßige Ausbreitung der Schwingung nach beiden Seiten erfolgt. Weiterhin sind die Zaunelemente nicht durch Aufschüttungen oder anlehnende Gegenstände beeinflusst. Die Wahl dieser Trainingsposition ist optimal, weil durch den einfachen Charakter des Aufbaus der umliegenden Bauzaunelemente sich dieser leicht wiederherstellen und in fast jedem Fall an jedem Ort des Bauzauns realisieren lässt. Für die spätere Ergebnisdarstellung soll hier auch die Kennung des Sensor-knotens angegeben werden, welcher an dem Zaunelement befestigt ist, an dem das Training durchgeführt wird. Es handelt sich hierbei um die Nummer 12.

Die maximale Dauer über alle Ereignisse wird mit 9 Sekunden abgeschätzt. Das Klettern stellt hierbei das am längsten dauernde Ereignis dar. Dieser Wert bestimmt die Dauer des *Samplingintervalls*.

### 6.1.4 Systemparameter

Die festgestellte maximale Dauer der Ereignisse hat Einfluss auf das *Samplingintervall*, also die Zeit, die zwischen der Aufnahme zweier Messwerte vergehen muss. Im Sensor-knotensystem ist es maximal möglich, 250 Messwerte zu speichern. Diese Anzahl  $N_{mess}$  muss gleichverteilt auf die Höchstdauer der Ereignisse  $t_{max}$  abgebildet werden. Dies geschieht mit der Formel

$$\Delta_s = \frac{t_{max}}{N_{mess}} \quad (13)$$

Bei 9 Sekunden und 250 Messwerten ergibt sich ein *Samplingintervall* von 0,036 Sekunden, welches einer Samplingfrequenz von 27,8 Hz entspricht.

### 6.1.5 Untersuchte Szenarien

Zur Analyse der Erkennungsgenauigkeit des Mustererkennungssystems werden 30 Muster pro Klasse getestet. Bei den hier verwendeten 4 Klassen werden also insgesamt 120 Muster erfasst und ausgewertet. Davon werden 60 Muster, also 15 pro Klasse, an dem Zaunelement getestet, an dem das Training stattgefunden hat. Dies soll zeigen, wie gut die Erkennung ohne Änderung der Trainingsbedingungen ist. Der Test der anderen 60 Muster soll die Ortsunabhängigkeit zeigen und wird zu 2/3 an einer schiefen Ebene, an welcher der Bauzaun einer Neigung ausgesetzt ist und zu 1/3 in unmittelbarer Nähe zum Trainingsort durchgeführt. Die merkliche Änderung der Umgebungssituation wird deshalb gewählt, um die Robustheit des Erkennungssystems zu testen.

Bei der Durchführung der Tests wird darauf geachtet, dass der Ablauf der Ereignisse im Vergleich zum Training gleich bleibt. Die natürliche Variation in der Intensität und Geschwindigkeit bei der Versuchsdurchführung soll und kann nicht unterbunden werden, um die Aussagekraft des Versuches unter realen Bedingungen herzustellen. Es soll untersucht werden, wie sich das System in Bezug auf die Ortsabhängigkeit (Versuchsort gleich Trainingsort), Ortsunabhängigkeit (Versuchsort ungleich Trainingsort) und die Umstände der realen Welt verhält. Dafür muss eine einheitliche Vorgehensweise bei der Ausführung der Ereignisse beibehalten werden. Die Erkennung von Ereignissen, welche nicht trainiert wurden, soll nicht betrachtet werden.

Die verteilte Ereigniserkennung wird auf den Sensorknoten durch die in den Abschnitten 4.5.3 und 4.5.4 beschriebene Merkmalsfusion und auf dem externen System durch die in Abschnitt 4.4.10 geschilderte Klassifikationsfusion durchgeführt. Erst nach Durchlaufen dieser beiden Stufen steht die endgültige Klassifikation eines zu erkennenden Ereignisses fest. Durch diese Aufteilung ist es sinnvoll, die Erkennungen auf den beiden Systemen getrennt zu betrachten.

Ein Ereignis wird im Sensornetz von mehreren Sensorknoten wahrgenommen und nach Abschluss der Merkmalsverteilung klassifiziert. Eine endgültige Klassifikation kann nur dann korrekt sein, wenn die Klassifikation vom Sensorknoten, wo das Ereignis ausgeführt wurde, stammt. Für eine absolute Korrektheit muss natürlich auch die Erkennung mit dem Ereignis übereinstimmen. Aufgrund der Tatsache soll nur die Erkennung auf den Sensorknoten betrachtet werden, an deren Zaunelement die Tests durchgeführt wurden (zentraler Knoten). Die Erkennungen anderer Knoten sind irrelevant, da sie immer als falsch eingestuft werden sollten. Die Klassifikation auf dem Sensorknotensystem wird durch die verwendete Zurückweisungsmethode beeinflusst. In diesem Zusammenhang sollen verschiedene Zurückweisungsmethoden mit bewertet werden.

Zusammengefasst sollen die folgenden Szenarien betrachtet werden:

- Merkmalsfusion im Sensornetz auf dem zentralen Knoten
  - ortsabhängig
  - ortsunabhängig
- Merkmalsfusion im Sensornetz und Klassifikationsfusion auf dem externen System
  - ortsabhängig
  - ortsunabhängig

Für jedes dieser Szenarien werden die Erkennungsraten ohne Zurückweisung, mit Disanz zurückweisung und mit Merkmalszurückweisung untersucht.

## 6.2 Kennwerte

Für die Auswertung der Erkennungsergebnisse werden statistische Kennwerte verwendet, um die Güte der Klassifikationen zu bewerten. Die verwendeten Kennwerte basieren auf dem Prinzip der bedingten Wahrscheinlichkeit. Grundlagen für die Werte sind die nachfolgenden Begriffe:

- richtig positiv: Ein gesuchtes Muster wird korrekt klassifiziert.
- richtig negativ: Ein Muster wird nicht klassifiziert und nicht gesucht.
- falsch positiv: Ein Muster wird klassifiziert und nicht gesucht.
- falsch negativ: Ein gesuchtes Muster wird falsch klassifiziert.

Die Sensitivität (*engl.: sensitivity*) wird als das Verhältnis von richtig erkannten Mustern einer Klasse zu allen bisher gesuchten Mustern dieser Klasse definiert. Dies entspricht der Wahrscheinlichkeit, dass ein gesuchtes Muster richtig erkannt wird.

$$\text{Sensitivität} = \frac{\# \text{richtig positiv}}{\# \text{richtig positiv} + \# \text{falsch negativ}}$$

Die Spezifität (*engl.: specificity*) wird als das Verhältnis von richtig als negativ erkannten Mustern einer Klasse zur Gesamtheit der negativen erkannten Muster dieser Klasse definiert. Dies entspricht der Wahrscheinlichkeit, dass es keinen Fehlalarm gibt.

$$\text{Spezifität} = \frac{\# \text{richtig negativ}}{\# \text{richtig negativ} + \# \text{falsch positiv}}$$

Die Relevanz (*engl.: positive predictive value*) beschreibt das Verhältnis von richtig als positiv erkannten Mustern einer Klasse zur Gesamtheit der als positiv erkannten Muster dieser Klasse definiert. Dies entspricht der Wahrscheinlichkeit, dass ein erkanntes Muster aus vielen verschiedenen Mustern wirklich das erkannte Muster ist.

$$\text{Relevanz} = \frac{\# \text{richtig positiv}}{\# \text{richtig positiv} + \# \text{falsch positiv}}$$

Die Segreganz (*engl.: negative predictive value*) ist das Verhältnis von richtig als negativ erkannten Mustern einer Klasse zu allen nicht erkannten Mustern dieser Klasse. Dies entspricht der Wahrscheinlichkeit, dass ein Muster, wenn es nicht als solches erkannt wurde, tatsächlich nicht das gesuchte Muster war.

$$\text{Segreganz} = \frac{\# \text{richtig negativ}}{\# \text{richtig negativ} + \# \text{falsch negativ}}$$

Die Korrektklassifikationsrate (*engl.: accuracy*) ist die Wahrscheinlichkeit für eine richtige Erkennung. Mit der Korrektklassifikationsrate werden die Aussagen der Kennwerte Sensitivität, Spezifität, Relevanz und Segreganz zusammengefasst.

*Korrektkl.-Rate*

$$= \frac{(\# \text{richtig positiv} + \# \text{richtig negativ})}{\# \text{richtig positiv} + \# \text{richtig negativ} + \# \text{falsch positiv} + \# \text{falsch negativ}}$$

## 6.3 Training

Während des Trainings werden auf dem externen System Daten aufgenommen und verifiziert und nach Abschluss des Trainings diese für die Berechnung von Klassifikationsparametern verwendet. Die relevanten Ergebnisse sollen hier dargestellt und ausgewertet werden.

### 6.3.1 Datenverifikation

Während des Trainings werden für jede Klasse nach Erreichen der definierten Trainingsmusterzahl die Sensorknoten bestimmt, welche die Relevanz-Anforderung erfüllen. In Abbildung 6.4 sind die Anzahl der Trainingsmuster pro Klasse dargestellt, welche die Sensorknoten bis zum Erreichen der definierten Trainingsmusterzahl pro Klasse (15) übermittelt haben.

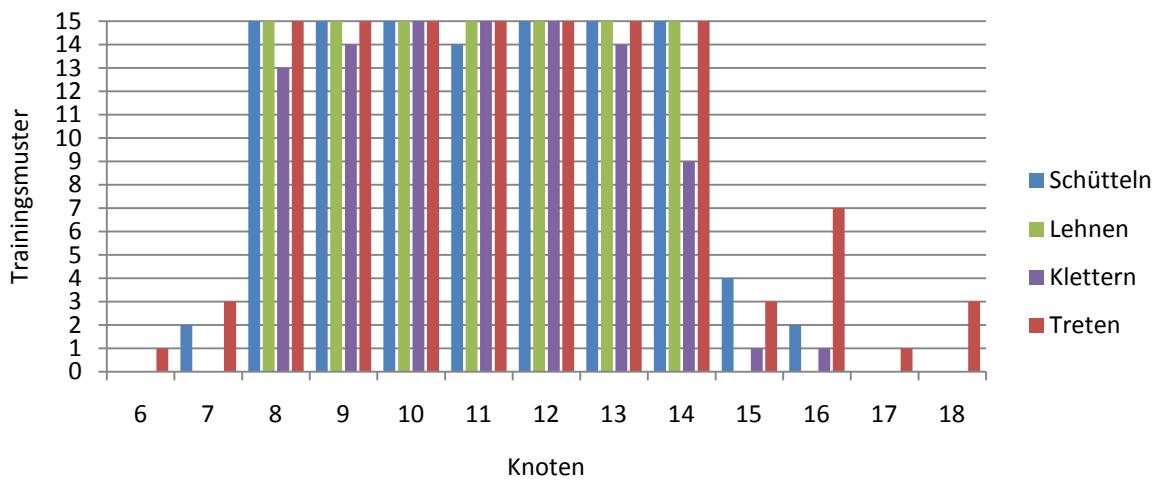


Abbildung 6.4: Relevante Knoten

Es werden von 7 Knoten relevante Daten für mindestens eine Klasse geliefert. Dabei ist zu beobachten, dass sich die Bewegungen, welche die Ereignisse auslösen, eher auf die links vom zentralen Zaunelement befindlichen Elemente übertragen. Dort werden 4 Knoten (8, 9, 10, 11) als relevant eingestuft, rechts davon nur 2 (13, 14). Eine Ursache dafür

kann sein, dass die Verbindungen zwischen den rechten Zaunelementen nicht genug für eine weite Übertragung der Schwingung ist. Besonders bei weiter vom Zentrum entfernten Zaunelementen macht sich das bemerkbar, da dort die Bewegung schon gedämpft ankommt und eine Weiterleitung über eine Kopplung möglichst einfach und ohne viel Verlust geschehen muss. Da für die weiter vom Zentrum entfernten Bewegungen angenommen wird, dass diese wenig aussagekräftige Daten liefern, soll der beschriebene Zustand nicht als Problem angesehen werden. Bei allen relevanten Knoten werden die Ereignisse Schütteln, Treten und Klettern in genügendem Ausmaß wahrgenommen. Das Klettern wird nur unregelmäßig von Knoten 14 wahrgenommen. Dies kann daher röhren, dass beim Hochklettern nur eine geringe Schwingung ausgelöst wird. In Addition mit einer schlechten Bewegungsübertragung kann es leicht möglich sein, dass teilweise von diesem Knoten nichts wahrgenommen wird. Es ist auch bei anderen Knoten (8, 9, 13) zu beobachten, dass dort dass Klettern weniger Werte als die anderen Ereignisse aufweist.

### 6.3.2 Merkmalsauswahl

Die Merkmalsauswahl, welche auf dem externen System nach der Fusion der Merkmalsvektoren abläuft, kann unter Verwendung zweier verschiedener Gütekriterien durchgeführt werden, wie in Abschnitt 4.3.2.1 beschrieben. Die ausgewählten Merkmale sollen demnach für beide Verfahren betrachtet werden.

Die Merkmalsauswahl wird für die Merkmale der 7 relevanten Knoten vorgenommen. Da auf jedem Knoten 16 Merkmale berechnet werden, sind dabei 112 Merkmale zu betrachten. Es werden zuerst Merkmale bis zur definierten Obergrenze von 10 Merkmalen ausgewählt und dann überprüft, ob an einem Punkt der Merkmalsauswahl eine Verschlechterung oder gleichbleibende Güte auftritt, wie in Abschnitt 4.3.2.2 erläutert.

#### 6.3.2.1 Kreuzvalidierung

Bei der Merkmalsauswahl, welche unter Verwendung der Kreuzvalidierung ausgeführt wird, zeigt sich nach Erreichen der maximalen Anzahl an Merkmalen, dass nach der Auswahl von 4 Merkmalen keine weitere Verbesserung der Güte eintritt. Wie Abbildung 6.5 belegt, stellt sich eine Fehlerrate von 0 ein. Daher werden nur die 4 Merkmale, die zu der Güte geführt haben, ausgewählt.

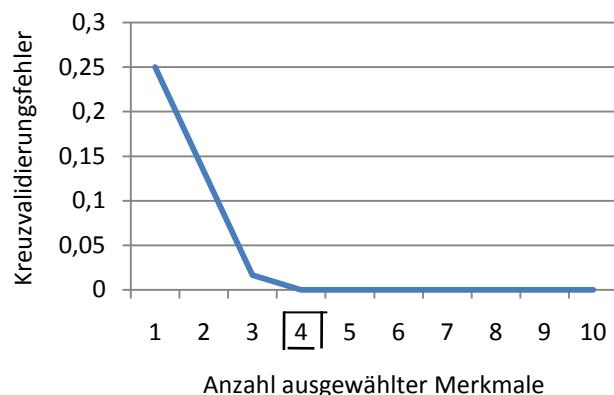
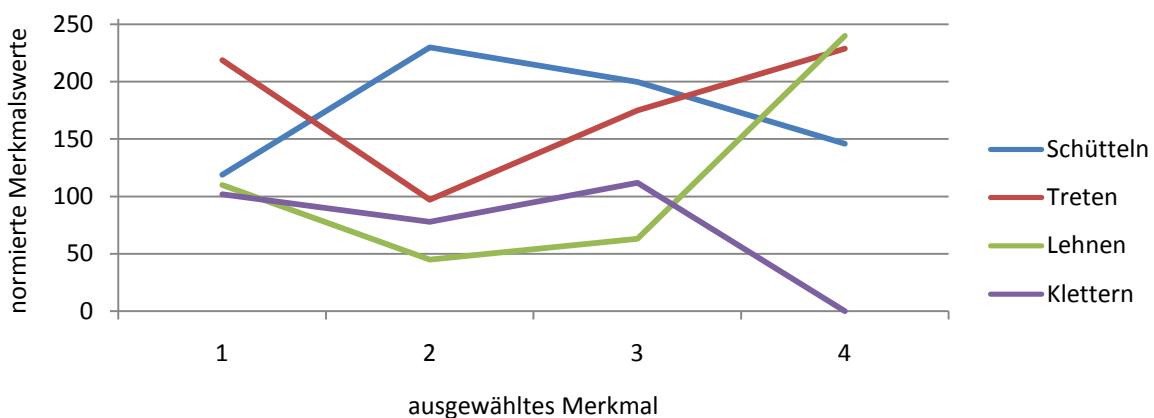


Abbildung 6.5: Fehlerrate bei Merkmalsauswahl mit Kreuzvalidierung

Die Verteilung der 4 Merkmale bei Verwendung der Kreuzvalidierung erstreckt sich über 4 Knoten (10, 12, 13, 14). Von jedem Knoten wird also ein Merkmal ausgewählt. Eines der Merkmale kommt dabei vom zentralen Knoten, bei dem auch aussagekräftige Merkmale erwartet werden. Ein weiteres Merkmal stammt vom direkten Nachbarknoten. Merkmale von weiter entfernten Knoten wie 8 und 9 wurden nicht gewählt. Dies kann ein Anzeichen dafür sein, dass die Werte von zu weit entfernten Knoten keine hohe Güte aufweisen. Die Verteilung über 4 Sensorknoten zeigt aber auch, dass die Merkmale von mehreren Knoten eine höhere Güte erreichen, als es beispielsweise Merkmale nur vom zentralen Knoten tun würden, da diese sonst vorher ausgewählt worden wären. Die Nutzung von Daten mehrerer Sensorknoten liefert in dem Fall eine bessere Erkennung als eine lokale Erkennung mit nur Daten eines Knotens.

