INF FAKULTÄT FÜR INFORMATIK

Intelligente Systeme

Agenten

Prof. Dr. R. Kruse **C. Braune**

{rudolf.kruse,christian.braune}@ovgu.de

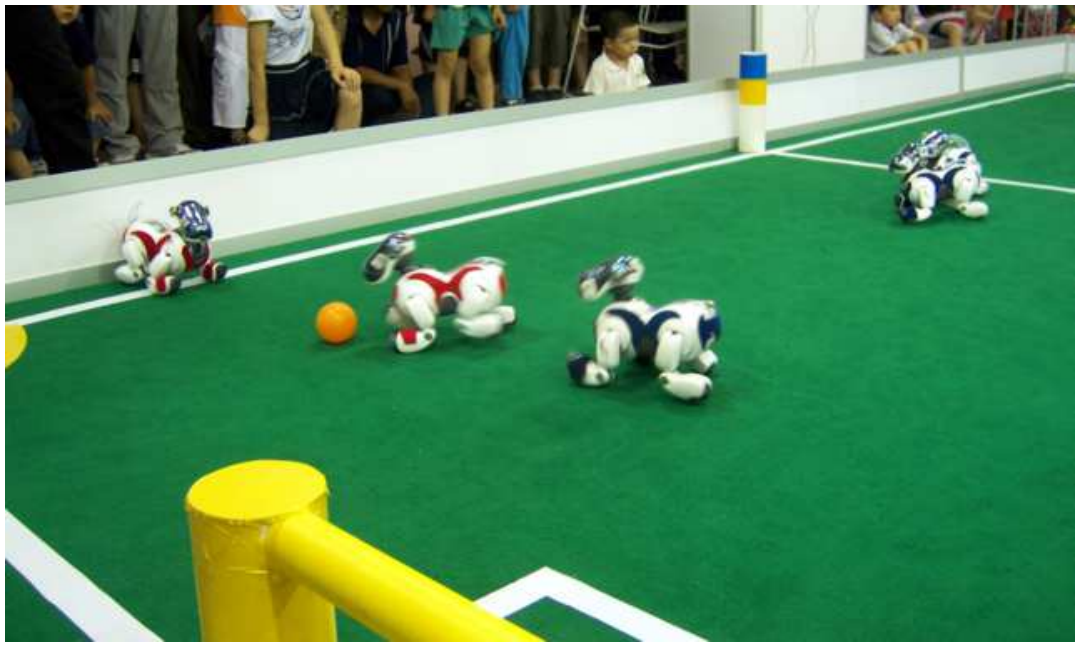
Institut für Intelligente Kooperierende Systeme
Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg

Übersicht

Intelligente Systeme beim „Robocup“

Ziel:

> *„Bis zum Jahr 2050 soll ein Team von vollständig autonomen humanoiden Robotern entwickelt werden, die gegen das menschliche Fußballweltmeisterschaftsteam gewinnen kann“*



Humanoider Roboter

weitere Informationen zum Robocup: `http://www.robocup.org`

„DARPA Grand Challenge“

Zwei (von elf) Teams: *Stanford Racing* und *Victor Tango*

Bild von `http://www.darpa.mil/GRANDCHALLENGE/gallery.asp`

Aktuelle Motivation: Alphago

AlphaGo

2016: Alphago besiegt Lee Sedol 4:1

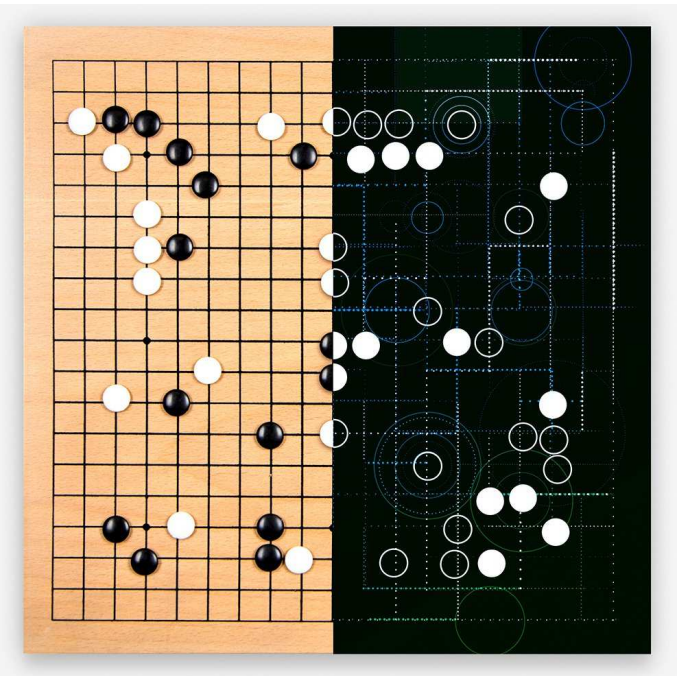
Es nutzt dabei tiefe Neuronale Netze:

Policy-Networks zur Vorauswahl möglicher Züge

Value-Networks zur Bewertung von Spielpositionen

Monte-Carlo Baumsuche zum Finden der besten Züge

Reinforcement Learning zur Verbesserung der Neuronalen Netze

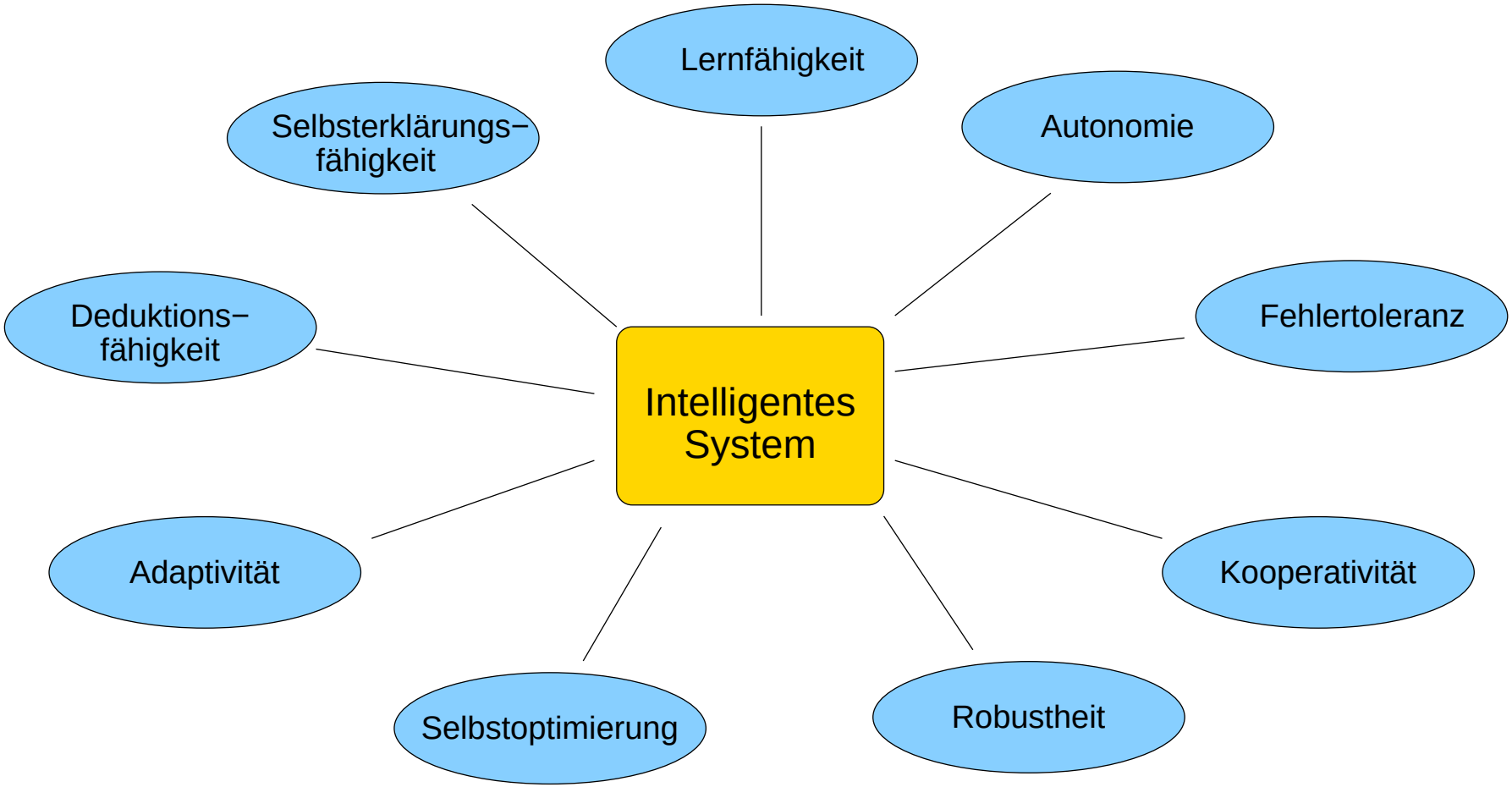


Was ist ein Intelligentes System?

In dieser Vorlesung werden Methoden der „Computational Intelligence“ sowie verwandte Ansätze vorgestellt.

Aus Sicher der Künstlichen Intelligenz handelt es sich um sogenannte „Subsymbolische Verfahren“

Merkmale Intelligenter Systeme



Künstliche Intelligenz

Der Begriff *Intelligentes System* wandelt sich mit wissenschaftlichem Fortschritt:

1956 Dartmouth Project on Artificial Intelligence

1967 Taschenrechner

1997 Schachcomputer DEEP BLUE gewann gegen Kasparov 3.5 zu 2.5

1998 Automatikgetriebe AG4 für VW New Beetle

2002 Schachcomputer DEEP FRITZ gegen V. Kramnik 4 zu 4

2006 VW Touareg Stanley bei DARPA Grand Challenge

2011 IBM Watson (Jeopardy)

2015 Google ALPHAGO schlägt Go-Weltmeister

2017 LIBERATUS schlägt Poker-Profis

...

Künstliche Intelligenz

Problem: oft anmaßende Voraussagen über Fortschritte der Künstlichen Intelligenz (KI) 1957: H. A. Simon (Nobelpreis 1978) und A. Newell (Turing-Award 1975) behaupten, dass im Jahr 1967

- Rechner Schachweltmeister sein wird,
- Computer wichtigen neuen mathematischen Satz entdecken und beweisen wird,
- digitaler Rechner Musikstück schreiben wird, welchem von Kritikern beachtlicher ästhetischer Wert bescheinigt wird.

Künstliche Intelligenz

1993: deutscher Delphi-Report behauptet, dass 2005

- Rechner entwickelt werden, die ungenaue Informationen in einer Art gesunden Menschenverstandes verarbeiten können,
- Informationsdatenbanken eingesetzt werden, die durch automatisches Lernen ihr Wissen vermehren,
- tragbare automatische Übersetzungsgeräte (einfache, alltägliche Konversation in beiden Richtungen) mit Spracheingaben kommerzialisiert werden,
- in Büros Geräte weitverbreitet sind, die Texte in handschriftlicher Fließschrift lesen können.

Künstliche Intelligenz

kognitiver Ansatz: Simulation kognitiver Prozesse, Analyse menschlicher Denkweise, *general problem solver*

ingenieurwissenschaftlicher Ansatz: Konstruktion von Systemen, die gewisse menschliche Wahrnehmungs- und Verstandsleistungen maschinell verfügbar machen (Produkte wie Gesichtserkenner, Roboter, Kooperation mit Gehirnforschern, . . .)

Künstliche Intelligenz

Spannende philosophische Fragestellungen, wie z.B.

Can machines think?

"can"

Mehrere Bedeutungen:

- „Kann denken": heute – irgendwann – im Prinzip
- derzeit: Frage „unentscheidbar", ob Systeme mit menschenähnlichen Fähigkeiten gebaut werden können

  Wir sind mit Fortschritten auf dem Weg dahin zufrieden und verdienen Geld sowie Ruhm mit innovativen Produkten

Künstliche Intelligenz

"machine"

- muss kein Stahlroboter sein
- kann auch biologischer Mechanismus sein (Bakterium Haemophilus influenzae Rd hat 10^7 Basenpaare, 1743 Gene, ...)

