



Verteilte Ereigniserkennung in Sensornetzen

Verteidigung der Diplomarbeit

Norman Dziengel

dziengel@inf.fu-berlin.de

10. Dezember 2007



Agenda

- Motivation, Grundlagen
- Verwandte Arbeiten
- Bewegungsmuster
- Lokale Erkennung, Klassifizierer
- Verteilte Erkennung
- Auswertung
- Zusammenfassung, Ausblick
- Demo



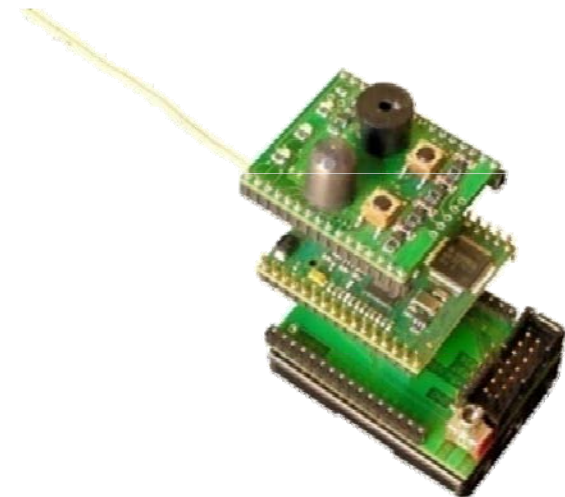
Motivation

- Sensornetzpotential nutzen (von zentraler zur dezentralen Auswertung)
- Sensornetze sollen autark (netzintern) und **verteilt** Ereignisse erkennen
 - ⇒ verringerte Sendelast
 - ⇒ verbesserte Ereigniserkennung im Vgl. zur lokalen Erkennung (?)
- Hardwareredundanz soll in der verteilten Ereigniserkennung wertschöpfend genutzt werden
- Optimierung der Erkennungsraten im Vgl. zum Fence Monitoring



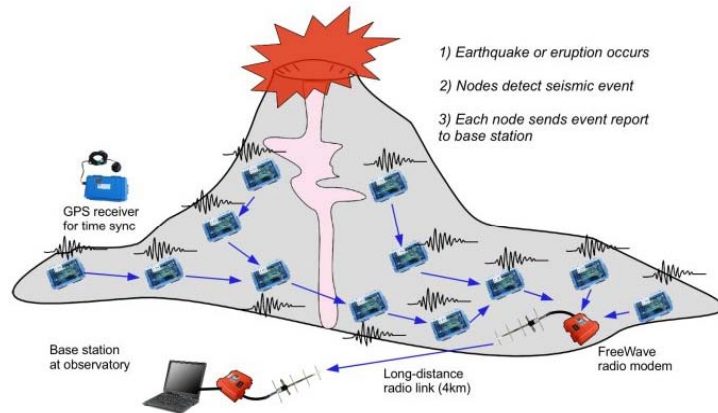
Grundlagen

- Scatterweb MSB-430 inkl. MSB-S, MSB-T
- Bewegungsmuster mit Daten des Beschleunigungssensors MMA7260 erkennen
- Automatische Kalibrierung des Sensors
 - Beschleunigung, Zitterbereich
- Anpassungsfähig
 - Variable Musteranzahl, Mustertypen, Merkmale
- Automatisiertes Training
- Lokale Ereigniserkennung ist Grundlage für verteilte Ereigniserkennung

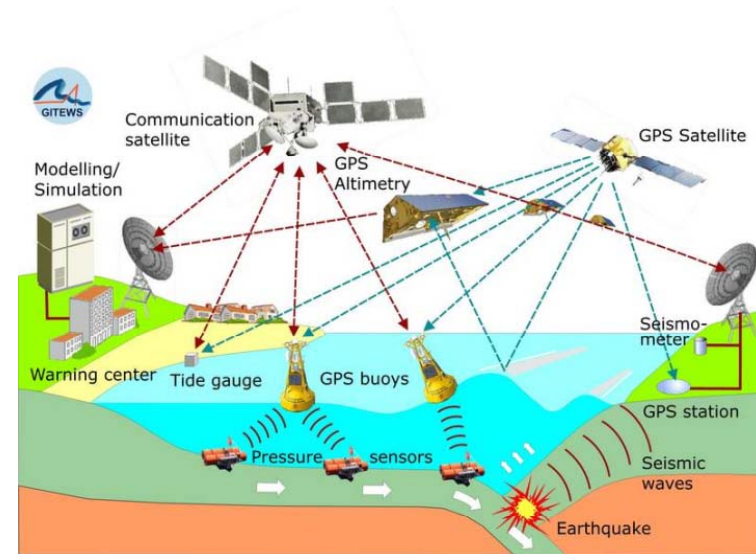




Verwandte Arbeiten / Zentrale Entscheidung



Überwachung von Vulkanaktivitäten [Werm06]



GITEWS [Rud06]

Quellen:

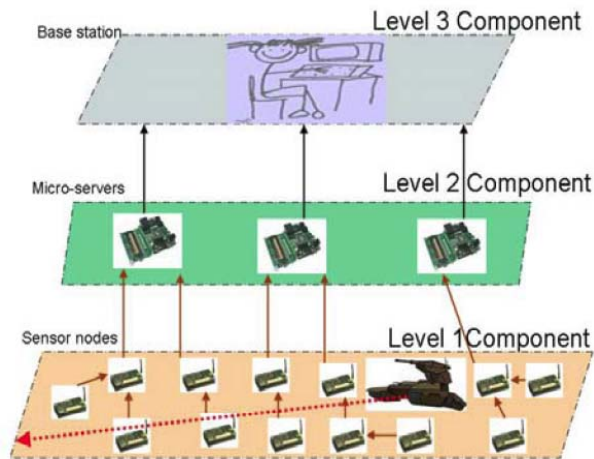
[Rud06] Rudloff, A., Lauterjung, J., Zschau, J. Der Deutsche Beitrag zur Einrichtung eines Tsunami-Frühwarnsystems. In: Notfallvorsorge (Walhalla-Verlag), Heft 1/2006, S. 10-12, 2006.

[Werm06] WernerAllen, G. Monitoring Volcanic Eruptions with a Wireless Sensor Network. 2006. <http://www.eecs.harvard.edu/~mdw/proj/volcano/>

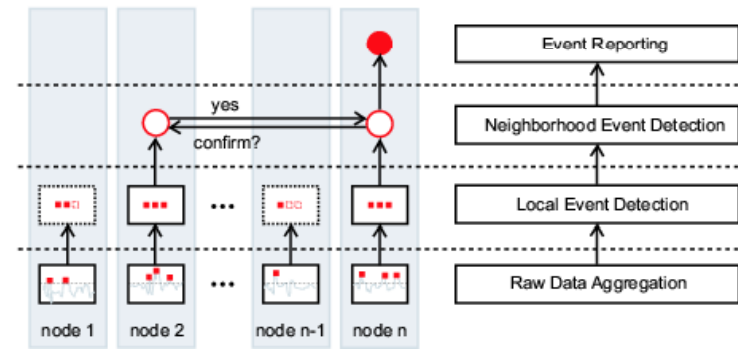
[Jaf06] Jafari, R., Noshadi, H., Ghiasi, S., Sarrafzadeh, M. Adaptive Medical Feature Extraction for Resource Constrained Distributed Embedded Systems. In: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops, Page: 506, 2006.



Verwandte Arbeiten / Dezentrale Entscheidung



Battlefield Surveillance [Bok06]



Fence Monitoring [Wit07]



SensIT [Dua07]

Quellen:

- [Bok06] Bokareva, T., Hu, W., Kanhere, S., Ristic, B., Gordon, N., Bessel, T., Rutten, M., Jha, S. Wireless Sensor Networks for Battlefield Surveillance. In: Proceedings of The Land Warfare Conference, LWC, 2006.
- [Dua07] Duarte, M. Sensor Networks Research Group, 2007. Electrical and Computer Engineering Department University of Wisconsin-Madison, 2007.
- [Wit07] Wittenburg G., Terfloth K., Villafuerte, F. L., Naumowicz, T., Ritter, H., Schiller, J. Fence Monitoring - Experimental Evaluation of a Use Case for Wireless Sensor Networks. In: European Workshop on Wireless Sensor Networks, Delft, Niederlande, 2007



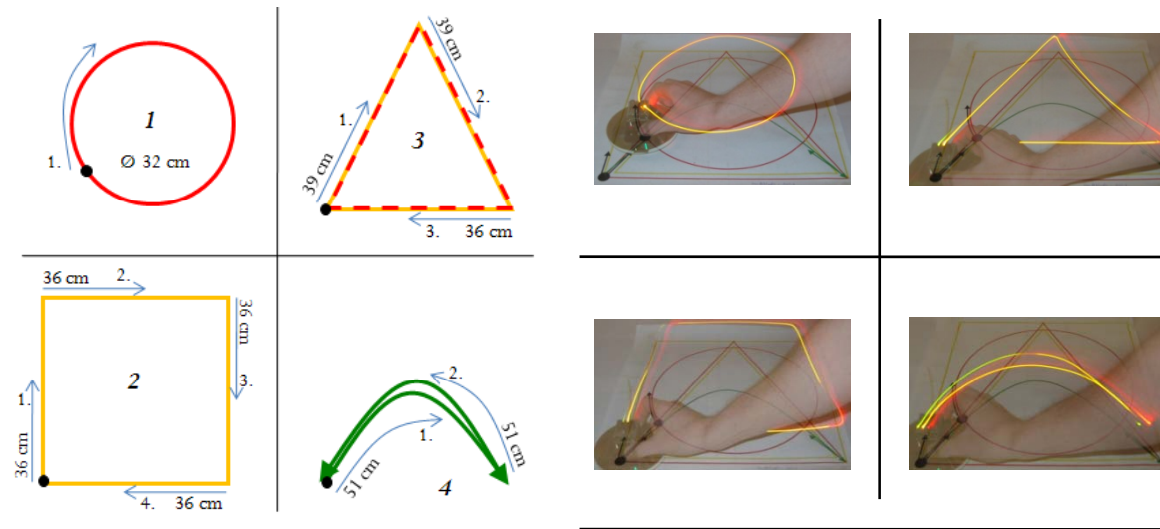
Agenda

- ✓ Motivation, Grundlagen
- ✓ Verwandte Arbeiten
- Bewegungsmuster
- Lokale Erkennung, Klassifizierer
- Verteilte Erkennung
- Auswertung
- Zusammenfassung, Ausblick
- Demo



Bewegungsmuster

- Muster sollen einfach sein => ermöglicht hohe Versuchszahl
- Muster sollen nachvollziehbar sein => geometrische 2D-Muster
- Vorrichtung zur Stabilisierung des Sensorknotens
- Musterdefinition: gleichförmige Bewegung ohne Pause, ohne Eigenrotation

