Institut für Informatik

Vorstellung der Lehre

SoSe 2023

Ablauf der Vorstellung

13:00 Begrüßung
13:10 AG Schömer
13:15 Weinzierl
13:20 AG Wand
13:25 AG Schuhknecht
13:30 AG Bouros
13:35 AG Kramer
13:45 Berger
13:50 Blumenstock

14:00 AG Schmidt
14:05 AG Brinkmann
14:10 AG Hildebrandt
14:20 Kuntze
14:30 Hoffmann/Oehl
14:40 v. Dziegielewski/Erbes

14:50 Spichtinger
14:55 AG Althaus
15:00 von Hippel
15:05 Fischer

Hiwis gesucht

- Praktikum Software Engineering (06.-17.03.)
- Studieneingangsprojekt (11.-14.04.)
- Lernwerkstatt (flexible Wochenstundenzahl)
- Einführung in die Programmierung
- Einführung in die Softwareentwicklung
- Technische Informatik
- Formale Sprachen und Berechenbarkeit
- Programmiersprachen
- Datenbanken
- Künstliche Intelligenz
- Einführung in die Computationale Logik
- IT-Sicherheit
- Kommunikationsnetze
- Kryptographie
- Machine Learning
- Strukturbasierte Bioinformatik
- High Performance Computing
- 3D Computer Vision
- Computergrafik II
- Praktikum mathematische Modellierung am Rechner

Bewerbung unter:
https://dethiwi.uni-mainz.de

Hinweis/Bitte aus dem Studienbüro

Bitte nicht nur für die Module, sondern auch für die Veranstaltungen (Vorlesungen etc.) anmelden.

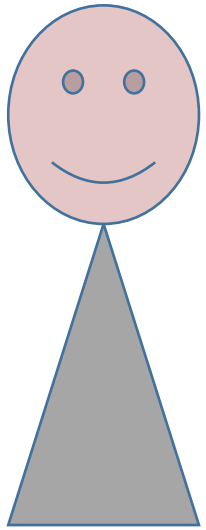
Ansonsten erhält man keinen Zugriff auf entsprechende LMS-Kurse und kann sich beispielsweise auch nicht für die entsprechende Klausur anmelden

Regelung Seminare

- Anmeldung freigeschaltet im Anschluss an die Vorstellung der Lehre
- Anmeldung ist offen für 2 Wochen
- Bei einer Überbelegung (> 15 Teilnehmer) werden nach Senatsrichtlinien Plätze vergeben.
  Kein first come, first serve!
- Anmeldung bedeutet implizite Prüfungsanmeldung.
  D. h. nach Vergabe der Themen ist keine Abmeldung mehr möglich

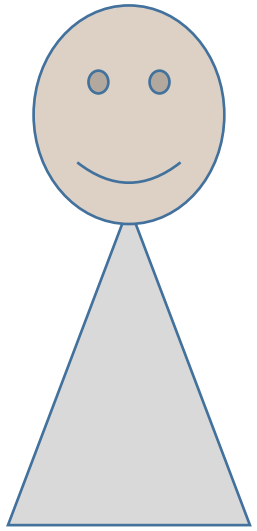
Vorlesung
Komplexitätstheorie II

Illustratives Beispiel



Alice

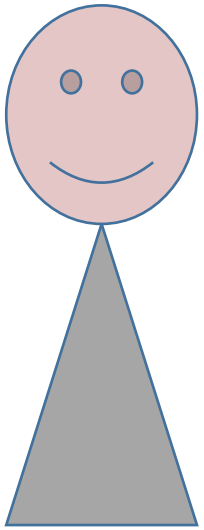
	3							
			1	9	5			
		8					6	
8				6				
4			8					1
				2				
	6					2	8	
			4	1	9			5
							7	



Bob

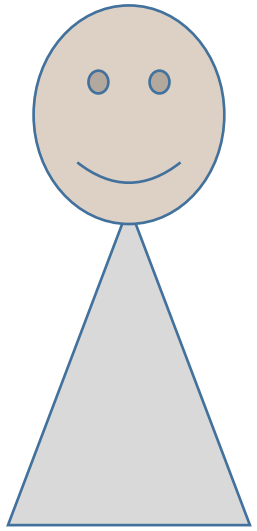
Alice möchte Bob davon überzeugen, dass das Sudoku lösbar ist.

Illustratives Beispiel



Alice

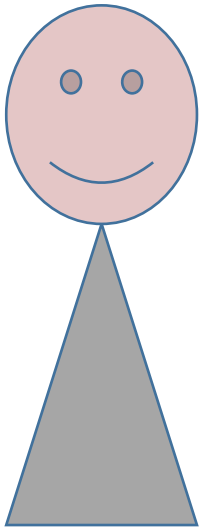
5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9



Bob

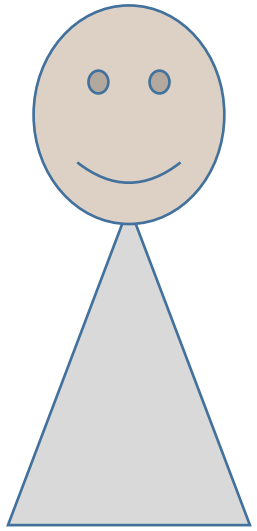
Sie füllt es aus und Bob verifiziert die Lösung (er prüft alle Zeilen, Spalten, Blöcke).

Illustratives Beispiel



Alice

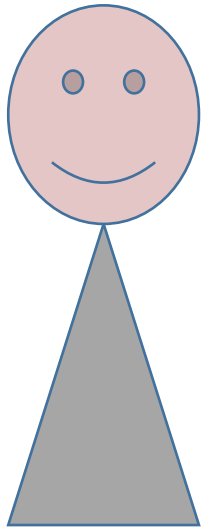
5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9



Bob

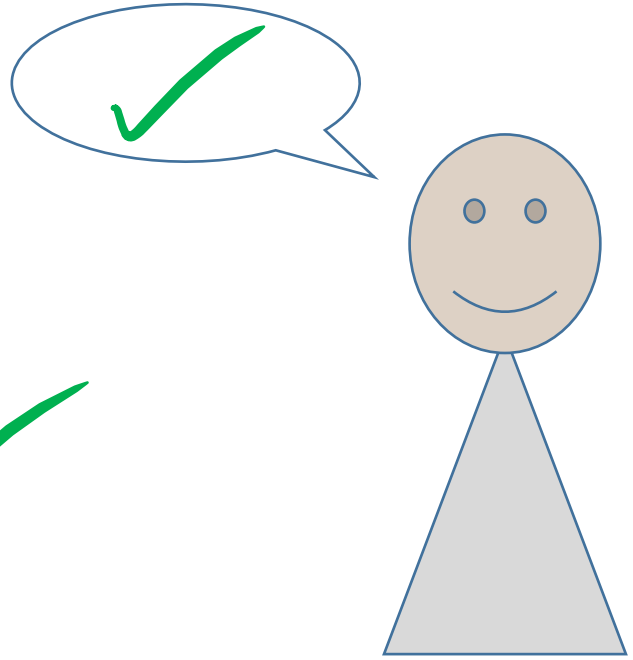
Was, wenn Bob faul ist? Er wählt zufällig nur 1/10 der Kästchen aus.
Er prüft, ob zwischen ihnen ein Fehler vorliegt, etwa 2 Fünfen in einer Zeile.

Illustratives Beispiel



Alice

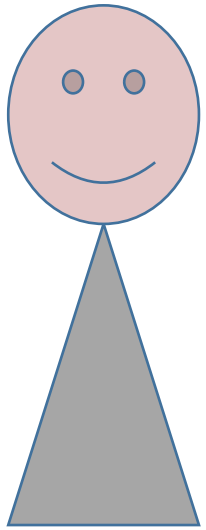
5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9



Bob

Eine korrekte Lösung akzeptiert er immer.

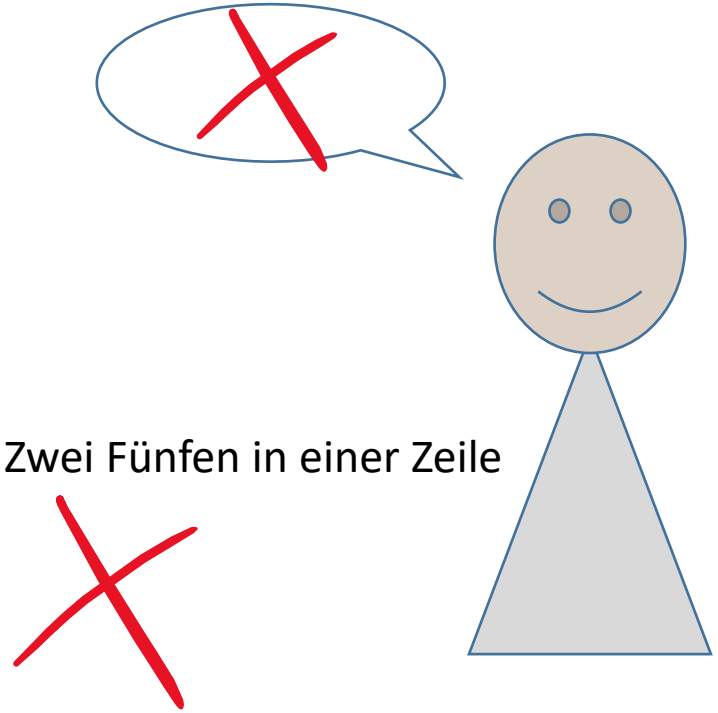
Illustratives Beispiel



Alice

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	5	2	4	8	5	6
9	6	1	9	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

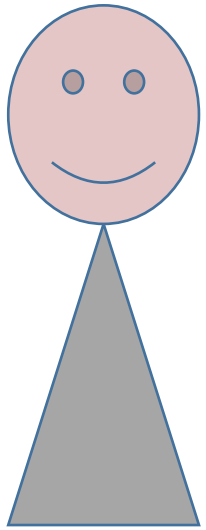
← Zwei Fünfen in einer Zeile



Bob

Manche falsche Lösungen wird er als falsch erkennen.