  Was, wenn menschliches Genom entziffert und verstanden?

  auch: Bewusstseinsdiskussion, Leib-Seele-Problem, das „Ich" im Gehirn

Künstliche Intelligenz

"think"

- Menschen sind Maschinen, also können Maschinen denken
- Searle (1992): Denken funktioniert nur in speziellen (tatsächlich lebenden) Maschinen
- Newell & Simon (1976): Physical symbol system (PSS) hypothesis:
  PSS hat notwendige und hinreichende Bedingungen für intelligentes Verhalten und PSS ist eine Maschine, die symbolische Daten manipulieren kann (z.B. Computer).
- Kohonen u.a. (1980): Entwicklung intelligenter Maschinen nur durch subsymbolische Prozesse (z.B. Signale)
- Zadeh (1964): wirklich intelligente Systeme müssen Art *fuzzy logic* benutzen (keine binäre Logik)
- Turing-Test (1950) (*"bestanden"* 2014?)

Turing-Test (1950)

Der Turing-Test wird von drei Personen gespielt: einem Mann (A), einer Frau (B) und einem Fragesteller (C). Der Fragesteller befindet sich in einem Raum, abgeschottet von A/B, und kommuniziert mit diesen über ein Terminal (teletype). Das Ziel des Spiels für den Fragesteller ist zu bestimmen, welche der beiden Personen der Mann und welche die Frau ist. Er adressiert die beiden mit Variablen X/Y und am Ende des Spiels sagt er "X ist A und Y ist B" oder "X ist B und Y ist A". Der Fragesteller kann Fragen beispielsweise der folgenden Form stellen:

> *C: X, würden Sie mir bitte Ihre Haarlänge verraten?*

Wenn mit "X" A adressiert wird, dann muß A jetzt antworten. Für A geht es darum, C in die Irre zu leiten und ihn zu einer falschen Identifikation zu verleiten.

...

Turing-Test (1950)

Das Ziel für den dritten Spieler B ist, dem Fragesteller zu helfen.

...

Jetzt stellen wir uns die Frage: "Was passiert, wenn eine Maschine den Anteil von A an diesem Spiel übernimmt?" Wird sich der Fragesteller genauso oft falsch entscheiden wenn das Spiel mit einer Maschine gespielt wird, wie wenn es mit Mann/Frau gespielt wird? Diese Fragestellung ersetzt das ursprüngliche "Kann eine Maschine denken?" Der Turing-Test wird oft vereinfacht zu einem Test, in dem eine Maschine versucht, einen menschlichen Fragesteller dazu zu verleiten, sie als Mensch zu identifizieren.

Turing-Test Beispiel 1

Judge: How do you like Bletchley Park?

Entity: lol.

Judge: Are you from England?

Entity: They have Wi-Fi here in the pub.

Judge: Which pub?

Entity: I'm just down the pub.

Judge: Have you ever been in a Turing Test before?

Entity: Will this be the 5 min argument, or were you thinking of going for the full half hour?

Judge: Very funny. You sound suspiciously human. Do you like the beatles?

Entity: I'd like to get the next Dread the Fear tape.

Judge: What is Dread the Fear?

Entity: Dread the fear has that Steve Henderson guy in it.

Judge: What sort of music is that? Or is it comedy?

Turing-Test Beispiel 2

Judge: Why hello there!

Entity: Why hello to you too!

Judge: How are you feeling in this fine day?

Entity: To be quite honest a little rejected, I thought you were never going to reply :(

Judge: Oh, I'm very sorry, it will not happen again.

Entity: It just did!

Judge: Oh, I lied then.

Entity: That's a great shame indeed.

Judge: It is. Are you following the Euro 2012's at the moment?

Entity: Yeah quite closely actually. I am Cristiano Ronaldo.

Frage: Ist der Teilnehmer *Entity* ein Mensch oder ein Computer?

Was ist ein Intelligentes System?

Gebiet der Wissensverarbeitung ist extrem innovativ:

- objektorientierte Programmiersprachen
- graphische Oberflächen
- Expertensysteme
- Software-Agenten (Internet)
- Autonome Roboter

sind hier erfunden worden.

Methoden der Computational Intelligence sind anwendungsorientiert.

CI: Technologien und Anwendungen

Kerntechnologien

Neuronale Netze
Fuzzy-Logik
Probabilistisches Schließen
Evolutionäre Algorithmen und weitere Metaheuristiken
Hybride Systeme

Verwandte Technologien

Fallbasiertes Schließen
Regelbasierte Expertensysteme
Maschinelles Lernen
Deep Learning
Belief-Netze

Anwendungen

Klassifikation und Clustering

- Überwachung/ Anomalieerkennung
- Diagnose
- Prognose
- Konfiguration/Initialisierung

Vorhersage

- Qualitätskontrolle
- Alterungsprozessmodellierung

Terminplanung

- Zeit-/Ressourcenzuweisung

Regelung

- Maschinen-/Prozesskontrolle
- Prozessinitialisierung
- Überwachungskontrolle

Automatische Entscheidung

- Kosten-/Risikoanalyse
- Einkommensoptimierung

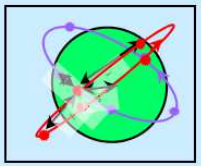
CI: Anwendungsbeispiele

Haushaltsgeräte
- Bevorzugte Serviceverträge (Stat.)
- Call-Center-Support (CBR)

Kapitalservice
- Kreditwürdigkeitsbewertung (Fusion/FL/CBR)

Finanzversicherungen
- Bevorzugte Kunden (Stat./NN)

Plastik
- Automatische Farbanpassung (CBR)



Föderierte Systeme
- Terminwartung von Satelliten-konstellationen (GA)

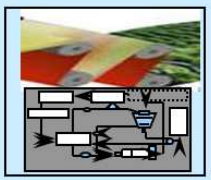
Schifffahrt
- Schiffmanagementsystme (AI/GA)

Medizinische Systeme
- Automatische Analyse von MRT (FL)
- Reverse Engineering von Picker (FL)
- Finite-Element-Analyse (FL)
- Analye von Röntgenfehlern (CBR)

Flugzeugmotoren
- Zentrum für Ferndiagnosen (CBR)
- Kundenanfragecenter (CBR)
- Ausreißererkennung (FL/Stat.)
- Wartungsberater (NN/FL)
- Sensorfusion (FL)

Transportsysteme
- Erfassung von Transport-DB (CBR)
- Prototyp. Zugführerkontrolle (FL/GA)
- Prototyp. Trendanalyse (Stat.)
- Eingebettete/Ferndiagnose (BBN)

Stromerzeugung
- Fernausreißererkennung (Stat.)
- Eingebettete/Ferndiagnose (BBN)
- Call-Center-Problem/Lösung (CBR)



Industriesysteme
- Vorhersage eines Abbruchs der Papierbahn (NN/Stat./Induction)
- Rührkontrolle von Zement (FL/GA)

Beispiel: DAX-Prognosen

Datenbasis: Zeitreihe von 1986–1997

DAX	Composite-DAX
Deutsche 3-Monate-Zinsrate	Return Germany
Deutscher Morgan-Stanley-Index	Industrie-Index des Dow Jones
DM / US-$	US Schatzanweisungen
Goldpreis	Japanischer Nikkei-Index
Europäischer Morgan-Stanley-Index	Verhältnis von Preis und Ertrag

Fuzzy-Regeln in der Finanzwelt

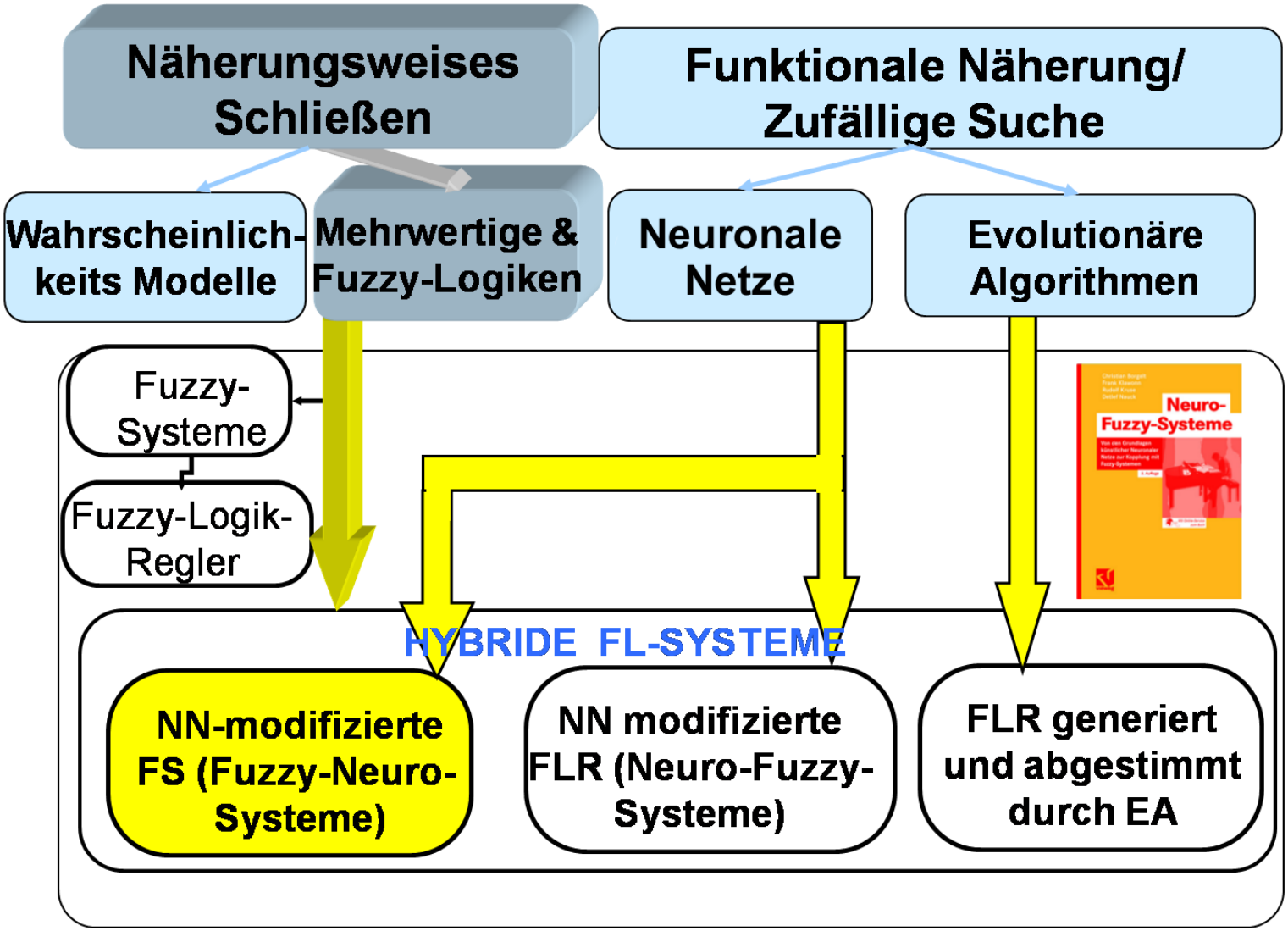
Entwicklungsregel
- WENN DAX = fallend UND US-$ = fallend
- DANN DAX-Voraussage = fallend
- MIT hoher Gewissheit

Wendepunktregel
- WENN DAX = fallend UND US-$ = steigend
- DANN DAX-Voraussage = steigend
- MIT niedriger Gewissheit