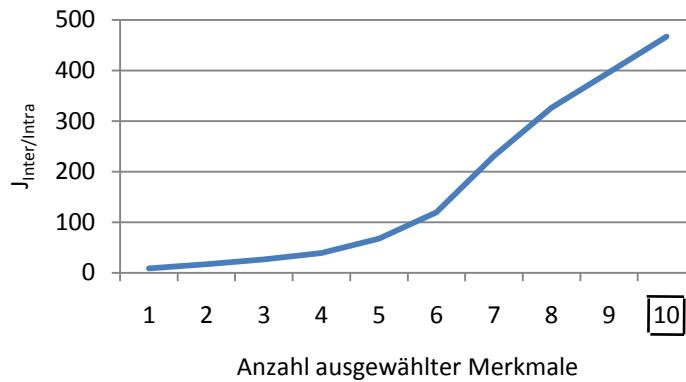
Zur Bewertung der ausgewählten Merkmale sollen die zugehörigen Referenzvektoren in Abbildung 6.6 betrachtet werden. Merkmal 1 (von Knoten 10, Histogramm-Merkmal) differenziert Treten gut von den anderen Klassen, deren Werte dafür bei diesem Merkmal eng beieinander liegen. Merkmal 2 (von Knoten 12, Intensitätsmerkmal) und 3 (von Knoten 13, Intensitätsmerkmal) zeigen eine Verteilung aller Daten über den Wertebereich, wodurch alle Klassen voneinander abgegrenzt werden. Durch das Merkmal 4 (von Knoten 14, Intensitätsmerkmal) hebt sich Klettern von den anderen Klassen ab, aber auch Schütteln hat einen guten Abstand zu den anderen Klassen. Die Werte von Treten und Lehnen liegen aber nah zusammen. Insgesamt gesehen, sind die einzelnen Klassen gut voneinander unterscheidbar.



**Abbildung 6.6: Vergleich der Referenzwerte für ausgewählte Merkmale durch Merkmalsauswahl mit Kreuzvalidierung**

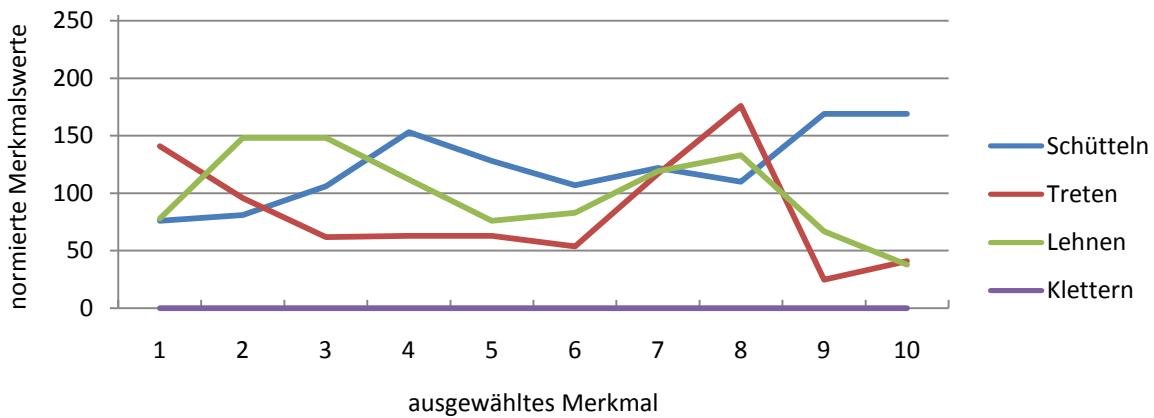
### 6.3.2.2 Inter/Intra-Klassenabstand

Im Gegensatz zur Verwendung der Kreuzvalidierung tritt bei dem Inter/Intra-Klassenabstand stetig eine Verbesserung der Güte mit jedem hinzugenommenen Merkmal auf, wie in Abbildung 6.7 zu erkennen ist. Es wird vermutet, dass ein asymptotisches Verhalten erst bei einer höheren Anzahl von ausgewählten Merkmalen auftritt. Da die Güte stetig steigt, werden alle ausgewählten Merkmale verwendet.



**Abbildung 6.7: Verteilungsverhältnis für die Merkmalsauswahl mit Inter/Intra-Klassenabstand**

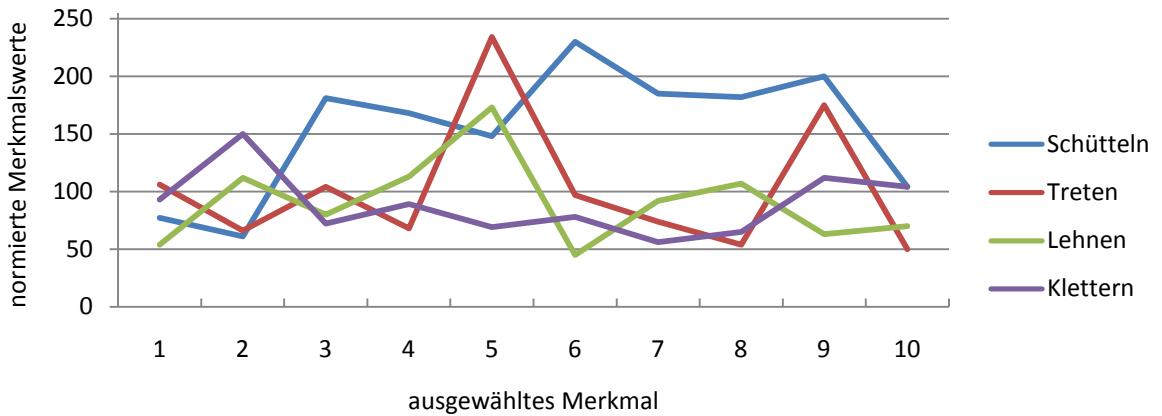
Die Merkmale werden nur von einem Knoten ausgewählt (Knoten 14). Die Werte der Referenzvektoren sollen betrachtet werden. Wie in Abbildung 6.8 zu sehen ist, sind durch dieses Verfahren solche Merkmale ausgewählt worden, wo die Werte von Klettern einer 0 entsprechen und die Werte der anderen Klassen relativ weit davon entfernt liegen. Grund dafür ist der Knoten 14, von dem diese Merkmale gewählt wurden. Dieser ist der einzige Knoten, welcher für eine Klasse, nämlich das Klettern, nicht als relevant angesehen wird. Die Referenzvektoren dieses Knotens für das Klettern werden alle mit 0 angenommen. Das Verhalten, das nur von diesem Knoten Merkmale ausgewählt wurden, hängt mit dem Berechnungsverfahren der Güte der Merkmalsmengen zusammen. Zur vereinfachenden Erklärung soll angenommen, dass die Inner-Klassen-Varianzen überall gleich sind. Dann spielt für die Güte nur die Varianzen zwischen den Referenzwerten der Klassen eine Rolle. Dabei wird eine höhere Varianz und somit eine höhere Güte erreicht, wenn zwischen wenigen Klassen ein hoher Abstand existiert, als wenn viele Klassen einen geringen Abstand zueinander haben. Dies wäre kein Problem, wenn dabei jede Klasse gleich gut unterschieden werden könnte, aber in diesem Fall hebt sich bei den meisten Merkmalen nur das Klettern von den anderen Klassen ab. Wäre dies auch für andere Knoten und andere Klassen geschehen, würde sich wahrscheinlich ein anderes Bild ergeben. In diesem Fall liegen die Werte der anderen Klassen relativ nah beieinander und sind schwer differenzierbar. Eine Ausnahme bildet nur das Merkmal 10. Dies kann aber nicht den Eindruck verbessern, dass die Merkmale, welche durch dieses Verfahren ausgewählt wurden, nicht geeignet für eine gute Erkennung aller Klassen zu sein scheinen.



**Abbildung 6.8: Referenzvektoren für ausgewählte Merkmale durch Merkmalsauswahl mit Inter/Intra-Klassenabstand**

Es ist anzunehmen, dass bei anderen Daten ohne solch eine Ausnahmestellung eine bessere Verteilung der ausgewählten Merkmale über alle Knoten erreicht wird. Allerdings sind solche Fälle nicht auszuschließen und es muss eine Möglichkeit gefunden werden, damit umzugehen. Eine Möglichkeit besteht darin, nur Knoten bei der Merkmalsauswahl unter Verwendung des Inter/Intra-Klassenabstands zu beachten, welche ausreichend Daten für alle Klassen liefern haben. Dann kann der Fall nicht auftreten, dass für eine Klasse nur 0-Werte existieren. Die Auswirkungen dieser Modifizierung sollen überprüft werden.

Nach Berechnung der Merkmale unter Ausschluss des Knotens 14 werden 10 Merkmale ausgewählt. Diese sind auf 6 Knoten verteilt (8, 9, 10, 11, 12, 13). Es zeigt sich, dass bei Daten, wo der Fall nicht auftritt, dass nur 0-Werte für einen Knoten und einer Klasse angenommen werden, die ausgewählten Merkmale auf die Knoten verteilt liegen. Die zugehörigen Referenzwerte sind in Abbildung 6.9 dargestellt. Bei manchen Merkmalen ist zu erkennen, dass diese zugehörigen Referenzwerte ein ähnliches Verhalten wie im vorher beschriebenen Fall aufweisen. Bei den Merkmalen 3, 4, 6, 7 und 8 liegen die Werte von drei Klassen sehr nah beieinander, während der Wert einer Klasse relativ weit entfernt ist. Es gibt aber auch einige Merkmale, wo sich die Referenzwerte gut voneinander abheben, wie bei Merkmal 5 und 9. Insgesamt betrachtet, scheinen durch die modifizierte Merkmalsauswahl in diesem Fall bessere Merkmale im Vergleich zur nicht modifizierten Merkmalsauswahl ausgewählt worden zu sein. Insbesondere verteilen sich die Merkmale sehr gut auf die Knoten. Da das Verhalten der Separierbarkeit von nur einer Klasse aber immer noch für einige Merkmale auftritt, scheinen die ausgewählten Merkmale unter Verwendung der Kreuzvalidierung für eine Erkennung besser geeignet zu sein.

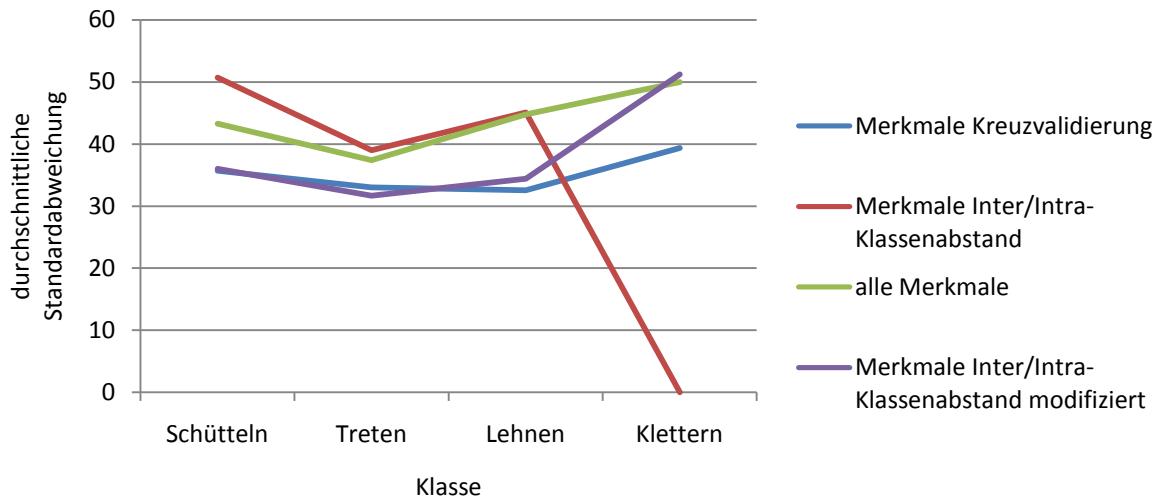


**Abbildung 6.9: Referenzvektoren für ausgewählte Merkmale durch modifizierte Merkmalsauswahl mit Inter/Intra-Klassenabstand**

### 6.3.2.3 Varianz innerhalb der Klassen

Bisher wurden nur die Referenzvektoren für die Analyse der ausgewählten Merkmale betrachtet. Die Varianz innerhalb der Klassen ist aber auch für eine gute Separierbarkeit notwendig. Aufgrund der besseren Darstellbarkeit soll hier die Standardabweichung anstelle der Varianz verwendet werden. Zum besseren Vergleich wird die durchschnittliche Standardabweichung über alle gewählten Merkmale betrachtet. In Abbildung 6.10 sind diese für die getroffene Auswahl unter Verwendung der Kreuzvalidierung und des Inter/Intra-Klassenabstands für die modifizierte und nicht modifizierte Merkmalsauswahl dargestellt. Zudem wird auch noch die durchschnittliche Standardabweichung über alle Merkmale zum Vergleich herangezogen.

Wie man sieht, sind die Werte für die Merkmale der Kreuzvalidierung und des modifizierten Inter/Intra-Klassenabstands für Schütteln, Treten und Lehnen etwa gleich und besser als die Werte für alle Merkmale. Für das Klettern ergibt sich ein besserer Wert für die Merkmale der Kreuzvalidierung. Die Werte für die ursprüngliche Anwendung des Inter/Intra-Klassenabstands sind für die ersten drei Klassen schlechter als die Werte für alle Merkmale. Nur beim Klettern liegt der Wert bei 0, da für alle Werte der Merkmalsvektoren eine 0 angenommen wird. Da für diese Klasse bei den entsprechenden Merkmalen keine Abweichung vorliegt, verstärkt das zum Teil zusätzlich die Güte dieser Merkmale, da die hohen Abweichungen bei den anderen Klassen ausgeglichen und die Varianz insgesamt noch besser werden kann als bei anderen Merkmalen, wo solch geringe Abweichungen nicht zu finden sind.