Illustratives Beispiel



Alice

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	1	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	5	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

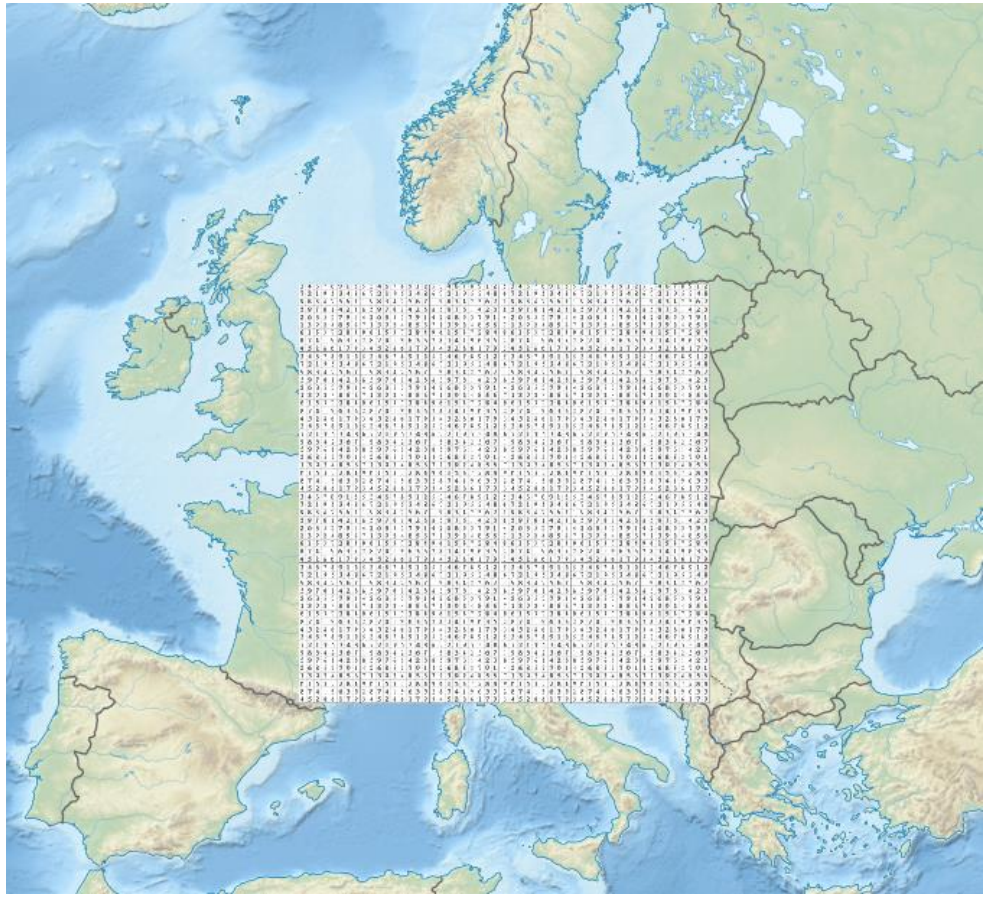
← Zwei Einsen in einer Zeile

← Zwei Fünfen in einer Zeile

Bob

Manche falsche Lösungen akzeptiert er als richtig.
Wie hoch ist die Wahrscheinlichkeit, dass das passiert? 1 %, 10 %, 90 %, 99 %?

Laaaaangweilig…wir machen es am besten gleich *richtig* schwierig



Karte von Benutzer TUBS,
https://commons.wikimedia.org/wiki/File:Germany_in_Europe_(relief).svg

Illustratives Beispiel

- Sagen wir, das Sudoku ist 100.000.000 x 100.000.000 Kästchen groß

  (bei Kästchengröße 1 cm^2 etwa dreimal so groß wie Deutschland)
- Bob darf sich 30 Kästchen ansehen und zu deren zufälligen Bestimmung 1000-mal eine Münze werfen.
- Dann muss er jede korrekte Lösung akzeptieren und eine falsche mit 99,9 % Wahrscheinlichkeit verwerfen.
- Ist das möglich?

Illustratives Beispiel

<u>Satz (PCP Theorem).</u>

Informell: Das geht, wenn Alice die Sudoku-Lösung vorher in einem bestimmten *fehlerverstärkenden* Format aufschreibt, sodass jeder vorhandene Fehler „potenziert“ wird.

- Mit dem fehlerverstärkenden Format ist die Überprüfung auf Fehlerfreiheit für Bob sehr einfach: Enthält Alices Lösung mindestens einen Fehler, dann ist die umgeschriebene Lösung durchsetzt mit Fehlern
- Der Haken: Er muss einen Teil seiner stark begrenzten Ressourcen aufwenden, um die Einhaltung des Formats durch Alice zu überprüfen
- Erstaunliche Anwendung des PCP Theorems: Man kann damit z. B. zeigen, dass Vertex Cover keinen 1,16-Approximationsalgorithmus besitzt (wenn $P \neq NP$)

Vorlesung Komplexitätstheorie II

- (Empfohlene) Voraussetzungen: Komplexitätstheorie, Mathematikkenntnisse
- Dozent inkl. Übung: Markus Blumenstock
- Vorlesung montags, 10-12 Uhr in 04-426
- Übungstermin wird mit Teilnehmern festgelegt
- Tafelvorlesung mit getextem Skript in Overleaf
- Organisation in LMS
- Mündliche Prüfung
- Literatur
  - S. Arora, B. Barak. Computational Complexity – A Modern Approach. Cambridge University Press, 2009.
  - Auszüge aus weiteren Büchern, Paper

Vorlesung Komplexitätstheorie II

Themen:

- Zeithierarchiesatz (P != EXP)
- Ladner's Theorem ("NP-Intermediate Problems")
- Zero-Knowledge Proofs
- Polynomialzeithierarchie
- Schaltkreiskomplexität
- Randomisierte Komplexität
- (Schwaches) PCP Theorem und Approximationsschwere

Soweit die Zeit reicht, weitere Themen:

- Complexity within P
- Expander-Graphen
- Derandomisierung
- Zählkomplexität
- PAC-Learning und Boosting

Computergrafik II = Computeranimation

E. Schömer

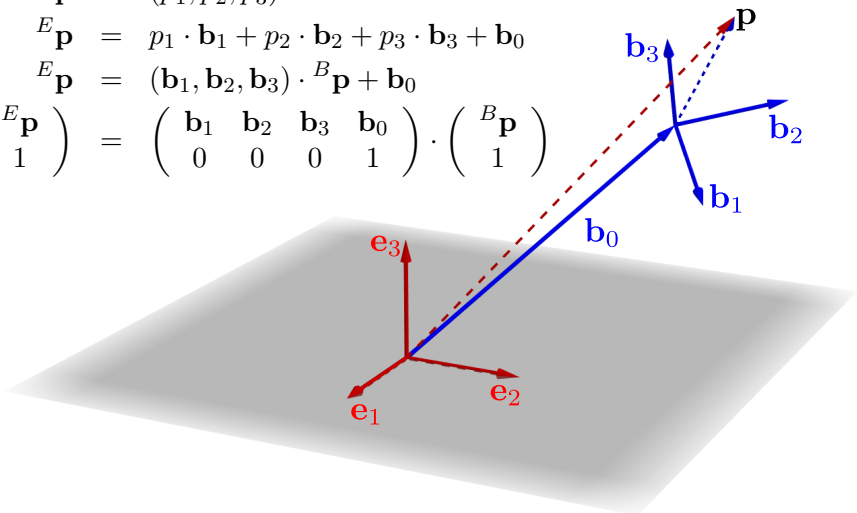
JOHANNES GUTENBERG UNIVERSITÄT MAINZ

INSTITUT FÜR INFORMATIK

Sommersemester 2023, Di 10-12 Uhr

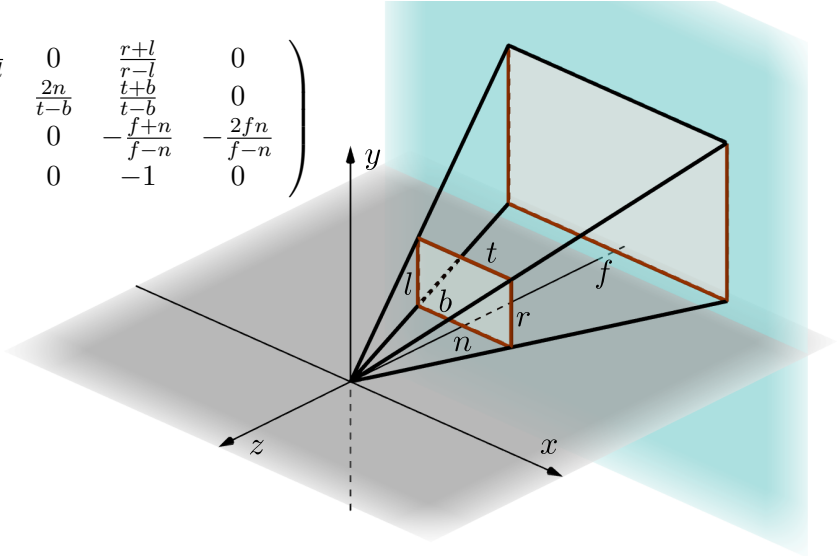
Koordinatentransformation

$$
\begin{aligned}
{}^B\mathbf{p} &= (p_1, p_2, p_3)^T \\
{}^E\mathbf{p} &= p_1 \cdot \mathbf{b}_1 + p_2 \cdot \mathbf{b}_2 + p_3 \cdot \mathbf{b}_3 + \mathbf{b}_0 \\
{}^E\mathbf{p} &= (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3) \cdot {}^B\mathbf{p} + \mathbf{b}_0 \\
\begin{pmatrix} {}^E\mathbf{p} \\ 1 \end{pmatrix} &= \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \mathbf{b}_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} {}^B\mathbf{p} \\ 1 \end{pmatrix}
\end{aligned}
$$