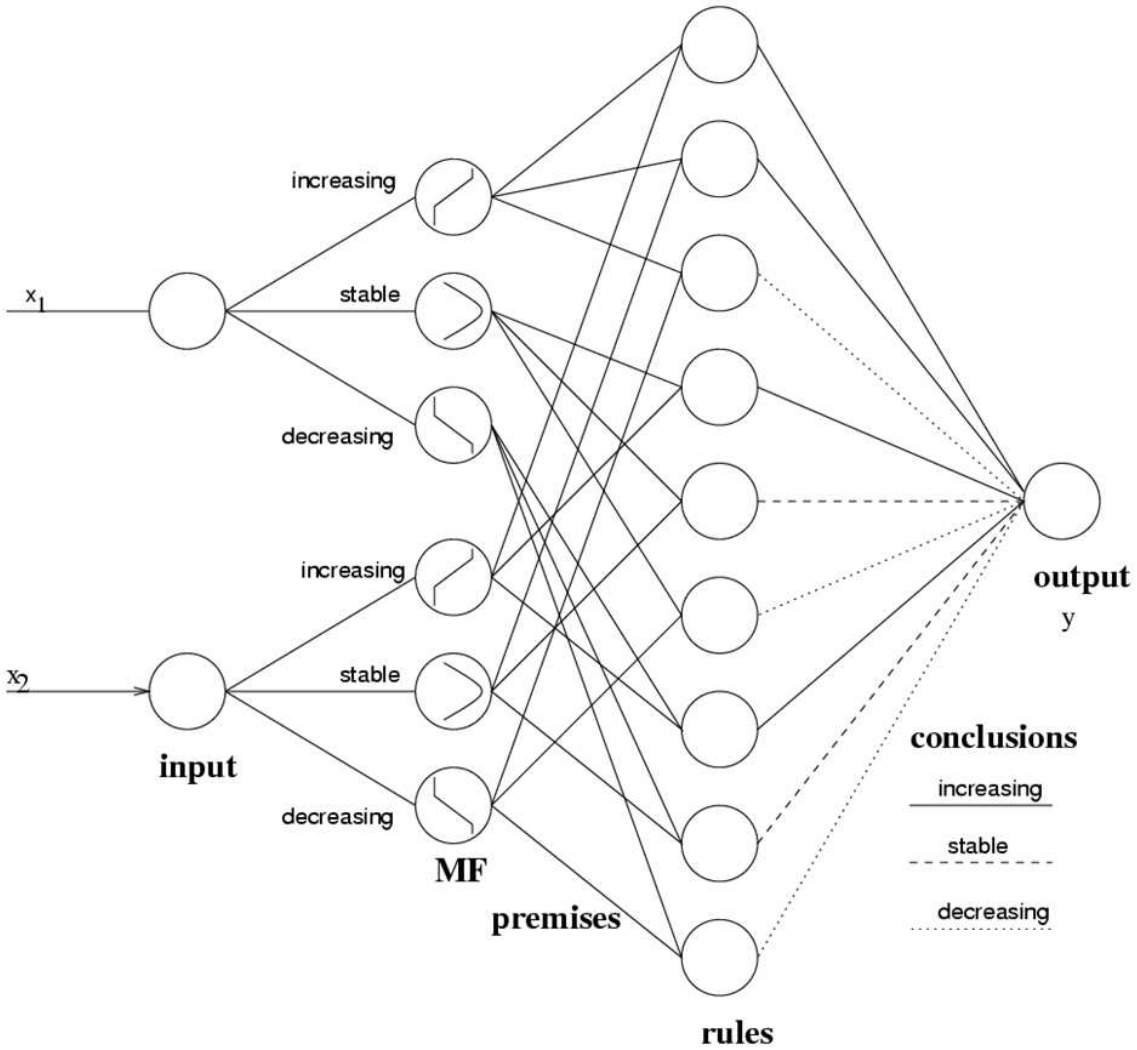
Verzögerungsregel
- WENN DAX = stabil UND US-$ = fallend
- DANN DAX-Voraussage = fallend
- MIT sehr hoher Gewissheit

generell
- WENN x_1 ist μ_1 UND x_2 ist μ_2 UND ... UND x_n ist μ_n
- DANN $y = \eta$
- MIT Gewicht k

Neuro-Fuzzy-Architektur



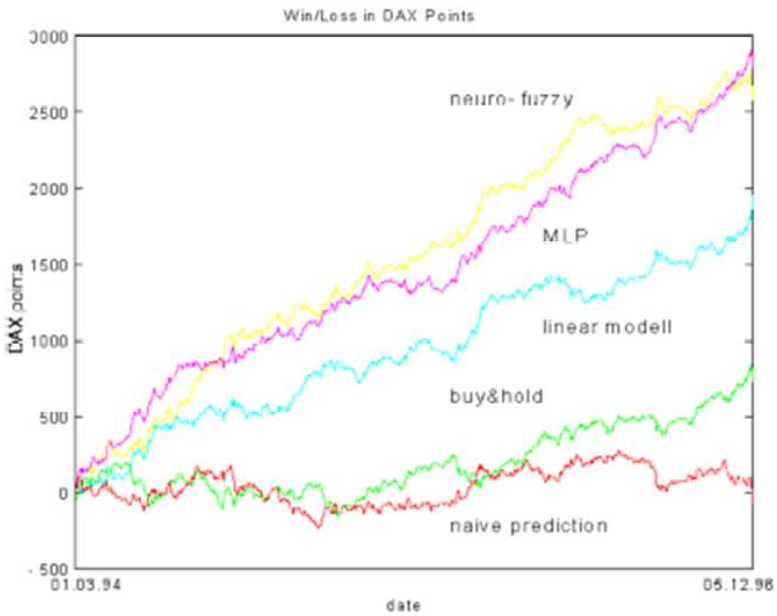
Neuro-Fuzzy-Architektur

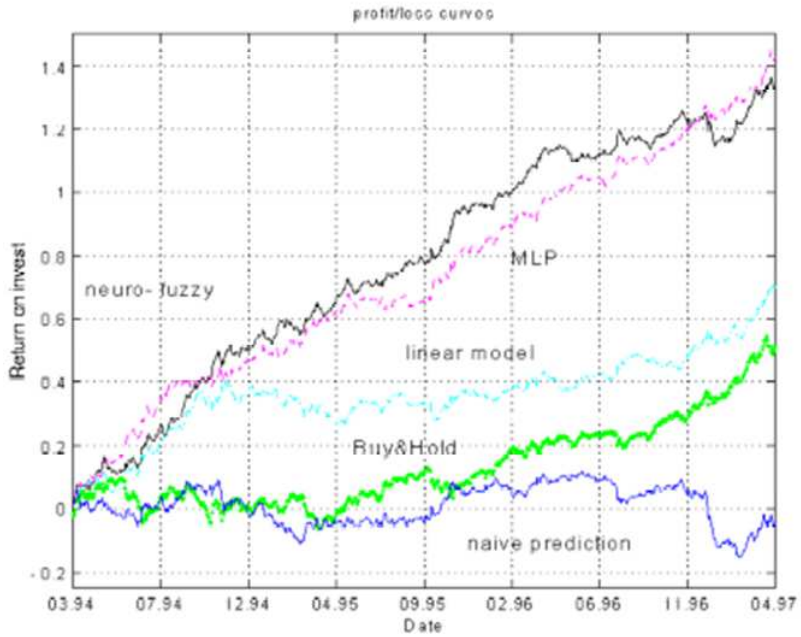


Return-on-Investment-Kurven

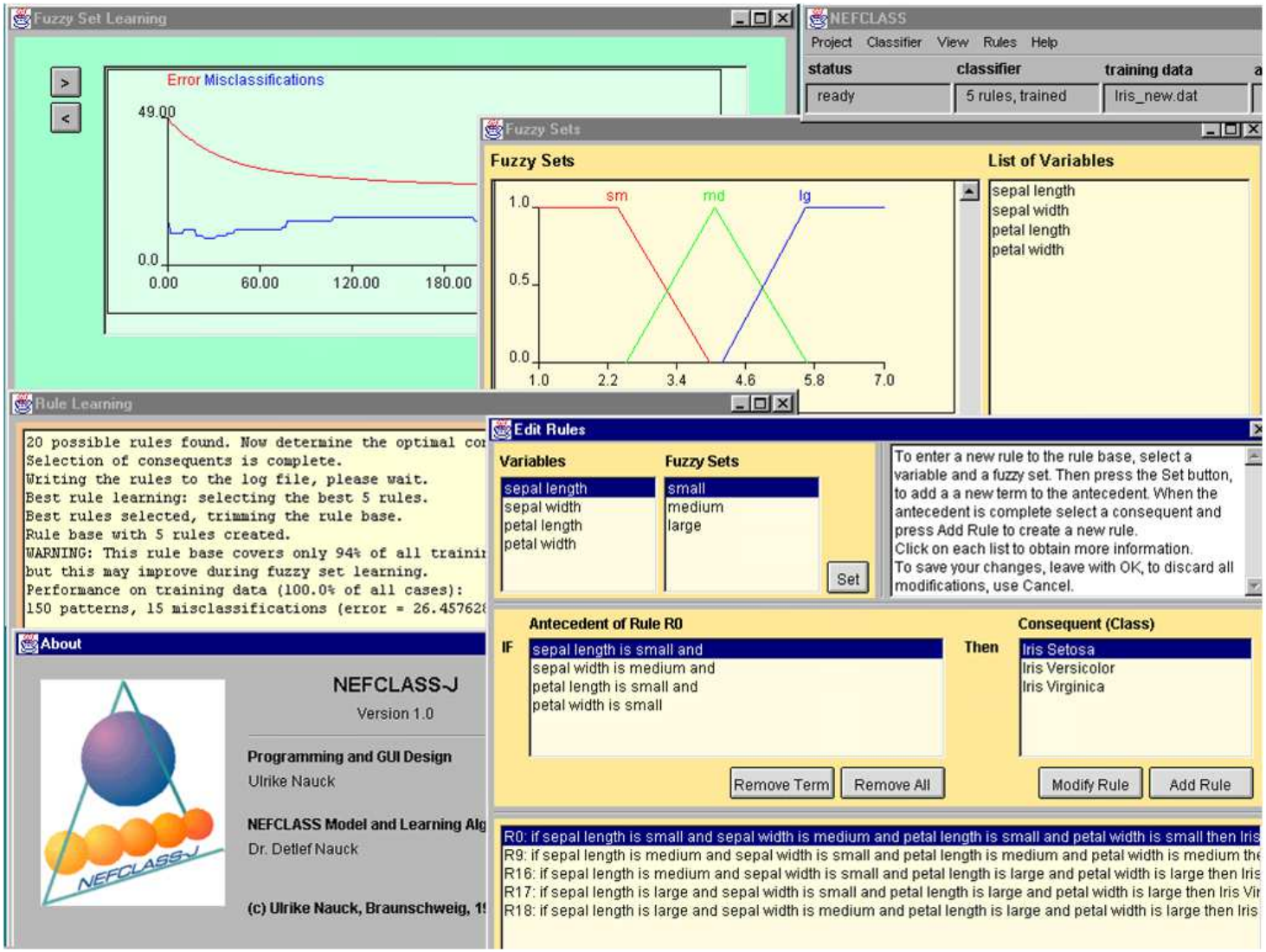
verschiedene Modelle

Validierungsdaten: 01. März 1994 bis April 1997





NEFCLASS-J


Fuzzy Set Learning
Error Misclassifications
49.00
0.0
0.00
60.00
120.00
180.00
NEFCLASS
Project Classifier View Rules Help
status
classifier
training data
ready
5 rules, trained
Iris_new.dat
Fuzzy Sets
Fuzzy Sets
List of Variables
sm
md
lg
1.0
0.5
0.0
1.0
2.2
3.4
4.6
5.8
7.0
sepal length
sepal width
petal length
petal width
Rule Learning
20 possible rules found. Now determine the optimal co
Selection of consequents is complete.
Writing the rules to the log file, please wait.
Best rule learning: selecting the best 5 rules.
Best rules selected, trimming the rule base.
Rule base with 5 rules created.
WARNING: This rule base covers only 94% of all traini
but this may improve during fuzzy set learning.
Performance on training data (100.0% of all cases):
150 patterns, 15 misclassifications (error = 26.45762
Edit Rules
Variables
Fuzzy Sets
sepal length
sepal width
petal length
petal width
small
medium
large
Set
To enter a new rule to the rule base, select a variable and a fuzzy set. Then press the Set button, to add a a new term to the antecedent. When the antecedent is complete select a consequent and press Add Rule to create a new rule.
Click on each list to obtain more information.
To save your changes, leave with OK, to discard all modifications, use Cancel.
About
NEFCLASS-J
Version 1.0
Programming and GUI Design
Ulrike Nauck
NEFCLASS Model and Learning Alg
Dr. Detlef Nauck
(c) Ulrike Nauck, Braunschweig, 1
Antecedent of Rule R0
IF
sepal length is small and
sepal width is medium and
petal length is small and
petal width is small
Then
Consequent (Class)
Iris Setosa
Iris Versicolor
Iris Virginica
Remove Term
Remove All
Modify Rule
Add Rule
R0: if sepal length is small and sepal width is medium and petal length is small and petal width is small then Iris
R9: if sepal length is medium and sepal width is small and petal length is medium and petal width is medium the
R16: if sepal length is medium and sepal width is small and petal length is large and petal width is large then Iris
R17: if sepal length is large and sepal width is small and petal length is large and petal width is large then Iris Vir
R18: if sepal length is large and sepal width is medium and petal length is large and petal width is large then Iris