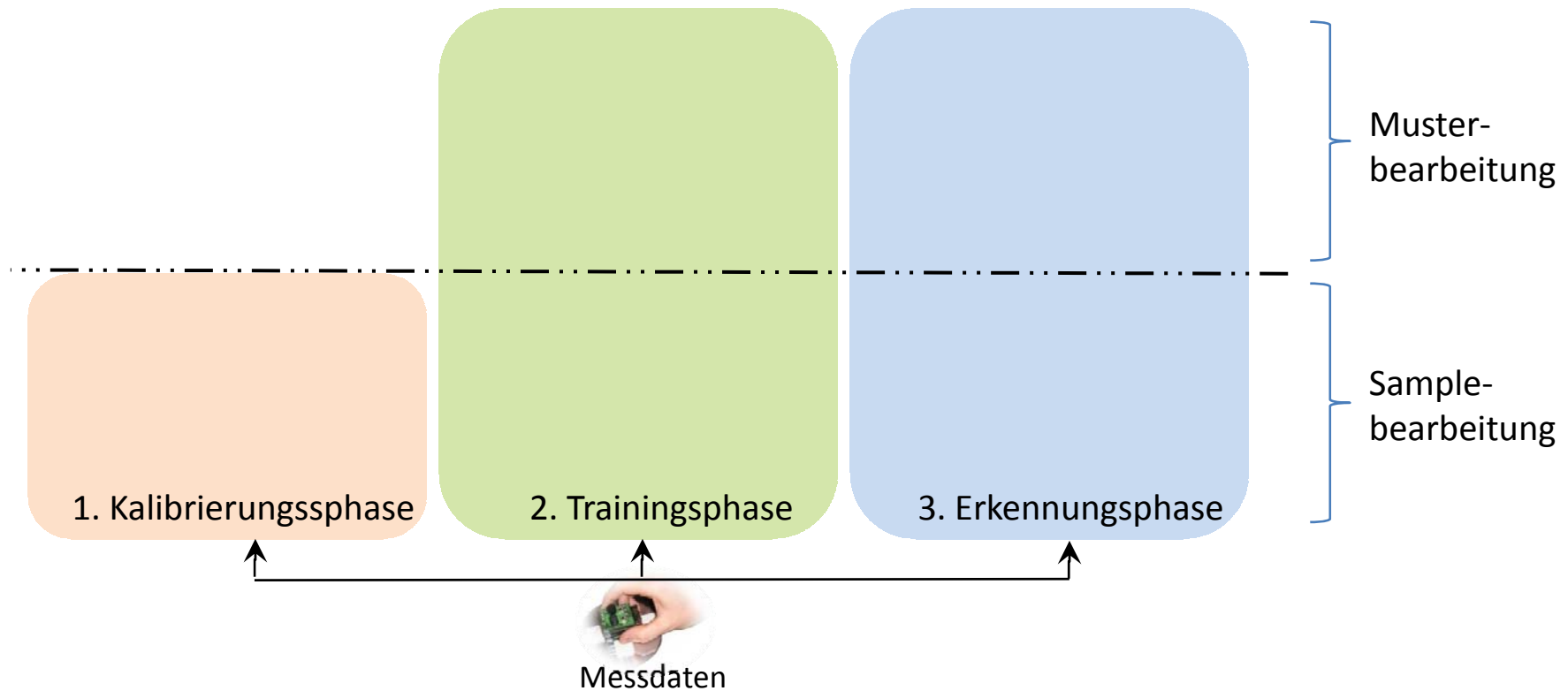


Musterdefinitionen

Ausführung der Musterbewegungen
auf gedruckter Vorlage

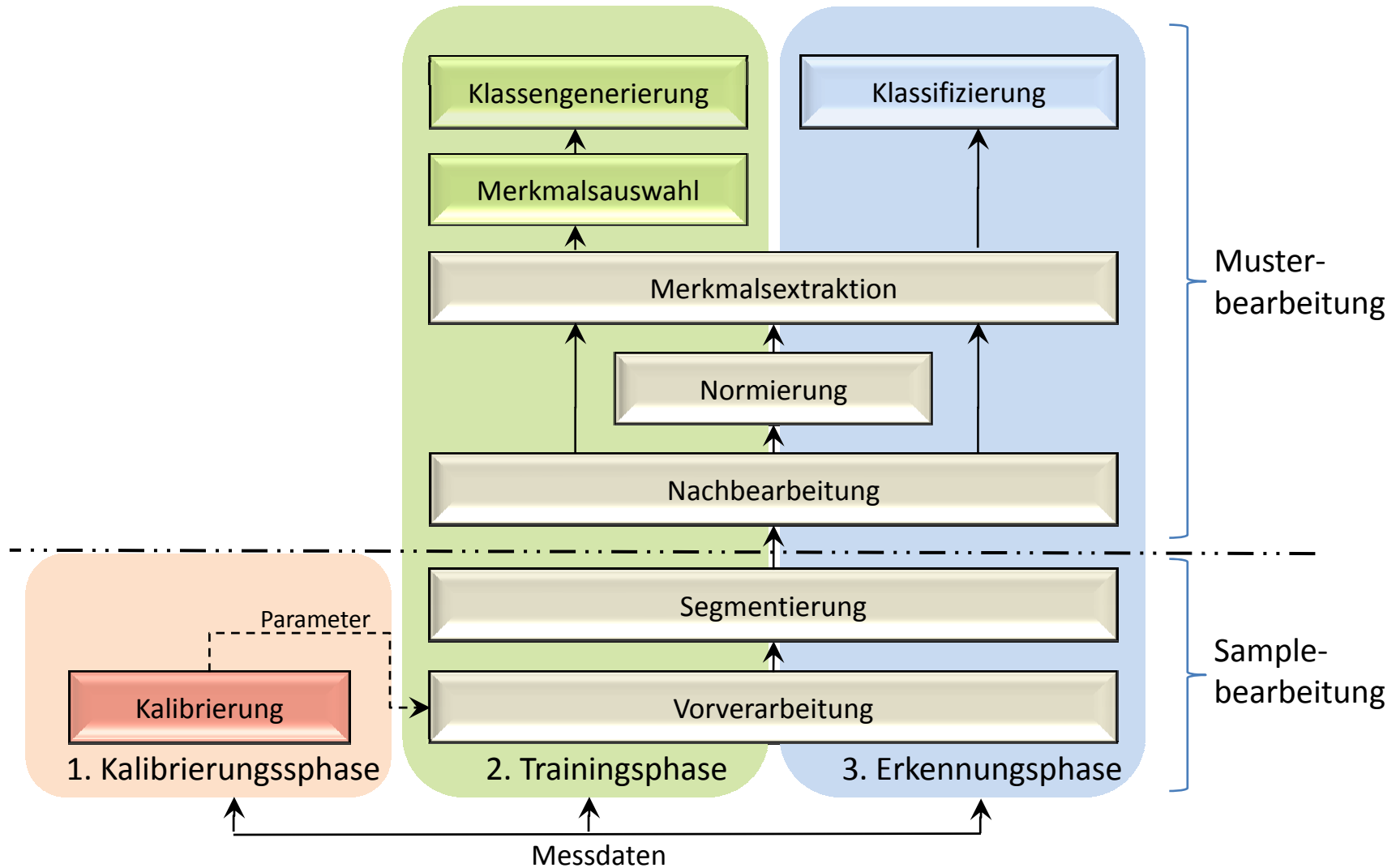


Lokale Erkennung - Grobkonzept





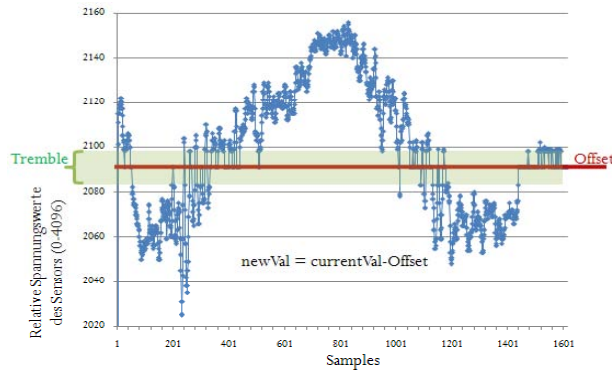
Lokale Erkennung - Konzept



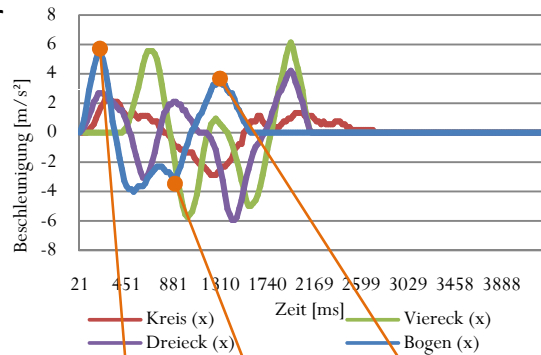


Lokale Erkennung – Datenverarbeitung

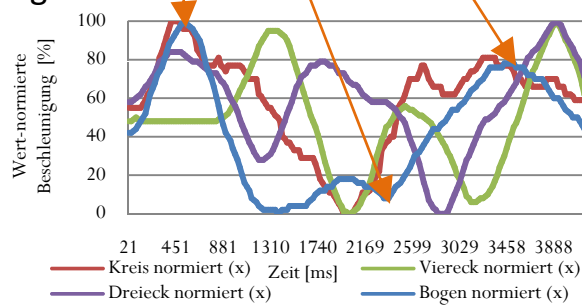
1. Rohdaten



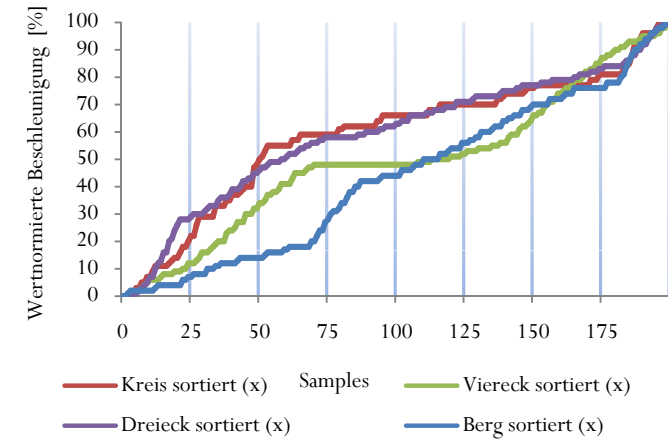
2. Tiefpassfilter



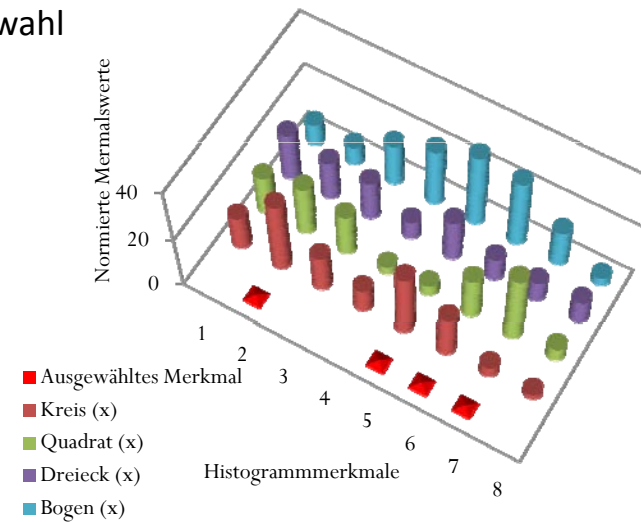
3. Normierung



4. Merkmalbildung



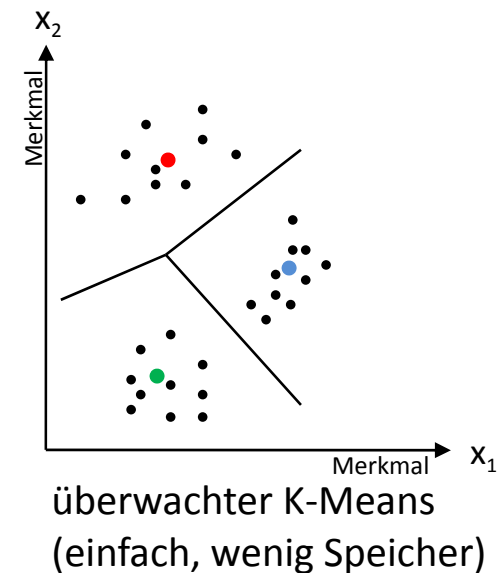
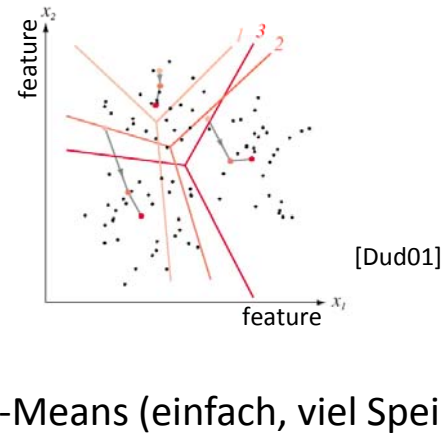
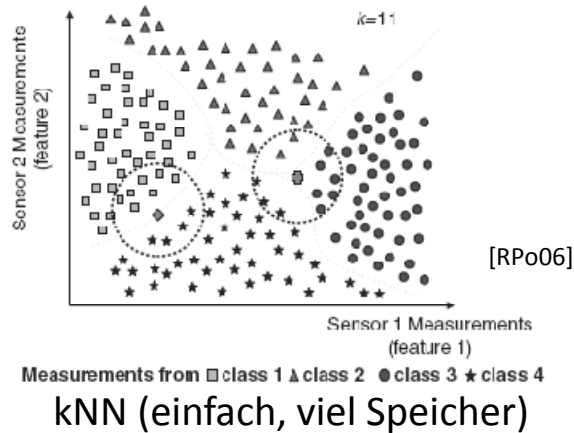
5. Merkmalsauswahl





Klassengenerator / Klassifizierer

- Alle Informationen für eine Klassifizierung auf einem Sensor
=> Trainingsdaten + Musterdaten
- Einfacher Algorithmus mit minimaler Fließkommazahlnutzung
- Leicht skalierbarer Algorithmus (Klassen, Merkmale)
=> Euklidischer Abstand geeignet (einfach + skalierbar)



Quellen:

[RPo06] Polikar, R. Pattern Recognition. In: Wiley Encyclopedia of Biomedical Engineering, 2006.

[Dud01] Duda, R. O., Hart, P. E., Stork, D. G. Pattern Classification. In: A Wiley-Interscience, 2001.



Agenda

- ✓ Motivation, Grundlagen
- ✓ Verwandte Arbeiten
- ✓ Bewegungsmuster
- ✓ Lokale Erkennung, Klassifizierer
- Verteilte Erkennung
- Auswertung
- Zusammenfassung, Ausblick
- Demo



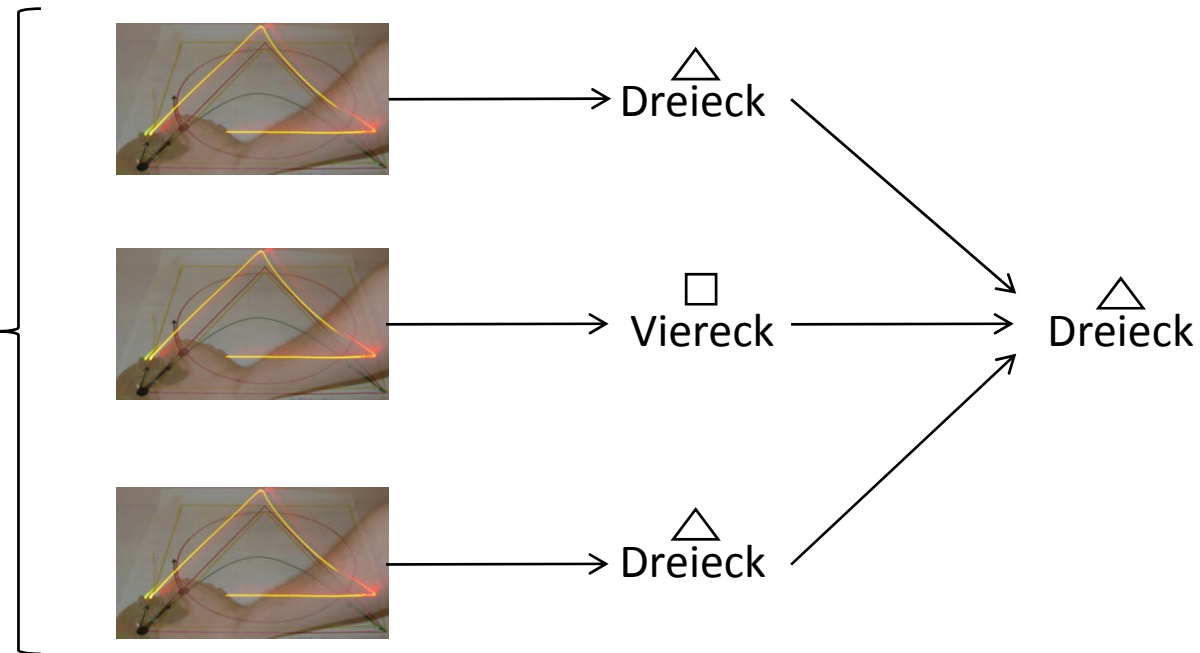
Verteilte Erkennung - Beispiel

Musterbewegung:

Klassifizierung:

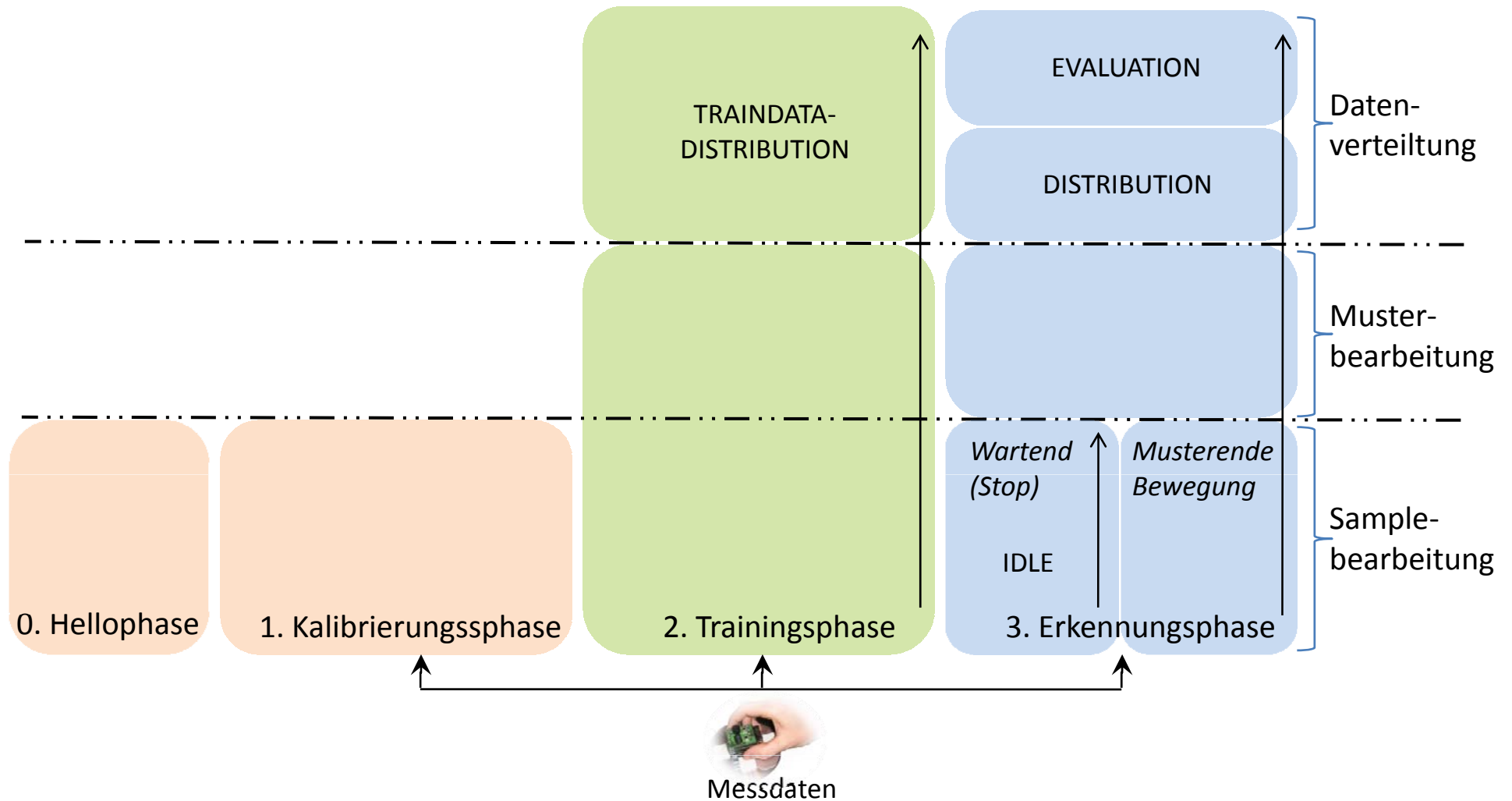
Verteilte Auswertung:

Beispielereignis:
Drei Dreiecke



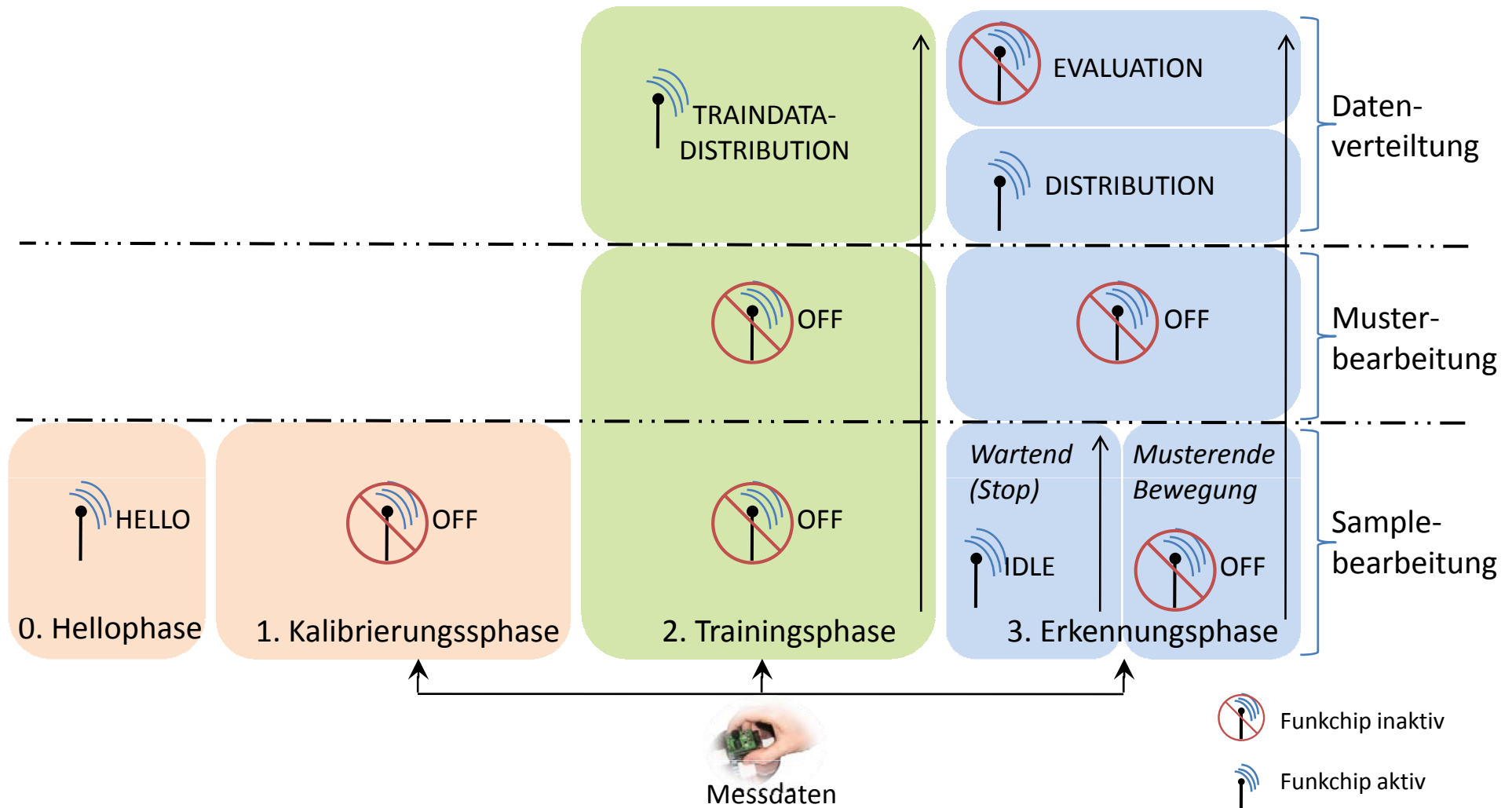


Verteilte Erkennung - Konzept





Verteilte Erkennung - Funkkonzept

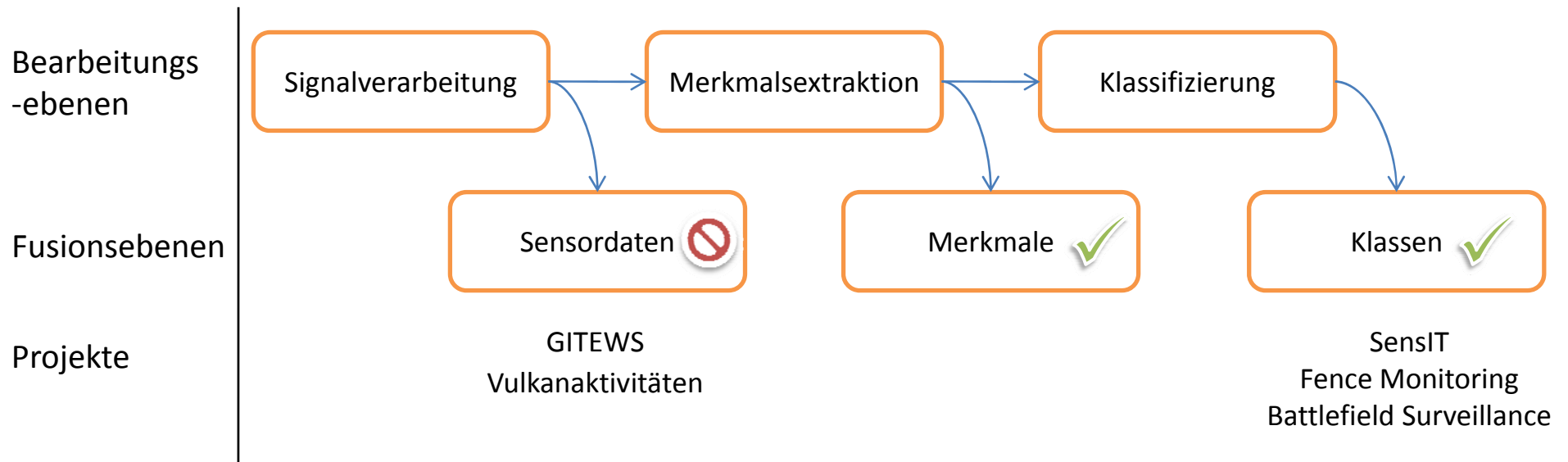




Verteilte Erkennung - Fusionsebenen

Omnibusmodell:

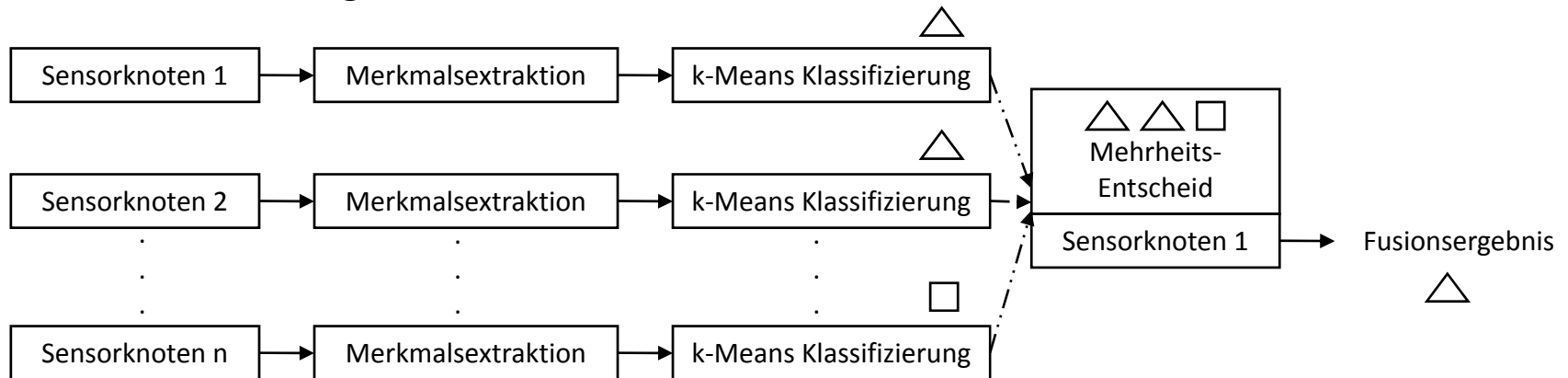
- Verwendet Fusionsebenen um verteilte Ansätze zu definieren



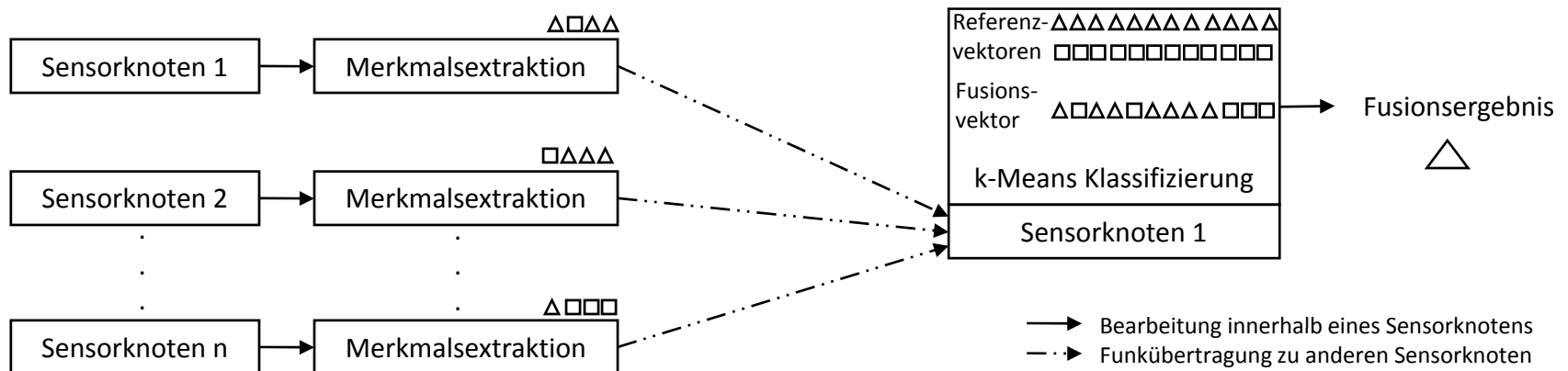


Verteilte Erkennung - Fusionsmethoden

Methode 1: Klassifizierungsfusion



Methode 2: Merkmalsfusion



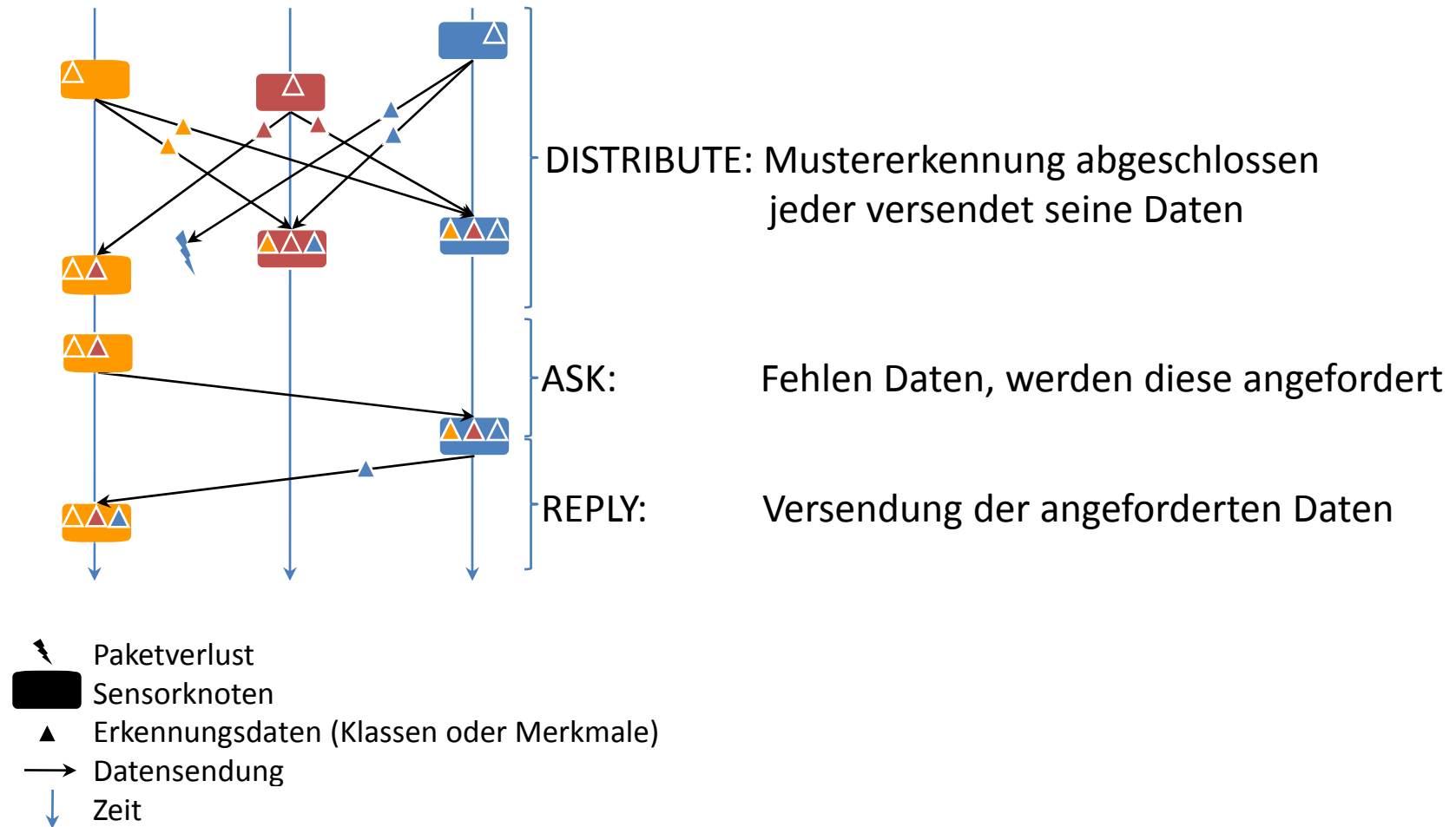


Verteilte Erkennung – Erweiterte Fusionsmethoden

- Methode 1 – Klassifikationsfusion $\triangle \triangle \square$
 - Geringer Kommunikationsaufwand (Klassen werden ausgetauscht 1 Byte)
 - Mehrheitsentscheid
- Methode 2 – Merkmalsfusion $\triangle \square \triangle \square \triangle \triangle \triangle \square \square \square$
 - Kommunikationsaufwand hängt von Merkmalszahl ab
 - Fusion aller Merkmale führt zur Berechnung einer gemeinsamen Klasse
- Methode 3 – Kooperative Fusion (nutzt Methode 1 und 2)
 - Keine eindeutige Mehrheit in der Klassifikationsfusion $\triangle \circ \square \Rightarrow \triangle \square \triangle \square \triangle \circ \square \square \square$
 - \Rightarrow Merkmalsfusion
- Methode 4 – Kooperative Fusion mit Vetorecht (nutzt Methode 1 und 2)
 - Bei minimaler Uneinigkeit der Sensorknoten $\triangle \triangle \square \Rightarrow \triangle \square \triangle \square \triangle \triangle \triangle \square \square \square$
 - \Rightarrow Merkmalsfusion

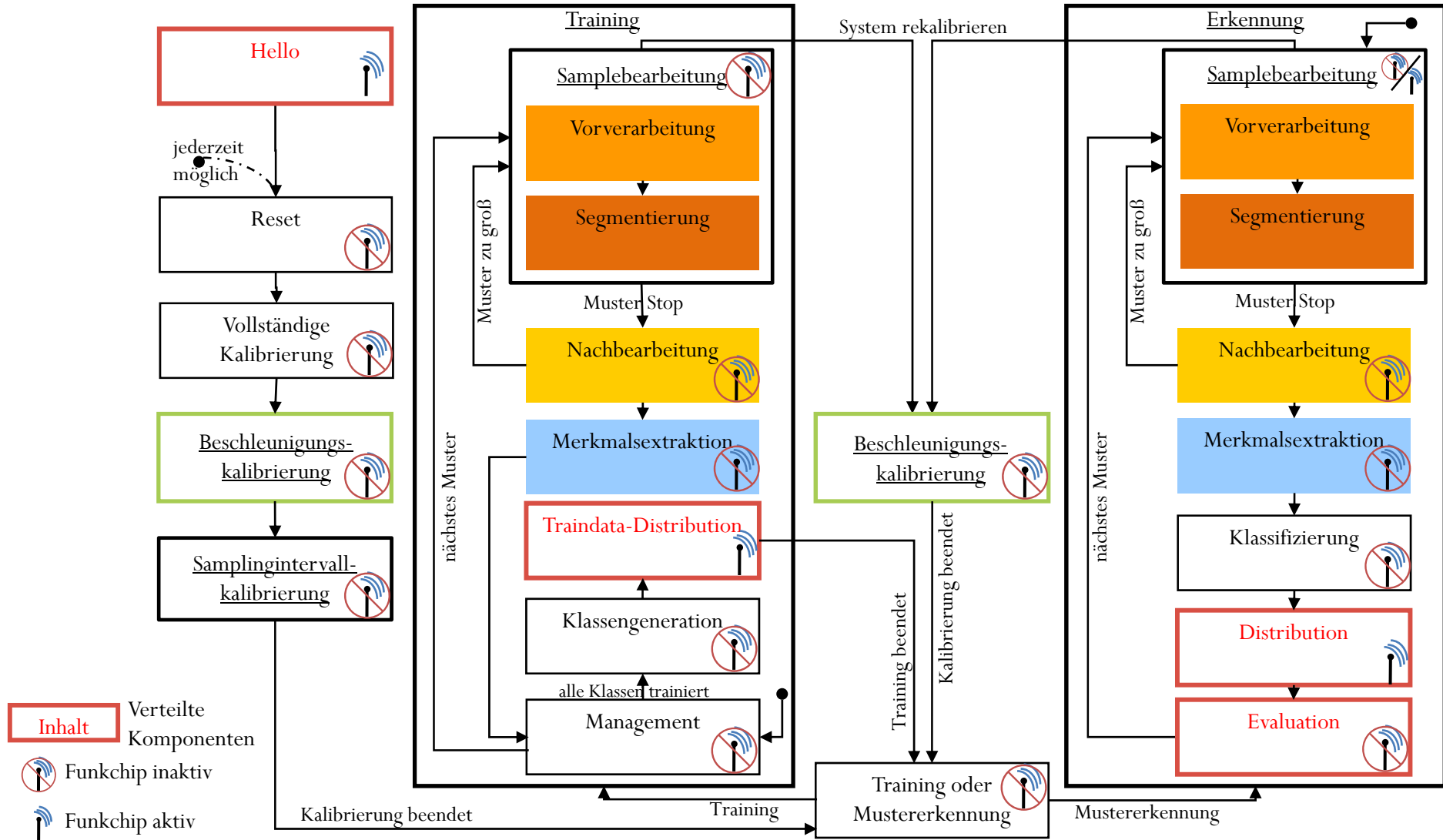


Verteilte Erkennung - Paketversendung





Verteilte Erkennung - Implementierung





Agenda

- ✓ Motivation, Grundlagen
- ✓ Verwandte Arbeiten
- ✓ Bewegungsmuster
- ✓ Lokale Erkennung, Klassifizierer
- ✓ Verteilte Erkennung
- Auswertung
- Zusammenfassung, Ausblick
- Demo



Auswertung - Versuchsaufbauten

Fence Monitoring (Vergleichsarbeit)

- Verteilte und lokale Ereigniserkennung durch Schwellenwertfindung
- Probanden müssen klettern, treten, lehnen etc.
- Überklettern löst Alarm aus, 90 Versuche



[Witf07]

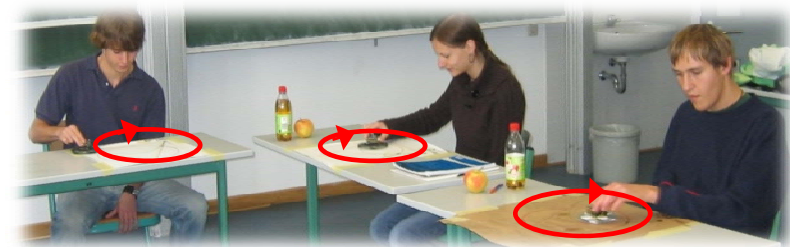
Lange Nacht der Wissenschaft 2007

- Stand der Entwicklung wird vorgestellt und evaluiert
- Probanden sind Besucher der Messe, 280 Versuche
(jeder Proband durfte alle vier Muster mind. einmal erzeugen)



Verteilter Versuch

- Projektfremde Probanden erzeugen Muster
- Lokale Auswertung, 480 Versuche
- Verteilte Auswertung, 160 Versuche



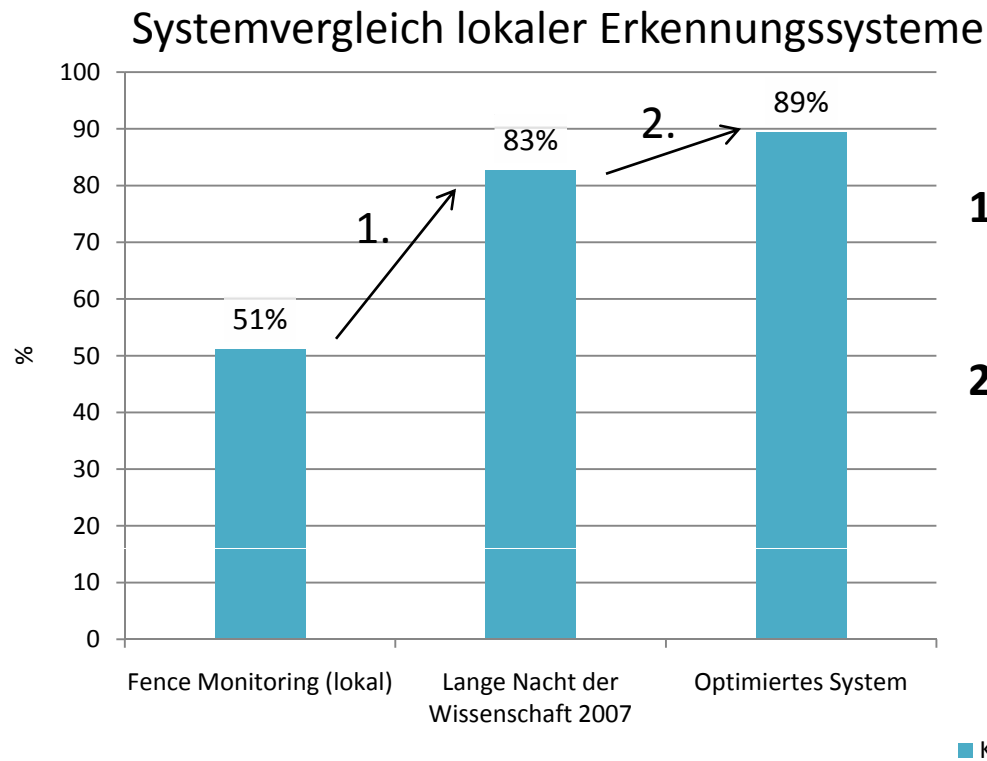
Quellen:

[Witf07] Wittenburg, G. Fence Monitoring - A Use Case for Wireless Sensor Networks - AG Computer Systems & Telematics - Freie Universität Berlin. <http://cst.mi.fu-berlin.de/projects/FenceMonitoring/>



Auswertung / Ergebnisse - Lokal

Die **Korrektklassifikationsrate** (*engl.: accuracy*) ist das Verhältnis aller korrekt klassifizierten Ereignisse zu allen Ereignissen.