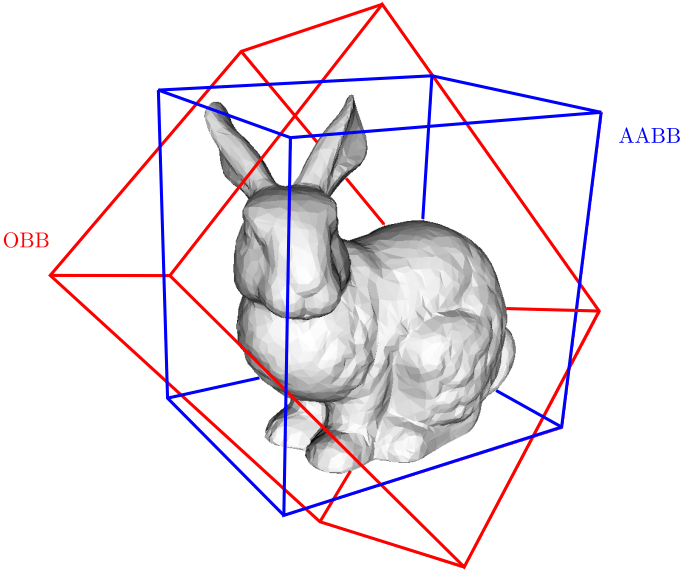


Perspektivische Projektion

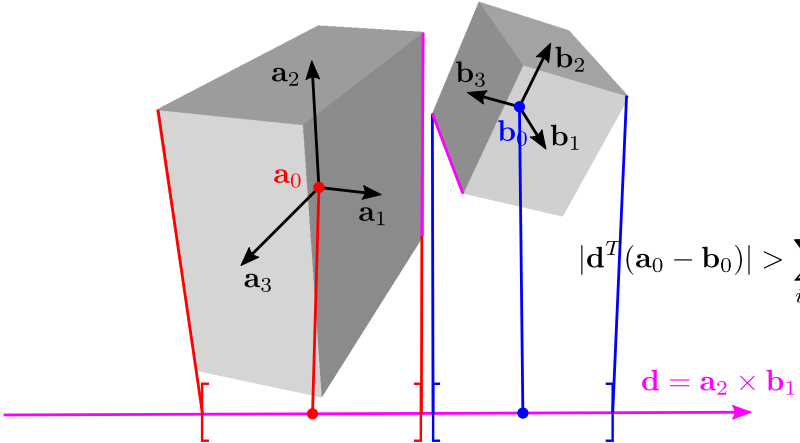
$$\mathbf{P} = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$



AABB vs OBB

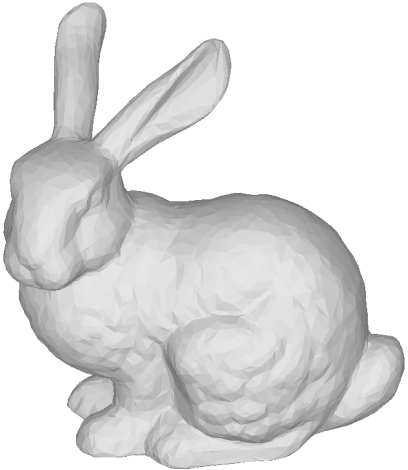


Kollisionstest für OBBs

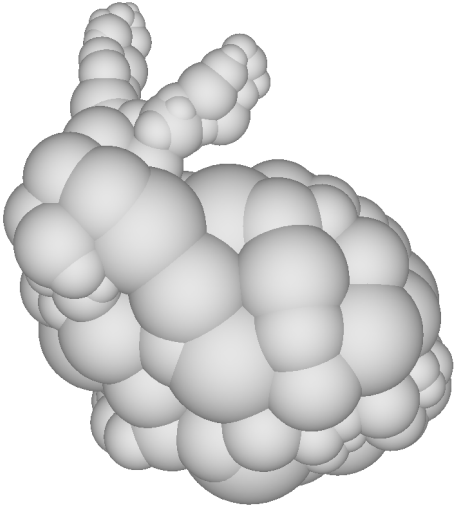


$$|\mathbf{d}^T(\mathbf{a}_0 - \mathbf{b}_0)| > \sum_{i=1}^{3} \left(|\mathbf{d}^T\mathbf{a}_i| + |\mathbf{d}^T\mathbf{b}_i|\right)$$

Hüllkörperhierarchien



Hüllkörperhierarchien

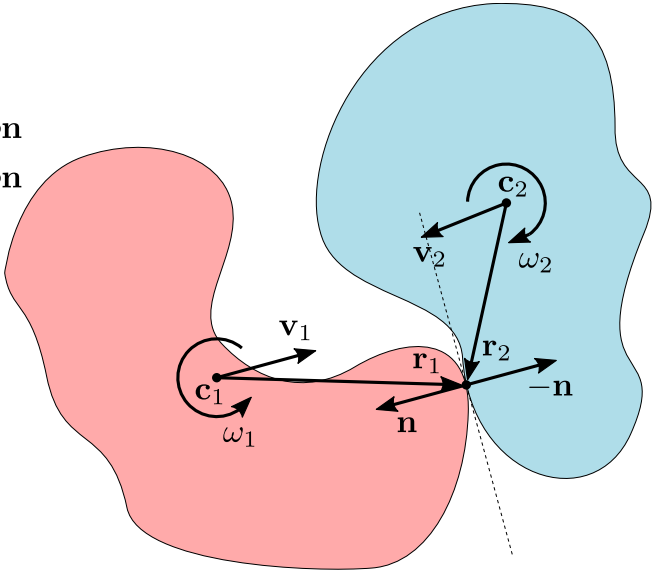


Kinematik und Dynamik starrer Körper

$\mathbf{c}$	$\mathbf{R}$	$\mathfrak{q}$
$\dot{\mathbf{c}} = \mathbf{v}$	$\dot{\mathbf{R}} = \omega^{\times} \cdot \mathbf{R}$	$\dot{\mathfrak{q}} = \frac{1}{2}\omega \cdot \mathfrak{q}$
m $\dot{m} = 0$	$\mathbf{I} = \mathbf{R} \cdot \mathbf{I}_0 \cdot \mathbf{R}^T$ $\dot{\mathbf{I}} = \omega^{\times} \cdot \mathbf{I} - \mathbf{I} \cdot \omega^{\times}$	$\mathbf{I} = \int_M \mathbf{r}^T \mathbf{r} \mathbf{E} - \mathbf{r}\mathbf{r}^T \, dm$
$\mathbf{p} = m\mathbf{v}$	$\mathbf{L} = \mathbf{I} \cdot \omega$	
$\dot{\mathbf{p}} = m\dot{\mathbf{v}}$	$\dot{\mathbf{L}} = \mathbf{I} \cdot \dot{\omega} + \omega^{\times} \cdot \mathbf{I} \cdot \omega$	
$\mathbf{F} = \dot{\mathbf{p}} = m\mathbf{a}$	$\mathbf{D} = \dot{\mathbf{L}}$	$\dot{\omega} = \mathbf{I}^{-1} \cdot (\mathbf{D} - \omega^{\times} \cdot \mathbf{I} \cdot \omega)$
$E_{kin} = \frac{1}{2} m \lvert\mathbf{v}\rvert^2$	$E_{rot} = \frac{1}{2}\omega^T \cdot \mathbf{I} \cdot \omega$	

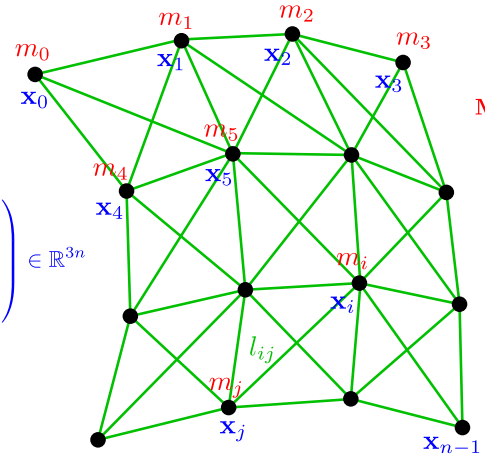
Elastischer/plastischer Stoß

$$
\begin{aligned}
m_1 \cdot \mathbf{v}_1' &= m_1 \cdot \mathbf{v}_1 + p\mathbf{n} \\
m_2 \cdot \mathbf{v}_2' &= m_2 \cdot \mathbf{v}_2 - p\mathbf{n} \\
\mathbf{I}_1 \cdot \omega_1' &= \mathbf{I}_1 \cdot \omega_1 + \mathbf{r}_1 \times p\mathbf{n} \\
\mathbf{I}_2 \cdot \omega_2' &= \mathbf{I}_2 \cdot \omega_2 - \mathbf{r}_2 \times p\mathbf{n}
\end{aligned}
$$