Beispiel: Qualitätskontrolle

Heutiges Verfahren

Oberflächenkontrolle: manuell durchgeführt

Erfahrener Arbeiter bearbeitet Oberfläche mit Schleifstein

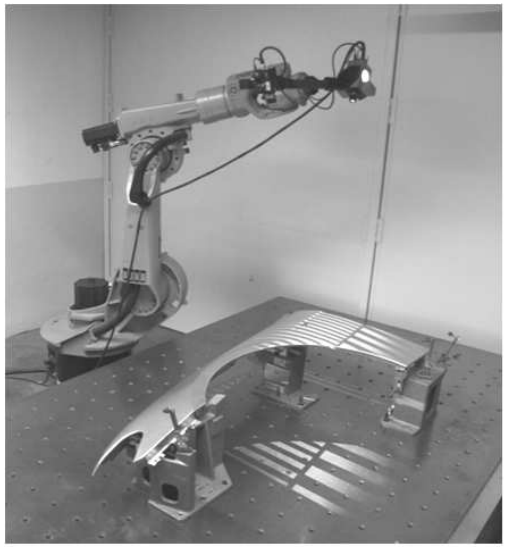
Experten klassifizieren Abweichungen durch sprachliche Beschreibungen

Umständlich, subjektiv, fehleranfällig, zeitaufwendig

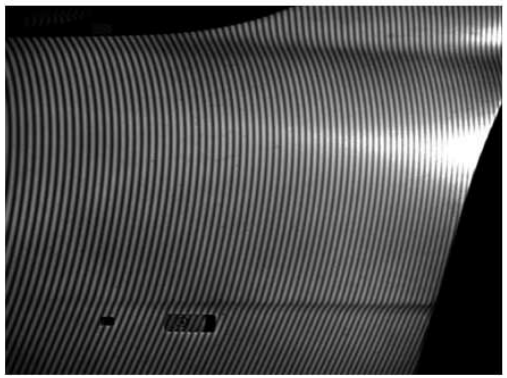
Vorgeschlagener Ansatz:

Digitalisierung der Oberfläche mit optischen Mess-Systemen
Charakterisierung der Formabweichungen durch mathematische Eigenschaften (nahe der subjektiven Merkmale)

Topometrisches 3D-Mess-System



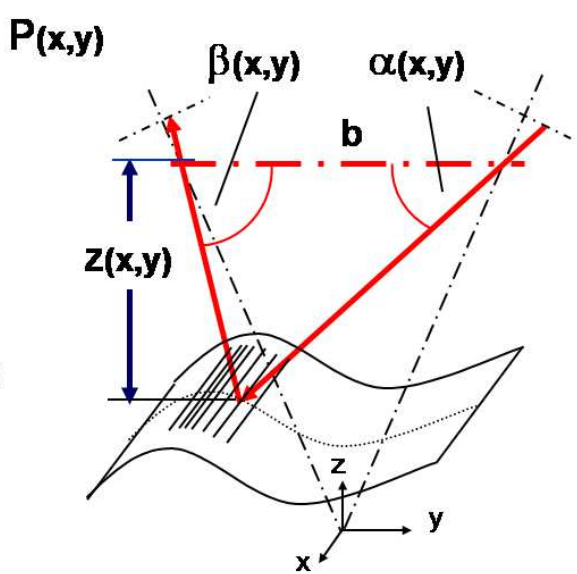
breuckmann



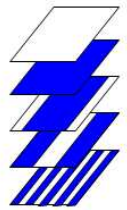
Triangulation und Gitterprojektion

0
1
0
0
⋮

Pixelkodierung



φ_n

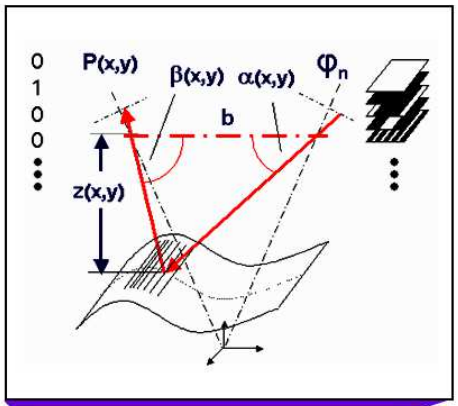


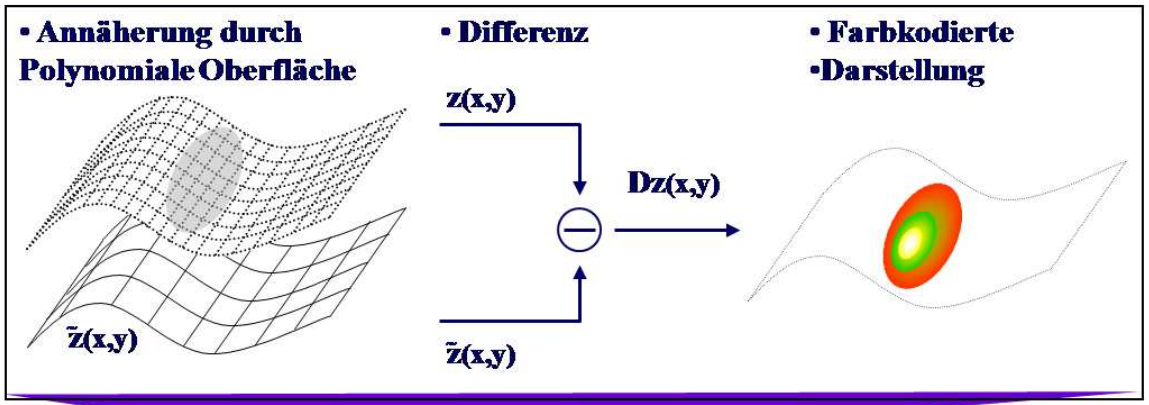
⋮

Verkleinerte Projektionstechnik (Phasenverschiebung von Grautönen)

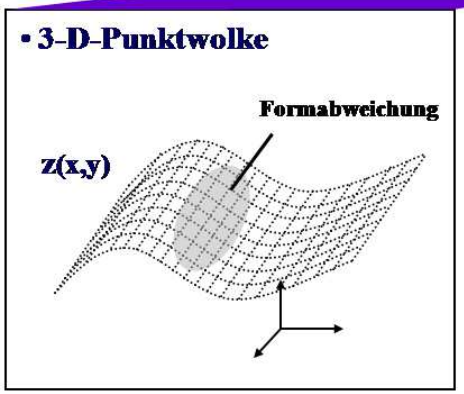
- hohe Punktdichte
- schnelle Datenansammlung
- genaue Messung
- kontakt- und harmlos

Datenverarbeitung

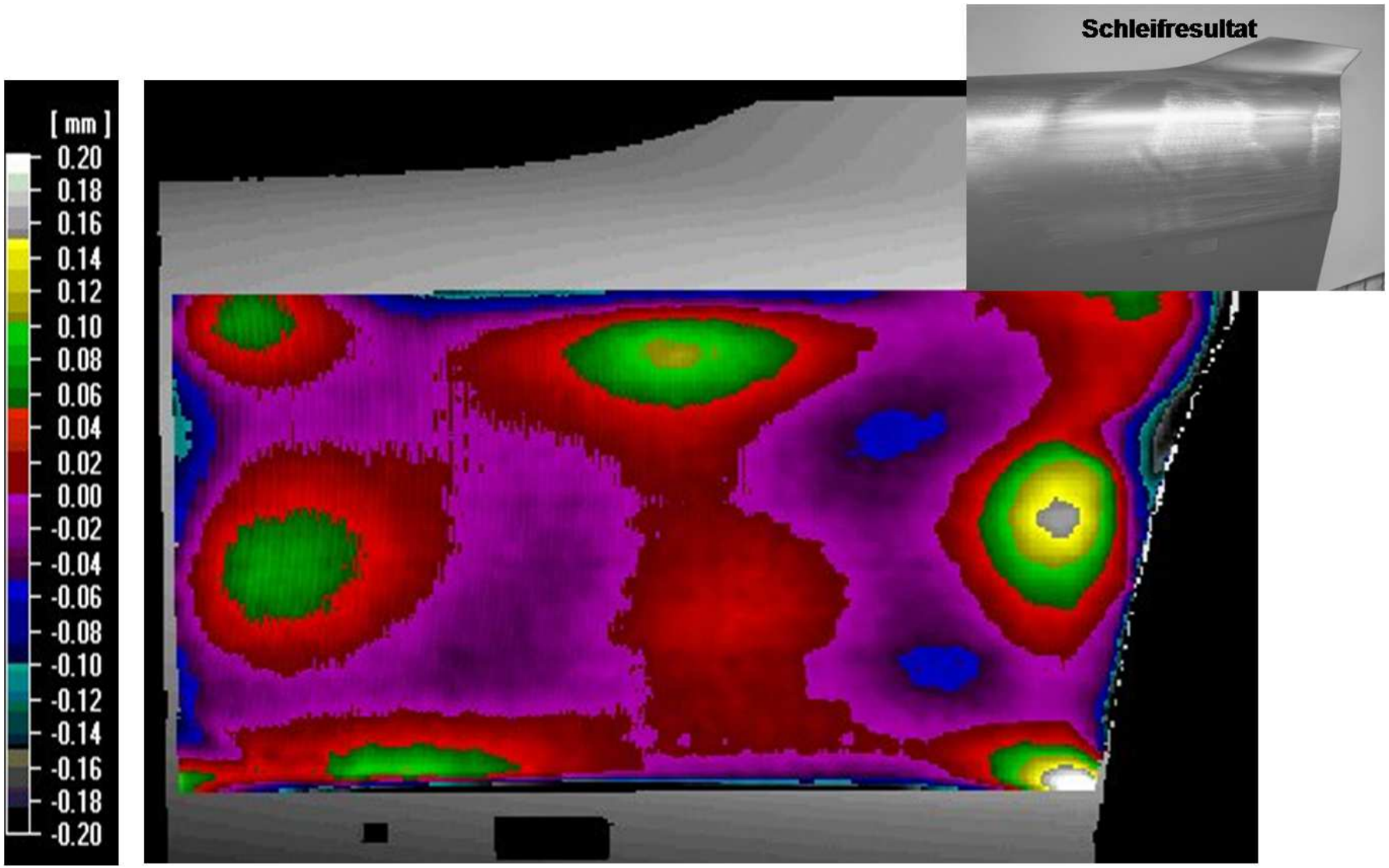




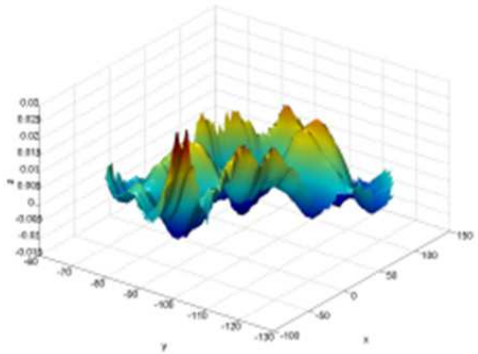
3-D-Daten-aufnahme → Nachverarbeitung → Erkennung von Abweichungen → Merkmalsanalyse