1. Halbzeit

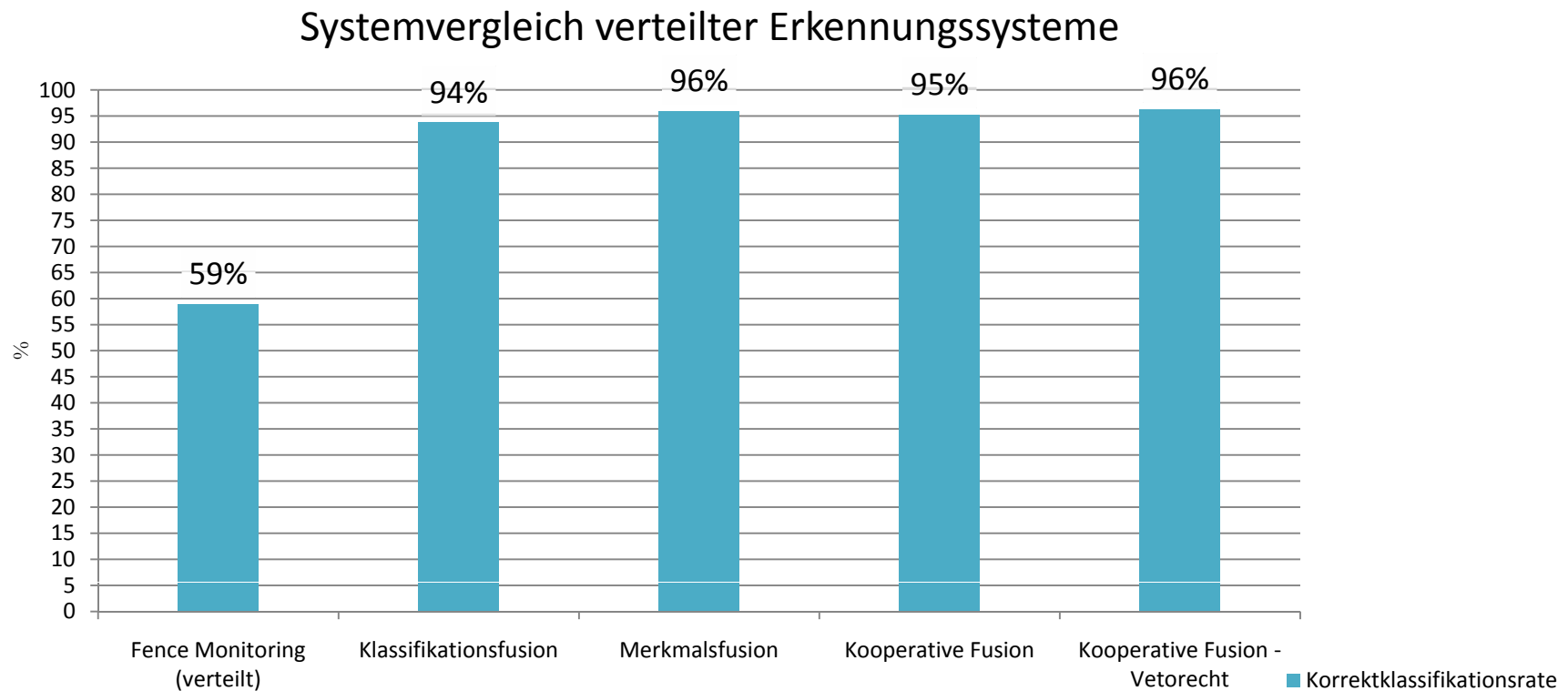
- echtes Mustererkennungssystem

2. Abgabe

- Normierungsbereich vergrößert
- div. Bugs behoben



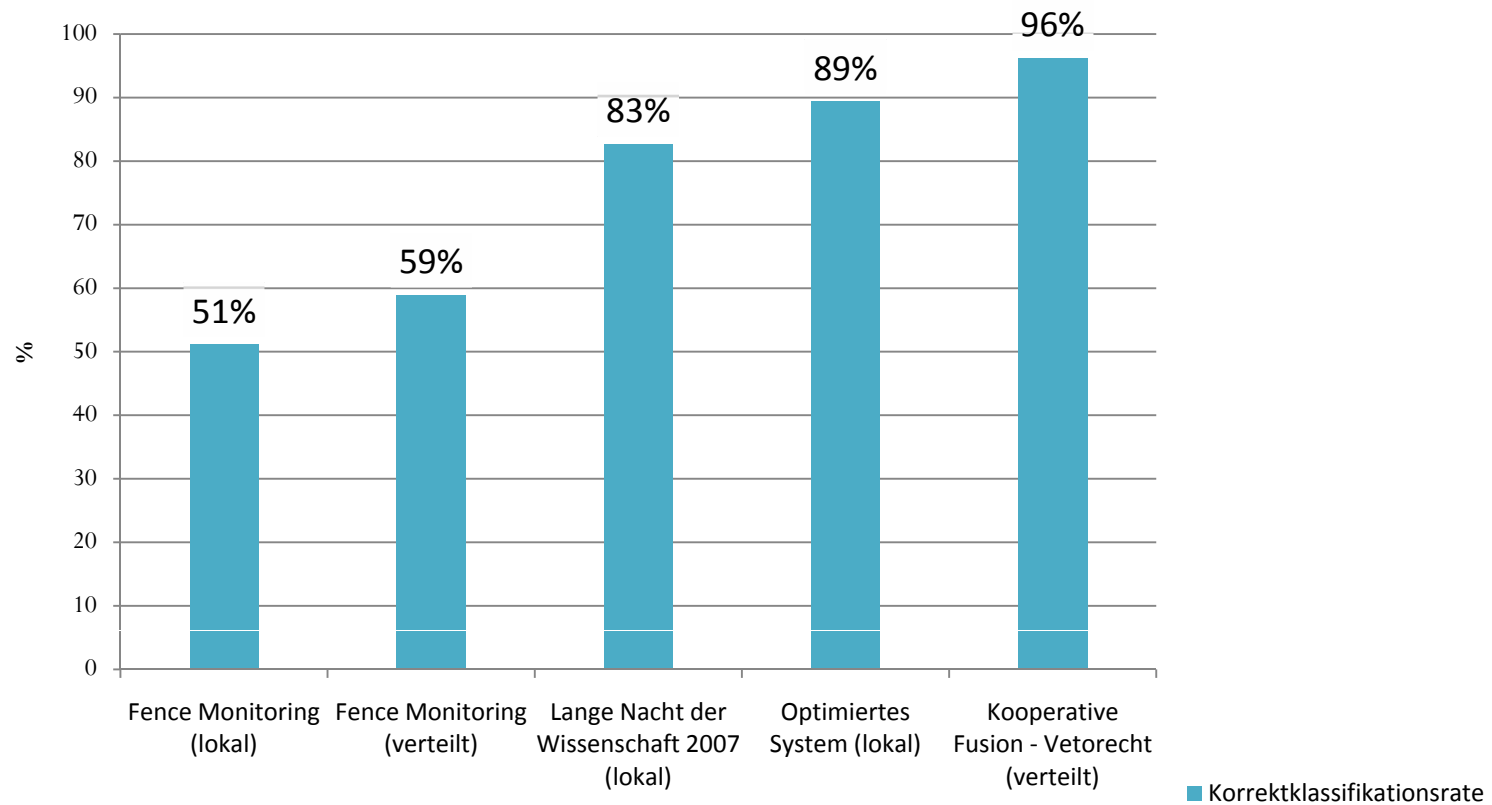
Auswertung / Ergebnisse - Verteilt





Auswertung / Ergebnis-Chronologie

Lokale vs. verteilte Erkennung





Ergebnisse – Theoretischer Kommunikationsaufwand

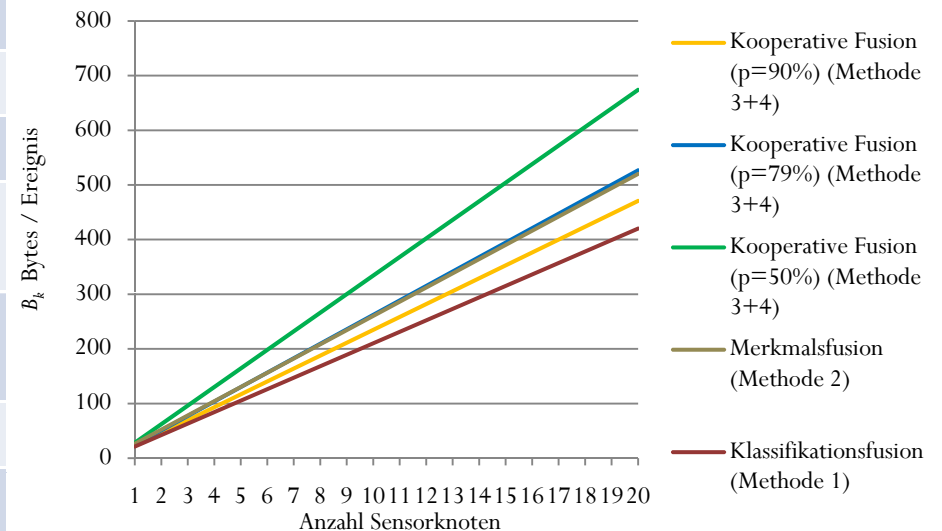
$$B_k = K * n + (1 - p) * (R + (M + m) * (n - 1))$$

Annotations for the equation above:

- Arrow pointing to $K * n$: Jeder Sensor versendet seine Klassifikationsdaten
- Arrow pointing to $(1 - p) * (R + (M + m) * (n - 1))$: Es müssen Merkmalsdaten angefordert werden, wenn $1 - p > 0$
- Arrow pointing to $(M + m) * (n - 1)$: Merkmalsdatenmenge hängt von Anzahl Knoten und Merkmalen ab
- Arrow pointing to $(M + m) * (n - 1)$: Merkmale werden von allen außer vom Anfordernden verschickt

Params	Beschreibung der anfallenden Bytegrößen
B_k	Anfallende Datenmenge bei der Kooperationsmethode
$K = 21 B$	Größe eines Paketes mit Klassifikationsdaten
n	Anzahl Sensorknoten
p	Wahrscheinlichkeit, dass der Sensorknoten mit Klassifikationsdaten die Klassifikationsfusion durchführt
$R = 14 B$	Größe eines Broadcast-Anfragepaketes, das eine Merkmalsfusion zusätzlich initiiert
$M = 20 B$	Paketgröße des Antwortpaketes ohne Merkmale
m	Anzahl Merkmale

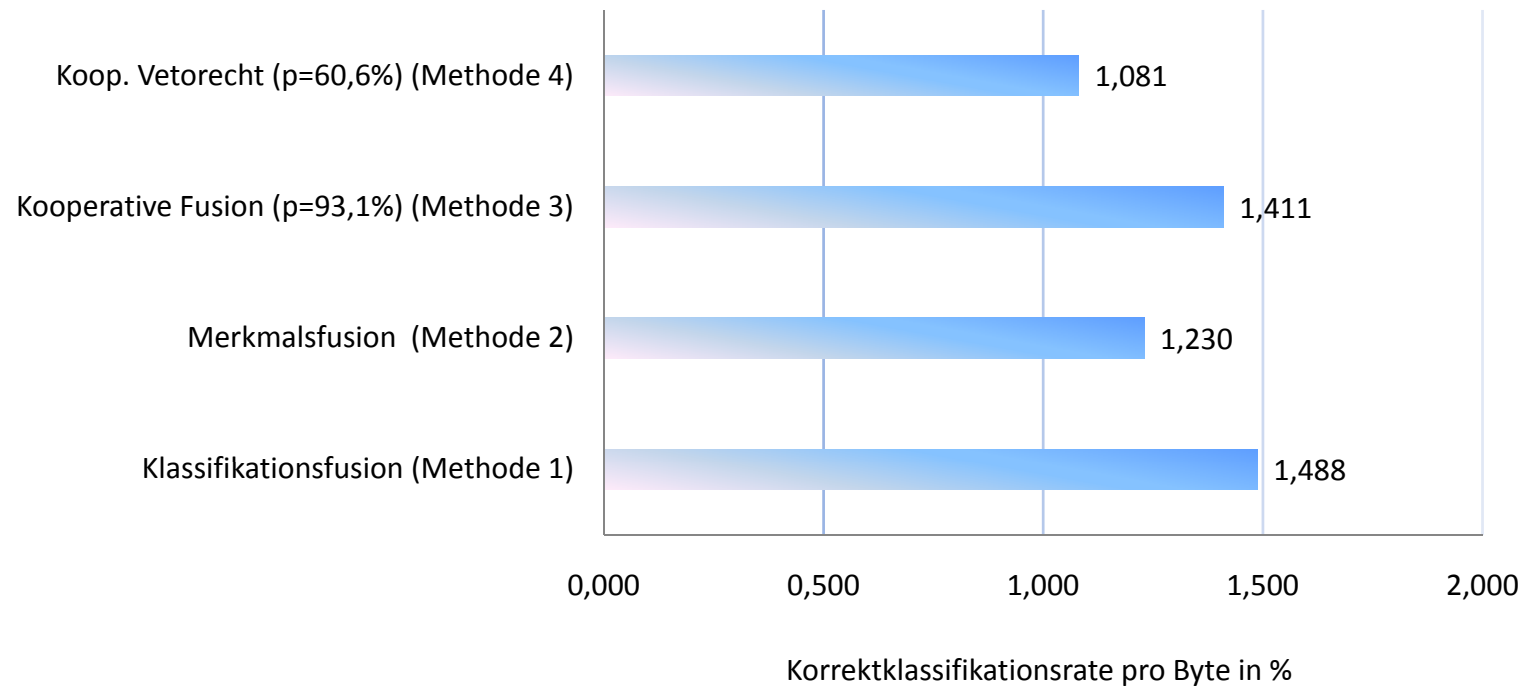
Methodenvergleich des Datenvolumen bei sechs Merkmalen





Auswertung / Ergebnisse – Rentabilität

Korrektklassifikationsrate pro versendetes Byte
in % bei drei Sensorknoten





Evaluation

- Verteilte Ereigniserkennung hat höhere Korrektklassifikationsrate als lokale Ereigniserkennung (+7% bei 160 Versuchen)
- Methode 1 Klassifikationsfusion
 - zeigt höchste Rentabilität aufgrund niedrigstem Kommunikationsaufwand
- Methode 2 Merkmalsfusion
 - Korrektklassifikationsrate (95,9%) wird empfohlen, verringert Kommunikationsaufwand
- Methode 3 Kooperationsfusion
 - Kompromiss zwischen Methode 1 und Methode 2
- Methode 4 Kooperationsfusion mit Vetorecht
 - Beste Korrektklassifikationsrate (96,3%), aber höchster Kommunikationsaufwand



Zusammenfassung

- Verteilte Ereigniserkennung in Sensornetzen ist realisierbar
 - Merkmalsfusion wird in bisherigen Projekten nicht oder nur inkonsequent untersucht
 - Merkmalsfusion erzielt geringfügig bessere Erkennungsraten als Klassifikationsfusion
 - Kooperationsfusionen können das Datenaufkommen optimieren und maximieren
- Korrektklassifikationsrate
- Methodenwahl hängt von p (*Wahrsch. erfolgr. Klassifizierung*) und m (*Merkmalszahl*) ab

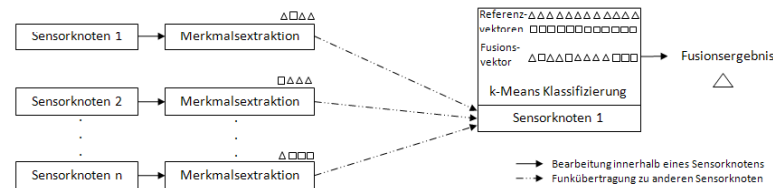
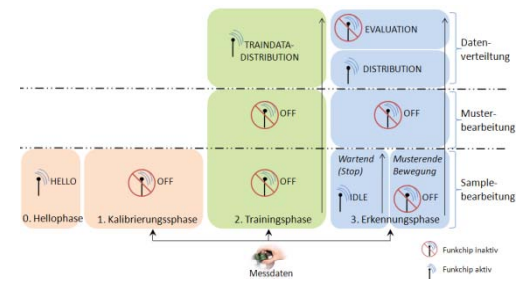
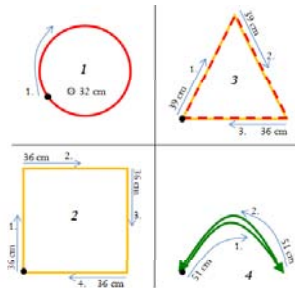


Ausblick

- Implementierung weiterer Merkmalstypen
- Externes Trainingssystem möglich
- Einbeziehung ergänzender Sensoren
- Dynamische Sensorknotenverwaltung
- Wahlsystem für Auswertungsknoten



Danke & Fragen ?

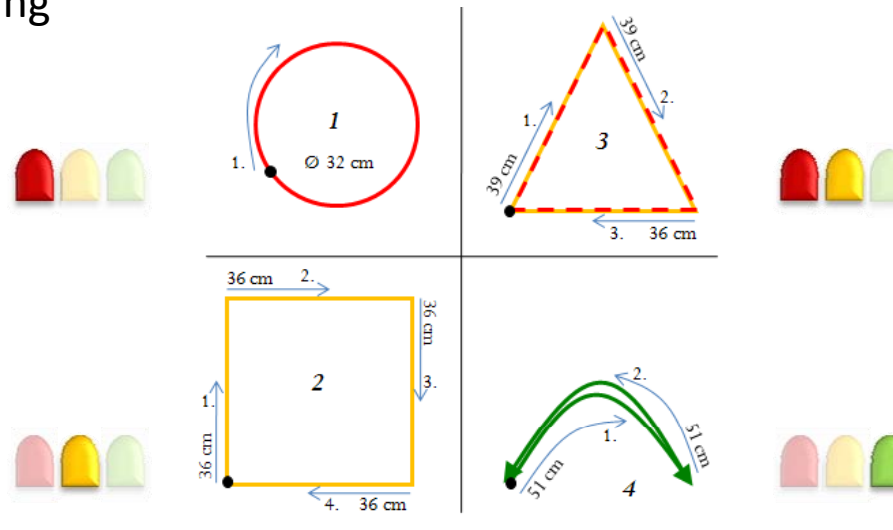


Demo =>



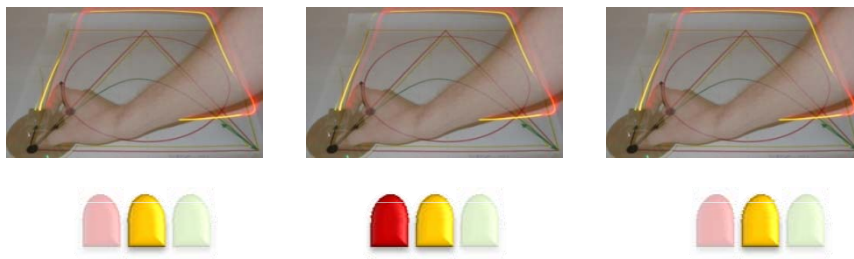
Demo - Merkmalsfusion

Lokale Ereigniserkennung

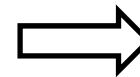


Musterdefinitionen

Verteilte Ereigniserkennung



Merkmalsfusion





Agenda

- ✓ Motivation, Grundlagen
- ✓ Verwandte Arbeiten
- ✓ Bewegungsmuster
- ✓ Lokale Erkennung, Klassifizierer
- ✓ Verteilte Erkennung
- ✓ Auswertung
- ✓ Zusammenfassung, Ausblick
- ✓ Demo
- Reservefolien