Feder-Masse-Systeme

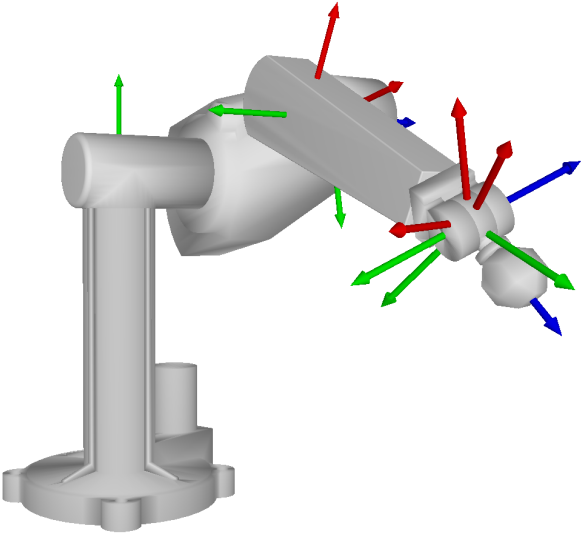
$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{n-1} \end{pmatrix} \in \mathbb{R}^{3n}$$



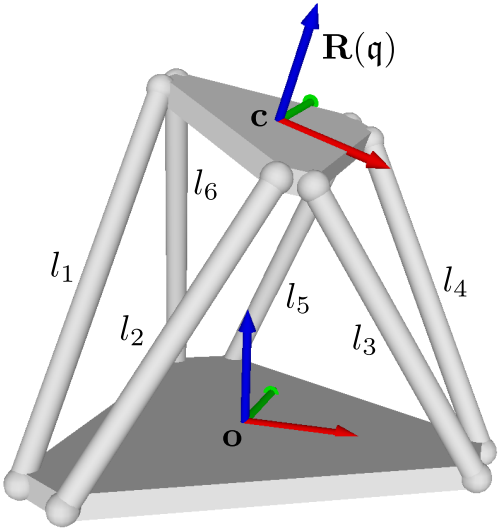
$$\mathbf{M} = \begin{pmatrix} m_0\mathbf{E} & \mathbf{0} & \dots & & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & & \\ \vdots & \ddots & m_i\mathbf{E} & \ddots & \vdots \\ & & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & & \dots & \mathbf{0} & m_{n-1}\mathbf{E} \end{pmatrix} \in \mathbb{R}^{3n\times 3n}$$

$$\begin{aligned} \mathcal{E} &= \{\{0,1\},\{0,4\},\{0,5\},\dots,\{i,j\},\dots\} \\ |\mathcal{E}| &= k \\ \mathbf{C} &= \begin{pmatrix} \vdots \\ |\mathbf{x}_i - \mathbf{x}_j| - l_{ij} \\ \vdots \end{pmatrix} \in \mathbb{R}^k \end{aligned}$$

Koordinatensysteme der Gelenke

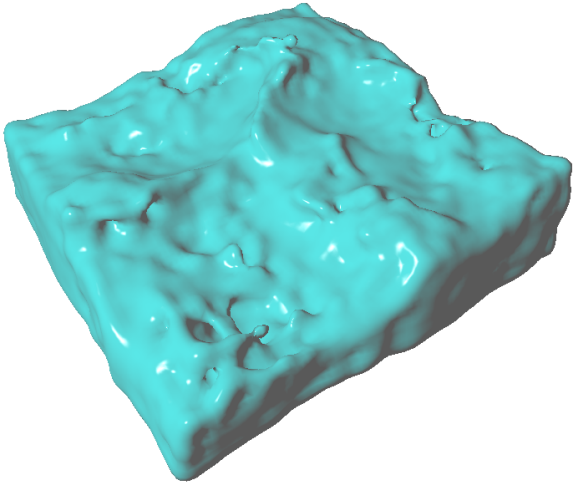


Stewart-Gough-Plattform



Smoothed Particle Hydrodynamics

Bachelorarbeit Julius Schellenberg



Kryptographie

Stefan Weinzierl

Institut für Physik, Universität Mainz

Sommersemester 2023

Motivation

Kryptographie gehört zum Alltag:

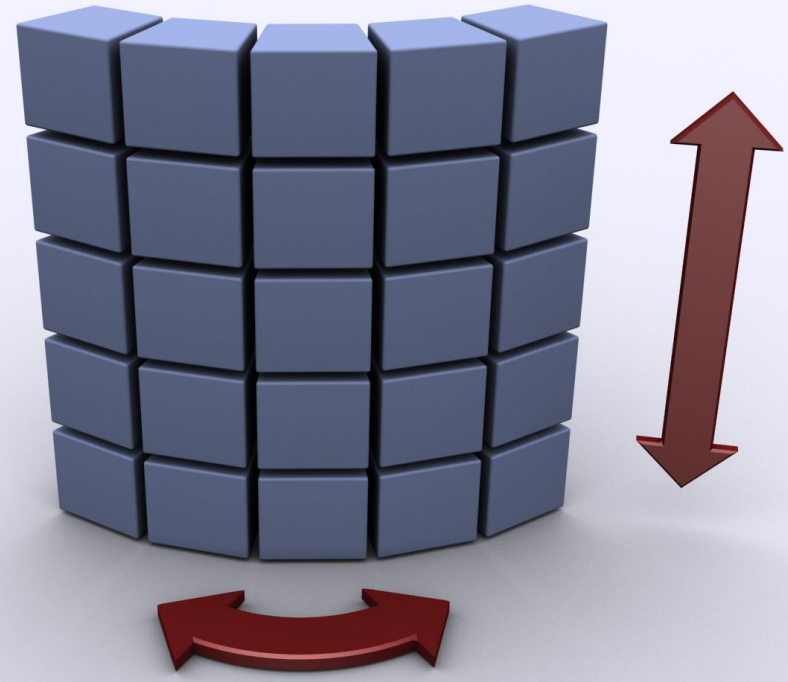
- Sichere Datenübertragung im Netz
- Elektronische Signatur
- Bitcoins, Blockchains, etc.
- Cyberangriffe mittels Verschlüsselungstrojaner

Inhalt

- Grundlagen:
  - Mathematische Grundlagen der Kryptographie
  - Definition von Sicherheit
- Praxis:
  - Private-Key-Verschlüsselung
  - Public-Key-Verschlüsselung
  - Digitale Signaturen
  - Zero-Knowledge-Verfahren
  - Mehr-Parteien-Berechnungen
- Ausblick:
  - Quantenalgorithmen
  - Post-Quantum-Verschlüsselung

Lehrangebot Visual Computing

Sommersemester 2023



JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Institut für Informatik

Veranstaltungen Visual Computing

Veranstaltungen

- Einführung in die Programmierung
- Modellierung II:
  Statistische Datenmodellierung
- Seminar Visual Computing
  - Aktuelle Forschung in Machine Learning & Computer Vision
  - 1 Woche im Block in den Semesterferien
  - Passend zu
    - Modellierung 1+2
    - 2D/3D Computer Vision

Vorlesungsreihe Modellierung

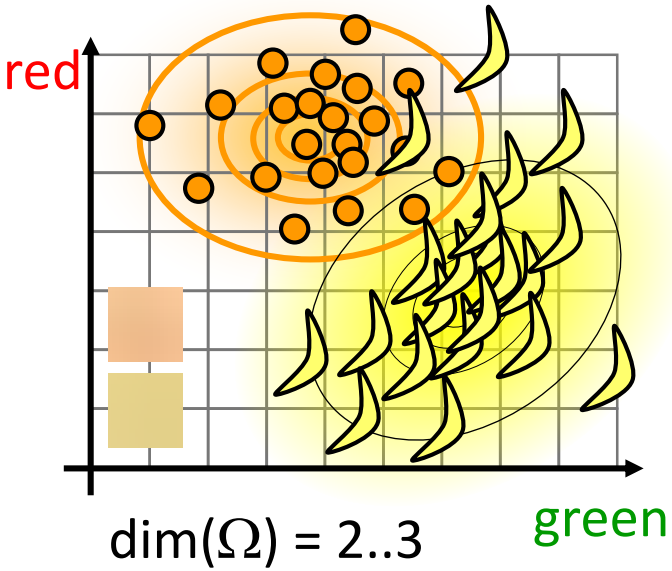
Reihe Modellierung

- Modellierung 1: Lineare Modellierung
  - Nächstes mal (vorauss.) im WS 2023/24
- Modellierung 2: Statistische Modellierung & Deep Learning
  - Neu seit SoSem 2021
  - Voraussetzung: Modellierung 1 empfohlen (oder ML/CV)
- Spezialveranstaltungen
  - Advanced Rendering – zuletzt WS 21/22
  - Statistical Geometry Processing – zuletzt WS 22/23

Modellierung II

Statistische Modellierung

- Wiederholung Grundlagen
  - Nach Absprache
- Deep Learning
- Verständnis tiefer Netze
  - Informationstheorie
  - Kernels + Gauß'sche Prozesse
  - Ideen aus der statistischen Physik
  - Aktuelle Forschungsfragen und Ergebnisse