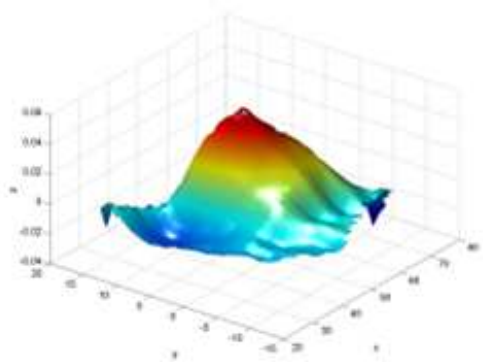
Farbkodierte Darstellung



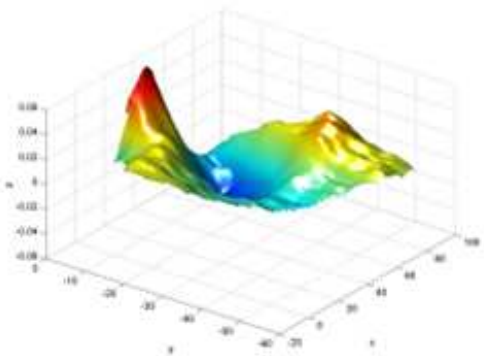
3D-Darstellung lokaler Oberflächendefekte



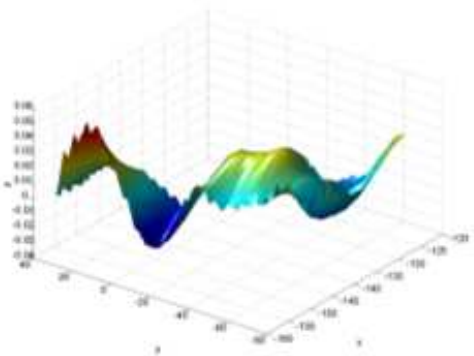
Unebene Oberfläche
Mehrere Einfallsstellen in Serie/benachbart



Walzenmarkierung
Lokale Glättung der Oberfläche



Einfallstelle
Leichte flach basierte Senke einwärts



Wellplatte
Mehrere schwerere Faltungen in Serie

Charakteristik der Daten

9 Meisterstücke mit insgesamt 99 Defekten analysiert

Für jeden Defekt wurden 42 Merkmale berechnet

Typen sind eher unbalanciert

Seltene Klassen wurden verworfen

Einige extrem korrelierte Merkmale wurden verworfen (31 übrig)

Rangfolge der 31 Merkmale nach Wichtigkeit

Geschichtete 4-fache Kreuzvalidierung fürs Experiment

Anwendung und Ergebnisse

Regelbasis für NEFCLASS:

Rule base
- Rule 1: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 1 AND max_extrema IS fun 1) THEN type IS press_mark
- Rule 2: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND max_extrema IS fun 2) THEN type IS sink_mark
- Rule 3: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
- Rule 4: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
- Rule 5: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND min_extrema IS fun 2) THEN type IS press_mark
- Rule 6: IF (max_distance_to_cog IS fun 3 AND all_extrema IS fun 2 AND max_extrema IS fun 3) THEN type IS uneven_surface
- Rule 7: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 3) THEN type IS uneven_surface

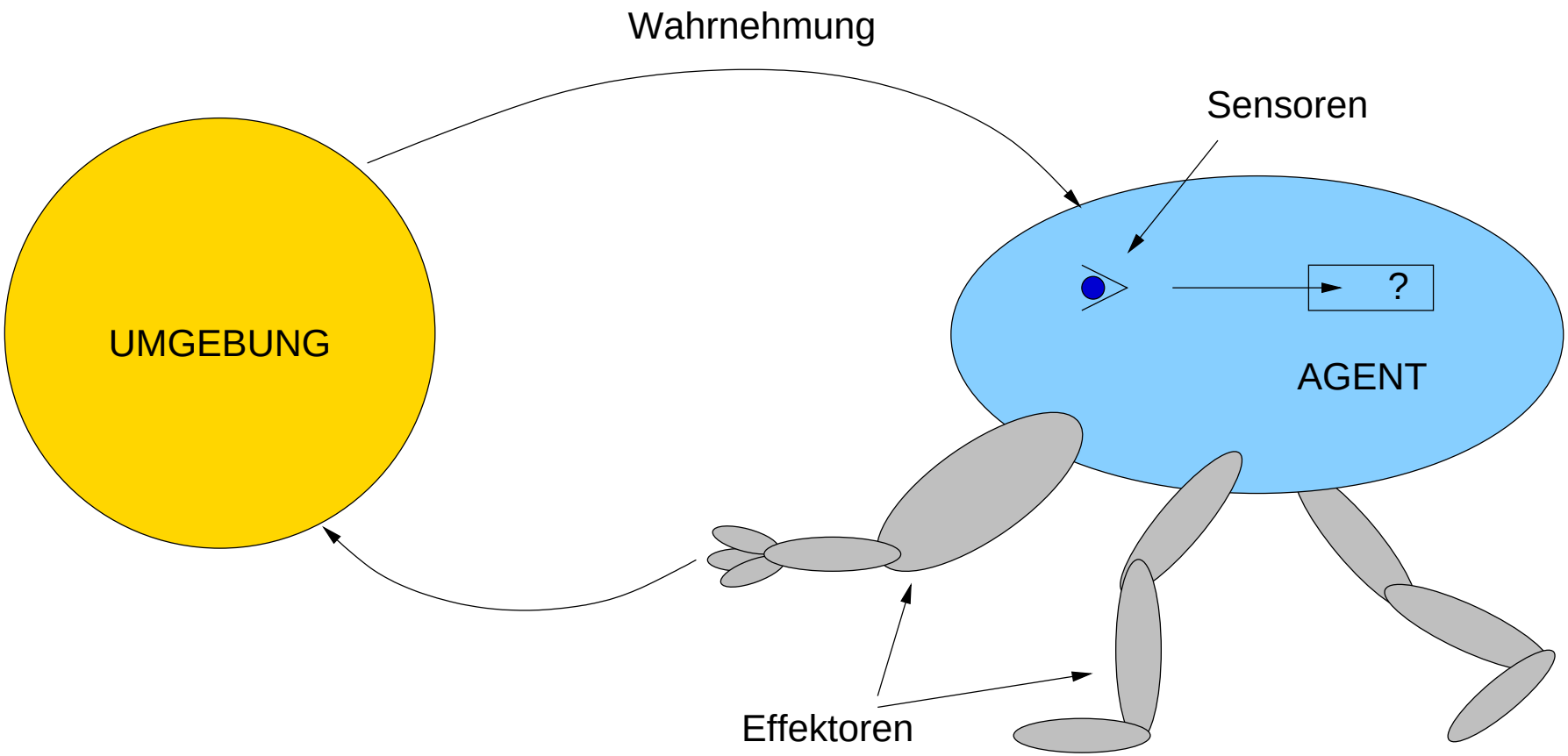
Klassifikationsgenauigkeit:

	NBC	DTree	NN	NEFCLASS	DC
Trainingsmenge	89.0%	94.7%	90%	81.6%	46.8%
Testmenge	75.6%	75.6%	85.5%	79.9%	46.8%

Übersicht

Agenten

Ein intelligenter Agent interagiert mit seiner Umgebung mittels Sensoren und Effektoren und verfolgt gewisse Ziele:

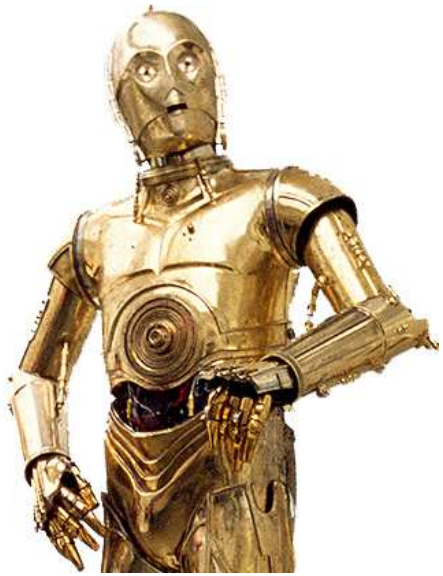


Beispiele für Agenten

Menschen und Tiere

Roboter und Software-Agenten (Softbots)

aber auch: Heizungen, ABS, ...



Agenten

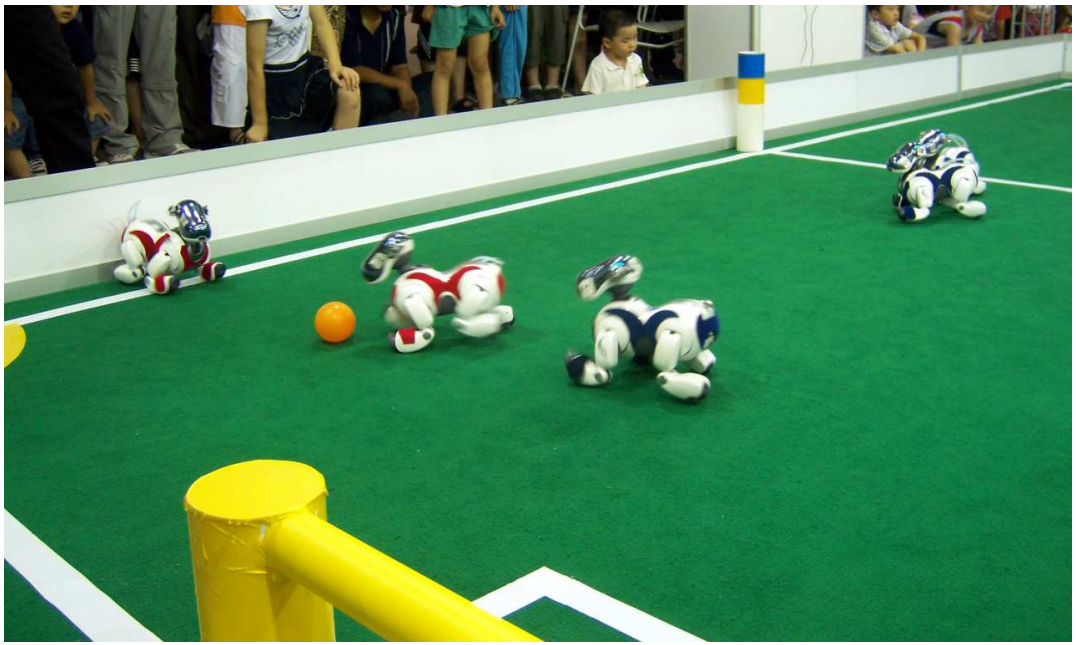
Auch andere Definitionen aus unterschiedlichen Fachgebieten, z.B.:

Ein Programm ist ein Softwareagent, wenn es korrekt in einer (Agenten-)Sprache wie ACL, KQML oder KIF kommuniziert.
BDI-Agenten werden durch Überzeugungen (*beliefs*), Wünsche (*desires*) und Absichten (*intentions*) beschrieben; praktisch werden sie mit einer Modallogik und speziellen Datenstrukturen implementiert.