Reservefolien



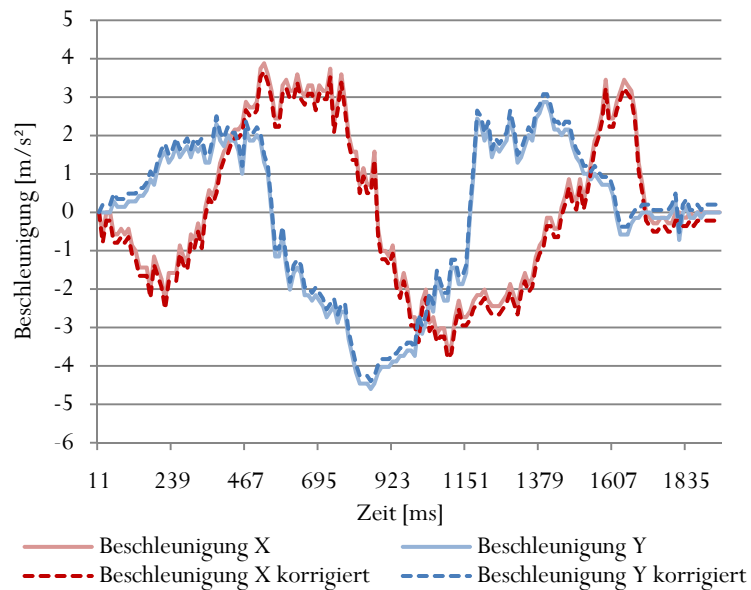
Postulate

- Postulat 1:
 - Annahme: Muster - Informationen werden durch Trainingsdaten erhalten
- Postulat 2:
 - Charakteristische Feature-Vektor für Muster können berechnet werden.
 - Warnung: curse of dimensionality
- Postulat 3:
 - Low intraclass distance - High interclass distance.
- Postulat 4:
 - Komplexe Muster können in kleinere, abhängige Teile zerlegt werden.
 - Beispiel: nebeneinander sitzende Tutoren...
- Postulat 5:
 - Komplexe Muster haben spezifische Strukturen
 - Beispiel :Gesichter im Raum finden
- **Postulat 6: Zwei Muster sind ähnlich, wenn Ihre Features ähnlich sind!**

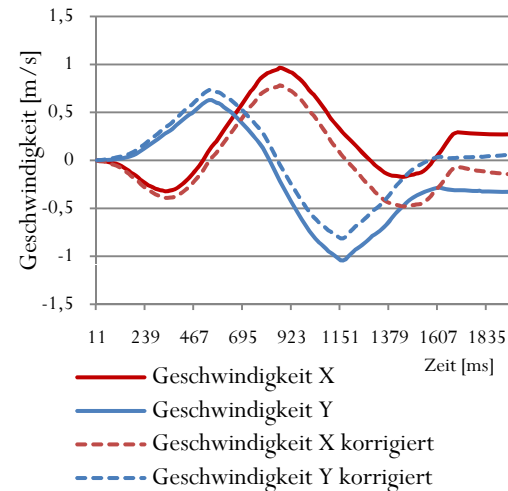


Originaldaten, Integration 1 und 2

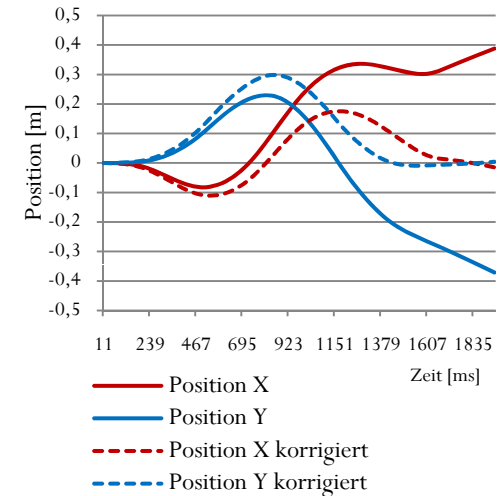
Beschleunigungsdaten vor und nach Offsetkorrektur



Erste Integration:
Geschwindigkeitsdaten
vor und nach
Offsetkorrektur



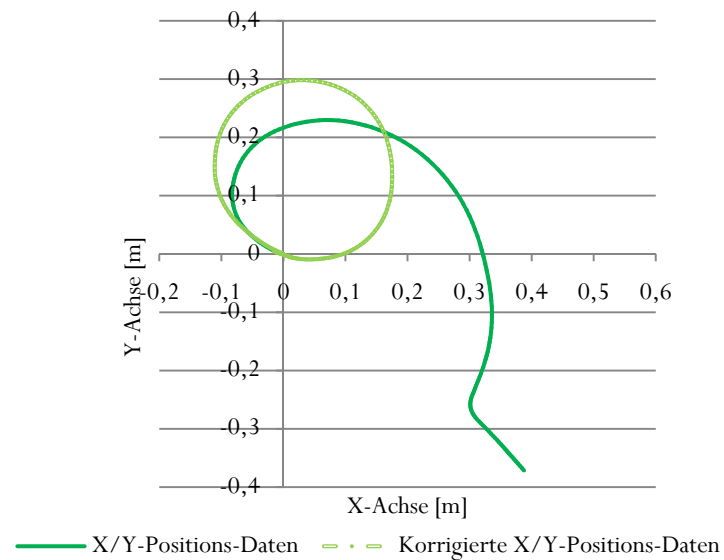
Zweite Integration:
Positionsdaten vor
und nach
Offsetkorrektur





Muster mit / ohne Offsetkorrektur

Ergebnis der gemeinsamen Achsendarstellung von X/Y: Kreisdarstellung vor und nach Offsetkorrektur (Durchmesser ca. 32 cm)



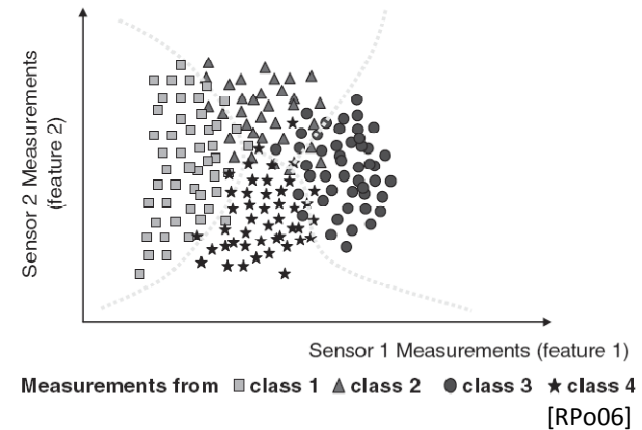
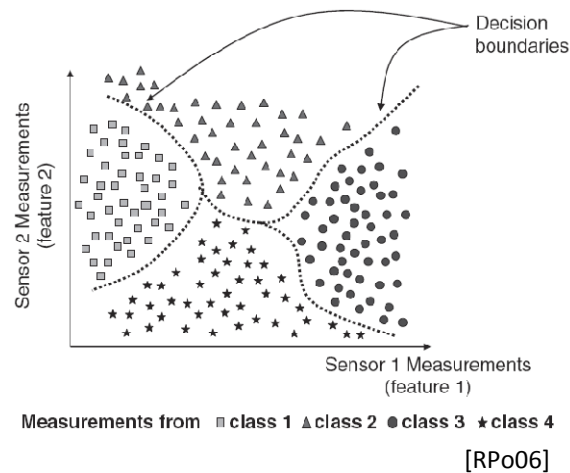


Quellen:

[RPo06]
 Polikar, R. Pattern Recognition.
 In: Wiley Encyclopedia of Biomedical Engineering, 2006.

Klassengrenzen

Beispiel für optimistische Klassengrenzen und 2D-Vektoren



Abgrenzungsproblem ist nicht einfach lösbar, da sich Klassengrenzen überschneiden

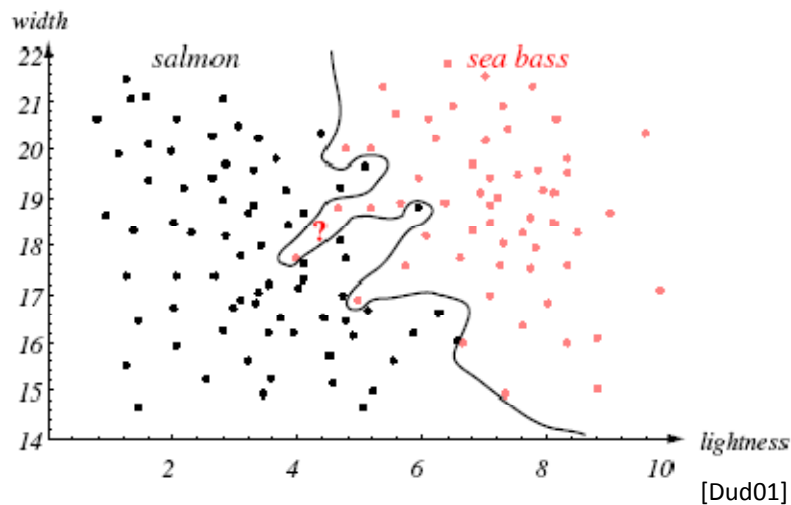


Quellen:

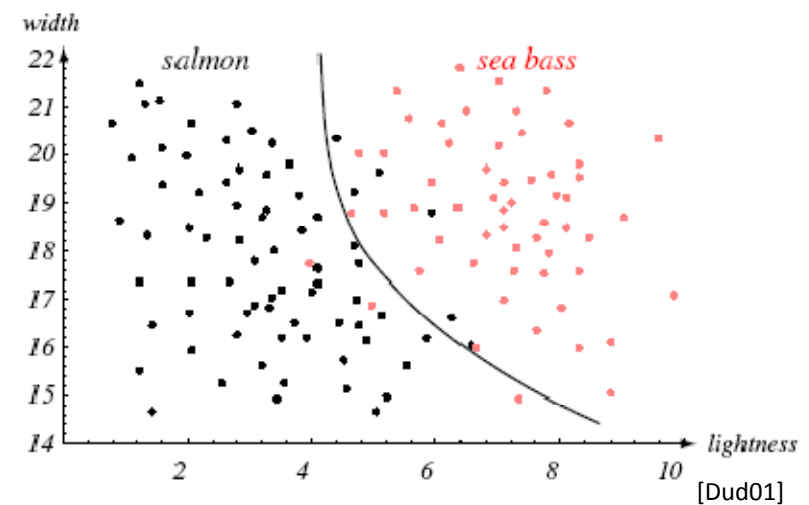
[Dud01]

Duda, R. O., Hart, P. E., Stork, D. G. Pattern Classification.
In: A Wiley-Interscience, 2001.

Trainingsumfang



Übertrainiertes System mit einer schlechten Performanz



Hohe Genauigkeit der Klassifizierung und hohe Performanz mittels generalisierter Klassengrenze

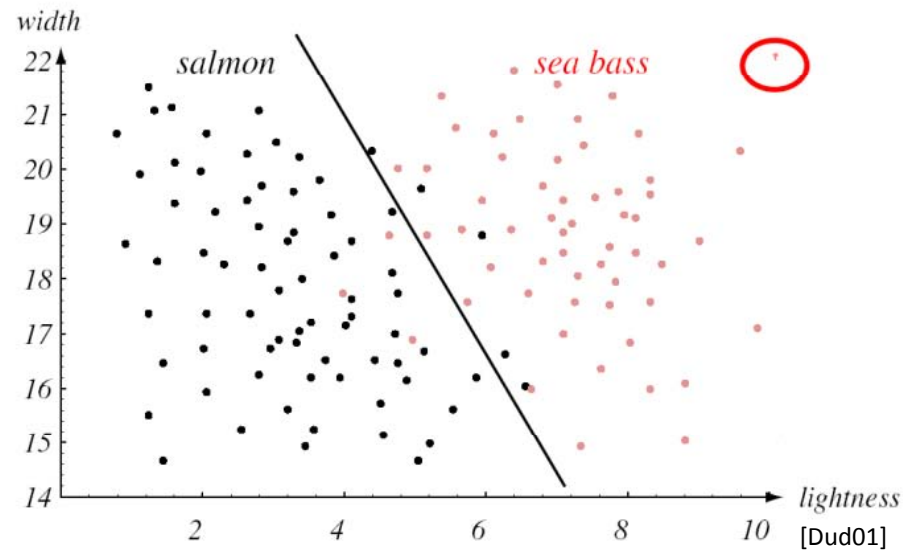


Quellen:

[Dud01]

Duda, R. O., Hart, P. E., Stork, D. G. Pattern Classification.
In: A Wiley-Interscience, 2001.

Außenseiter



Ein Außenseiter (rot umrandet) muss korrekt zugeordnet oder zurückgewiesen werden



Klassifizierer

kNN-Algorithmus

1. Initialisierung: Vorgabe der Cluster mit E Elementen
2. Abstandsberechnung: Abstand unbekanntes Muster zu allen Mustern berechnen
3. Sortiere: Alle Abstände werdend er Größe nach sortiert
4. kNN: finde die k kürzesten Abstände zu einer Klasse

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

k-means-Algorithmus

1. Initialisierung: (Zufällige) Auswahl von k (gegebenen) Clusterzentren
2. Zuordnung: Jedes Muster wird nächsten liegenden Clusterzentrum zugeordnet
3. Neuberechnung: Für jedes Cluster Clusterzentren neu berechnen
4. Wiederholung: Zuordnung der Objekte geändert? Zuordnung, Abbruch sonst

- Cluster kann leer bleiben => man startet den *k-means-Algorithmus* neu.
- „quick'n'dirty“, praktisch fast immer gute Resultate



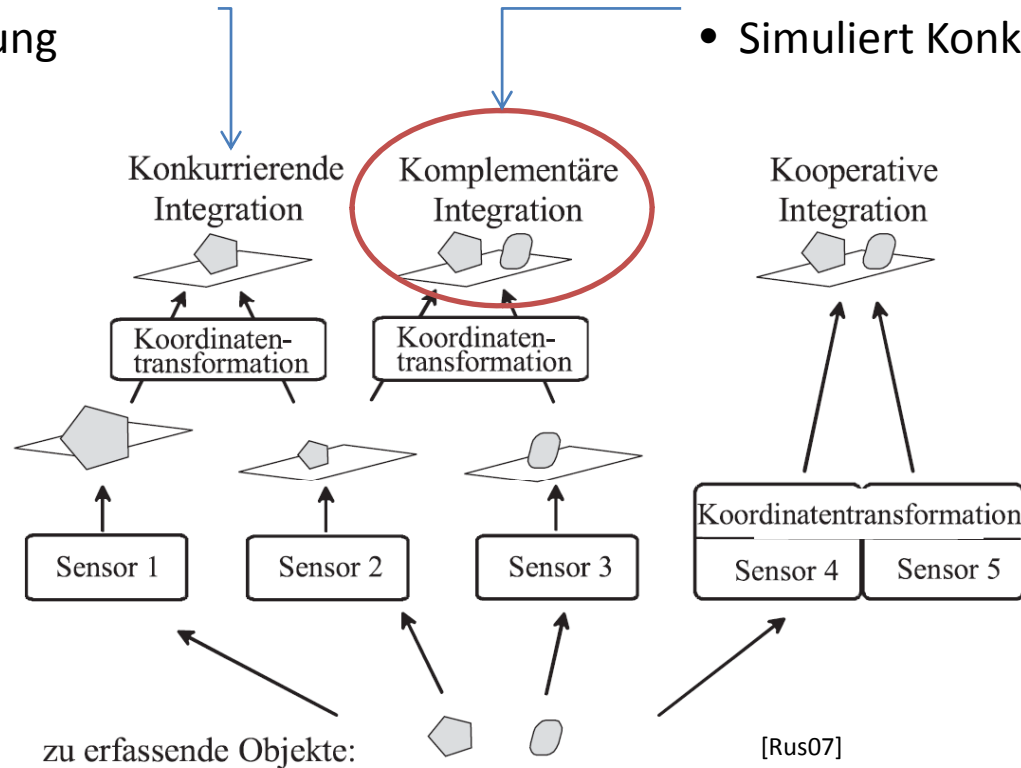
Grundlegende Fusionskonzepte

Gleiche Nutzinformationen

- Mittelwertbildung

Verschiedene Nutzinformationen

- Simuliert Konkurrierende Integration

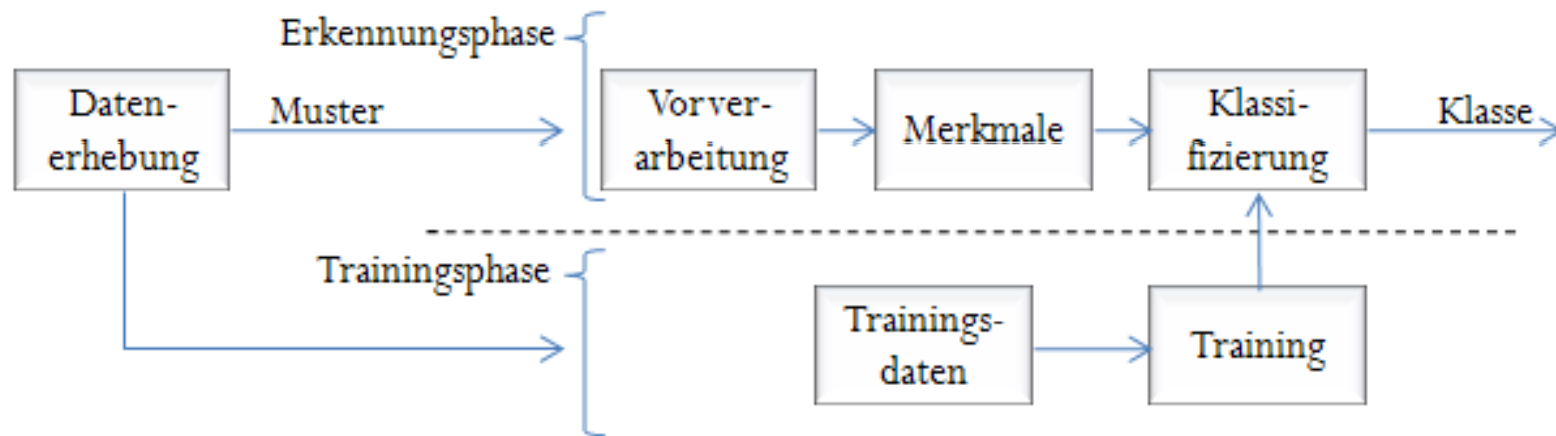


Quellen:
[Rus07]

Ruser, H., Puente León, F. Informationsfusion - Eine Übersicht.
In: Technisches Messen 74, Oldenburg Verlag, 2007.