**Abbildung 6.10: Vergleich der Standardabweichung bei verschiedenen ausgewählten Merkmalen**

Bei der Verwendung des Verfahrens der Kreuzvalidierung wurden also nicht nur Merkmale ausgewählt, welche die Referenzwerte gut voneinander trennen, sondern auch die Unterschiede innerhalb der Klassen sind für diese Merkmale geringer als gesamtdurchschnittlich. Ähnlich sieht es für die Merkmale der modifizierten Merkmalsauswahl mit Inter/Intra-Klassenabstand bis auf eine Klasse aus. Die hohe Standardabweichung bei den Merkmalen der nicht modifizierten Variante scheint sich nicht sonderlich auf die Güte auszuwirken und ist ein weiterer Grund, dass diese Merkmale keine hohe Erkennungsgenauigkeit vermuten lassen.

## 6.4 Erkennung

Die Ergebnisse der Klassifikationen im Sensornetz und auf dem externen System für den ortsabhängigen und ortsunabhängigen Fall werden in diesem Kapitel dargestellt.

Weiterhin werden die Erkennungswerte der Klassifikation auf dem externen System in Vergleich mit dem dieser Arbeit zugrunde liegenden und einem verwandten Erkennungssystem und zu den Ergebnissen auf im Sensornetz gestellt.

Wie in Abschnitt 6.3.2 beschrieben, scheinen die ausgewählten Merkmale durch Verwendung der Kreuzvalidierung eine bessere Unterscheidbarkeit der Klassen als die Merkmale, welche durch die Bewertung der Verteilung der Daten im Raum gewählt wurden, zu gewährleisten. Deshalb sollen diese für den praktischen Versuch verwendet werden.

### 6.4.1 Erkennung im Sensornetz

Es soll die Erkennung auf dem zentralen Knoten für den ortsabhängigen und ortsunabhängigen Fall untersucht werden. Zusätzlich zur Klassifikation kann diese auch zurückgewiesen werden. Dabei werden die Methoden Distanz- und Merkmalszurückweisung verwendet. Die Auswirkungen dieser Methoden sollen für den zentralen Knoten im Detail bewertet werden.

#### 6.4.1.1 Ortsabhängige Erkennung

Die Ergebnisse der Testreihe, bei dem der zentrale Knoten dem zentralen Knoten des Trainings entspricht, sollen untersucht werden. In Abbildung 6.11 sind die Erkennungsraten ohne Anwendung einer Zurückweisung dargestellt. Das Ereignis Klettern hat die geringste Sensitivität und wurde somit am häufigsten nicht erkannt. Der Wert unterscheidet sich aber nicht signifikant von den anderen. Die Spezifität ist bei allen Ereignissen fast gleich hoch, so dass nur eine geringe Anzahl von Fehlalarmen ausgelöst wird und diese auf alle Ereignisse gleich verteilt sind. Die Relevanz und Segreganz sind bei Lehnen und Klettern aufgrund der geringeren Sensitivitäten geringer im Vergleich zu Schütteln und Treten. Im Mittel lässt sich eine hohe Spezifität und Segreganz feststellen. In Zusammenhang mit der relativ niedrigen Sensitivität hat auch die Relevanz einen ähnlich geringen Wert. Die mittlere Korrektklassifikationsrate aller Ereignisse liegt bei 88,3%.

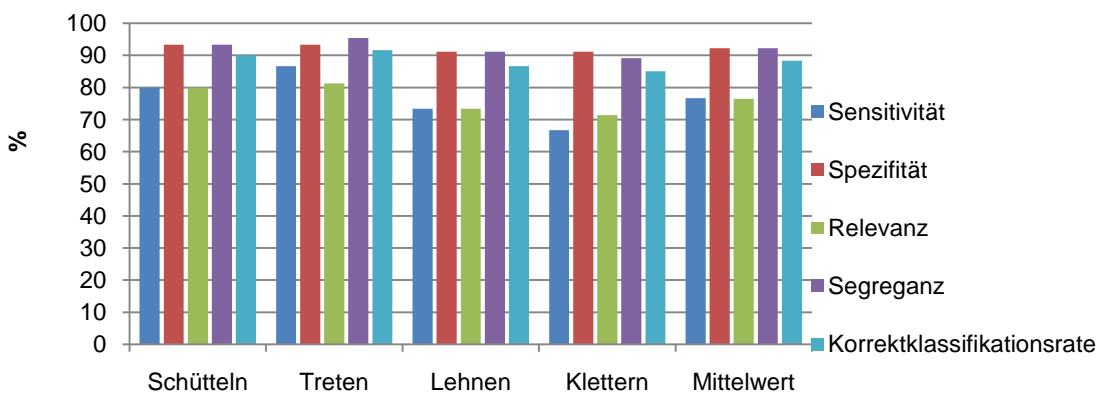
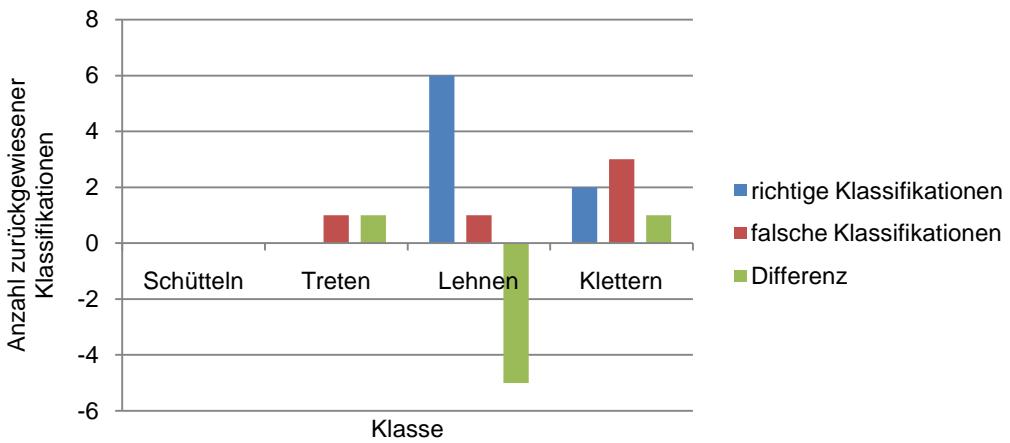


Abbildung 6.11: Ortsabhängige Kennwerte auf dem zentralen Sensorknoten ohne Zurückweisung

Die Auswertung der Erkennungsraten bei der Anwendung von Zurückweisungsmethoden unterscheidet sich von der Auswertung der Erkennung ohne Zurückweisung darin, dass richtige Klassifikationen, die zurückgewiesen werden, als nicht erkannt (false negative) gewertet werden. Zurückgewiesene falsche Klassifikationen werden auch als nicht erkannt eingestuft, allerdings gibt es aufgrund der Zurückweisung keinen Fehlalarm (false positive) bei der fälschlich erkannten Klasse.

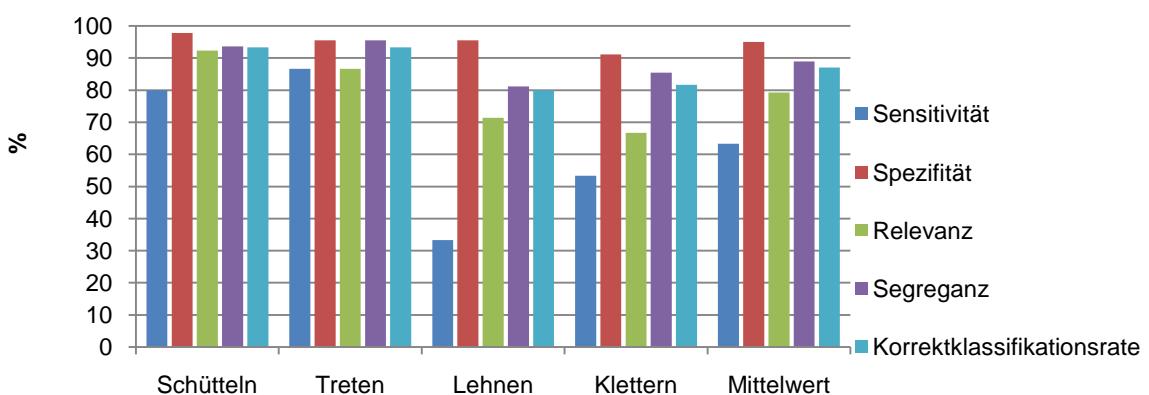
Die Differenzen der zurückgewiesenen richtigen und falschen Klassifikationen sind in Abbildung 6.12 dargestellt. Negative Werte bei der Differenz bedeuten, dass mehr richtige als falsche Klassifikationen verworfen wurden. Für Schütteln, Treten und Klettern zeigen sich annähernd gleiche Werte. Beim Lehnen werden sehr viele richtige Klassifikationen verworfen. Es scheint, als treten bei dieser Klasse Streuungen der Daten auf, die während des Trainings nicht erfasst wurden.



**Abbildung 6.12: Rückgang richtiger und falscher Klassifikationen auf dem zentralen Sensorknoten bei der Distanzzurückweisung (ortsabhängig)**

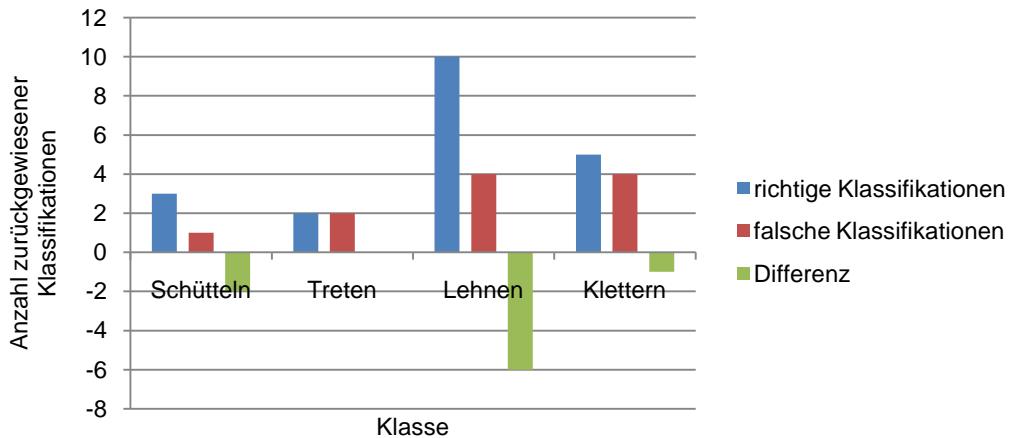
Die Ergebnisse bei der Verwendung der Distanzzurückweisung sind in Abbildung 6.13 zu sehen. Durch den Rückgang einiger falscher Klassifikationen hat auch die Anzahl der Falschalarme während der Erkennung abgenommen. Die Relevanz und Spezifität für Schütteln und Treten haben sich dadurch leicht verbessert. Die Sensitivität ist beim Ereignis Lehnen von 73% auf 33% zurückgegangen, da für dieses Ereignis viele richtige Klassifikationen zurückgewiesen wurden. Ein Rückgang der Sensitivität ist auch beim Klettern zu beobachten. Auch die Relevanz hat sich durch eine gleichbleibende Anzahl an Falschalarmen für diese Kasse bei sinkenden richtigen Klassifikationen verschlechtert. Im Mittel sind alle Kennwerte bis auf die Sensitivität um 1 bis 4% gesunken. Die Sensitivität ist aufgrund der hohen Zahl der zurückgewiesenen richtigen Klassifikationen besonders beim Treten im Mittel um 13% zurückgegangen.

Es zeigt sich, dass die Zurückweisungsgrenzen der Distanzzurückweisung, die während des Trainings ermittelt wurden, für die Klassen Schütteln und Treten auch für die Erkennung zutreffen. Es sind also keine großen Abweichungen der Daten im Verlauf der Erkennung im Vergleich zum Training festzustellen, was auch durch die guten Erkennungswerte der beiden Klassen ohne Zurückweisung unterstrichen wird. Für die Klasse Lehnen scheinen die Grenzen durch das Training nicht ausreichend bestimmt worden zu sein.



**Abbildung 6.13: Ortsabhängige Kennwerte auf dem zentralen Sensorknoten mit Distanzzurückweisung**

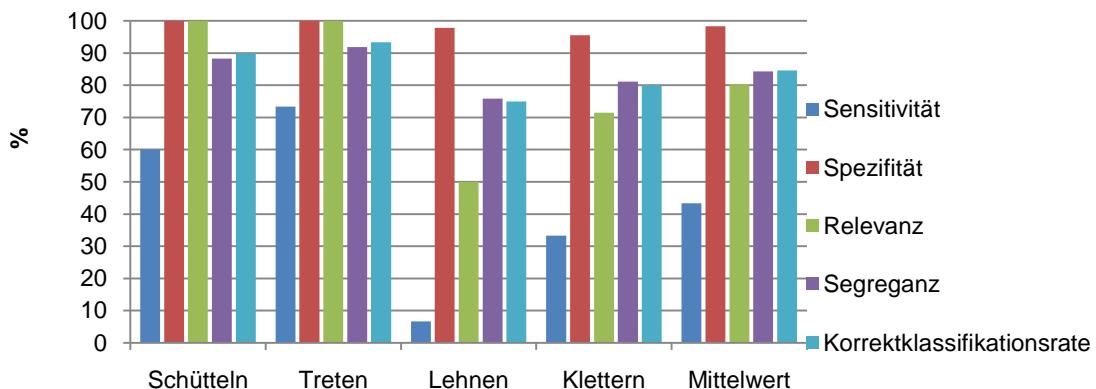
Bei der Verwendung der Merkmalszurückweisung werden mit Ausnahme des Treten mehr richtige als falsche Klassifikationen verworfen. Das ist dadurch bedingt, dass die Merkmalszurückweisung restriktiver arbeitet. Es werden viele der ursprünglich hohen Anzahl an richtigen Klassifikationen verworfen, was nicht durch die Verringerung der ursprünglich geringen Anzahl an falschen Klassifikationen ausgeglichen wird (siehe Abbildung 6.14).



**Abbildung 6.14: Rückgang richtiger und falscher Klassifikationen auf dem zentralen Sensorknoten bei der Merkmalszurückweisung (ortsabhängig)**

Die Kennwerte bei Anwendung der Merkmalszurückweisung sind in Abbildung 6.15 dargestellt. Für die Klassen Schütteln und Treten zeigt sich eine hohe Relevanz und Spezifität. Diese kommt dadurch zustande, dass für die Klassen keine Fehlalarme existieren. Durch die Merkmalszurückweisung werden mehr falsche Klassifikationen im Gegensatz zur Distanzzurückweisung verworfen, was mit der Verringerung der Anzahl der Fehlalarme einhergeht. Die Sensitivität sinkt im Gegensatz zu den Werten der Erkennung ohne Zurückweisung deutlich bei der Klasse Klettern um 34% und Lehnen um 70%. Die Sensitivität für Schütteln und Treten sinkt auch, aber nicht so deutlich. Bezogen auf die Werte ohne Zurückweisung, sinken die Werte im Mittel über alle Klassen für die Sensitivität, Segreganz und Korrektklassifikationsrate. Aufgrund der geringeren Anzahl an Fehlalarmen steigen Relevanz und Spezifität leicht an.

Wie schon bei der Distanzzurückweisung beobachtet werden konnte, sind auch für die Merkmalszurückweisung die Grenzen für die Klassen Schütteln und Treten besser als die Grenzen für Lehnen und Klettern definiert worden. Allerdings zeigt sich hier erstmals ein Rückgang der richtigen Klassifikationen. Für die beiden Klassen Lehnen und Klettern scheint das Training nicht ausreichend gewesen zu sein, um die Merkmalsgrenzen so zu wählen, dass alle Merkmale bei richtigen Klassifikationen innerhalb der Grenzen bleiben. Die Streuung der Daten ist zu groß.