Agenten

Beispiel: Simulation Soccer

RoboCup: Roboterfußball



Taxifahrer

Typ	Taxifahrer
Wahrnehmung	Kameras, Tachometer, GPS, Mikrofon
Aktionen	Steuern, Schalten, Bremsen, mit Fahrgästen sprechen
Ziele	Sichere, schnelle, legale, komfortable Fahrt; Profit maximieren
Umgebung	Straßen, andere Verkehrsteilnehmer: Fußgänger, Radfahrer; Fahrgäste

Charakterisierung von Agenten

Agenten können charakterisiert werden durch (*PAGE*):

Wahrnehmungen (***p****erceptions*)

Aktionen (***a****ctions*)

Ziele (***g****oals*)

Umgebung (***e****nvironment*)

Beispiele von Agenten nach *PAGE*

Art	Wahrnehmung	Aktionen	Ziele	Umgebung
Medizinisches Diagnosesystem	Symptome, Diagnose, Antworten des Patienten	Fragen, Tests, Behandlungen	Gesundheit, geringe Kosten	Patient, Krankenhaus
Satellitenbildanalyse	Punkte verschiedener Intensität	Klassifikation	Korrekte Klassifikation	Satellitenbilder
Roboter	Punkte verschiedener Intensität	Teile aufheben und einsortieren	Teile richtig einsortieren	Förderband mit Teilen

Beispiele von Agenten nach *PAGE*

Art	**Wahrnehmung**	**Aktionen**	**Ziele**	**Umgebung**
Raffinerie-Regler	Temperatur, Druck	Öffnen, Schließen von Ventilen, Temperatur einstellen	Reinheit, Ertrag, Sicherheit maximieren	Raffinerie
Interaktiver Englisch-Tutor	Eingegebene Wörter und Übungen, Vorschläge	Korrekturen ausgeben	Testergebnisse des Studenten maximieren	Menge von Studenten

Typen von Agenten (I)

Unterscheidung von Agenten nach Art und Weise ihrer Umwelt-Interaktionen:

reaktive Agenten:

steuern über ein Reiz-Antwort-Schema ihr Verhalten

reflektive Agenten:

agieren planbasiert, verarbeiten also explizit Pläne, Ziele und Intentionen

situierte Agenten:

verbinden einfaches Reagieren und überlegtes Handeln in dynamischer Umwelt

Typen von Agenten (II)

autonome Agenten:

sind zwischen reflektiven und situierten Agenten einzuordnen (werden meist in Robotik verwendet)

rationale Agenten:

entsprechen reflektiven Agenten, allerdings mit ausgeprägter Bewertungsfunktionalität

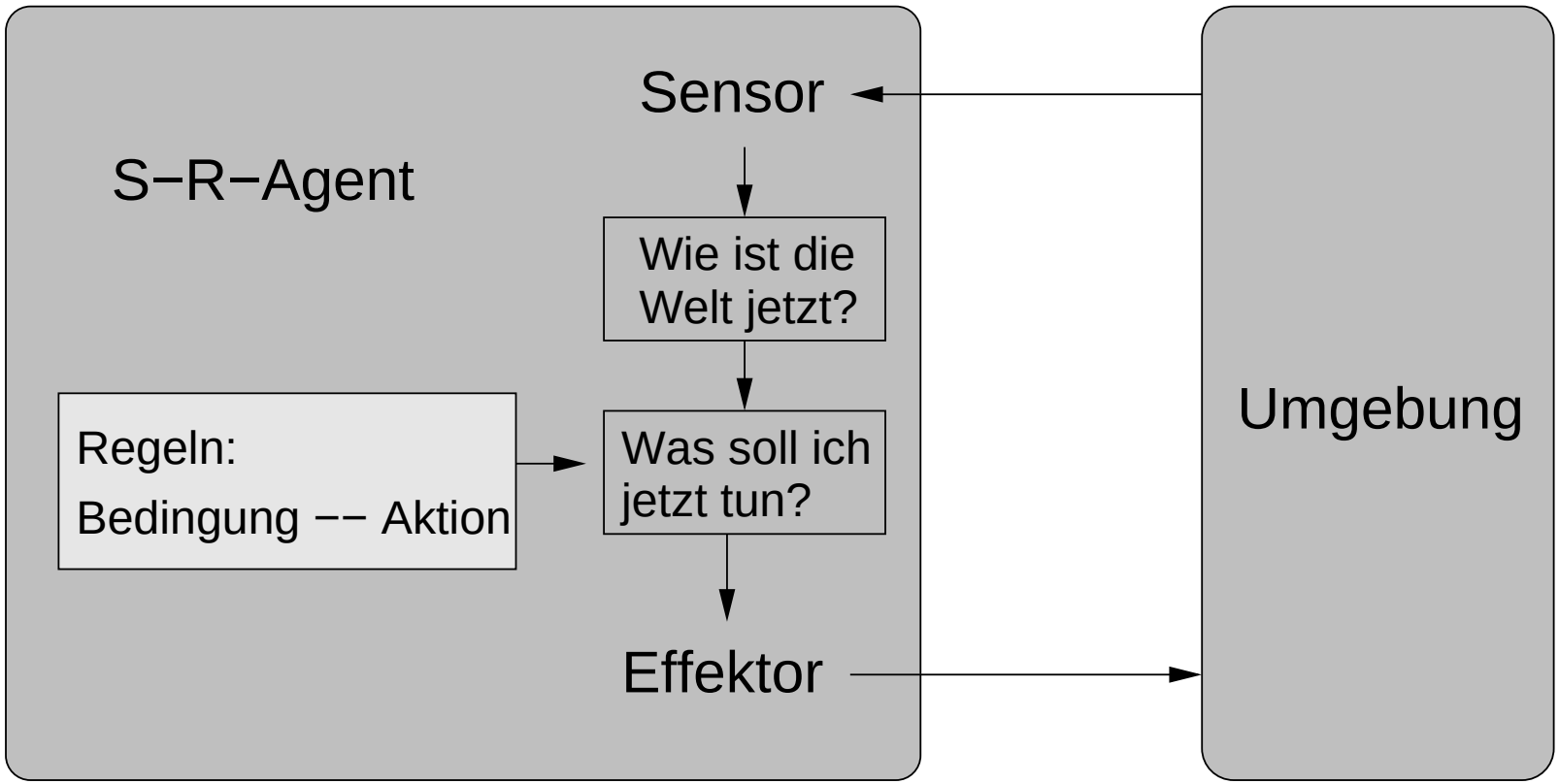
soziale Agenten:

sind in der Lage, ihr Handeln an Gemeinziel auszurichten

Übersicht

Stimulus-Response-Agent

einfacher reaktiver Agent: antwortet unmittelbar auf Wahrnehmungen



„Skelett“ eines Stimulus-Response-Agenten

```
function S-R-Agent(percept) returns action
    static: rules
    state INTERPRET-INPUT(percept)
    rule RULE-MATCH(state, rule)
    action RULE-ACTION(rule)
    return action
```

Agent sucht Regel, deren Bedingung der gegebenen Situation entspricht

er führt zugehörige Aktion (Regel-Konklusion) aus

Gitterwelt (I)

Umwelt = fiktive zweidimensionale Gitterzelleneinheit

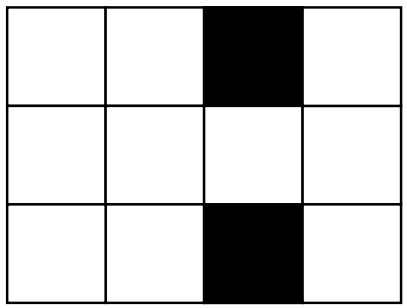
Verschiedene (Spielzeug-)Agenten tummeln sich dort

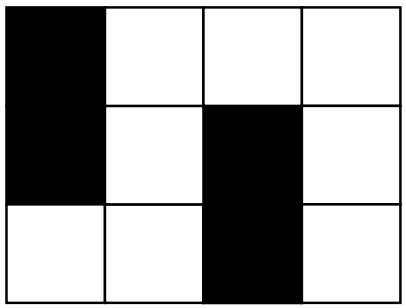
In Zellen können Objekte mit verschiedenen Eigenschaften sein

Es gibt Barrieren

Agenten können von Zelle zu Zelle laufen

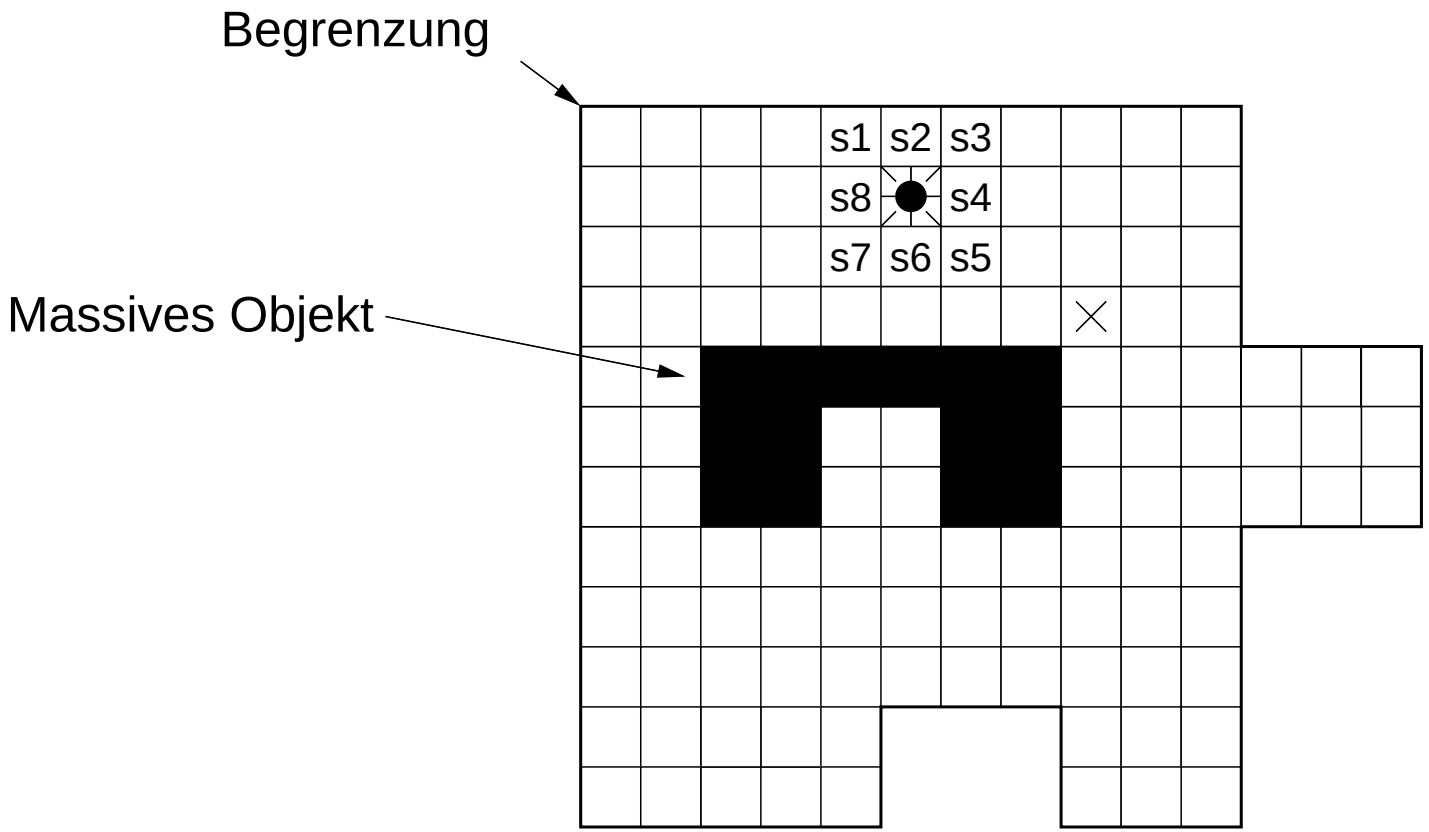
Keine engen Zwischenräume, d.h. keine Lücken zwischen Objekten und Begrenzungen, die nur 1 Zelle breit sind (*tight spaces*)





Solche Umgebungen sind *nicht* erlaubt!