Standardmodell der Mustererkennung



[Nie03]

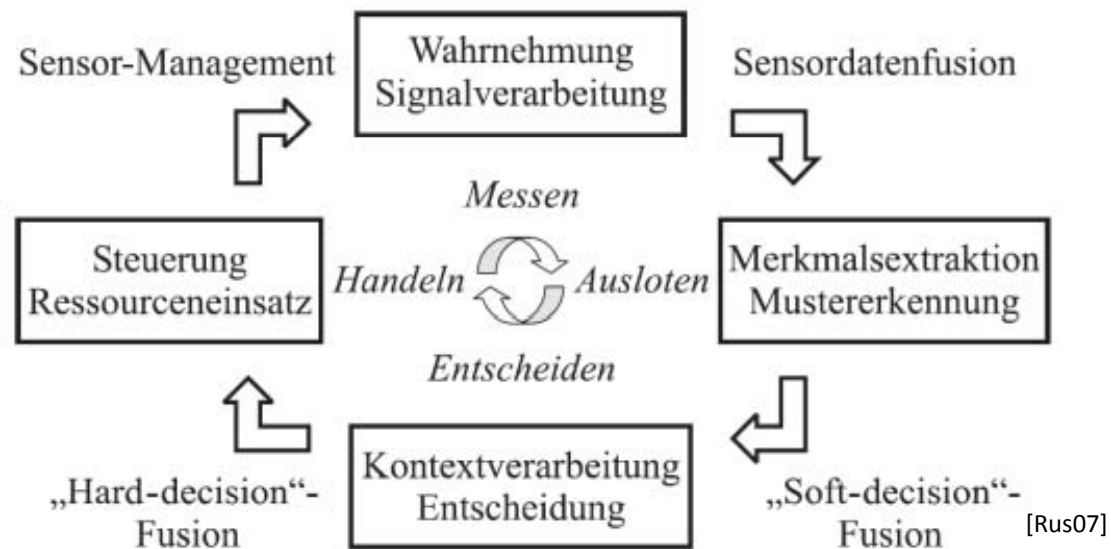
[Nie03]
 Niemann, H. Klassifikation von Mustern.
 Springer-Verlag, Berlin 1983.



Omnibusmodell

Omnibusmodell:

- Basiert auf Klassischem Mustererkennungsmodell
- Verwendet Fusionsebenen um verteilte Ansätze zu definieren

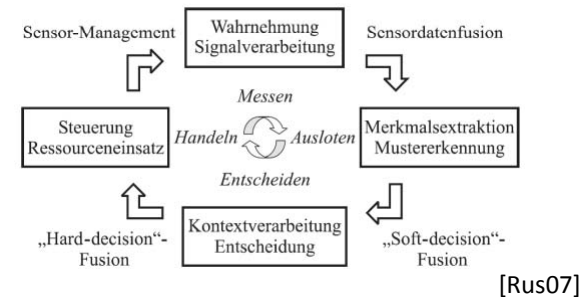
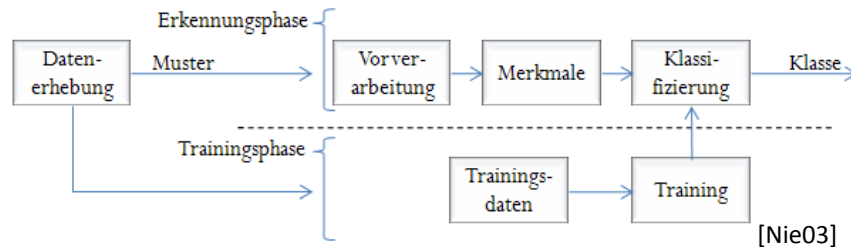


Quellen:
[Rus07]

Ruser, H., Puente León, F. Informationsfusion - Eine Übersicht.
In: Technisches Messen 74, Oldenburg Verlag, 2007.



Abbildung von Omnibusmodell auf Erkennungssystem

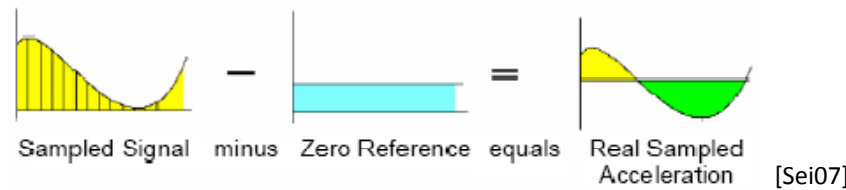
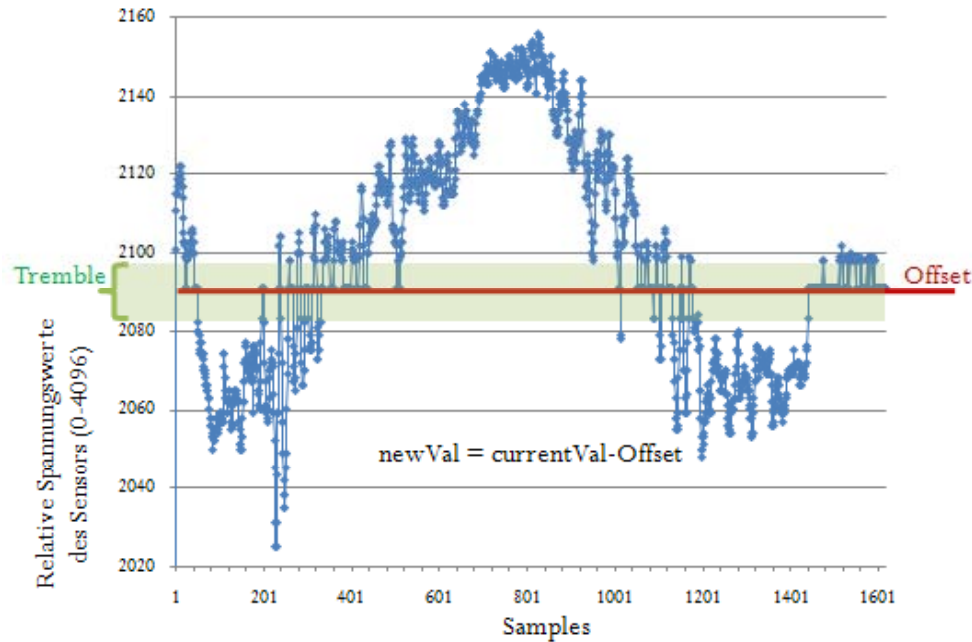


Omnibusmodell	Mustererkennungssystem
Elemente Wahrnehmung Signalverarbeitung	Vorverarbeitung Segmentierung
Merkmalsextraktion Mustererkennung	Nachbearbeitung Normierung Merkmalsextraktion
Kontextverarbeitung Entscheidung	Klassifizierung
Steuerung Ressourceneinsatz	beliebiges übergeordnetes System zur Verwaltung von Ressourcen sowie die Alarm- und Handlungsinstanz

Quellen:
 [Rus07] Ruser, H., Puente León, F. Informationsfusion - Eine Übersicht. In: Technisches Messen 74, Oldenburg Verlag, 2007.
 [Nie03] Niemann, H. Klassifikation von Mustern. Springer-Verlag, Berlin 1983.



Offsetbildung



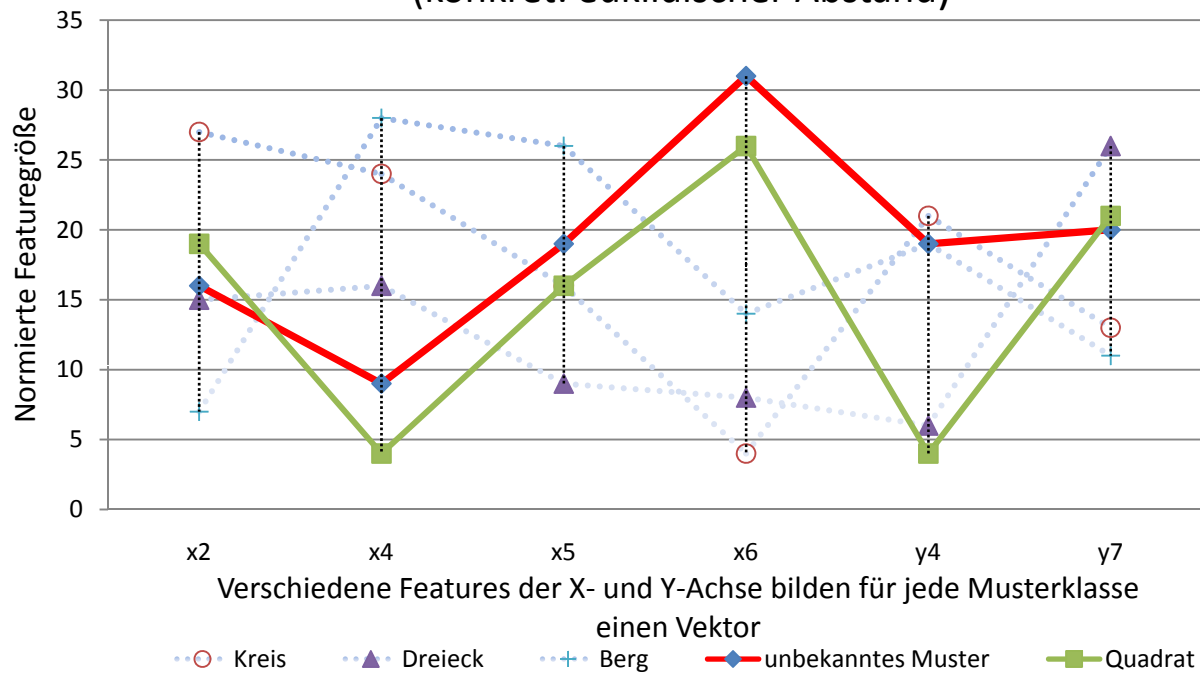
Vereinfachte Darstellung des Beschleunigungssignals vor und nach der Offsetbildung

Seifert, K., Camacho, O. Implementing Positioning Algorithms Using Accelerometers. Application Note. In: Freescale Semiconductor, Inc., 2007.



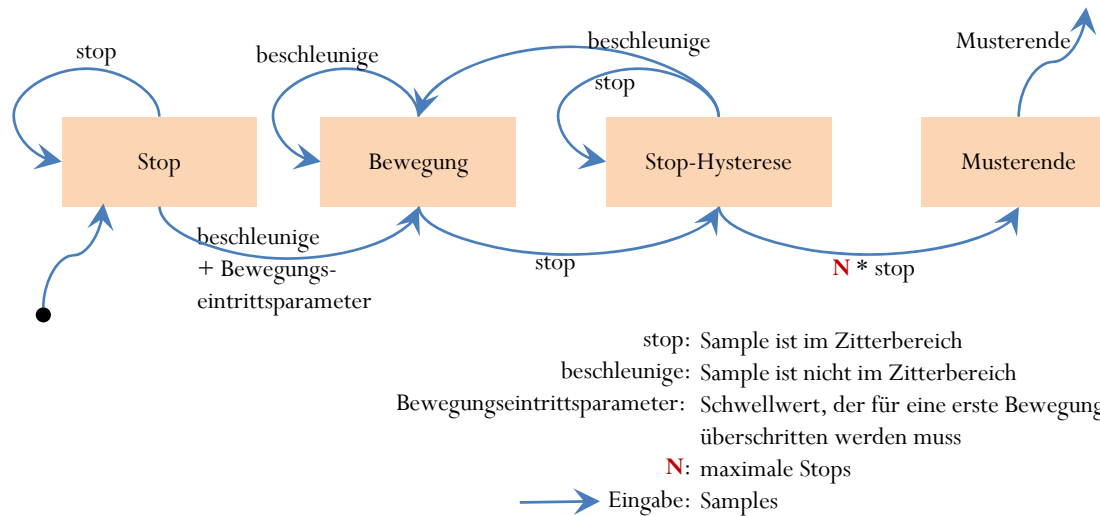
Idee der Klassifizierung

Klassifikation unbekannter Muster mit Featurevektoren
(konkret: euklidischer Abstand)



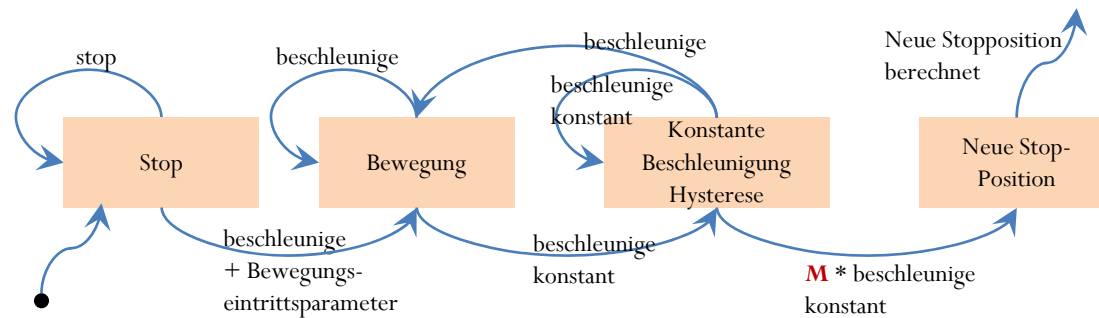


Stop-Hysterese





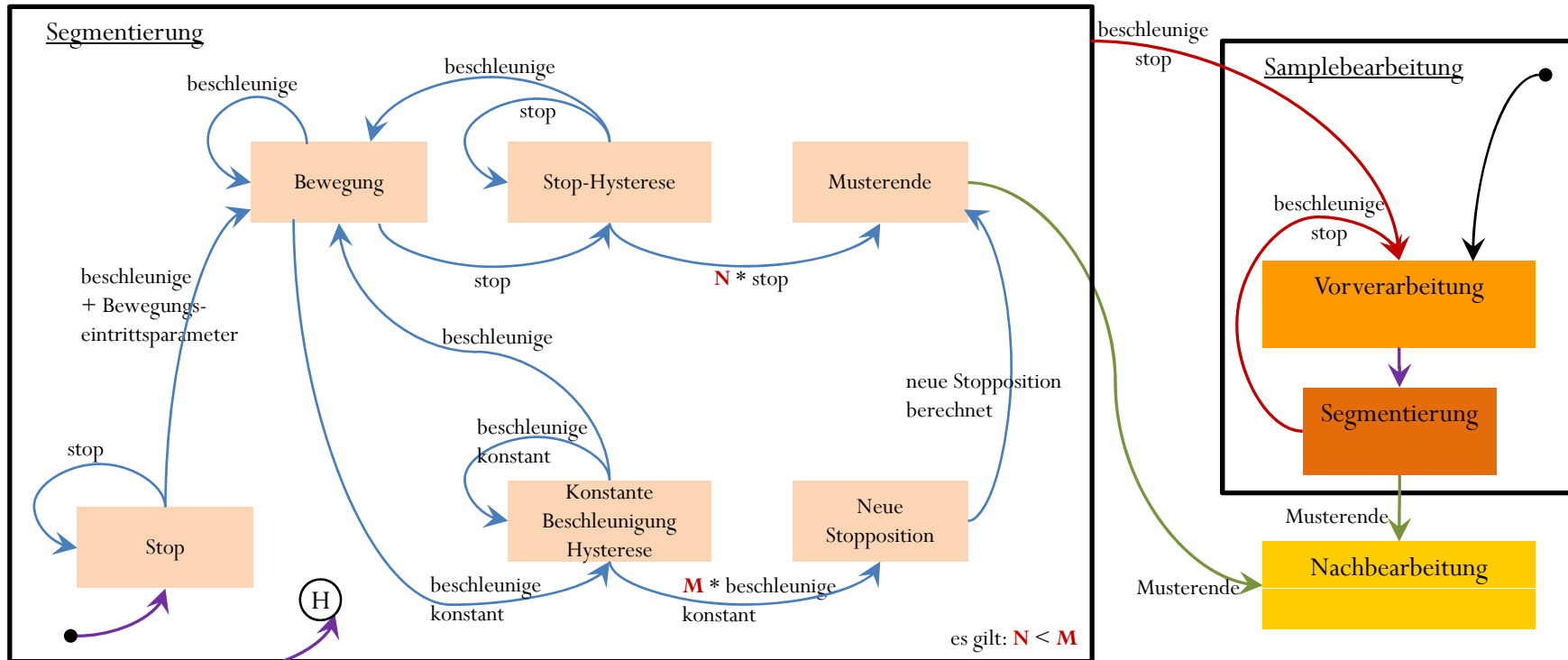
Konstante Beschleunigungshysterese



Beschleunige konstant: Beschleunige genauso wie im letzten Sample
 beschleunige: Sample ist nicht im Zitterbereich
 Bewegungseintrittsparameter: Schwellwert, der für eine erste Bewegung überschritten werden muss
M: maximale konstante Beschleunigungen
 → Eingabe: Samples



Automat - Segmentierung

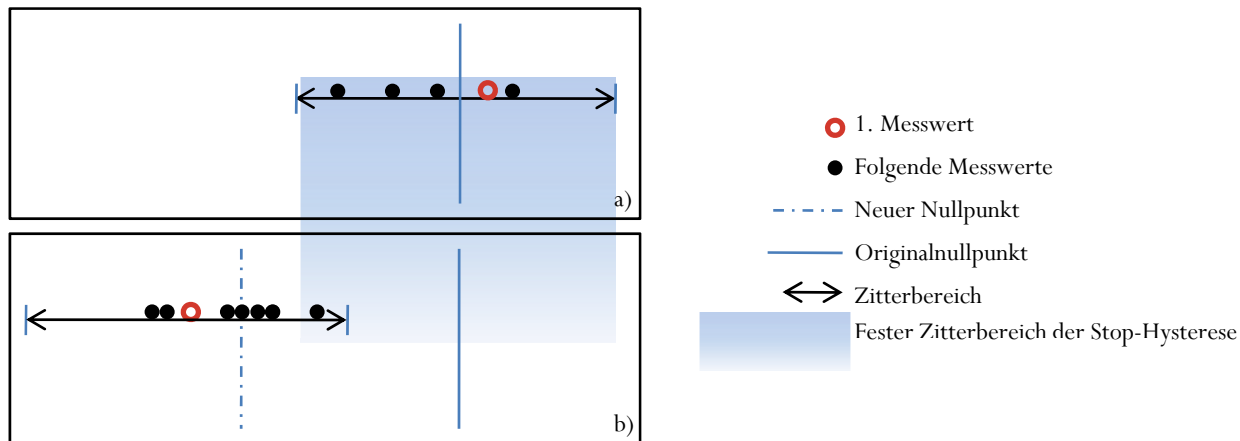


- stop: Sample ist im Zitterbereich
- beschleunige konstant: Beschleunige genauso wie im letzten Sample
- beschleunige: Sample ist nicht im Zitterbereich
- Bewegungseintrittsparemeter: Schwellwert, der für eine erste Bewegung überschritten werden muss
- N**: maximale Stops
- M**: maximale konstante Beschleunigungen
- Eingabe: Samples
- (H) Historienfunktion: merke letzten Zustand

- Alle grünen Pfeile entsprechen exakt dem selben Vorgang: Musterende
- Alle roten Pfeile entsprechen exakt dem selben Vorgang: Beschleunigungs- oder Stopsample trifft ein
- Alle violetten Pfeile entsprechen exakt dem selben Vorgang: verlasse Vorverarbeitung und betrete Segmentierung