**Abbildung 6.15 Ortsabhängige Kennwerte auf dem zentralen Sensorknoten mit Merkmals-zurückweisung**

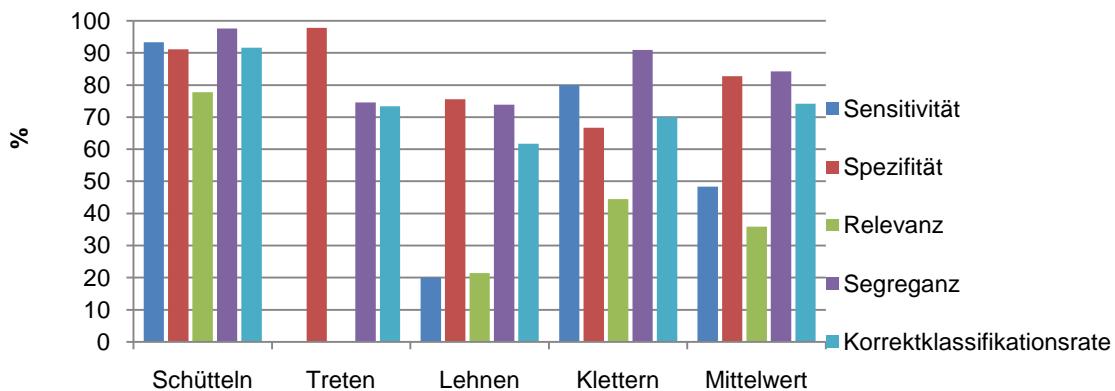
Insgesamt werden für alle untersuchten Fälle Korrektklassifikationsraten von über 84% erreicht. Die Distanzzurückweisung erweist sich für alle Klassen bis auf eine als geeignet, da nur keine oder sehr wenige richtige Klassifikationen verworfen werden und zudem die Anzahl der Fehlalarme durch Verringerung der falschen Klassifikationen abnimmt. Es scheint, als sei die Merkmalszurückweisung für den praktischen Fall nicht gut geeignet, da deren hohes Restriktionsverhalten mit der Streuung der Daten, die in der realen Welt häufig auftreten, nicht vereinbar ist.

#### 6.4.1.2 Ortsunabhängige Erkennung

In der zweiten Testreihe, bei dem der Ort der Ausführung der Ereignisse nicht dem des Trainings entspricht, soll die Auswirkung auf die Erkennung untersucht werden. Der zentrale Knoten, dessen Werte hier betrachtet werden, entspricht also nicht dem zentralen Knoten des Trainings.

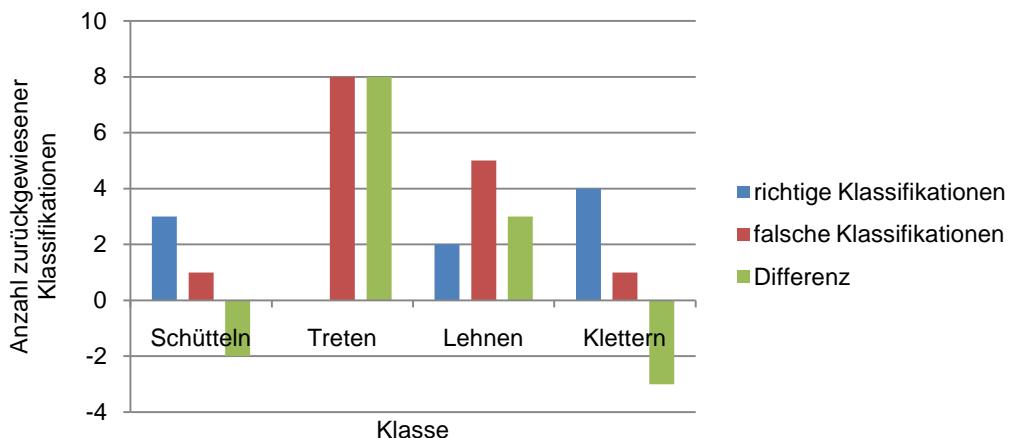
Zuerst sollen wieder die Ergebnisse ohne Zurückweisung betrachtet werden. Wie in Abbildung 6.16 zu sehen ist, weisen die Ereignisse Schütteln und Klettern eine hohe Sensitivität auf. Treten wurde in keinem Fall richtig und Lehnen nur in wenigen Fällen richtig erkannt. Für das Schütteln wurden, wie an der hohen Spezifität zu sehen ist, nur geringe Fehlalarme gemeldet. Diese Klasse scheint gut trainiert und von den anderen Klassen unterscheidbar zu sein. Fehlalarme treten häufig bei Lehnen und Klettern auf, was an der geringen Spezifität erkennbar ist. Die Klassen Treten, Lehnen und Klettern werden nur schlecht voneinander unterschieden. Die hohe Sensitivität bei Klettern ist dadurch zu stande gekommen, da diese Klasse sehr oft fälschlich erkannt wird und somit auch der Test dieser Klasse viele richtige Klassifikationen dieser Klasse liefert. Über alle Klassen wird eine Sensitivität von 50% erreicht. Dadurch wird auch die Relevanz verringert. Die Wahrscheinlichkeit eines Fehlalarms liegt bei 17%. Die Korrektklassifikationsrate liegt im Mittel bei 74,2%. Es sieht so aus, als sei das Training nicht robust genug, um mit veränderten äußerlichen Gegebenheiten zurechtzukommen. Für das Schütteln scheint das gut möglich zu sein, aber für die anderen 3 Klassen werden keine zufriedenstellenden Ergebnisse geliefert. Es scheinen bei veränderter Umgebung in diesen Fällen solche Daten aufzutreten, welche nicht denen des Trainings entsprechen und wobei die Klassen untereinander falsch erkannt werden. Die meisten Muster der Testreihe für das Treten, Leh-

nen und Klettern scheinen dabei den trainierten Mustern des Kletterns zu entsprechen. Treten wird so gut wie gar nicht mehr, auch nicht fälschlicherweise, erkannt.



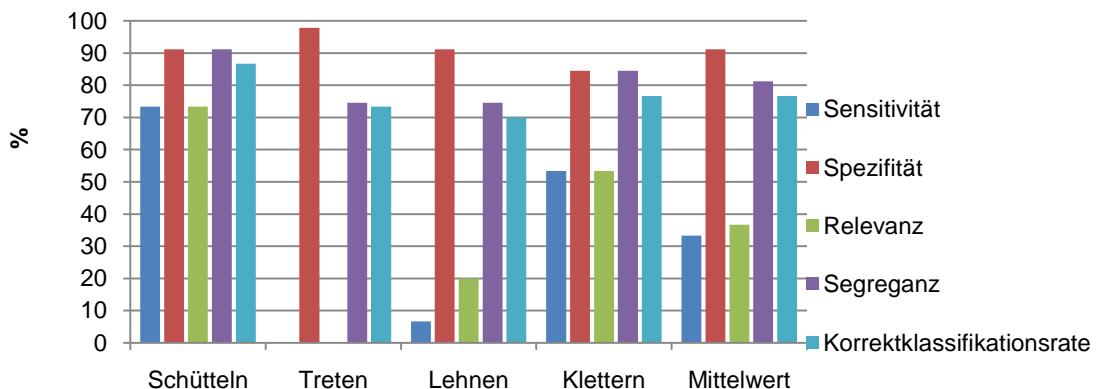
**Abbildung 6.16: Ortsunabhängige Kennwerte auf dem zentralen Sensorknoten ohne Zurückweisung**

Bei der Anwendung der Distanzzurückweisung lässt sich beobachten, dass beim Schütteln und Klettern ein relativ hoher Rückgang der richtigen Klassifikationen im Vergleich zu den falschen auftritt (siehe Abbildung 6.17). Der positive Wert von Treten und Lehnen ist nicht aussagekräftig, da diese schon vor der Zurückweisung fast nur falsche Klassifikationen aufweisen. Richtige Klassifikationen können also nicht oder nur in geringer Zahl verworfen werden.



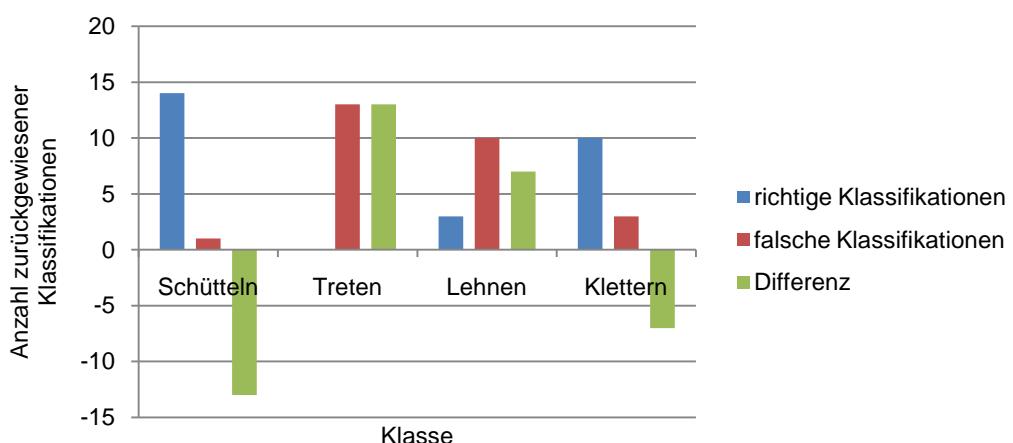
**Abbildung 6.17: Rückgang richtiger und falscher Klassifikationen auf dem zentralen Sensorknoten bei der Distanzzurückweisung (ortsunabhängig)**

Die Kennwerte sind in Abbildung 6.18 dargestellt. Wie erwartet, bleiben die Ergebnisse für Schütteln relativ stabil. Durch den starken Rückgang an falschen Klassifikationen verbessern sich die Kennwerte außer der Sensitivität für Lehnen und Klettern. Da insgesamt mehr falsche als richtige Klassifikationen zurückgewiesen werden, steigen die Werte für Relevanz, Spezifität und Korrektklassifikationsrate im Mittel leicht an. Die Anwendung der Distanzzurückweisung scheint bei diesen Daten sinnvoll zu sein.



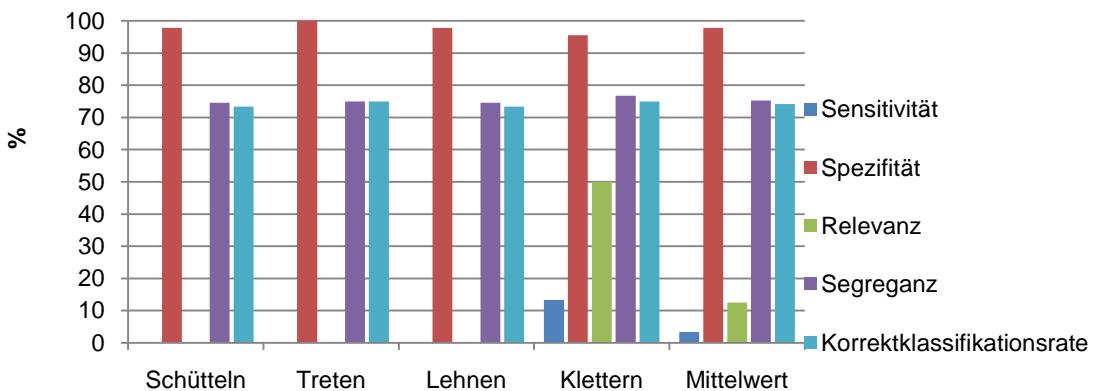
**Abbildung 6.18: Ortsunabhängige Kennwerte auf dem zentralen Sensorknoten mit Distanz-zurückweisung**

Bei der Merkmalszurückweisung zeigt sich aufgrund der restiktiven Verhaltens ein hoher Rückgang aller Klassifikationen (siehe Abbildung 6.19). Bei dem bisher so robusten Schütteln werden alle richtigen Klassifikationen zurückgewiesen. Bei allen Klassen werden so gut wie alle Klassifikationen verworfen. Hier wird nochmals deutlich, dass sich die wahrgenommenen Daten von den trainierten Daten unterscheiden.



**Abbildung 6.19: Rückgang richtiger und falscher Klassifikationen auf dem zentralen Sensorknoten bei der Merkmalszurückweisung (ortsunabhängig)**

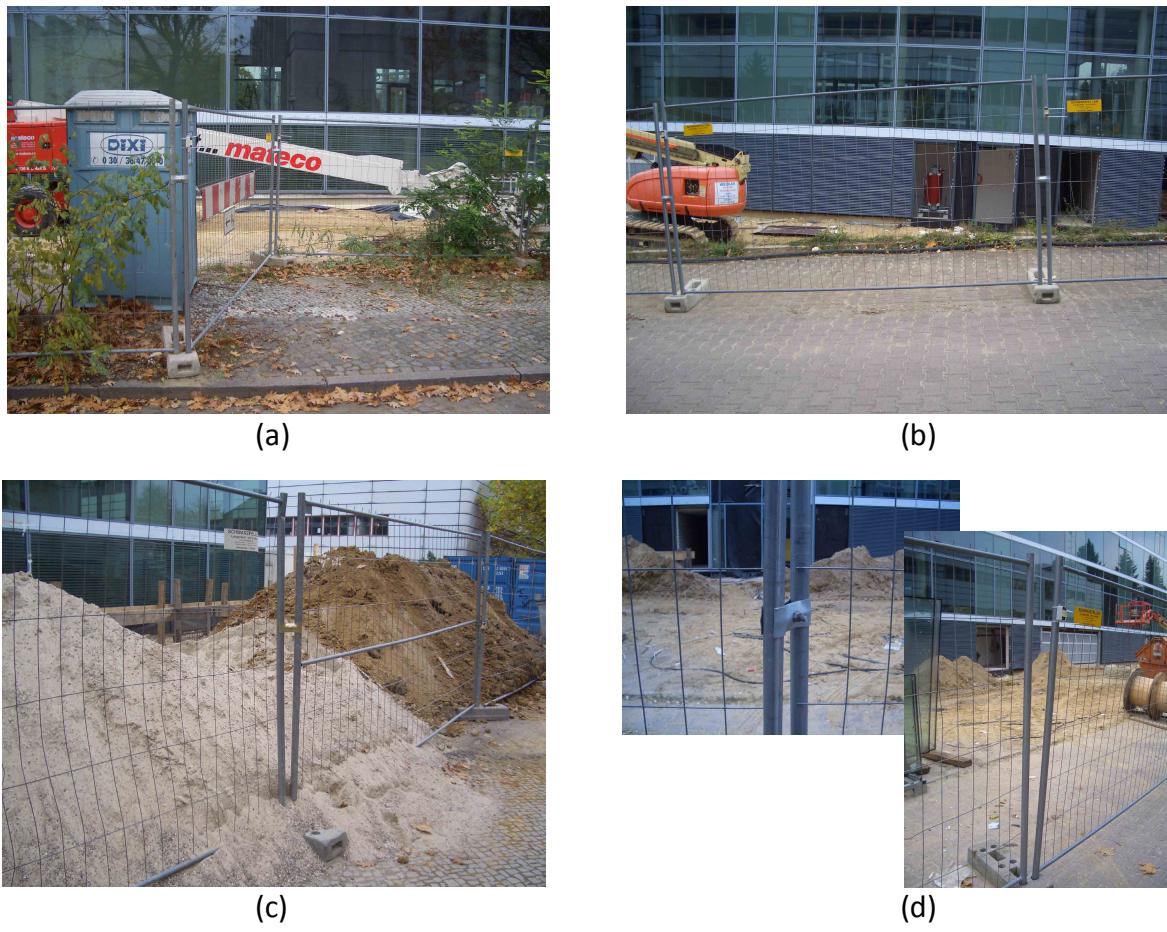
Durch die geringe Anzahl verbleibender Klassifikationen ergeben sich Kennwerte, welche keine Aussagekraft mehr haben. Der Vollständigkeit halber sollen sie hier trotzdem präsentiert werden und sind in Abbildung 6.20 dargestellt.