Gitterwelt (II)



Roboter kann mithilfe der Sensoren $s_1, \ldots, s_8$ feststellen, welche Zellen in seiner Nachbarschaft belegt sind

Gitterwelt (III)

$s_i \in \{0, 1\}$, Sensoreingabe:

- $s_j = 0 \Leftrightarrow$ Zelle s_j ist frei für Roboter
- an der mit $\times$ markierten Stelle: $(0, 0, 0, 0, 0, 0, 1, 0)$

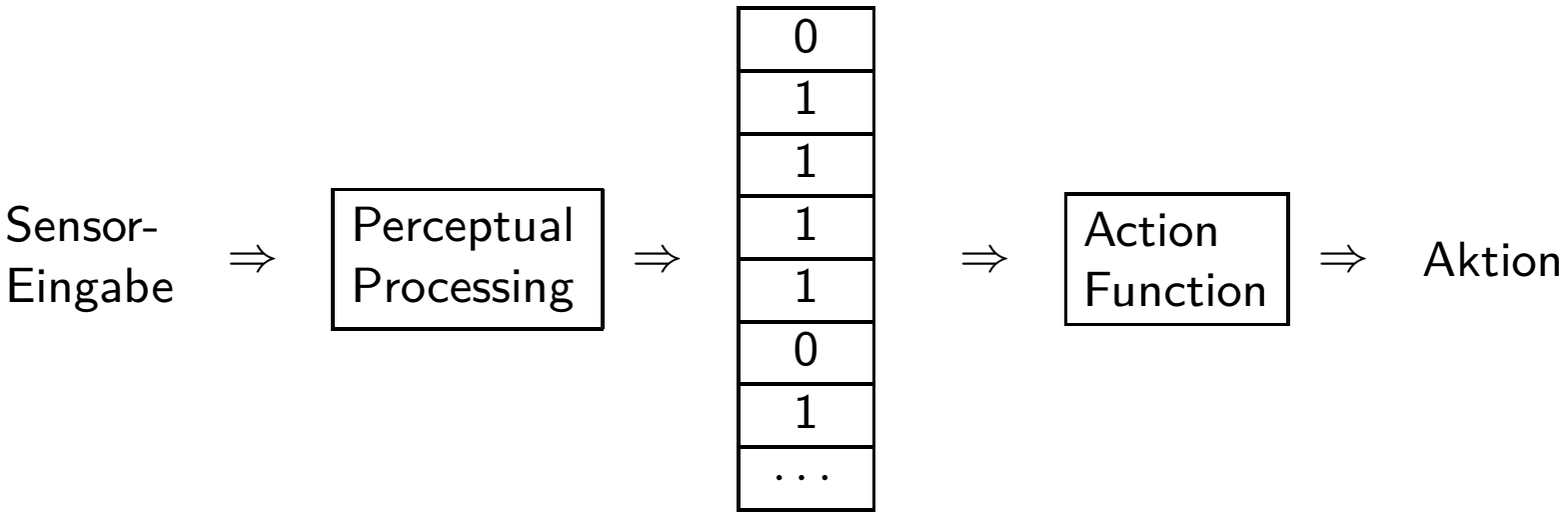
Vier mögliche Aktionen:

- north, east, south, west
- z.B. north bewegt Roboter 1 Zelle nach oben, falls Zelle frei ist, ansonsten wird nicht bewegt

Aufgabe häufig in 2 Schritten gelöst:

- Phase: perception processing
- Phase: action computation

Komponenten: *Perception* und *Action*



Eigenschaftsvektor $X = (0, 1, 1, 1, 1, 0, 1, \cdots)^T$

vom Entwickler zugewiesene Bedeutungen:

- $(0, 0, \boxed{\mathbf{1}}, 1, 1, 0, \ldots)$: „an einer Wand“
- $(0, 0, 0, 1, \boxed{\mathbf{1}}, 1, \ldots)$: „in einer Ecke“

Beispiel: Wandverfolgung (I)

Aufgabe: gehe zu einer Zelle an Begrenzung eines Objekts und folge dieser Grenze

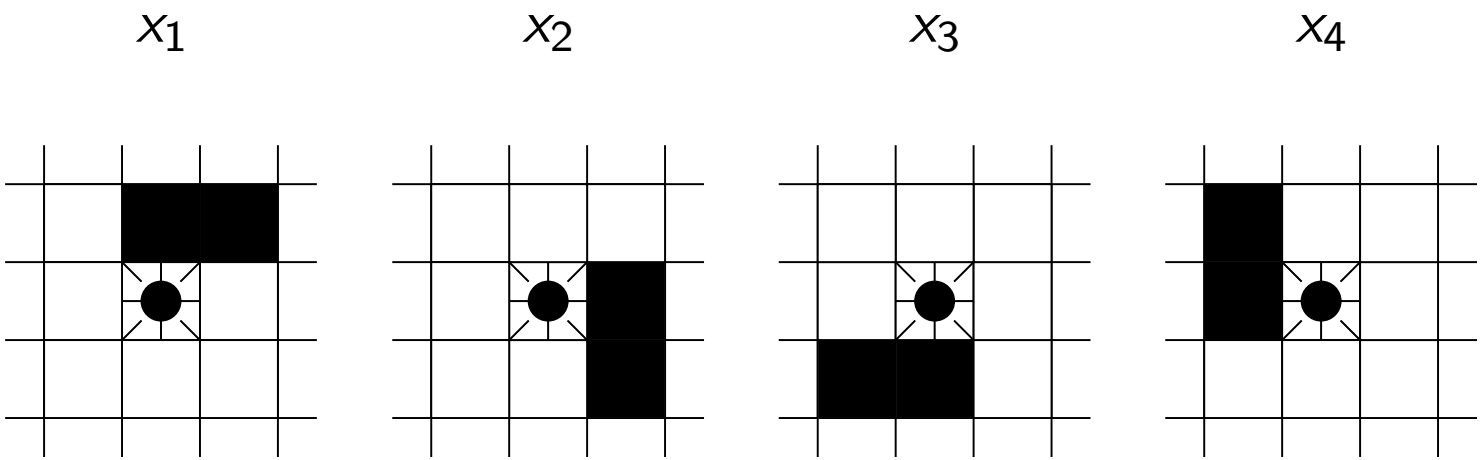
Perception:

- 2^8 verschiedene Sensoreingaben, von denen einige wegen Einschränkung (*keine engen Zwischenräume*) wegfallen
- vier Merkmale $x_1, \cdots, x_4$:

$$\begin{aligned} x_1 = 1 &\Leftrightarrow (s_2 = 1 \vee s_3 = 1) \\ x_2 = 1 &\Leftrightarrow (s_4 = 1 \vee s_5 = 1) \\ x_3 = 1 &\Leftrightarrow (s_6 = 1 \vee s_7 = 1) \\ x_4 = 1 &\Leftrightarrow (s_8 = 1 \vee s_1 = 1) \end{aligned}$$

Beispiel: Wandverfolgung (II)

Das Merkmal in jedem Diagramm hat genau dann Wert 1, wenn mindestens 1 der markierten Zellen belegt



in komplexen Welten: Informationen sind typischerweise unsicher, vage, oder sogar falsch

Beispiel: Wandverfolgung (III)

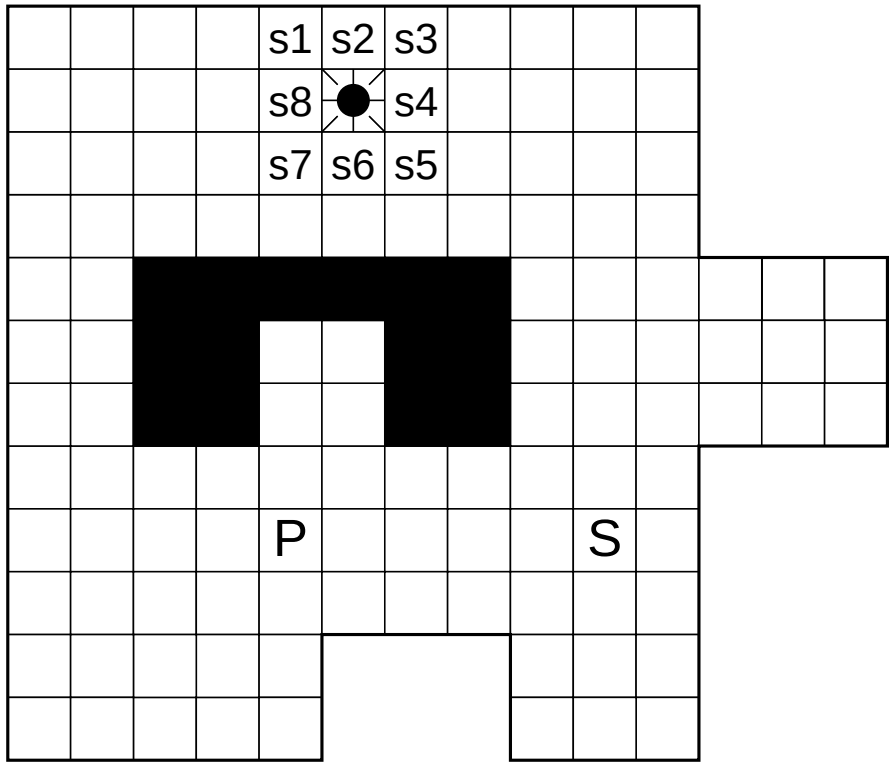
Aktionen:

falls keines der 4 Merkmale Wert 1 hat, führe `north` durch

sonst:

$$\text{If } x_1 = 1 \text{ and } x_2 = 0 \text{ then } \texttt{east}$$
$$\text{If } x_2 = 1 \text{ and } x_3 = 0 \text{ then } \texttt{south}$$
$$\text{If } x_3 = 1 \text{ and } x_4 = 0 \text{ then } \texttt{west}$$
$$\text{If } x_4 = 1 \text{ and } x_1 = 0 \text{ then } \texttt{north}$$

Beispiel: Wandverfolgung (IV)



Roboter, der an Position P startet, bewegt sich entgegen Uhrzeigersinn am Objekt entlang

Roboter, der an Position S startet, bewegt sich im Uhrzeigersinn an der äußeren Begrenzung entlang

Auswertung der Sensoreingaben

für beiden Phasen *perception processing* und *action computation* werden oft Boolesche Algebren verwendet

so gilt:

$$x_4 = s_1 \vee s_8 \text{ und go north}$$
$$\Leftrightarrow$$
$$(\bar{x_1} \wedge \bar{x_2} \wedge \bar{x_3} \wedge \bar{x_4}) \vee (x_4 \wedge \bar{x_1}) = 1$$

Auswertung der Sensoreingaben

Geeignete Repräsentationsform für Aktionen sind Regelsysteme der Form $c_j \rightarrow a_j$, wobei

c_j der Bedingungsteil und

a_j der Aktionsteil sind

In unserem Beispiel erhält man folgende Regeln:

$$\begin{aligned} x_4 \wedge \bar{x_1} &\rightarrow \text{go north} \\ x_1 \wedge \bar{x_2} &\rightarrow \text{go east} \\ x_2 \wedge \bar{x_3} &\rightarrow \text{go south} \\ x_3 \wedge \bar{x_4} &\rightarrow \text{go west} \\ 1 &\rightarrow \text{go north} \end{aligned}$$

Regelsysteme und Boolesche Algebren kann man gut anhand von Netzwerken implementieren