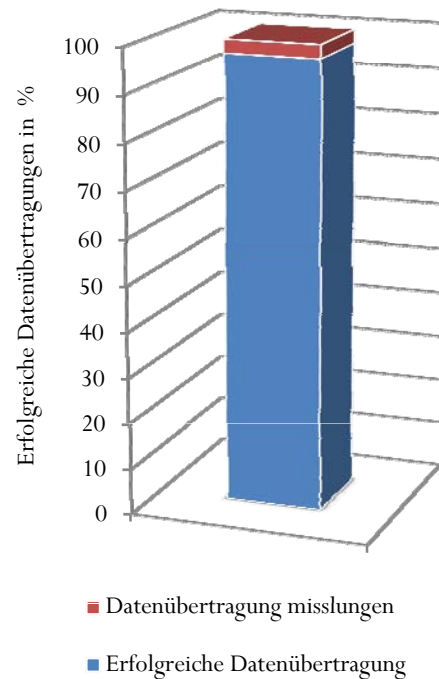
Ruhepunktfindung



- a) beschreibt schematisch die Berechnung des Nullpunktes mit der „Stop-Hysteresis“
- b) beschreibt schematisch die Berechnung des Nullpunktes mit der „Konstante Beschleunigung-Hysteresis“



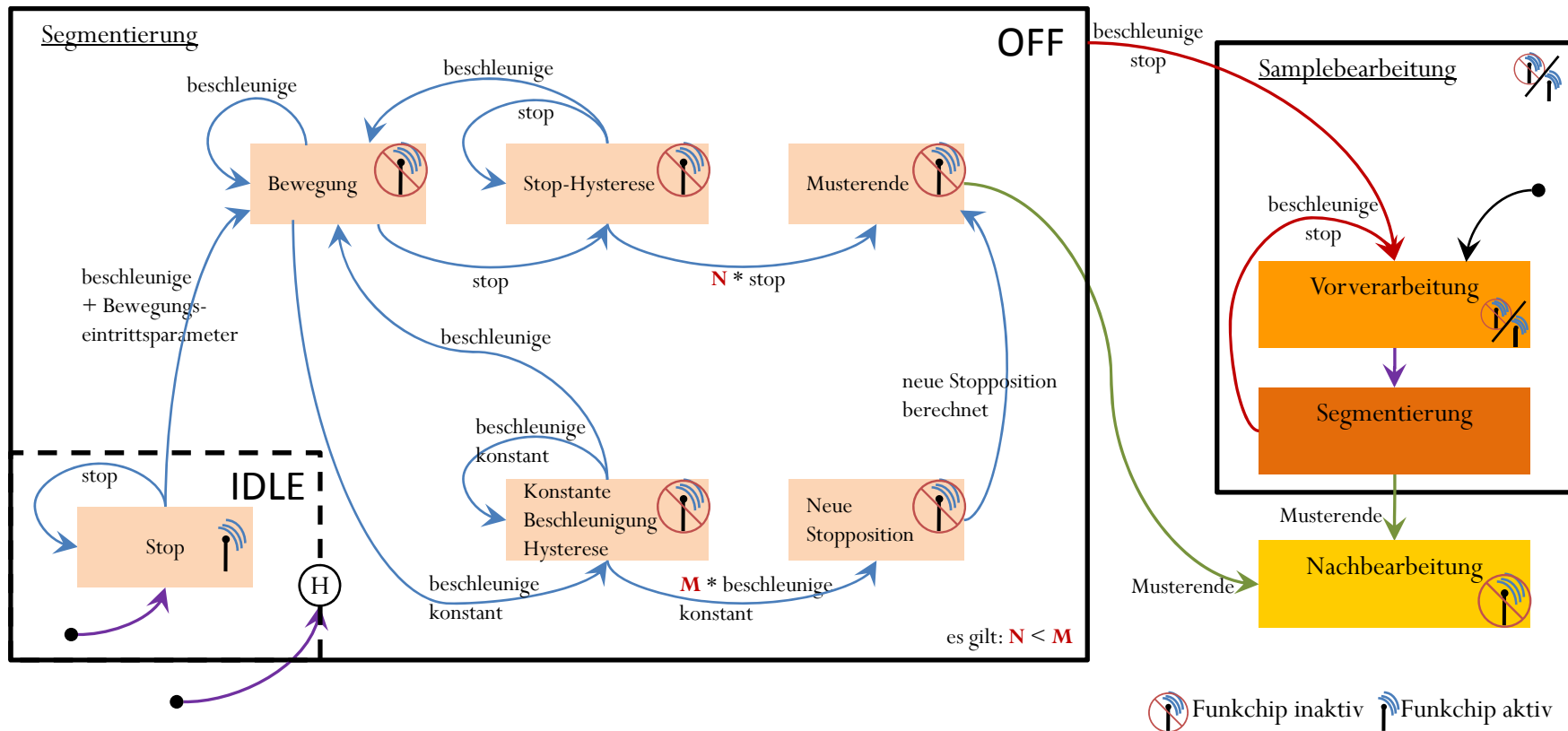
Datenübertragungsqualität (verteilt)



97 % aller Versuche der verteilten Erkennung konnten nach erfolgreicher Datenübertragung ausgewertet werden.



Automat – Segmentierung (inkl. Radio)



Die Samplebearbeitung der Erkennungsphase ist um die verteilten Erkennungszustände „Idle“ und „Off“ erweitert. Zusätzlich werden die Funkchipaktivitäten aufgezeigt.



Kennwerte

- richtig positiv: wird verwendet, wenn ein gesuchtes Muster korrekt klassifiziert wird
- richtig negativ: wird verwendet, wenn ein gesuchtes Muster nicht vorliegt und dies korrekt festgestellt wird
- falsch positiv: wird verwendet, wenn ein gesuchtes Muster gefunden wird, obwohl es nicht vorliegt
- falsch negativ: wird verwendet, wenn das gesuchte Muster nicht korrekt klassifiziert wird, obwohl es vorliegt



Kennwerte

Die **Sensitivität** (*engl.: sensitivity*) wird als das Verhältnis von richtig erkannten Mustern eines Typs zu allen bisher aufgetretenen Mustern dieses Typs definiert. Dies entspricht der Wahrscheinlichkeit, dass ein gesuchtes Muster richtig erkannt wird wenn es auftritt.

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

Die **Spezifität** (*engl.: specificity*) wird als das Verhältnis von richtig erkannten anderen Mustern zu allen anderen erzeugten Mustern 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 erkannten Mustern eines Typs zu allen bisher richtig oder falsch erkannten Mustern dieses Typs. Dies entspricht der Wahrscheinlichkeit, dass ein gesuchtes Muster, wenn es erkannt wird, richtig erkannt werden wird.

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

Die **Segreganz** (*engl.: negative predictive value*) ist das Verhältnis von korrekt nicht erkannten eines Typs zu allen bisher korrekt oder nicht korrekt erkannten Nichtauftreten des Musters. Dies entspricht der Wahrscheinlichkeit, dass ein gesuchtes Muster, wenn es als „nicht erkannt“ bewertet wird, diese Aussage richtig ist.

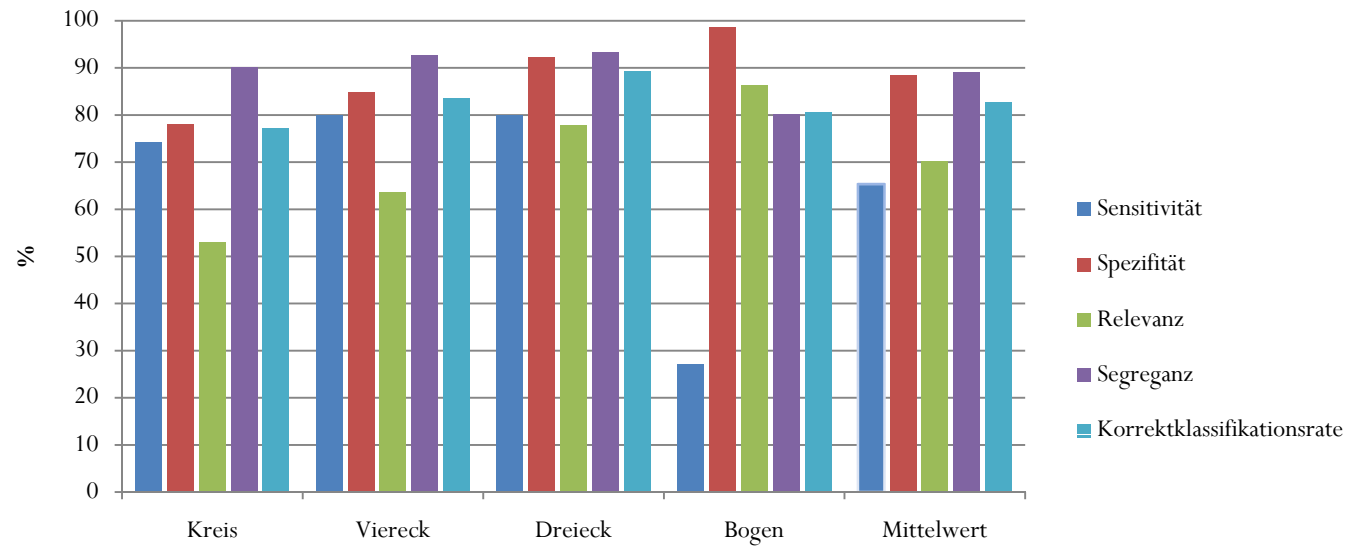
$$\text{Segreganz} = \frac{\# \text{richtig negativ}}{\# \text{richtig negativ} + \# \text{falsch negativ}}$$

Die **Korrektklassifikationsrate** (*engl.: accuracy*) ist der Anteil aller richtig klassifizierten Objekte. Mit der Korrektklassifikationsrate werden die Aussagen der Kennwerte Sensitivität, Spezifität, Relevanz und Segreganz zusammengefasst.

$$\text{Korrektkl. -Rate} = \frac{(\# \text{richtig positiv} + \# \text{richtig negativ})}{\# \text{richtig positiv} + \# \text{richtig negativ} + \# \text{falsch positiv} + \# \text{falsch negativ}}$$

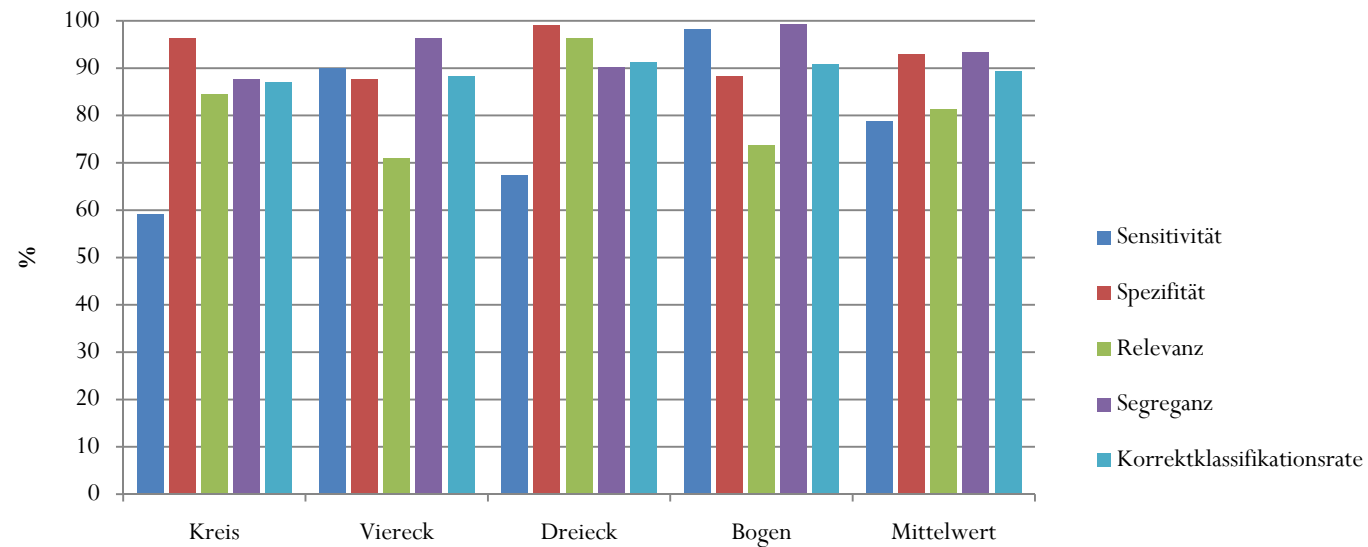


Kennwerte - Lange Nacht 2007 (lokal)



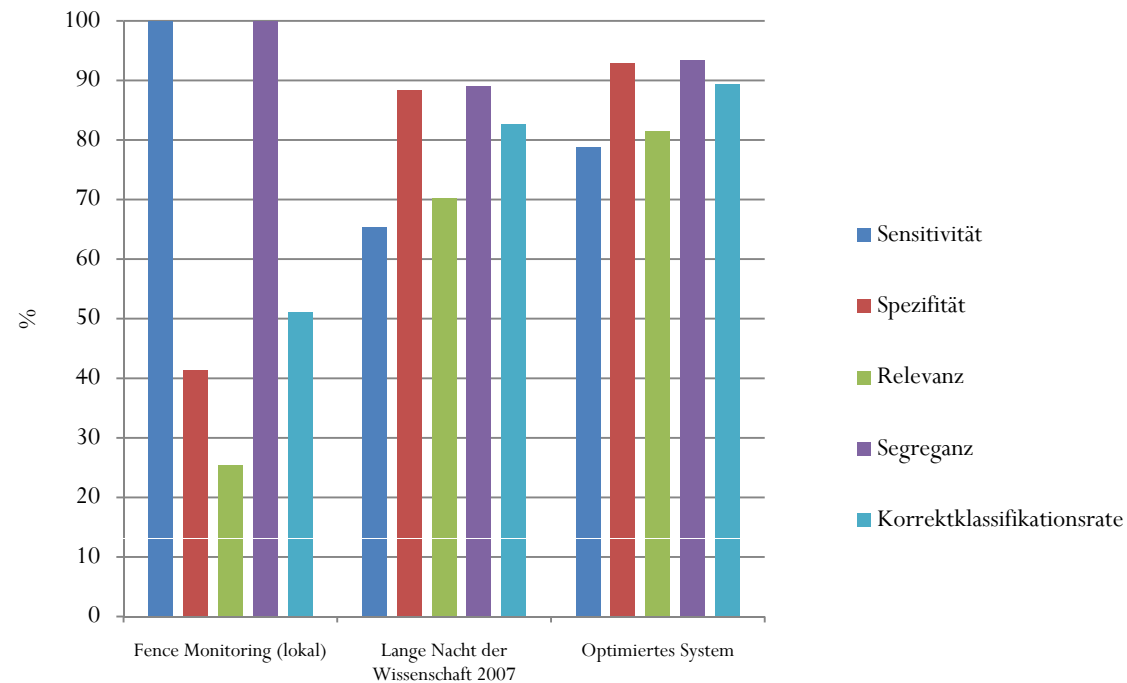


Kennwerte – optimiertes System (lokal)



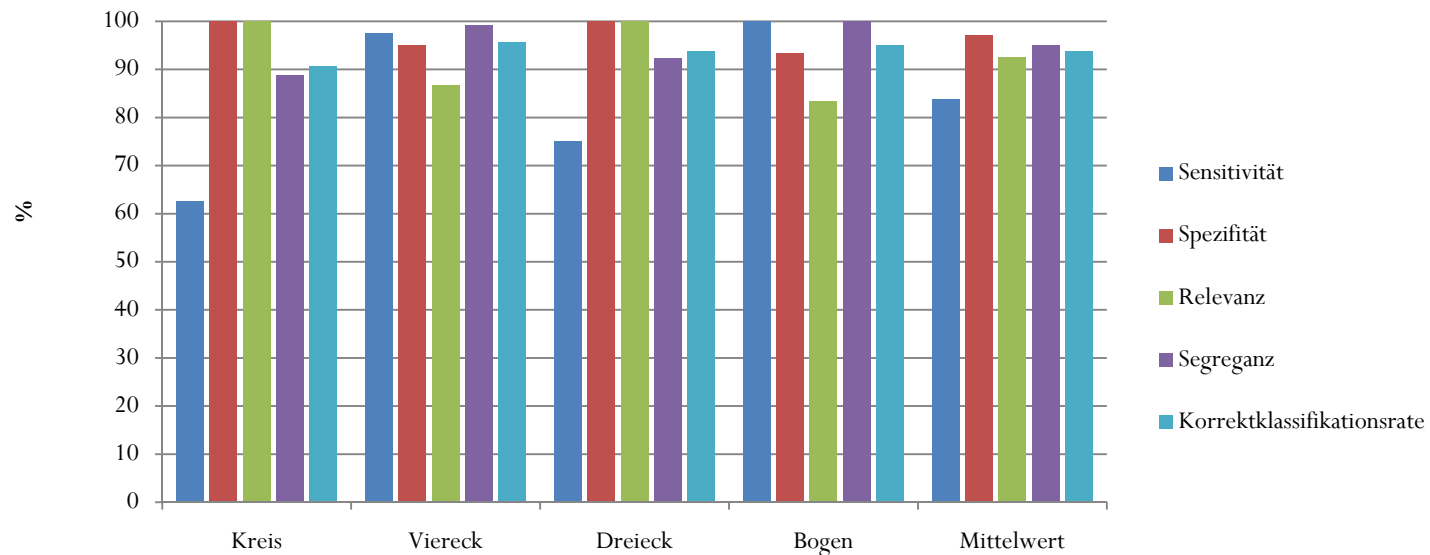


Kennwerte – Vergleich lokaler Systeme



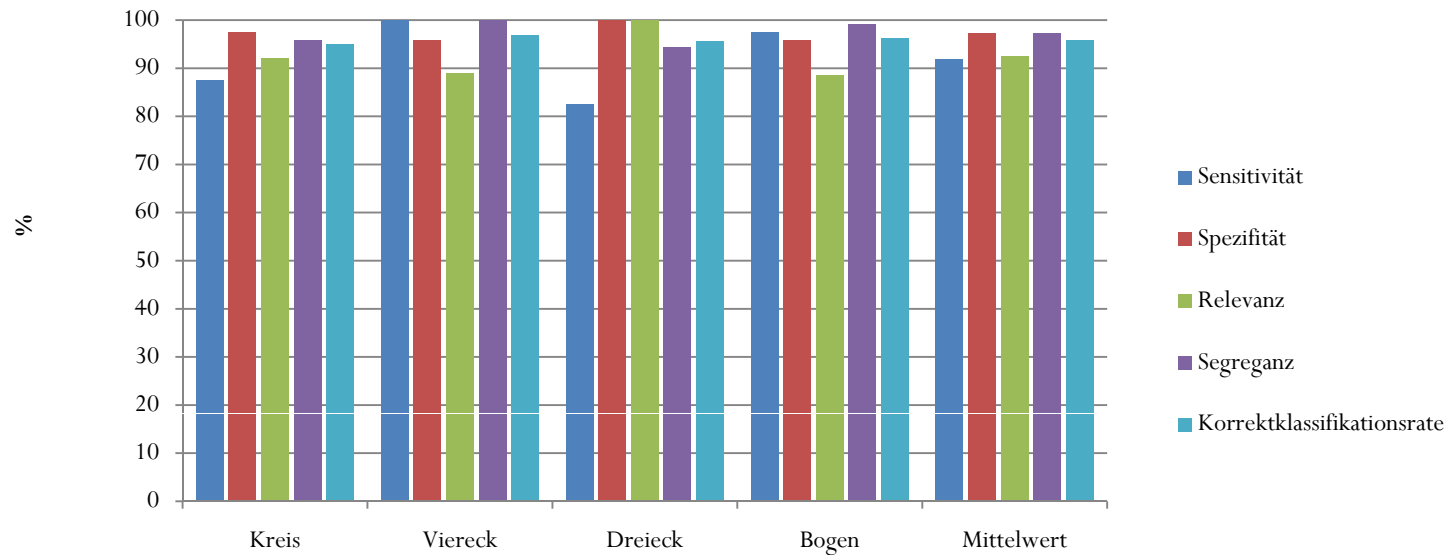


Auswertung Klassifikationsfusion (verteilt)