**Abbildung 6.20: Ortsunabhängige Kennwerte auf dem zentralen Sensorknoten mit Merkmalszurückweisung**

Durch die Auswertung der Testreihen für die ortsunabhängige Erkennung kann man erkennen, dass die wahrgenommenen Daten nur teilweise den Trainingsdaten entsprechen. Für manche Klassen werden passable Ergebnisse geliefert, während die Erkennung für andere Klassen sehr schlecht ist.

Grund dafür ist, dass der Bauzaun nicht überall gleichförmig aufgebaut ist, wie in Abbildung 6.21 zu sehen ist. An nicht allen Seiten besteht er aus einem zusammenhängenden Stück, sondern ist teilweise durch Tore unterbrochen. Weiterhin sind die Elemente nicht immer in einer geraden Linie angeordnet, an einigen Stellen sind Einbuchtungen eingelassen oder es gibt Kurven im Verlauf (a). Die Zaunelemente stehen nicht immer auf einer ebenen Fläche, sondern sind an einigen Stellen einer Neigung, bedingt durch abschüssiges Gelände, unterworfen (b). An manchen Stellen befindet sich auf den Sockeln, in denen die Seitenrohre der Zaunelemente eingelassen sind, aufgeschüttete Erde, so dass eine Bewegung des Bauzauns eingeschränkt ist (c). Zum Teil sind auch Gegenstände an Zaunelemente angelehnt, so dass derselbe Effekt auftritt. Koppelemente bestehen aus verschiedenen Materialien und verbinden die Zaunelemente auf unterschiedliche Arten und sind selten gar nicht vorhanden (d).



**Abbildung 6.21: Ungleicher Aufbau des Bauzauns**

Die Erkennung wurde für den ortsunabhängigen Fall an einer Stelle mit einer Neigung des Bauzauns sowie an einer Stelle mit schlechten Kopplungen der Bauzaunelemente ausgeführt. Die Unterschiede im Aufbau zum Trainingsort bewirken ein anderes Bewegungsverhalten des Bauzauns, als wie es während des Trainings verzeichnet wurde.

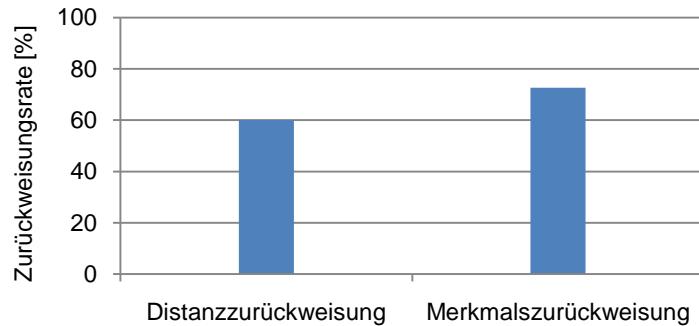
## 6.4.2 Zurückweisung über alle Sensorknoten

Bisher wurde nur die Erkennung des zentralen Knotens unter Berücksichtigung der Zurückweisungsmethoden im Detail betrachtet. Dabei liefern alle Sensorknoten, welche ein Ereignis wahrnehmen, eine Klassifikation. Die Zurückweisung soll dazu dienen, die Klassifikationen der nicht-zentralen Knoten zu verwerfen. Im Idealfall sollten alle Klassifikationen dieser Knoten zurückgewiesen werden. Zur Bewertung der beiden Zurückweisungsmethoden wird der prozentuale Anteil der Zurückweisungen (Zurückweisungsrate) für die nicht-zentralen Knoten untersucht. Weiterhin soll die Anzahl der Klassifikationen dargestellt werden, welche nach Anwendung der verschiedenen Zurückweisungsmethoden von allen Knoten an das externe System geschickt werden. Diese werden in Vergleich zu der Anzahl gestellt, die ohne Zurückweisung versendet werden würden.

### 6.4.2.1 Ortsabhängige Erkennung

Für die Testreihe am Ort des Trainings wird für die Distanzzurückweisung eine Zurückweisungsrate von 61% bei den nicht-zentralen Knoten erreicht. Die Distanzzurückweisung

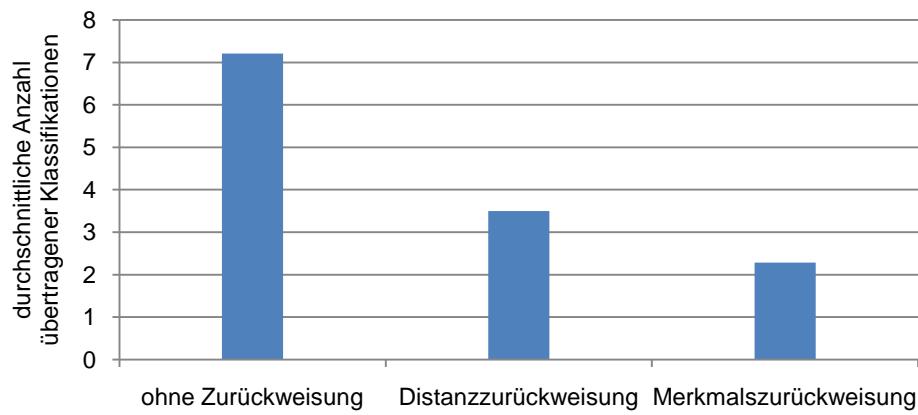
ist wenig restriktiv, so dass bei den nicht-zentralen Knoten viele Klassifikationen erhalten bleiben. Bei der Merkmalszurückweisung, die deutlich restriktiver arbeitet, werden mehr Erkennungen verworfen. Der Idealfall von 100% wird aber nicht erreicht.



**Abbildung 6.22: Ortsabhängige Zurückweisungsrate der nicht-zentralen Sensorknoten**

Wie zu erwarten ist, werden im Durchschnitt pro Ereignis bei der Merkmalszurückweisung am wenigsten Klassifikationen übermittelt. Eine Übersicht findet sich in Abbildung 6.23. Insgesamt wird ein Ereignis durchschnittlich von etwa sieben Knoten erkannt und deren Klassifikationen versendet. Durch die Distanzzurückweisung sinkt diese Rate auf etwa die Hälfte. Durch die Anwendung einer Zurückweisung lässt sich somit eine deutliche Verringerung der zu versendeenden Klassifikationsergebnisse erreichen. Es ist aber nicht so, dass der Idealfall erreicht wird und nur eine Klassifikation übermittelt wird. Das zeigt die Notwendigkeit einer Klassifikationsfusion.

Durch die geringe Anzahl an zu übertragenden Klassifikationen scheint die Merkmalszurückweisung besser geeignet zu sein. Wie aber bei der Bewertung der Erkennung auf dem zentralen Knoten festgestellt werden konnte, werden bei dieser Methode mehr richtige Klassifikationen des zentralen Knoten verworfen als bei der Distanzzurückweisung. Da nur diese durch die Klassifikationsfusion auf dem externen System als korrekt eingestuft werden, ist es also besser, die Distanzzurückweisung zu verwenden.



**Abbildung 6.23: Durchschnittliche Anzahl übertragener Klassifikationen für alle Sensorknoten (ortsabhängig)**

#### 6.4.2.2 Ortsunabhängige Erkennung

An den Orten ungleich des Trainings steigt die Rate der Zurückweisungen im Vergleich zum ortsabhängigen Fall bei der Merkmalszurückweisung für die nicht-zentralen Knoten leicht an und nimmt bei der Distanzzurückweisung leicht ab (siehe Abbildung 6.24).

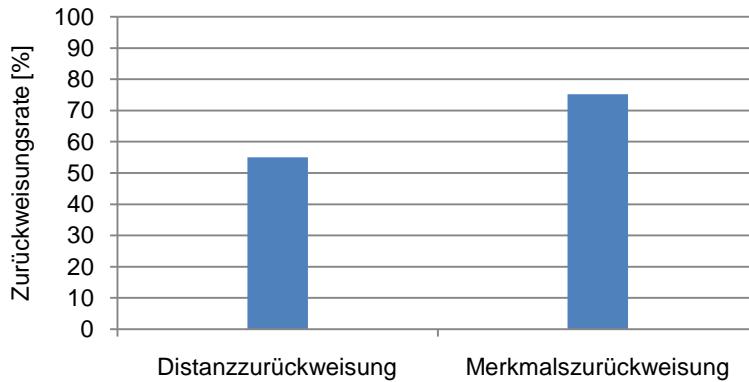


Abbildung 6.24: Ortsunabhängige Zurückweisungsrate der nicht-zentralen Sensorknoten

Die Anzahl der übertragenen Klassifikationen ohne Zurückweisung und bei Anwendung des Distanzverfahrens ist in etwa den Werten bei der ortsabhängigen Erkennung. Im Vergleich sinkt die Anzahl für die Merkmalszurückweisung, da weniger Klassifikationen vom zentralen Knoten und von den nicht-zentralen Knoten übermittelt werden. Auch hier ist eine Klassifikationsfusion notwendig.

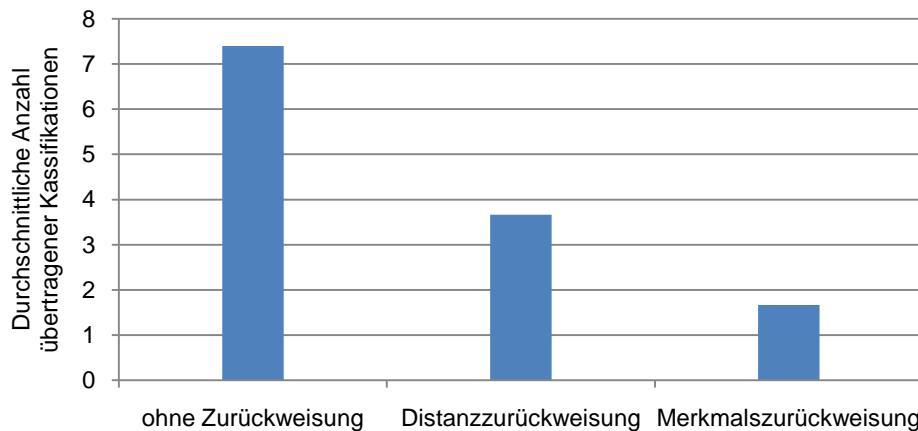


Abbildung 6.25: Durchschnittliche Anzahl übertragener Klassifikationen für alle Sensorknoten (ortsunabhängig)

#### 6.4.3 Erkennung auf dem externen System

Wie in Abschnitt 6.4.2 festgestellt, ist die Notwendigkeit einer Klassifikationsfusion gegeben. In diesem Fall soll das Ergebnis der Fusion aller eingetroffenen Klassifikationen, welche von den Sensorknoten anhand der Merkmalsfusion durchgeführt wurde, bewertet werden. Das Ergebnis wird nur dann als korrekt eingestuft, wenn die Klassifikation vom zentralen Knoten stammt und mit dem zu erkennenden Ereignis übereinstimmt. Wenn die Klassifikation von einem nicht-zentralen Knoten kommt, ist sie immer falsch, auch wenn

sie dem Ereignis entspricht. Es soll wieder zwischen den verschiedenen Zurückweisungsmethoden unterschieden werden.

#### 6.4.3.1 Ortsabhängige Erkennung

Für die Testreihe am Trainingsort sind die Kennwerte ohne Anwendung einer Zurückweisung in Abbildung 6.26 dargestellt. In die Klassifikationsfusion, die bewertet werden soll, fließen somit in diesem Fall die Klassifikationen aller an einem Ereignis teilnehmenden Sensorknoten ein. Die Sensitivität ist bei allen Klassen bis auf Treten sehr gering. Es ist eine hohe Wahrscheinlichkeit von Fehlalarmen gegeben. Die ist für alle Klassen etwa gleich. Die Relevanz ist für das Anlehnen und Klettern sehr gering. Im Mittel über alle Klassen ist die Wahrscheinlichkeit, dass ein gesuchtes Muster auch korrekt erkannt wird, unter 50%. Zusammen mit der gleichverteilten hohen Fehlalarmzahl, welche sich in der niedrigen Relevanz und Spezifität wiederspiegelt, ist eine Erkennung bei alle Klassen mit einer hohen Unsicherheit verbunden. Insgesamt wird eine Korrektklassifikationsrate von 74,77% erreicht. Ein Vergleich zu den Werten, welche auf dem zentralen Knoten erzielt wurde, soll im späteren Verlauf des Kapitels durchgeführt werden.

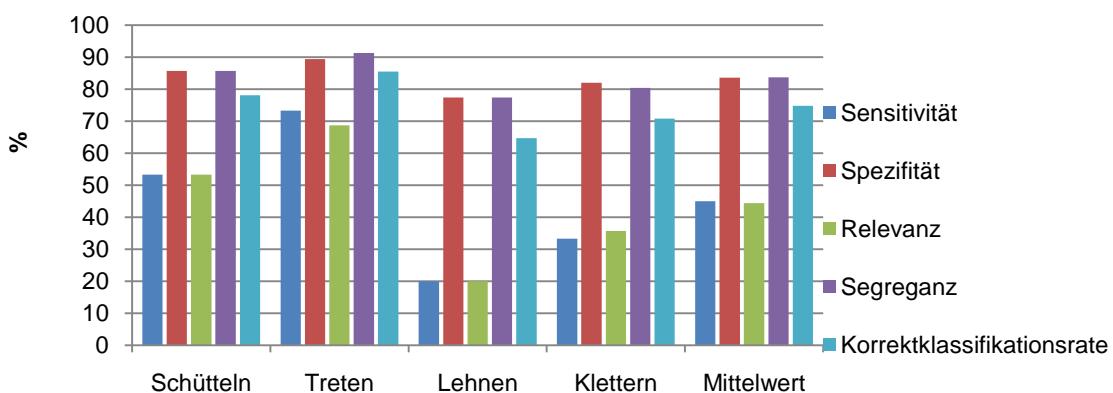
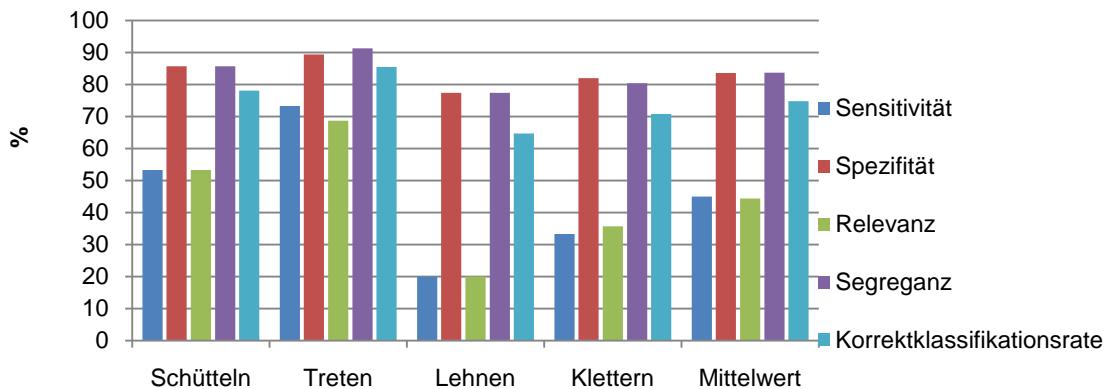


Abbildung 6.26: Ortsabhängige Kennwerte auf dem externen System ohne Zurückweisung

Bei der Anwendung der Distanzzurückweisung lassen sich in etwa die gleichen Ergebnisse vermuten. Sollte eine richtige Klassifikation vom zentralen Knoten verworfen werden, so bedeutet das eine hohe Abweichung vom Mittelwert. Diese Klassifikation wäre somit durch die Klassifikationsfusion nicht als beste Klassifikation ausgewählt worden, da es mit hoher Wahrscheinlichkeit noch bessere Klassifikationen nicht-zentraler Sensorknoten gibt, die durch die Distanzzurückweisung nicht zurückgewiesen wurden. Die Zurückweisung der richtigen Klassifikation des zentralen Knotens sollte daher keine Verschlechterung der Ergebnisse bewirken. Die Zurückweisung von falschen Klassifikationen bedeutet wiederum keine Verbesserung, da dadurch eine Klassifikation eines nicht-zentralen Knotens ausgewählt wird, welche auch als falsch angesehen wird.