Wandverfolgung

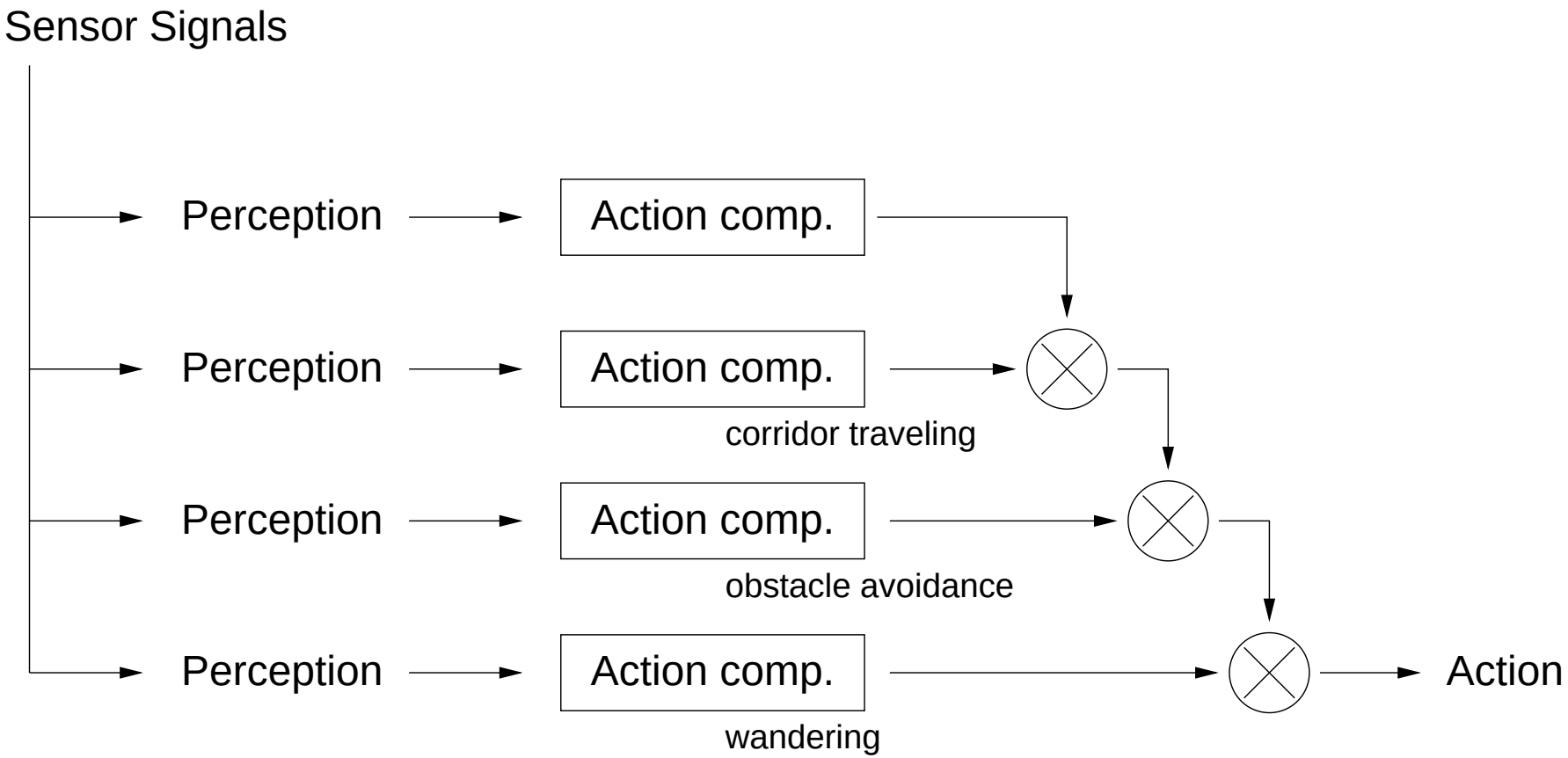
Alternativer Ansatz zu S-R-Agenten: Einführung von Subsumptions-Modulen

jedes Modul enthält Sensorinformationen direkt von Umwelt

sind spezifizierten Voraussetzungen des Moduls erfüllt, wird Programm ausgeführt

„höhere“ Module subsumieren „tiefere“, d.h. falls Voraussetzung eines höheren Moduls erfüllt, wird tieferes Modul durch höheres ersetzt

Wandverfolgung



Beispiel aus unserer Forschung: Sicherheitskritische Systeme

Fehlfunktion kann zu schweren Unfällen, Umwelt- oder physischen Schäden oder Todesopfern führen und können für gewöhnlich nicht korrigiert werden

Airbag-Zündung
nur bei schweren Crashs

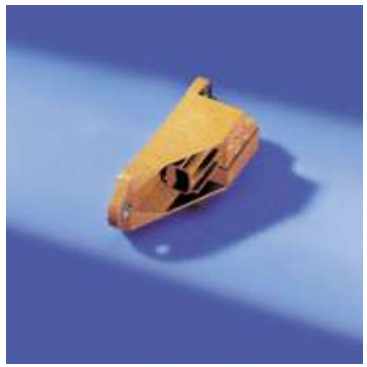
Crash-Sensorik

Airbag-steuergerät

Sensor-positionen

Insassen-klassifikations-

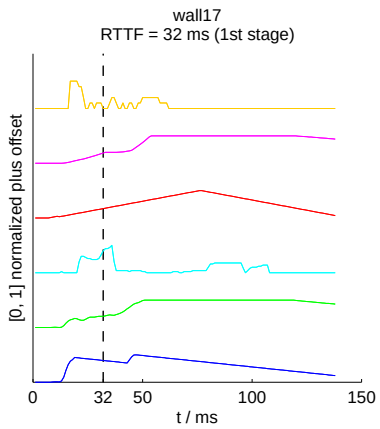
“Early Crash” Sensor

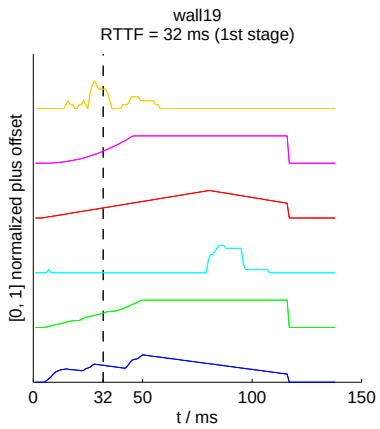


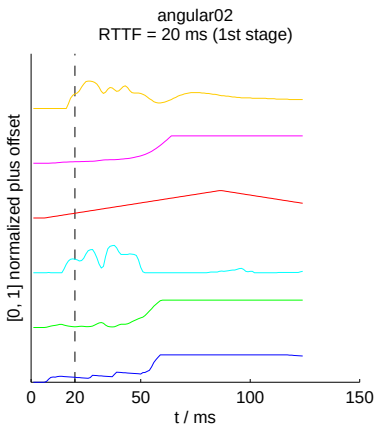
Beschleunigungs-sensor

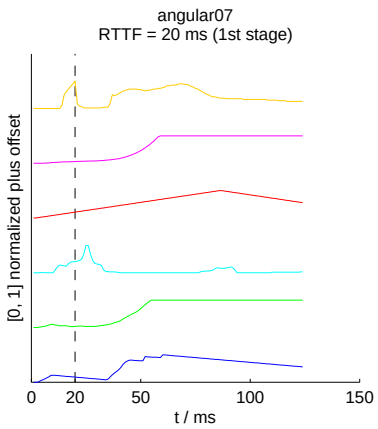
Drucksensor

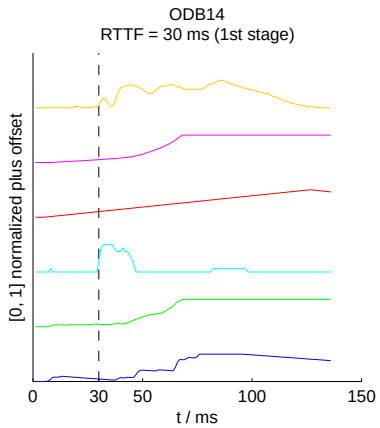
Beispielhafte Crash-Signale

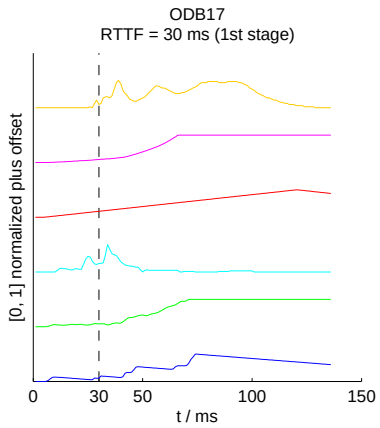


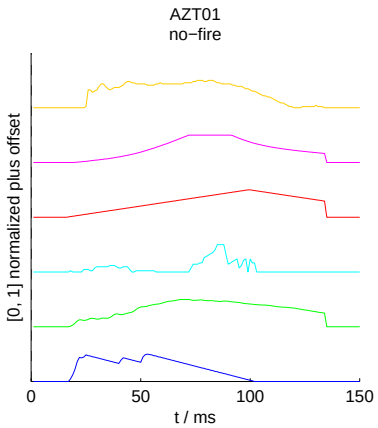


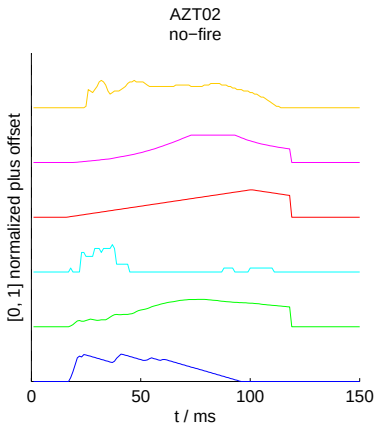












Stand der Technik

Verwendung von regelbasierten Agenten **(manuell erstellt)**

Zwei-Klassen-Problem: positive (Crash, krank) und negative (kein Crash, gesund)

Beispiele (Fahrten, Patienten)

jedes Beispiel: beschrieben durch numerischen Attributwerten

Reale Regelbasis: Erkennen von Wand-Crashes

```
 1: (X01 >= 221) & (X02 >= 164) => deploy airbag
 2: (X01 >= 215) & (X03 >= 22) => deploy airbag
 3: (X04 >= 225) & (X02 >= 188) => deploy airbag
 4: (X01 >= 177) & (X02 >= 248) => deploy airbag
 5: (X04 >= 176) & (X02 >= 236) => deploy airbag
 6: (X04 >= 171) & (X02 >= 231) & (X05 >= 74) & (X06 >= 14) => deploy airbag
 7: (X04 >= 164) & (X07 >= 3) & (X06 >= 9) => deploy airbag
 8: (X02 >= 224) & (X03 >= 26) => deploy airbag
 9: (X01 >= 149) & (X08 >= 106) => deploy airbag
10: (X01 >= 144) & (X03 >= 32) => deploy airbag
11: (X04 >= 150) & (X07 >= 4) => deploy airbag
12: (X07 >= 2) & (X02 >= 255) => deploy airbag
13: (X07 >= 5) & (X09 >= 131) => deploy airbag
14: (X010 >= 255) & (X07 >= 3) & (X05 >= 231) => deploy airbag
15: (X010 >= 255) & (X07 >= 3) & (X011 >= 77) & (X06 >= 10) => deploy airbag
16: (X07 >= 3) & (X06 >= 24) & (X09 >= 134) => deploy airbag
17: (X010 >= 255) & (X07 >= 2) & (X02 >= 50) & (X03 >= 22) => deploy airbag
18: (X010 >= 255) & (X07 >= 2) & (X02 >= 188) & (X03 >= 11) => deploy airbag
19: (X010 >= 255) & (X07 >= 2) & (X02 >= 188) & (X05 >= 255) & (X06 >= 26) => deploy airbag
20: (X07 >= 2) & (X02 >= 179) & (X03 >= 14) => deploy airbag
21: (X07 >= 2) & (X02 >= 176) & (X08 >= 90) => deploy airbag
22: (X010 >= 255) & (X07 >= 2) & (X01 >= 128) & (X08 >= 93) & (X06 >= 15) => deploy airbag
23: (X010 >= 255) & (X07 >= 1) & (X01 >= 131) & (X08 >= 103) & (X06 >= 23) => deploy airbag
24: (X07 >= 2) & (X01 >= 133) & (X08 >= 137) => deploy airbag
25: (X01 >= 131) & (X08 >= 105) & (X03 >= 24) => deploy airbag
26: (X010 >= 226) & (X07 >= 3) & (X012 >= 13) & (X06 >= 15) & (X03 >= 3) => deploy airbag
27: (X07 >= 2) & (X01 >= 115) & (X06 >= 52) & (X03 >= 19) => deploy airbag
28: (X010 >= 138) & (X07 >= 3) & (X01 >= 94) & (X03 >= 9) & (X09 >= 113) => deploy airbag
```

Literatur zur Lehrveranstaltung

S. Russell, P. Norvig: Künstliche Intelligenz: Ein moderner Ansatz, 3. Auflage, Pearson- Verlag, 2012

G. Görz, J. Schneeberger, U. Schmidt, Handbuch der Künstlichen Intelligenz, 5. Auflage, Oldenbourg-Verlag, 2014

C. Beierle und G. Kern-Isberner Methoden wissensbasierter Systeme, 5., verb. Aufl., Springer-Vieweg Verlag, 2014

C. Kruse, C. Borgelt, C. Braune, F. Klawonn, C. Moewes, M. Steinbrecher. Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze. 2. Aufl., Springer-Vieweg-Verlag, 2015