Weitere Infos

Weitere Infos

- Modellierung 2:
  luna.informatik.uni-mainz.de/mod2-23

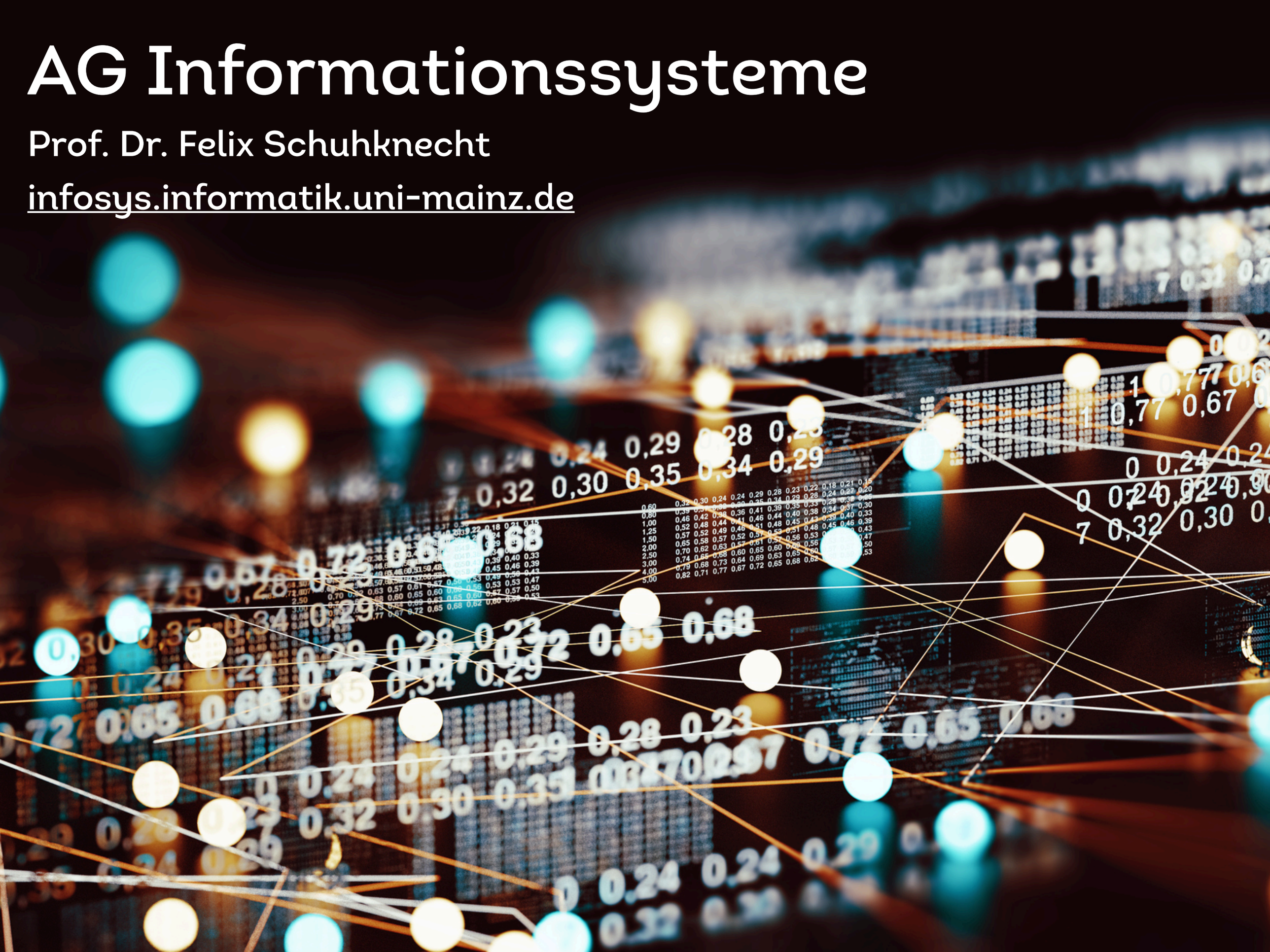
- Seminar:
  luna.informatik.uni-mainz.de/sem-prak-vc-23

(alles spätestens April 2023 online)

AG Informationssysteme

Prof. Dr. Felix Schuhknecht

infosys.informatik.uni-mainz.de



Einführung in die Softwareentwicklung

- Direkte Anknüpfung an "Einführung in die Programmierung"
- Entwurf und Umsetzung komplexer Softwaresysteme

- Schwerpunkte:
  - Zerlegung komplexer Projekte in einzelne Komponenten und Schnittstellen → modularisierter Code
  - Objektorientierte Programmierung
  - Statische Typisierung
  - Funktionale Programmierung
  - Nebenläufige Programmierung
  - Ausführung und Performance

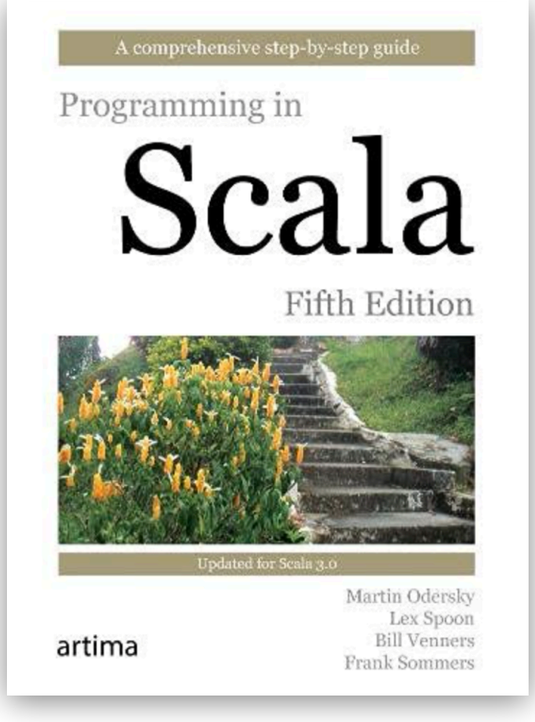
- Projektorientierte Struktur

Einführung in die Softwareentwicklung

- Wir erproben die Konzepte anhand der Programmiersprache Scala

- Warum Scala?
  - Kompakte Syntax
  - Sehr konsequente Objektorientierung
  - Funktionale Programmierung möglich (und erwünscht)
  - Statische Typisierung mit Typinferenz
  - Java Unterbau

- Literatur:
  Martin Odersky, Lex Spoon, Bill Venners:
  Programming in Scala, Fifth Edition



Einführung in die Softwareentwicklung

- Vorlesung + Übung:
  - Vorlesung: Montag 14-16 Uhr
  - Vorlesung/Übung auf Deutsch

- Erste Vorlesung:
  - 17.04.2023 um 14:15 Uhr im Audi Max

- Voraussetzung: Einführung in die Programmierung

- Übung:
  - Wöchentliche Übungsblätter mit praktischen Programmieraufgaben
  - Assistent: Justus Henneberg
  - Tutoren gesucht!

    (Keine Angst: Wir veranstalten einen "Crashkurs" zu Scala in den Semesterferien, falls Sie von einer anderen OO-Sprache kommen)

Datenbanken Seminar

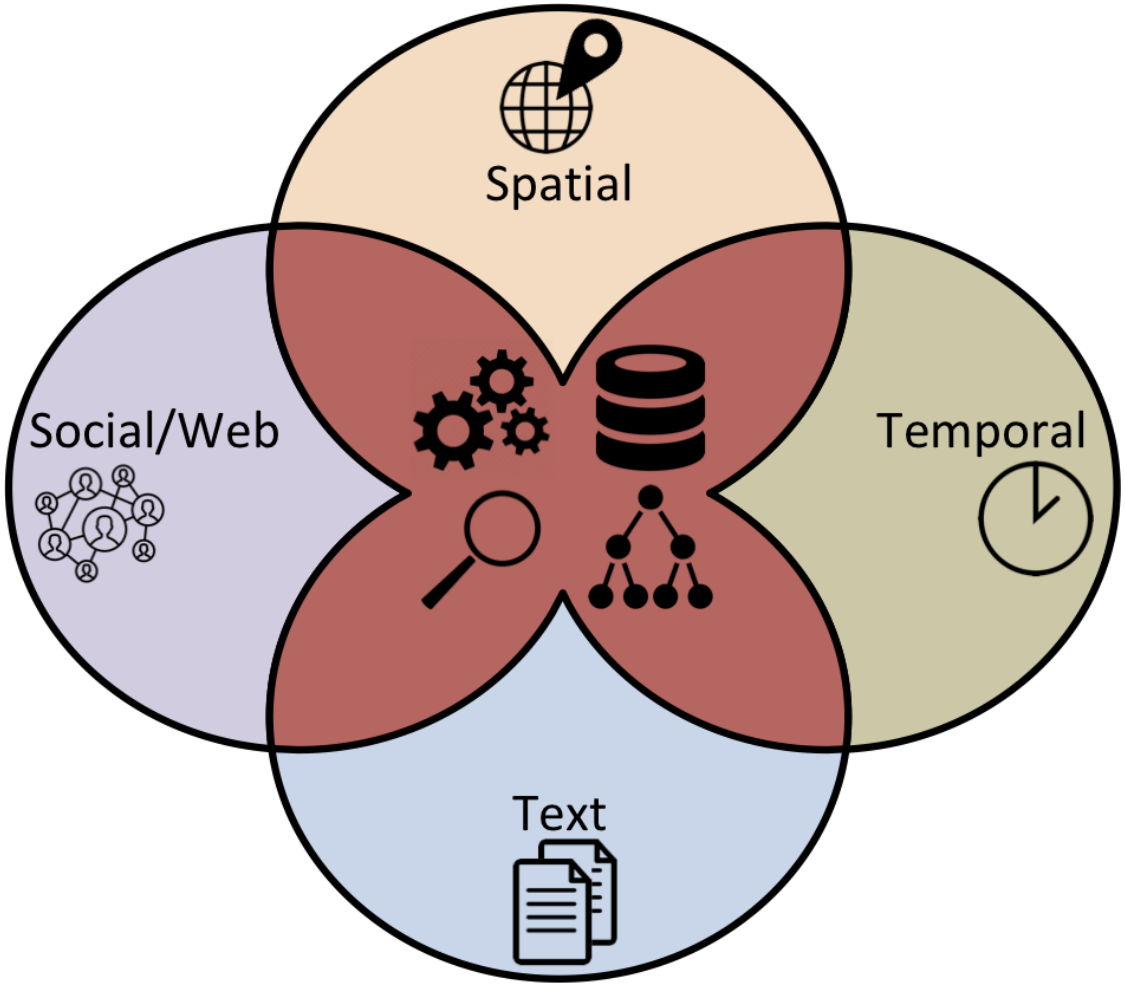
- Präsentation und Diskussion aktueller Forschungspapiere.
- Blockseminar, Präsentationen im Anschluss an die Vorlesungszeit.
- Themenausgabe und Bearbeitung während der Vorlesungszeit.