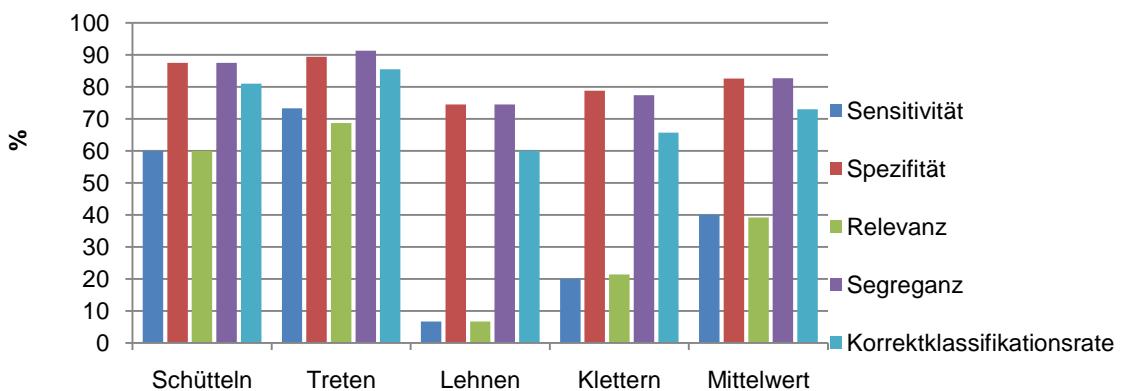
Die Vermutung, dass bei Anwendung der Distanzzurückweisung in etwa die gleichen Werte wie bei der Erkennung ohne Zurückweisung auftreten, bestätigt sich, wie in Abbildung 6.27 zu sehen ist. Die Ergebnisse bleiben sogar für alle Kennwerte gleich.



**Abbildung 6.27: Ortsabhängige Kennwerte auf dem externen System mit Distanzzurückweisung**

Der Charakter der Merkmalszurückweisung könnte eine Veränderung der Ergebnisse bewirken. Eine richtige Klassifikation des zentralen Knotens muss noch lange nicht bedeuten, dass die Distanz zum Mittelwert groß ist. Es kann sich auch nur um eine geringe Streuung vielleicht eines Merkmals handeln, die beim Training nicht erfasst wurde. Somit können hier unter Umständen Klassifikationen verworfen werden, welche positiv nach der Klassifikationsfusion bewertet würden. Es kann natürlich auch sein, dass Klassifikationen von nicht-zentralen Knoten verworfen werden, wodurch die richtigen Klassifikationen des zentralen Knotens besser bewertet werden.

In diesem Fall kommt es zu einer Verschlechterung der Werte. Die Sensitivität und Relevanz sinken im Mittel auf 40%, die Korrektklassifikationsrate auf 73% (siehe Abbildung 6.28).



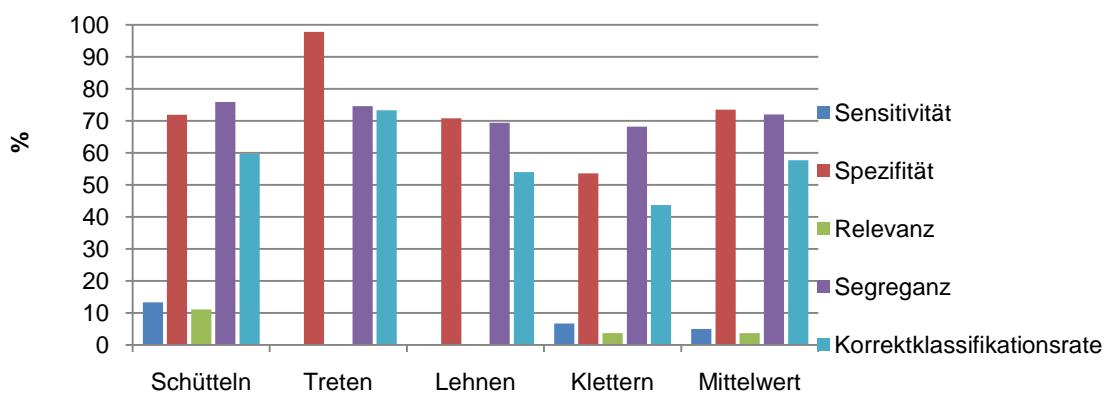
**Abbildung 6.28: Ortsabhängige Kennwerte auf dem externen System mit Merkmalszurückweisung**

Aufgrund der Unsicherheit bei der Anwendung der Merkmalszurückweisung, bei der durch geringe Streuungen der Daten des zentralen Knotens relativ leicht schlechtere Werte entstehen können, wie auch an den Ergebnissen zu sehen ist, soll anhand der Daten der Testreihe die Distanzzurückweisung als beste Methode angesehen werden. Die Daten

werden im Vergleich zu keiner Zurückweisung zwar nicht verbessert, dafür sinkt der Kommunikationsaufwand für zu versendende Klassifikationsergebnisse deutlich. Würde die Merkmalszurückweisung auch auf dem zentralen Sensorknoten gute und sichere Ergebnisse liefern, könnte diese auch hier bessere Ergebnisse bei noch geringer Kommunikation liefern. Dies müsste mittels eines umfangreicheren Trainings untersucht werden.

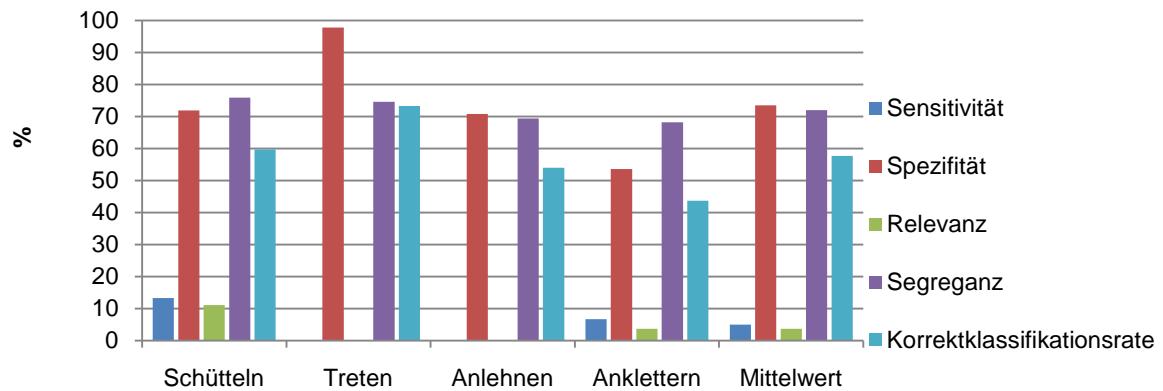
#### 6.4.3.2 Ortsunabhängige Erkennung

Die Ergebnisse der Testreihe an den Orten ungleich des Trainings sind in Abbildung 6.29 dargestellt. Wie man sehen kann, werden Ereignisse so gut wie nie richtig erkannt. Nur das Schütteln und Klettern weisen eine geringe Sensitivität auf. Für das Klettern werden die meisten Fehlalarme geliefert. Die Relevanz ist im Mittel bei 3,7%. Es kann nicht entschieden werden, ob eine Erkennung auch wirklich das zu erkennende Ereignis darstellt. Die mittlere Korrektklassifikationsrate liegt bei 57,67%.

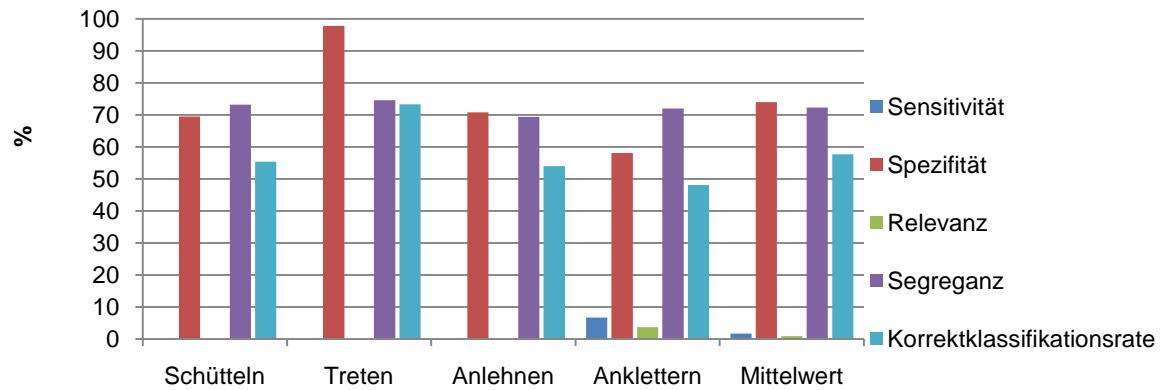


**Abbildung 6.29: Ortsunabhängige Kennwerte auf dem externen System ohne Zurückweisung**

Wie auch bei der ortsabhängigen Erkennung, verändern sich die Werte durch die Distanzzurückweisung kaum. Die Merkmalszurückweisung liefert sogar noch leicht schlechtere Werte. Aufgrund der Ähnlichkeit mit den Werten, die ohne Zurückweisung erzielt wurden, wird auf eine Beschreibung verzichtet und die Daten nur in Abbildung 6.30 (Distanzzurückweisung) und Abbildung 6.31 (Merkmalszurückweisung) dargestellt.



**Abbildung 6.30: Ortsunabhängige Kennwerte auf dem externen System mit Distanzzurückweisung**



**Abbildung 6.31: Ortsunabhängige Kennwerte auf dem externen System mit Merkmalszurückweisung**

Das Problem der schlechten Erkennungsraten, welches schon während der Auswertung der Erkennung auf dem zentralen Knoten festgestellt wurde, zeigt sich hier noch verstärkt. Durch die veränderten Bewegungsabläufe des Bauzauns liefern nicht-zentrale Knoten gute Erkennungen, wodurch die positiven Effekte nach der Klassifikationsfusion gering ausfallen.

#### 6.4.4 Vergleiche

Die Ergebnisse der ortsabhängigen und ortsunabhängigen Erkennung auf dem zentralen Knoten und dem externen System werden verglichen. Die Erkennungswerte der Klassifikation auf dem externen System werden in Vergleich mit dem dieser Arbeit zugrunde liegenden und einem verwandten Erkennungssystem gestellt. Verbesserungsmöglichkeiten, die sich aus diesen Daten ergeben, werden später in Abschnitt 7.2 diskutiert.

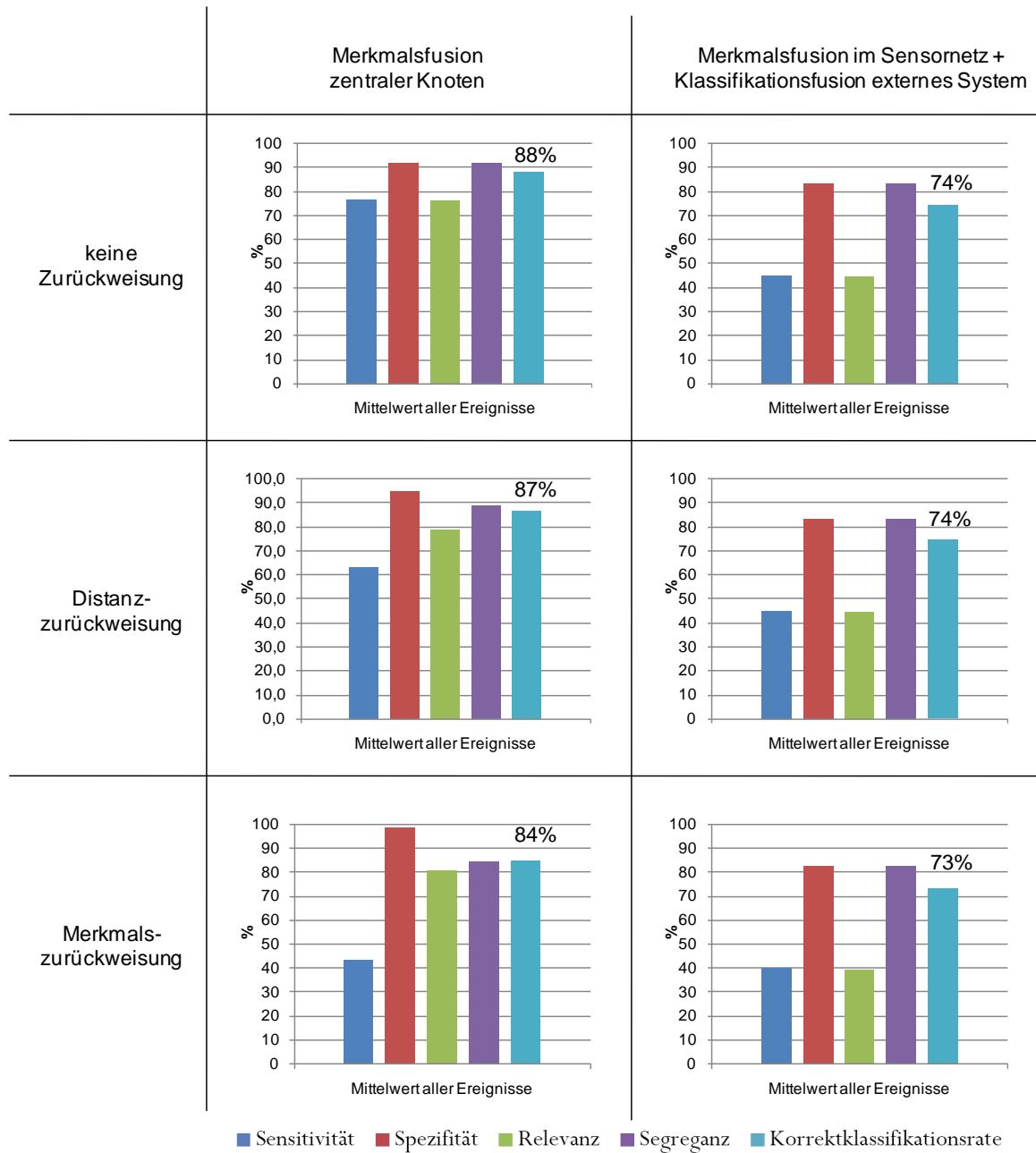
##### 6.4.4.1 Ortsabhängige Erkennung

Für die ortsabhängige Erkennung soll ein Vergleich der Mittelwerte der Ergebnisse auf dem zentralen Sensorknoten zu denen des externen Knotens für alle Zurückweisungsme-

thoden gezogen werden. Wie in Abbildung 6.32 zu erkennen ist, gibt es für den Fall, dass keine Zurückweisung verwendet wird, einen starken Rückgang der Sensitivität und Relevanz von 76% auf 45% für beide Kennwerte. Die Spezifität und Segreganz verringern sich jeweils um 9%. Die Korrektklassifikationsrate sinkt von 88,3% auf 74,8%. Im Fall der Distanzzurückweisung geht die Sensitivität von 63% auf 45% zurück. Die Relevanz sinkt deutlich von 79% auf 44%. Die Korrektklassifikationsrate verringert sich von 87% auf 74%. Durch die Merkmalszurückweisung wird für den zentralen Knoten eine Sensitivität von 43%, auf dem externen System von 40% erreicht. Die Korrektklassifikationsrate sinkt von 84% auf 73%.