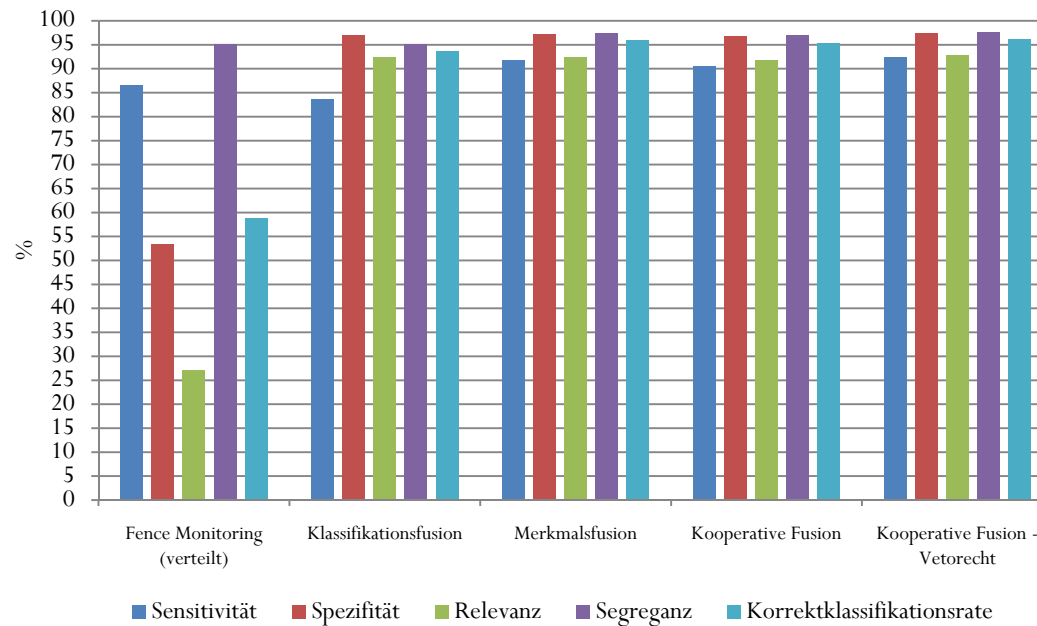


Auswertung Merkmalsfusion (verteilt)





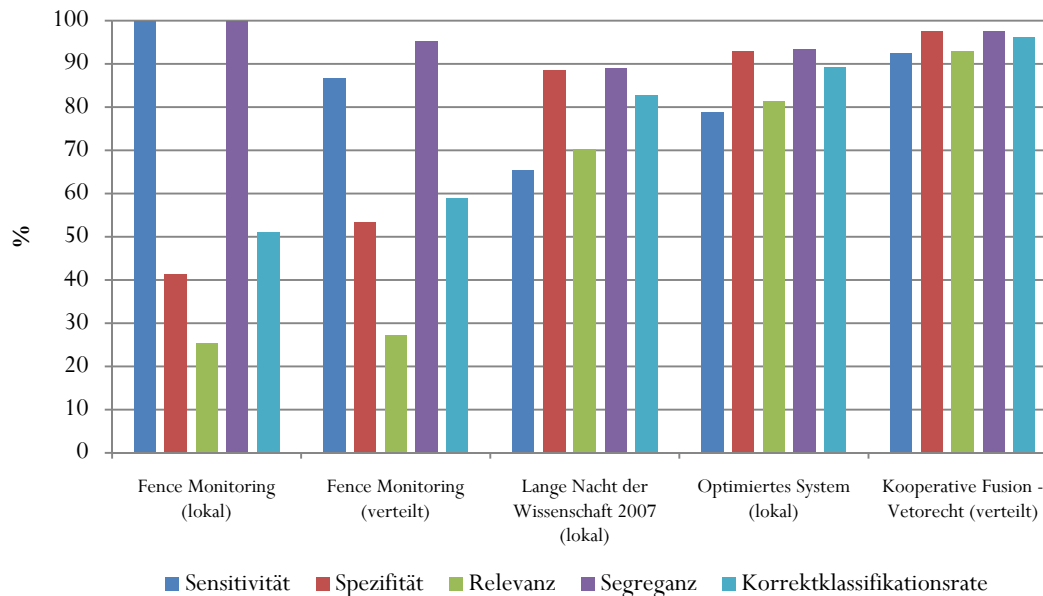
Auswertung Klassifikationsfusion (verteilt)



Kennwertevergleich zur Bewertung der vorgestellten verteilten Methoden dieser Arbeit mit Fence Monitoring



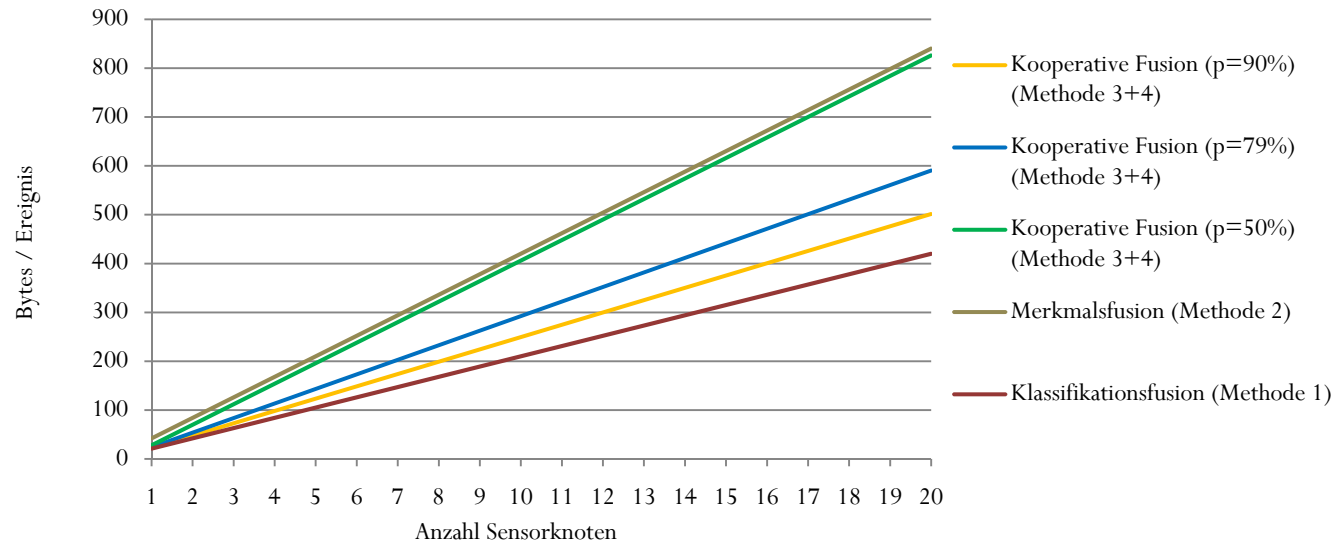
Kennwerte – Vergleich aller Systeme



Vergleich des gesamten Fence Monitoring Systems mit den in dieser Arbeit vorgestellten lokalen und erkenntungstechnisch besten verteilten System.



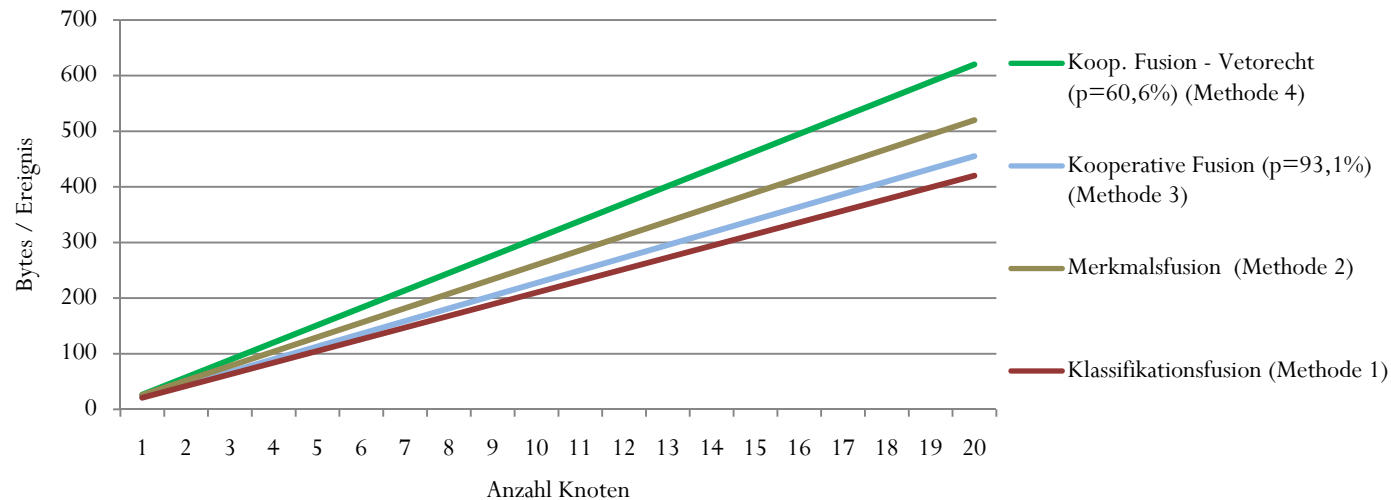
Datenvolumen bei 22 Merkmalen (theoretisches p)



Spezifischer Methodenvergleich des Datenvolumens bei 22 Merkmalen mit theoretischer primärer Auswertungsquote der Methode 3 bzw. 4



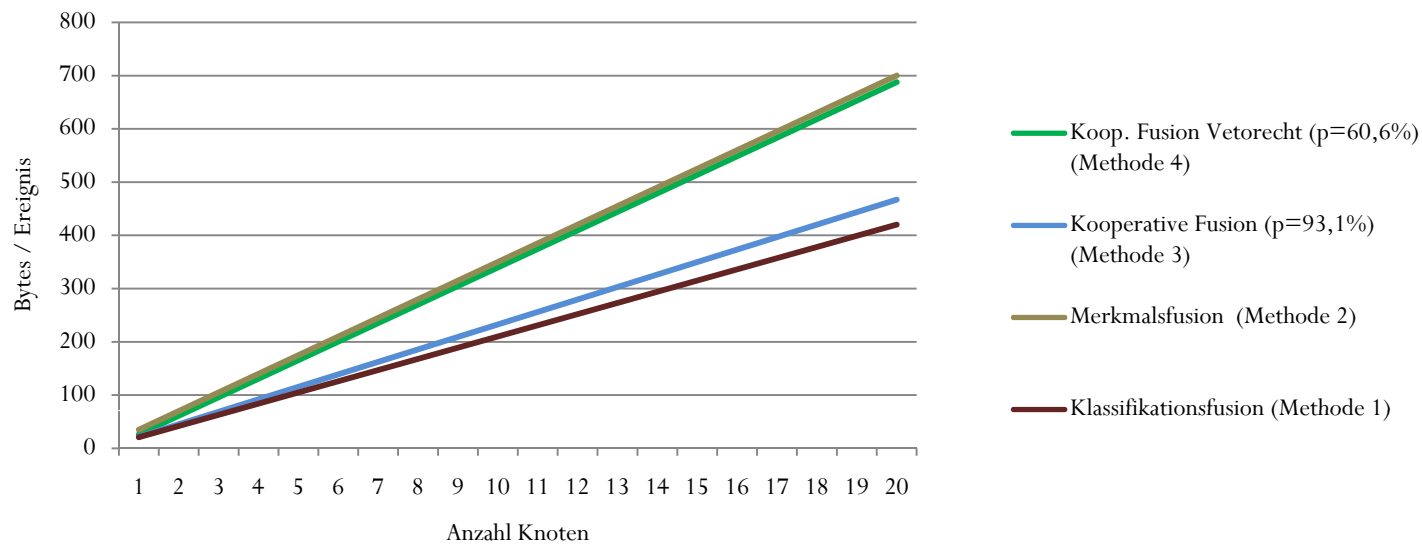
Datenvolumen bei sechs Merkmalen (spezifisches p)



Spezifischer Methodenvergleich des Datenvolumens bei sechs Merkmalen.
 p wird mittels Versuchen konkret ermittelt



Datenvolumen bei 15 Merkmalen (spezifisches p)

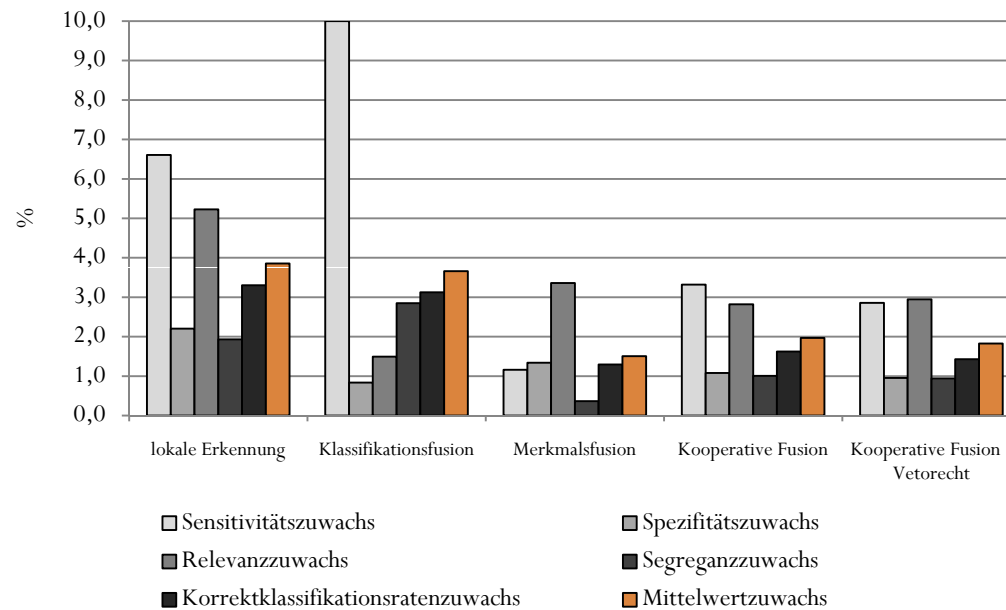


Spezifischer Methodenvergleich des Datenvolumens bei 15 Merkmalen mit spezifischem p der Methode 3 und 4



Projektvertraute Personen verbessern Auswertung

Vergleich von Kennwertverbesserungen (in Prozentpunkten)
durch projektvertraute Personen

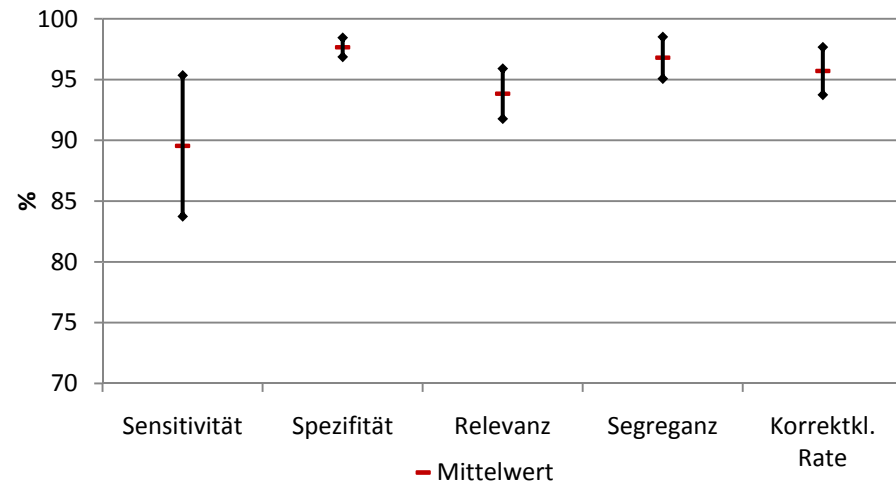
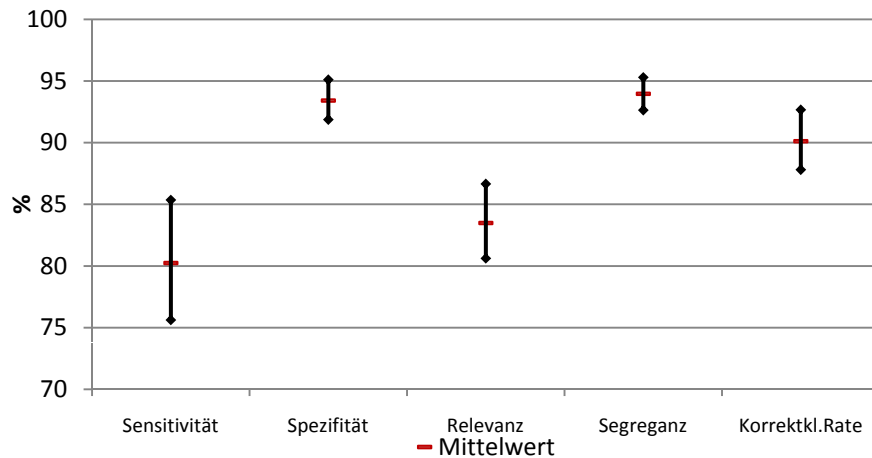


Das System wurde von der projektinterne Person trainiert und erreicht durch diese Person die besten Kennwerte.



Kennwertstreuweiten lokal und verteilt

Streuweite erzielter Kennwerte der lokalen Erkennung der jeweiligen Probanden (ein projektinterner Proband, drei projektfremde Probanden)



Streuweite erzielter gemittelter Kennwerte der verteilten Erkennung der jeweiligen Probanden (ein projektinterner Proband, drei projektfremde Probanden)