- Themengebiete:
  - Storage
  - Indizierung
  - Anfrageverarbeitung und -optimierung
  - Systemdesign

- Kickoff-Meeting:
  20.04.2023 um 16:15 Uhr zur Vorstellung der Themen.

Welcome to SoSe 2023

Panagiotis Bouros,
Jun.-Prof. Dr.
Head of the Data Management group



Datenbanken I

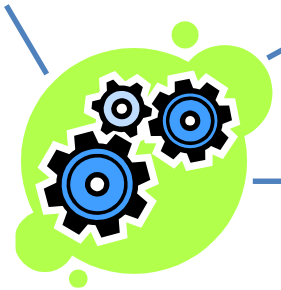
- Vorlesung
  - Tuesdays, 12:00 – 14:00, Rm 00-521 (N1)
- Übung
  - Deepen your understanding, gain practical experience
  - Weekly exercises (handins assignments) and quizzes
  - Live meeting and chatting
  - Appointments to be determined
- Unterrichtssprache
  - English & German (tutorials)
- Studiengang
  - B. Sc. / M. Ed.
- Fora
  - Moodle, https://lms.uni-mainz.de/moodle/course/view.php?id=94181
  - MS Teams

Datenbanken I

- Course objectives
  - Effective and efficient management of data
  - Databases as part of modern information systems
  - Learn how to model and develop effective database systems
  - SQL for working with databases
  - Applications from various areas, including the web
  - Apply in your future work

Development: analysis, conceptual model

Databases: entity-relationship model, relational model, SQL, …



Applications: standalone, internet and the web, …

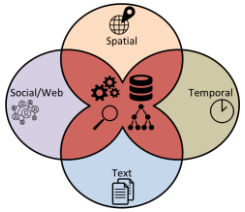
Datenbanken Praktikum

- Termin
  - TBD
  - Two or three weeks before the start of WiSe 2023/24
- Inhalte
  - Hands-on experience, practical aspects
  - Building a functional and practical UI on top of a DBMS
  - Web programming
- Unterrichtssprache
  - English/German
- Studiengang
  - B. Sc. / M. Ed.
- Fora
  - Moodle, https://lms.uni-mainz.de/moodle/course/view.php?id=91769
  - MS Teams

Datenbanken II - Seminar

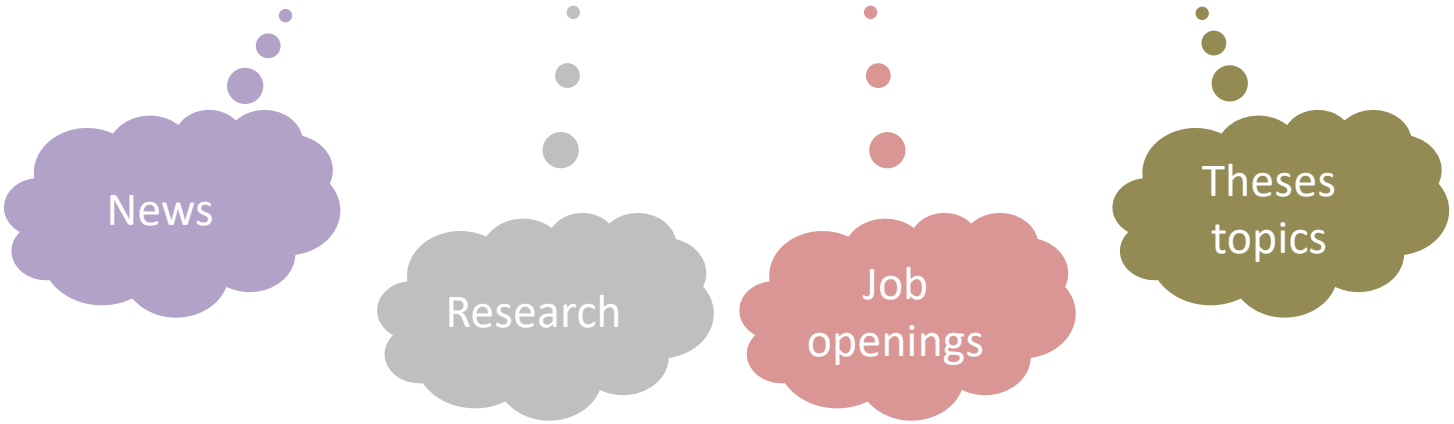
- Termin
  - TBD
  - Meetings and chats either physically held or on MS Teams
- Inhalte
  - Advanced topics on data management
  - How to critically read and review research literature
  - How to write a report on research literature
  - How to present research literature
- Unterrichtssprache
  - English
- Studiengang
  - B. Sc. / M. Ed.
- Fora
  - Moodle, https://lms.uni-mainz.de/moodle/course/view.php?id=93080
  - MS Teams

AG Data Management



For more information about the group, visit

https://datamanagement.cs.uni-mainz.de/



Machine Learning

Vorlesung

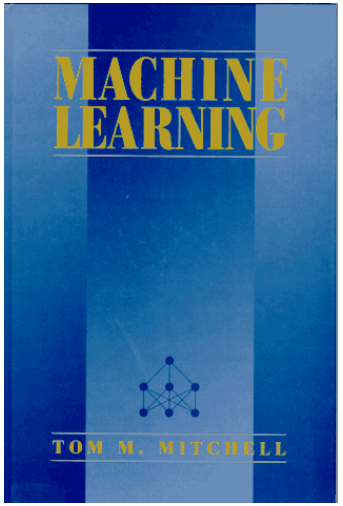
- Donnerstag 14 – 16 Uhr
- Material auf Englisch