Da durch die Distanzzurückweisung dieselbe Korrektklassifikationsrate wie ohne Anwendung einer Zurückweisung erreicht wird und zudem auch, wie in Abschnitt 6.4.2.1 beschrieben, eine Verringerung der zu übertragenden Klassifikationen erreicht wird, scheint diese die geeignetste Zurückweisungsmethode zu sein. Die zum Teil geringeren Kennwerte auf dem zentralen Knoten bei der Distanzzurückweisung haben keinen Einfluss auf die Kennwerte auf dem externen System im Vergleich zur Anwendung keiner Zurückweisung.

Als Ursache für den Rückgang der Korrektklassifikationsrate kann gesehen werden, dass nicht-zentrale Sensorknoten zum Teil bessere Klassifikationen als der zentrale Knoten liefern. Um genauere, mehr dem Training entsprechende Daten zu erhalten, könnte das Training umfangreicher gestaltet werden. Bessere Zurückweisungen, welche die Grenzen der Trainingsdaten besser beschreiben, könnten mehr Klassifikationen von nicht-zentralen Knoten verwerfen. Umso restriktiver die Zurückweisung ist, umso umfangreicher muss das Training durchgeführt werden.



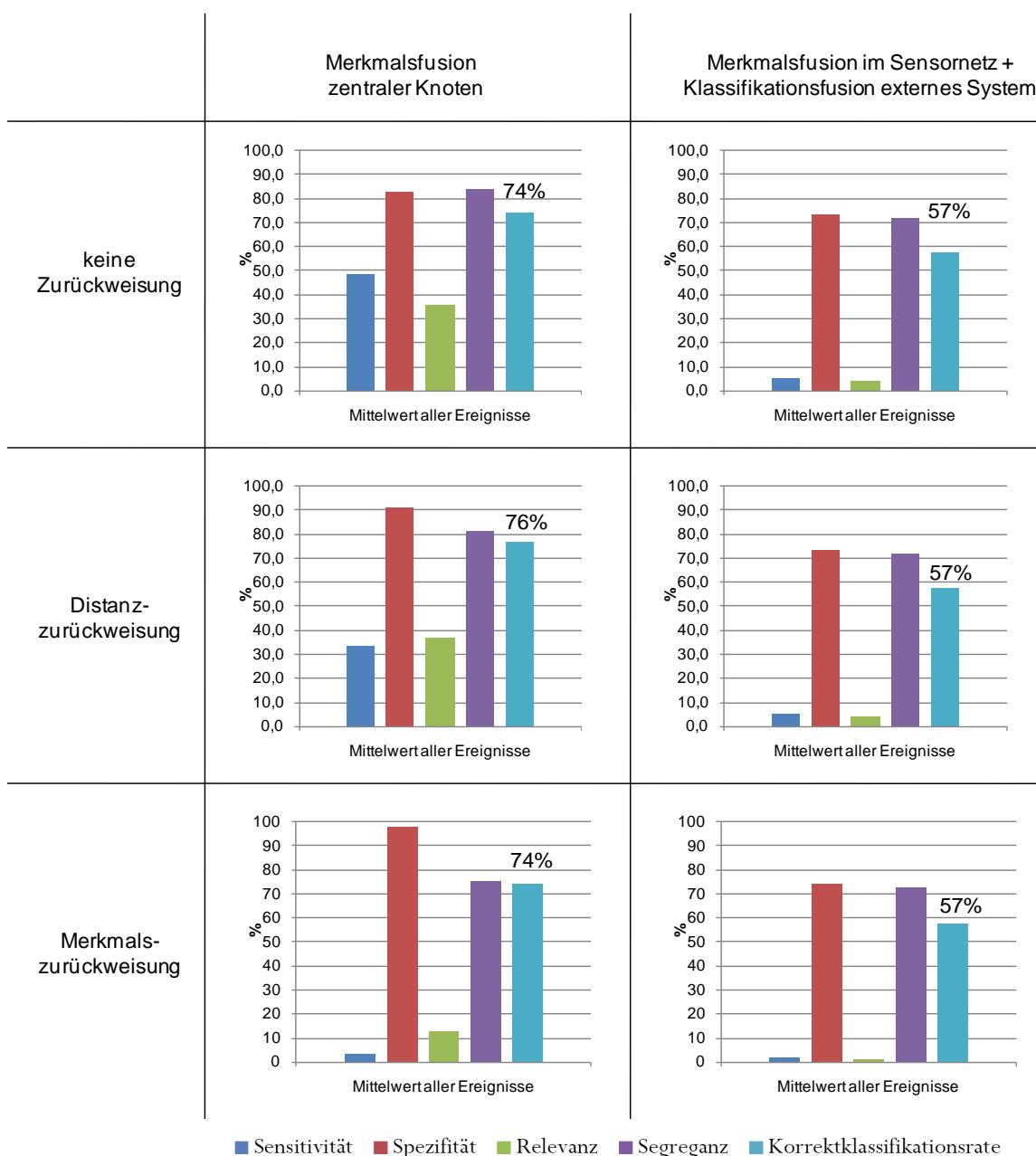
**Abbildung 6.32: Vergleich der Ergebnisse vom zentralen Knoten und externem System (ortsabhängig)**

#### 6.4.4.2 Ortsunabhängige Erkennung

Der Vergleich der Mittelwerte über alle Klassen zwischen zentralem Knoten und externem System soll für die ortsunabhängige Erkennung durchgeführt werden. Abbildung 6.33 zeigt, dass die Sensitivität und Relevanz für den Fall keiner Zurückweisung von 48,3% auf 5% bzw. 35,9% auf 3,7% zurückgeht. Während auf dem zentralen Knoten eine Korrektklassifikationsrate von noch 74% zu beobachten ist, sinkt diese auf dem externen System auf 57%. Dies ist bedingt durch die kaum noch vorhandene richtige Klassifikation von Ereignissen auf dem externen System. Bei der Distanzzurückweisung wird eine Korrektklassifikationsrate von 76% erreicht, auf dem externen System 57%. Die Werte für die Merkmalszurückweisung sind aufgrund zu geringer Sensitivität auf beiden Systemen zu vernachlässigen. Wie auch im ortsabhängigen Fall kann die Distanzzurückweisung auf-

grund der Werte auf dem externen System und der geringen Anzahl zu versendender Klassifikationen als geeignet angesehen werden.

Aufgrund der Ergebnisse, die durch die Klassifikationsfusion erreicht werden, ist in diesem Fall keine Ortsunabhängigkeit gegeben. Es können nur gute Ergebnisse erzielt werden, wenn die der Erkennung zugrunde liegenden Muster auch in etwa denen des Trainings entsprechen. Wenn sich durch andere Bewegungen des Bauzauns, obwohl das Ereignis gleich zum Training ausgeführt wird, aufgrund von äußerlichen Gegebenheiten wie Neigung, Bewegungseinschränkungen oder ähnliches andere Musterverläufe ergeben, dann können diese auch nicht mehr korrekt erkannt werden.



**Abbildung 6.33: Vergleich der Ergebnisse vom zentralen Knoten und externem System (ortsunabhängig)**

#### 6.4.4.3 Vergleich mit anderen Systemen

Abschließend soll ein Vergleich der Ergebnisse zu den Arbeiten von Wittenburg et. al [Wit07] und Norman Dziengel [Dzi07] erfolgen. In diesen beiden Arbeiten wurde die Erkennung nur ortsabhängig betrachtet. Es soll genügen, die Ergebnisse, die in der vorliegenden Arbeit durch Anwendung der Distanzrückweisung erreicht wurden, zu verwenden, da diese Methode sich als geeignet herausgestellt hat. Die in der Arbeit von Dziengel als zuverlässigste und genaueste angesehene Methode der Kooperativen Fusion mit Votorecht liefert die besten Ergebnisse. Die Erkennung wurde aber auch unter Laborbedingungen und anhand weniger Sensorknoten durchgeführt. Zudem kann bei der Arbeit jeder Sensorknoten das Ereignis richtig erkennen. Die zusätzlichen Anforderungen einer realen Umgebung, die durch die vorliegende Arbeit bewältigt werden müssen, führen zu einer deutlich geringen Erkennungsrate. In [Wit07] (Fence Monitoring) wurden nur die Ergebnisse für die Klasse Klettern dargestellt. Die geringe Spezifität in Verbindung mit der hohen Sensitivität lässt vermuten, dass diese Klasse sehr oft erkannt wird, auch wenn das Ereignis gar nicht vorliegt. Die höhere Sensitivität im Gegensatz zum Mittelwert des externen Systems hat somit nur eine geringe Aussagekraft. Durch die häufigen Fehlalarme ist beim Fence Monitoring nur eine geringe Relevanz sichtbar. Für die ortsabhängige Erkennung ist diese deutlich höher. Die Korrektklassifikationsrate ist für die ortsabhängige Klassifikationsfusion auf dem externen System niedriger als bei der Arbeit von Dziengel, aber um 16% höher als beim Fence Monitoring. Im Vergleich zu dieser Arbeit, welche auch an einem Bauzaun getestet wurde, lässt sich somit eine Verbesserung der Erkennung erkennen. Die Ergebnisse der ortsunabhängigen Erkennung ergeben in etwa dieselbe Korrektklassifikationsrate, diese ist aber aufgrund der geringen Sensitivität und Relevanz nicht mit dem Wert beim Fence Monitoring vergleichbar.

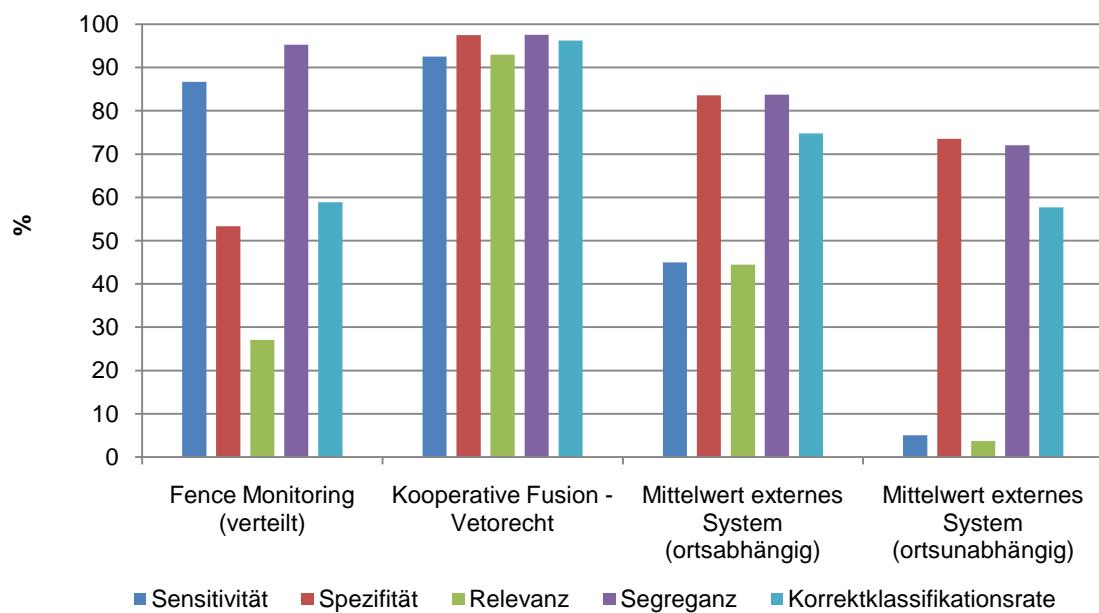


Abbildung 6.34: Vergleich mit anderen Systemen

## 7 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein System zur Erkennung von Ereignissen in drahtlosen Sensornetzwerken weiterentwickelt. Das zugrunde liegende Erkennungssystem kann Ereignisse lokal und verteilt klassifizieren. Für die lokale Entscheidungsfindung werden die Daten eines einzelnen Sensorknotens verwendet, im verteilten Fall werden die Daten mehrerer Sensorknoten zusammen bewertet. Die verteilte Entscheidungsfindung zeigt im ursprünglichen System höhere Erkennungsraten im Vergleich zum lokalen Fall. Das System weist aber Beschränkungen auf, die eine praktische Anwendbarkeit erschweren oder in manchen Fällen unmöglich machen. Solch ein Fall ist die Überwachung eines Bauzauns, welcher den Anwendungsfall dieser Arbeit darstellt. Die verteilte Entscheidungsfindung des Systems musste daher weiterentwickelt werden, um den entstehenden Anforderungen gerecht zu werden.

### 7.1 Zusammenfassung

Die verteilte Entscheidungsfindung wird auf den Sensorknoten anhand einer Merkmalsfusion durchgeführt. Zur Bewältigung der damit verbundenen hohen Dimension der Referenzvektoren wurde ein externes System entwickelt, welches anhand von gespeicherten Trainingsdaten eine automatische Merkmalsauswahl durchführt. Es wird eine Ortsunabhängigkeit der Ereignisse erreicht, indem die Merkmalsvektoren während der Erkennung von den Sensorknoten relativ zur eigenen Kennung angeordnet werden. Es wurden verschiedene Zurückweisungsmethoden eingeführt, wodurch falsche Klassifikationen verworfen werden sollen. Die Parameter dafür werden auch auf dem externen System berechnet. Durch die Verwendung einer Zurückweisung werden Fehlalarme von den Sensorknoten verhindert, die sich an Nachbarzaunelementen des Zaunelements befinden, an dem das Ereignis auftritt. Somit wird sichergestellt, dass ein Ereignis ohne Zuhilfenahme des externen Systems im Sensornetz verteilt klassifiziert werden kann. Des Weiteren kann zur Evaluierung der Funktionsweise der Zurückweisung eine Klassifikationsfusion auf dem externen System zum Einsatz kommen, die bei perfekter Funktionalität die gleiche Korrektklassifikationsrate leisten sollte wie das Sensornetz. Ort und Art des Ereignisses werden auf dem externen System auf einer Umgebungskarte angezeigt.

Um die Erkennungsraten des weiterentwickelten Systems zu testen, wurden Testreihen ortsabhängig an der Stelle des Trainings und ortsunabhängig an zwei anderen Stellen auf einer echten Baustelle mit 100 Sensorknoten aufgenommen. Für den ortsabhängigen Fall zeigt sich für die Erkennung auf dem zentralen Knoten eine Korrektklassifikationsrate von 88,3% ohne Zurückweisung, von 87,1% mit Distanzzurückweisung und von 84,6% mit Merkmalszurückweisung. Die Rate von 96,3% des dieser Arbeit zugrunde liegenden Systems [Dzi07] , bei welchem keine Zurückweisung umgesetzt ist, wird aber nicht erreicht. Dies ist durch den komplexeren Versuchsaufbau und die Auswirkungen der realen Umwelt zu begründen.

Im Sensornetz werden nicht nur vom zentralen Knoten, sondern auch von anderen Sensorknoten Klassifikationen gemeldet, die das zu erkennende Ereignis wahrgenommen haben. Nach Ausführung der Fusion dieser Klassifikationen sinkt die Korrektklassifi-

tionsrate, unter vorheriger Anwendung der oben genannten Zurückweisungsmethoden auf den einzelnen Sensorknoten, auf 74,8%, 74,8% sowie 73%. Dies ist damit zu begründen, dass die Klassifikationswerte des zentralen Knotens in manchen Fällen nicht gut genug sind und Entscheidungen anderer Knoten besser eingeschätzt werden. Im Vergleich zu der verteilten Erkennung in [Wit07] ist eine Steigerung um 15,9 Prozentpunkte erreicht worden. Dabei ist kein Unterschied zur Distanzzurückweisung und zur Anwendung keiner Zurückweisung zu beobachten.

Für den ortsunabhängigen Fall liegen die Korrektklassifikationsraten für die verschiedenen Zurückweisungsmethoden für den zentralen Knoten bei 74,2%, 76,7% und 74,2%. Hier lässt sich erkennen, dass eine Änderung des Ortes eine Verschlechterung der Erkennungsrate zur Folge hat. Die andere Umgebungssituation verursacht eine Veränderung in der Bewegung des Bauzauns, welche sich zum Teil von der Bewegung während des Trainings unterscheidet. Nach der Klassifikationsfusion sind so gut wie keine positiven Erkennungen mehr zu verzeichnen, da es fast immer einen Sensorknoten gibt, der eine bessere Klassifikation als der zentrale Knoten liefert.