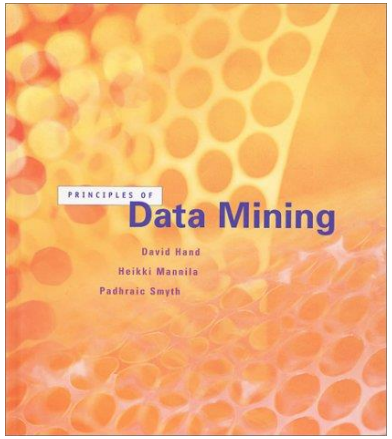
Übung

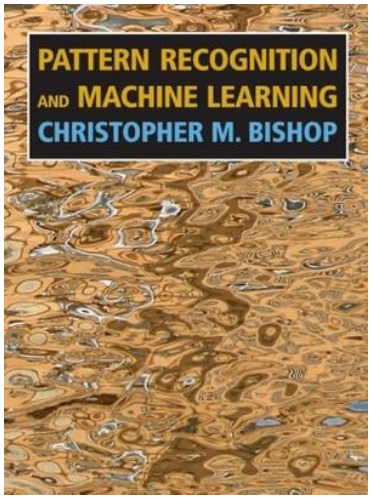
- Zeit: TBA

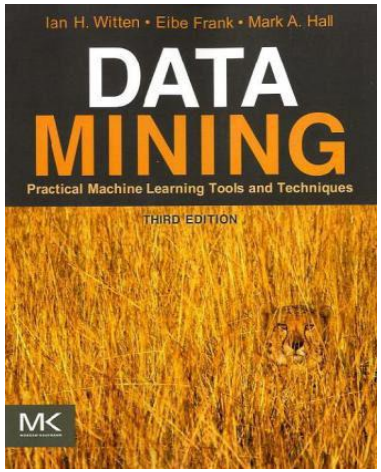
Prüfung

- mündlich oder schriftlich



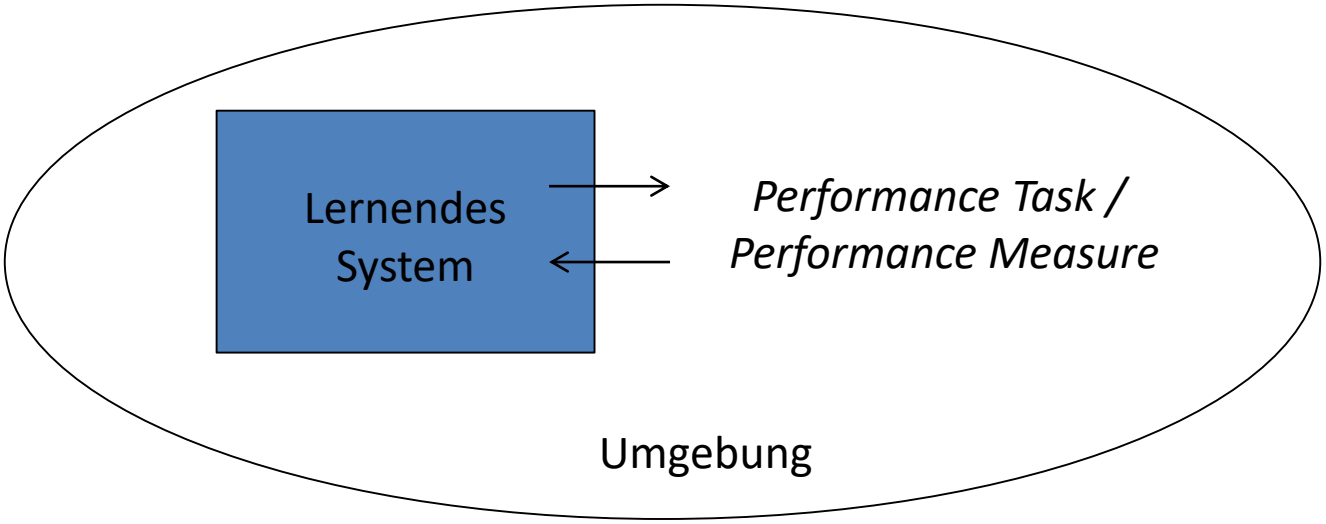






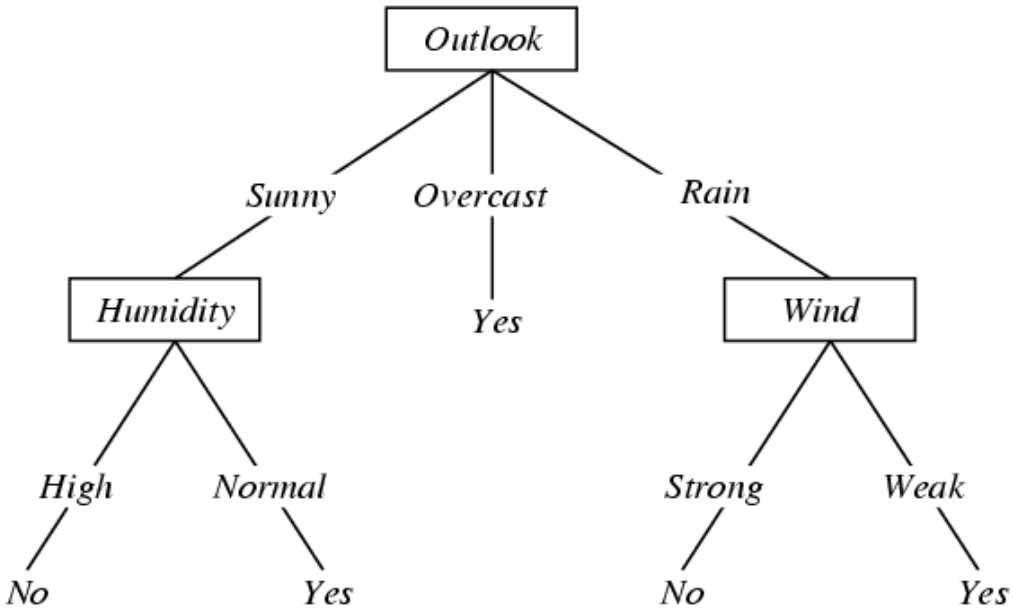
Machine Learning

- Lernen = sich durch **Erfahrung** bei einer **Aufgabe verbessern**
  - Aufgabe (*performance task*) **T**
  - hinsichtlich eines Performancemaßes (*performance measure*) **P**
  - beruhend auf Erfahrung **E**.



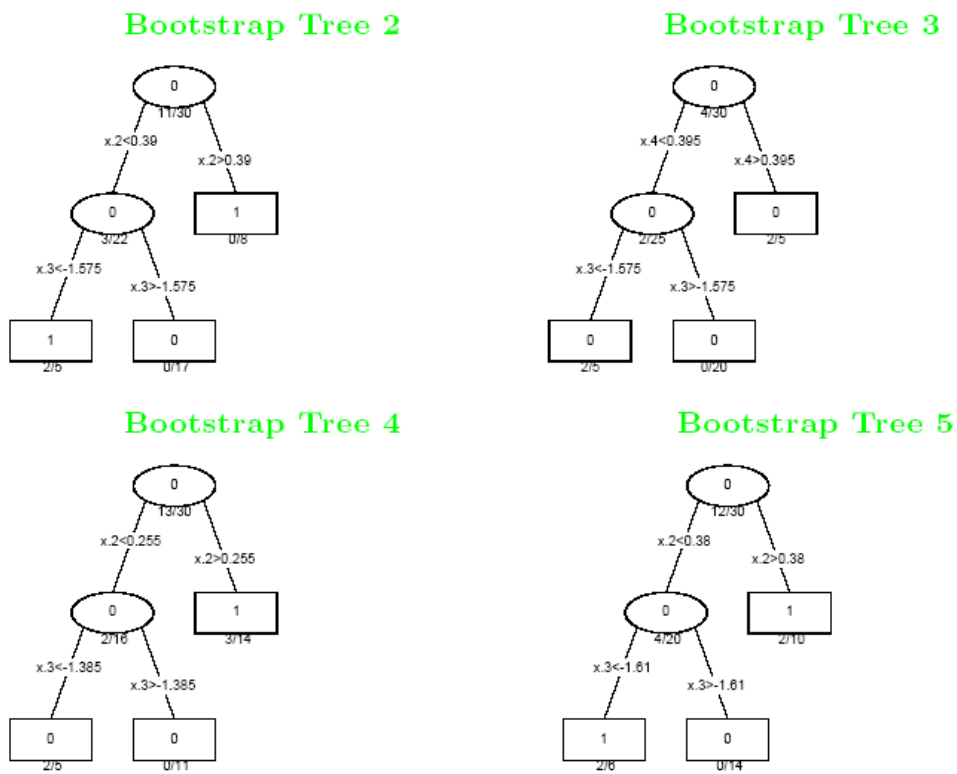
Machine Learning Course Overview

- Decision trees
- Ensemble methods
- Bayesian learning
- Linear models
- Neural networks
- Instance-based learning
- SVMs and kernels
- Deep learning



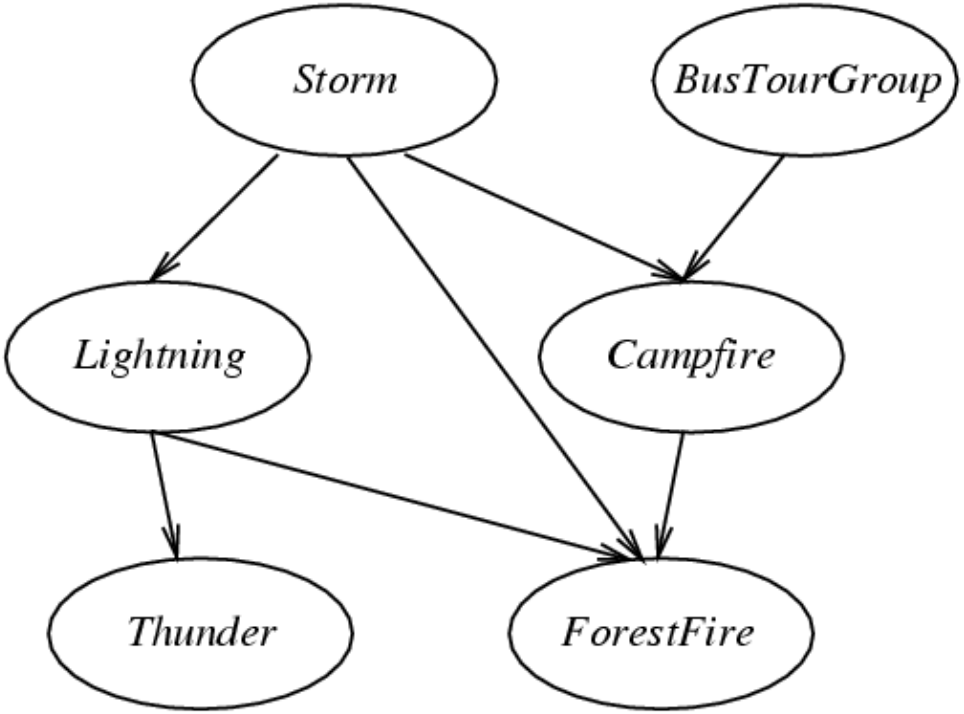
Machine Learning Course Overview

- Decision trees
- Ensemble methods
- Bayesian learning
- Linear models
- Neural networks
- Instance-based learning
- SVMs and kernels
- Deep learning



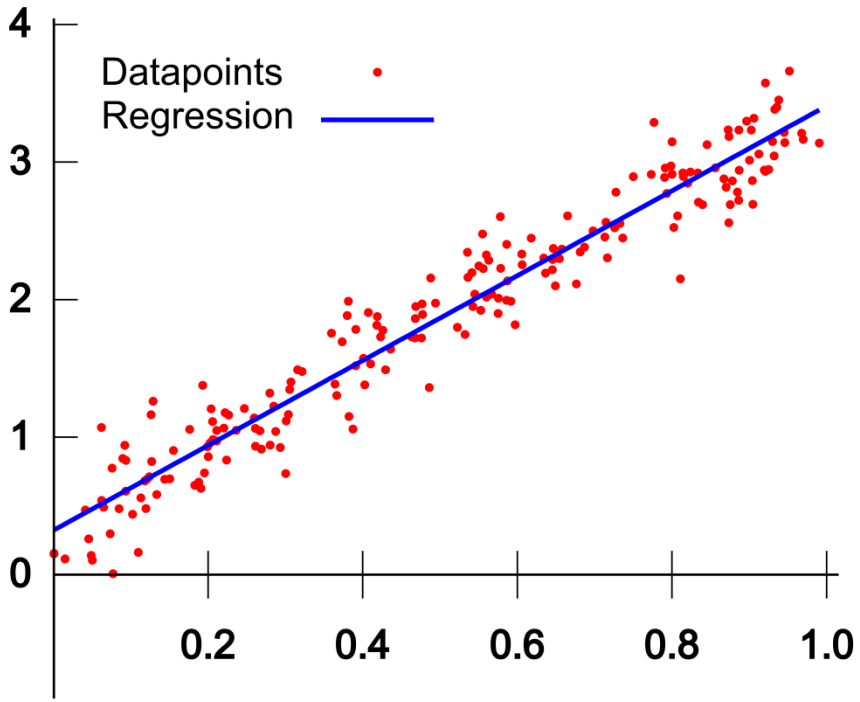
Machine Learning Course Overview

- Decision trees
- Ensemble methods
- Bayesian learning
- Linear models
- Neural networks
- Instance-based learning
- SVMs and kernels
- Deep learning