Es muss festgestellt werden, dass die Anwendung der verteilten Ereigniserkennung in der Praxis schlechtere Ergebnisse als unter Laborbedingungen liefert. Dies war auch zu erwarten, da sich die Ereignisse in ihrer Art deutlich voneinander unterscheiden und in der Praxis häufig Schwankungen ausgesetzt sind. Somit wird eine Unterscheidung der verschiedenen Ereignisse erschwert. Für den ortsabhängigen Fall wird im Vergleich zu Fence Monitoring eine Steigerung der Erkennungsgenauigkeit erreicht. Es zeigt sich weiterhin, dass die Klassifikationsparameter nicht robust genug gegen den inkonsistenten Zaunaufbau sind, damit eine ortsunabhängige Erkennung erfolgreich durchgeführt werden kann. Das ist nicht auf einen Fehler im System zurückzuführen, sondern auf die Tatsache, dass sich die wahrgenommene Bewegung im Zuge der Umgebungsänderung auch verändert.

## 7.2 Weiterführende Arbeiten

Um die Erkennungsrate im Allgemeinen zu optimieren, kann überprüft werden, wie sich ein umfangreicheres Training auswirkt. Auch die Erhöhung der Samplingfrequenz kann Vorteile bringen, da hiermit eine bessere Darstellung der aufgenommenen Muster erzielt werden kann. Es ist zu prüfen, ob dies mit den bestehenden Speicher- und Rechenressourcen möglich ist oder ob eine leistungsfähigere Hardware notwendig ist. Durch eine größere Anzahl an Trainingsdaten lassen sich auch die Zurückweisungsgrenzen besser definieren. Besonders die Merkmalszurückweisung sollte dann getestet werden. Wenn diese für den zentralen Knoten durch bessere Anpassung der Merkmalsgrenzen gute Ergebnisse liefert, können die Ergebnisse der Klassifikationsfusion aufgrund der hohen Restriktion der Methode deutlich ansteigen.

Die Bewertung der Knoten, welche während des Trainings für eine Klasse eine unzureichende Anzahl an Daten liefern, sollte durch Testverfahren überprüft werden. Dafür ist es wichtig festzustellen, welche Gründe für das Ausbleiben der Daten vorliegen. Danach

kann entschieden werden, wie mit den Daten des Knotens verfahren wird oder ob die Daten solcher Knoten überhaupt weiterhin verwendet werden sollten.

Die Datenübertragung zwischen den Sensorknoten kann durch zusätzliche Verfahren gesichert werden. Eine Möglichkeit ist, mit Bestätigungen zu arbeiten. Dies erfordert Nachbarschaftsprotokolle, weil sonst sehr viele Bestätigungen versendet werden. Es ist zu überprüfen, inwieweit sich verschiedene Verfahren auf Kommunikationsaufwand, Zeitbedarf und Übertragungsfehlerrate auswirken. Für die Übertragung der Klassifikationsparameter und Klassifikationsergebnisse durch das Sensornetz könnte ein Routingverfahren eingesetzt werden, damit nicht jeder Knoten die Pakete weiterleiten muss, sofern häufig Ereignisse erkannt werden müssen.

Der Kommunikationsaufwand kann dadurch reduziert werden, indem während der Merkmalsverteilung der Erkennungsphase nur die Merkmale übertragen werden, welche auch von mindestens einem Knoten für die Erkennung verwendet werden. Zusätzlich wird der Rechenaufwand verringert, wenn nur noch die zu versenden Merkmale berechnet werden.

Die Ortsunabhängigkeit wurde bisher nur anhand einer merklich veränderten Umgebungssituation getestet. Es sollte untersucht werden, wie sich eine Umgebung ähnlich der des Trainings auf die Erkennungsrate auswirkt. Werden gute Ergebnisse erzielt, kann als Bedingung gesetzt werden, dass der Aufbau des Bauzauns möglichst gleichförmig sein sollte. Dadurch ist an allen Stellen eine annähernd gleiche Bewegung des Bauzauns gegeben. Eine andere Möglichkeit ist, das Training an verschiedenen Stellen des Bauzauns durchzuführen, die jeweils eine andere Aufbausituation darstellen, so dass alle möglichen Bewegungsverhalten abgedeckt sind. Durch dieses Vorgehen kann die Anzahl der zu trainierenden Klassen stark ansteigen, was zu einem Speicher- und Rechenproblem auf den Sensorknoten führen könnte.

Durch das entwickelte System ist die Möglichkeit gegeben, Ereignisse an einem beliebigen Punkt des Netzwerks verteilt zu erkennen. Erstmals konnten mit einem aus 100 Knoten bestehenden Sensornetzwerk Erkenntnisse über die verteilte Ereigniserkennung gewonnen werden. Durch eine Optimierung des Systems lassen sich weitere Fortschritte in der verteilten Ereigniserkennung und ein großer Nutzen für die Gesellschaft erzielen.



## 8 Literaturverzeichnis

- [Aky02] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E. *Wireless sensor networks: A survey*. Computer Networks, 38: 393-422, 2002.
- [Bak96] Baker, S., Nayar, S. K. *Algorithms for pattern rejection*. Proceedings of the 13th International Conference on Pattern Recognition, 2(B): 869-874, 1996.
- [Bha02] Bharathidasan, A., Anand, V. *Sensor networks: An overview*. Technical Report, Dept. of Computer Science, University of California at Davis, 2002.
- [Che76] Chen, C.H. *On information and distance measures, error bounds, and feature selection*. Information Sciences, 10: 159–173, 1976.
- [Chi05] Chipcon. 2005. CC1020 datasheet. Oslo : Chipcon AS, 2005.
- [Cho03] Chong, C-Y., Kumar, S. P. *Sensor Networks: Evolution, Opportunities, and Challenges*. Proceedings of the IEEE, 91(8): 1247-1256, 2003.
- [Das00] Dash, M., Huan, L., Hiroshi, M. *Consistency Based Feature Selection*. Lecture Notes In Computer Science, 1805: 98-109.
- [Dev82] Devivjer, P. A., Kittler, J. *Pattern Recognition: A Statistical Approach*. Prentice Hall, London, 1982.
- [Dua04] Duarte, M.F., Hu, Y. H. *Vehicle Classification in Distributed Sensor Networks*. Journal of Parallel and Distributed Computing, 64(7): 826-838.
- [Dud01] Duda, R. O., Hart, P. E., Stork, D. G. *Pattern Classification*. Wiley, New York, 2001.
- [Dzi07] Dziengel, N. *Verteilte Ereigniserkennung in Sensornetzen*. Freie Universität Berlin, Institut für Informatik, 2007.
- [Fer03] Ferber, R. *Information Retrieval*. dpunkt.verlag, Heidelberg, 2003.
- [Fre06] FreeScale. <http://www.freescale.com/>.  $\pm 1.5g$  -  $6g$  Three Axis Low-g Micromachined Accelerometer. *Technische Daten*. 2007. [Online]
- [Fuk90] Fukunaga, K. *Introduction to Statistical Pattern Recognition*. Academic Press, London, 1990.
- [Gu05] Gu, L., Jia, D., Vicaire, P. , Yan, T. , Luo, L., Tirumala, A. , Cao, Q., He, T., Stankovic, J. , Abdelzaher, T., Krogh, B. *Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments*. Proceedings of the 3rd international conference on Embedded networked sensor systems, 205-217, 2005.
- [Gut2] Gutierrez-Osuna, R. *Lecture on "Intelligent Sensor Systems"*. Wright State University.

[Gut] —. Texas A&M University. "Lecture on „Introduction to Pattern Analysis“.

[Hal97] Hall, M.A., Smith, L.A. *Feature Subset Selection: A Correlation Based Filter Approach*. International Conference on Neural Information Processing and Intelligent Information Systems, 855-858, 1997 : s.n.

[Hei04] Heijden, F. van der, Ridder, D. de, Tax, D. M. J. *Classification, parameter estimation and state estimation*. Wiley, London, 2004.

[Kal01] Kalton, A., Langley, P., Wagstaff, K., Yoo, J. *Generalized Clustering, Supervised Learning, and Data Assignment*. Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 299-304, 2001.

[Kle93] Klein, L.A. *Sensor and Data Fusion Concepts and Applications*. SPIE Optical Engineering Press, 1993.

[Lli90] Llinas, J., Hall, D.L. *Multisensory Data Fusion*. Artech House Inc., 1990.

[Mou06] Mouchere, H., Anquetil, E. *A Unified Strategy to Deal with Different Natures of Reject*. Proceedings of the 18th International Conference on Pattern Recognition, 2: 792-795, 2006.

[Nar77] Narendra, P. M., Fukunaga, K. *A branch and bound algorithm for feature subset selection*. IEEE Transactions on Computers, 26:917–922, 1977.

[Nie83] Niemann, H. *Klassifikation von Mustern*. Springer Verlag, Berlin, 1983.

[Pud94] Pudil, F., Ferri, F., Novovicova, J., Kittler, J. *Floating search methods for feature selection with nonmonotonic criterion functions*. Proceedings of the International Conference on Pattern Recognition, 2: 279–283, 1994.

[Rus07] Ruser, H., Puente-Leon, F. *Informationsfusion - Eine Übersicht*. Technisches Messen, 74(3): 93-102, 2007.

[Sch05] Schiller, J., Liers, A., Ritter, H. 2005. *ScatterWeb: A wireless sensor network platform for research and teaching*. 2005. Computer Communications, 28(13): 1545-1551, 2005.

[Ste76] Stearns, S.D. *On selecting features for pattern classifiers*. Proceedings of the Third International Conference on Pattern Recognition, 71–75, 1976.

[Tav05] Tavakoli, A., Zhang, J., Son, S. H. *Group based event detection in undersea networks*. Proceedings of the 2nd International Workshop for Networked Sensing Systems, 2005.

[Tex06] Texas Instruments. *MSP430F1612 datasheet (slas368e) and user's guide (slau049f)*. Texas Instruments Inc., Dallas, 2006.

[The03] Theodoridis, S., Koutroumbas, K. *Pattern Recognition*. 4th ed., Elsevier, 2003.

[Tub03] Tubaishat, M., Madria, S. *Sensor networks: an overview*. IEEE Potentials, 22(2): 20-23, 2003.

[Wan03] Wang, H., Elson, J., Girod, L., Estrin, D., Yao, K. *Target classification and localization in habitat monitoring*. s.l.: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, 2003.

[Web02] Webb, A. *Statistical Pattern Recognition*. 2nd ed., Wiley, 2002.

[Whi71] Whitney, A. W. *A direct method of nonparametric measurement selection*. IEEE Transactions on Computers, 20: 1100–1103, 1971.

[Wit07] Wittenburg G., Terfloth K., Villafuerte, F. L., Naumowicz, T., Ritter, H., Schiller, J. 2007. *Fence Monitoring - Experimental Evaluation of a Use Case for Wireless Sensor Networks*. 2007. Proceedings of the Fourth European Conference on Wireless Sensor Networks (EWSN '07), 163-178, 2007.

[Yan08] Yang, A.Y., Iyengar, S., Sastry, S., Bajcsy, R., Kuryloski, P., Jafari, R. *Distributed Segmentation and Classification of Human Actions Using a Wearable Motion Sensor Network*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 1-8, 2008.

[Yu03] Yu, L., Liu, H. *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*. Proceedings of The Twentieth International Conference on Machine Learning (ICML-03), 856-863, 2003.



## **Erklärung**

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig angefertigt und ohne fremde Hilfe verfasst habe, keine außer den von mir angegebenen Hilfsmitteln und Quellen dazu verwendet habe und die den benutzten Werken inhaltlich oder wörtlich entnommen Stellen als solche kenntlich gemacht habe.

Berlin, den 19. Dezember 2008

Christian Wartenburger



# Aufgabenstellung

## Experimentelle Analyse verteilter Ereigniserkennung in Sensornetzen

### Überblick

Die Verarbeitung und Aggregation von Daten ist ein integraler Bestandteil der Vision von drahtlosen Sensornetzen, da man sich von der Reduktion des Datenvolumens eine längere Lebensdauer des Netzes verspricht. In den vergangenen Jahren wurden unterschiedlichste Ansätze zur Datenvorverarbeitung in Sensornetzen vorgestellt, und am Fachbereich wurde ein auf Mustererkennung basierendes System zur verteilten Ereigniserkennung entwickelt. Das System arbeitet vollständig autark, also ohne Basisstation oder andersartige externe Rechenleistung, und erzielt unter Laborbedingungen eine Erkennungsgenauigkeit, die weltweit führend ist.

Ein Sensornetz zur verteilten Ereigniserkennung kann in verschiedenen Anwendungsszenarien zum Einsatz kommen. Eines dieser Szenarien ist die Überwachung eines Areals mit Hilfe von am Begrenzungszaun befestigter Sensorknoten, die mit einem Beschleunigungssensor ausgestattet sind. Anhand der gemessenen Beschleunigungswerte kann das Sensornetz feststellen, ob ein sicherheitsrelevantes Ereignis stattgefunden hat, wie beispielsweise das Überklettern des Zaunes. Hierbei ist eine ausreichende Genauigkeit der Ereigniserkennung des Gesamtsystems erforderlich, damit das Sensornetz zuverlässig und unbeaufsichtigt seine Aufgabe erfüllen kann.

Ziel dieser Arbeit ist es, in ein bestehendes System zur verteilten Ereigniserkennung bis zur Praxistauglichkeit weiterzuentwickeln. Dies beinhaltet sowohl die Anpassung und Erweiterung der Mustererkennungsalgorithmen, als auch die Integration von netzwerktechnischen Komponenten, wie dem Routing von erkannten Ereignissen zu einer Basisstation. Die Leistungsfähigkeit des zu erstellende Sensornetzes soll anhand eines realistischen Anwendungsfalles experimentell beurteilt werden.

### Aufgaben

Eine Diplom- oder Masterarbeit in diesem Bereich umfasst folgende Punkte:

- Einarbeitung in die ScatterWeb Plattform und den bestehenden Ansatz zur verteilten Ereigniserkennung
- Anpassung und Erweiterung der verteilten Ereignis- bzw. Mustererkennung für den Anwendungsfall, sowie Integration von netzwerktechnischen Komponenten
- Experimentelle Evaluation der verteilten Ereigniserkennung bezüglich ihrer Genauigkeit und der Laufzeiteigenschaften des Sensornetzes anhand eines groß angelegten Feldversuches
- Implementierung einer graphischen Benutzeroberfläche, die die vom Sensornetz erkannten Ereignisse im Bezug auf das zu überwachende Areal anzeigt (optional)

### Umfeld

Verteilte Ereigniserkennung in Sensornetzen ist derzeit ein aktuelles Thema in der Forschung. Die grundsätzliche Möglichkeit der Ereigniserkennung mit Hilfe der ScatterWeb Plattform wurde am Fachbereich bereits gezeigt, die Vorgehensweise ist gut dokumentiert und auf der Fence Monitoring Homepage beschrieben. Für den praktischen Teil der Arbeit stehen ScatterWeb Sensorknoten zur Verfügung, die am Fachbereich entwickelt wurden und ebenfalls auf der ScatterWeb Homepage voll dokumentiert sind.

Zur Vernetzung der Aktivitäten der Arbeitsgruppe gibt es regelmäßige Treffen aller Studien- und Diplomarbeiter sowie der wissenschaftlichen Mitarbeiter der Arbeitsgruppe montags von 10:00 bis 12:00 Uhr.

Bei Interesse entweder bei Georg Wittenburg ([wittenbu@inf.fu-berlin.de](mailto:wittenbu@inf.fu-berlin.de)) melden oder einfach in Raum 150 vorbeikommen.