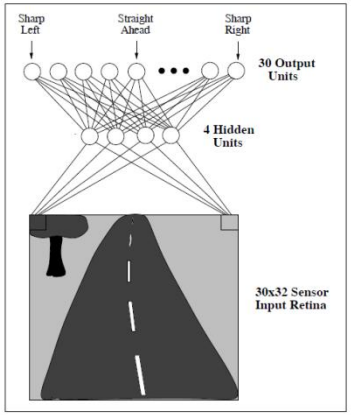
Machine Learning Course Overview

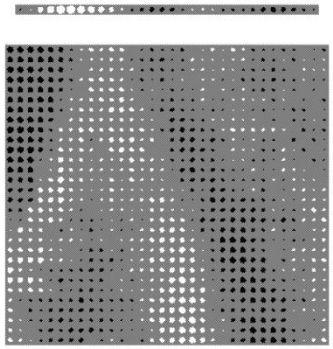
- Decision trees
- Ensemble methods
- Bayesian learning
- Linear models
- Neural networks
- Instance-based learning
- SVMs and kernels
- Deep learning



Machine Learning Course Overview

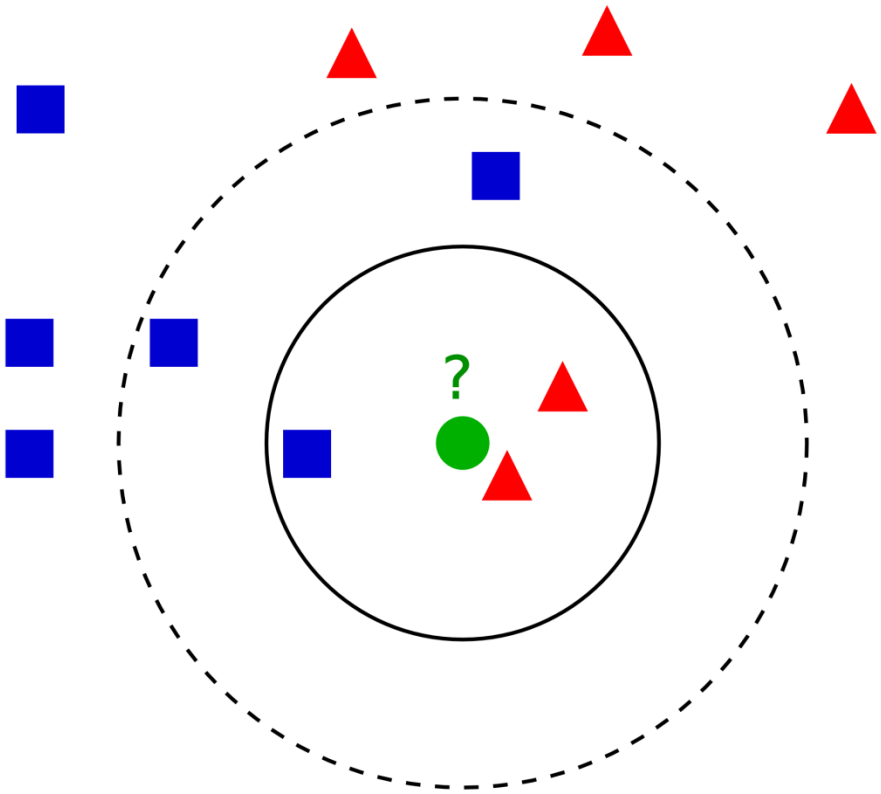
- Decision trees
- Ensemble methods
- Bayesian learning
- Linear models
- Neural networks
- Instance-based learning
- SVMs and kernels
- Deep learning





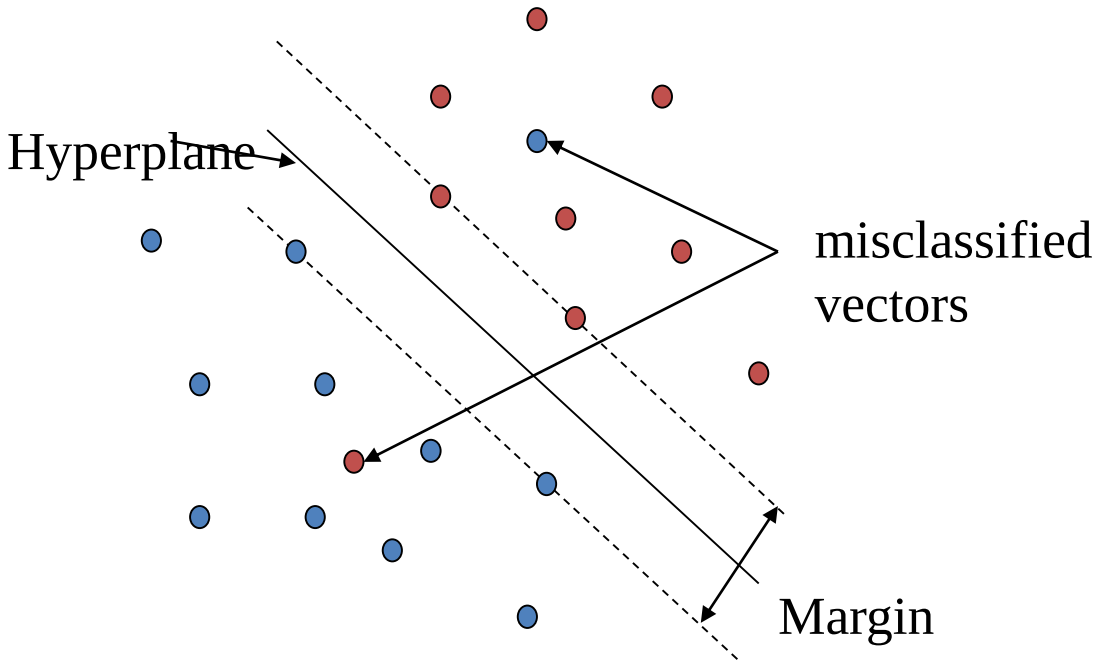
Machine Learning Course Overview

- Decision trees
- Ensemble methods
- Bayesian learning
- Linear models
- Neural networks
- Instance-based learning
- SVMs and kernels
- Deep learning



Machine Learning Course Overview

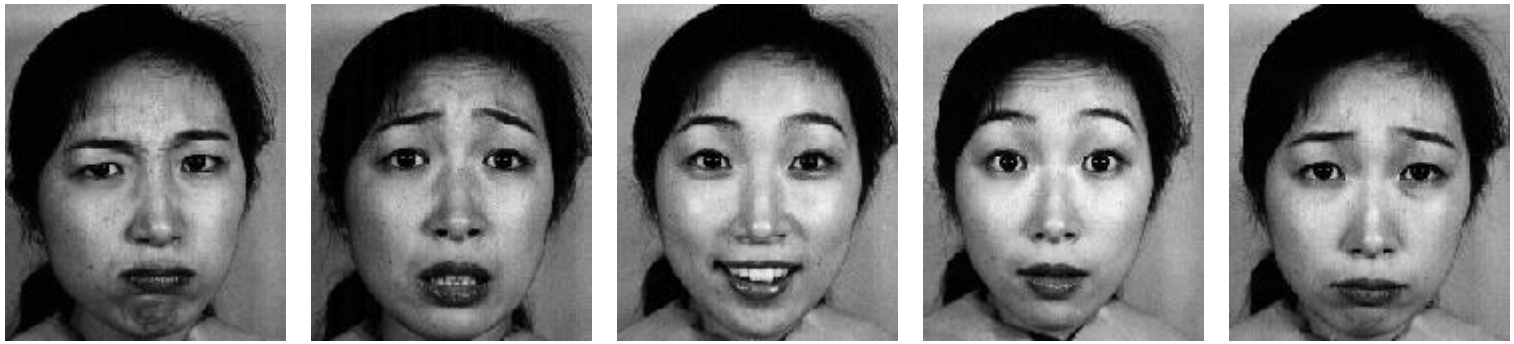
- Decision trees
- Ensemble methods
- Bayesian learning
- Linear models
- Neural networks
- Instance-based learning
- SVMs and kernels
- Deep learning



Data Mining Praktikum

Ablauf

- 3er Gruppen
- ein Datensatz pro Gruppe
- Daten analysieren
- wöchentliche Treffen



Einführung in die Computationale Logik

Vorlesung

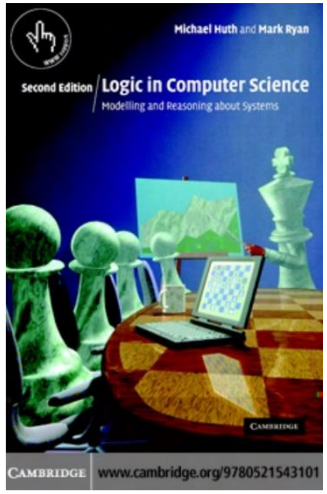
- Montag 16 – 18 Uhr
- Material auf Deutsch und Englisch

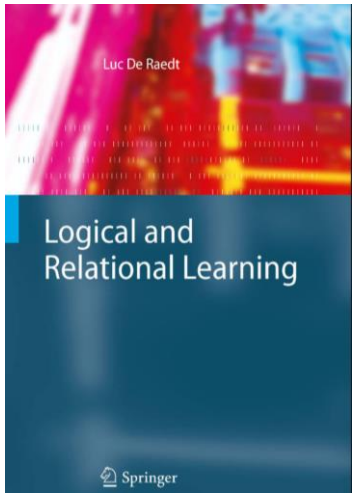
Übung

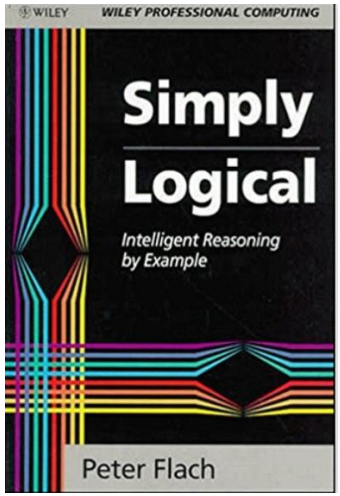
- Zeit: TBA

Prüfung

- mündlich oder schriftlich







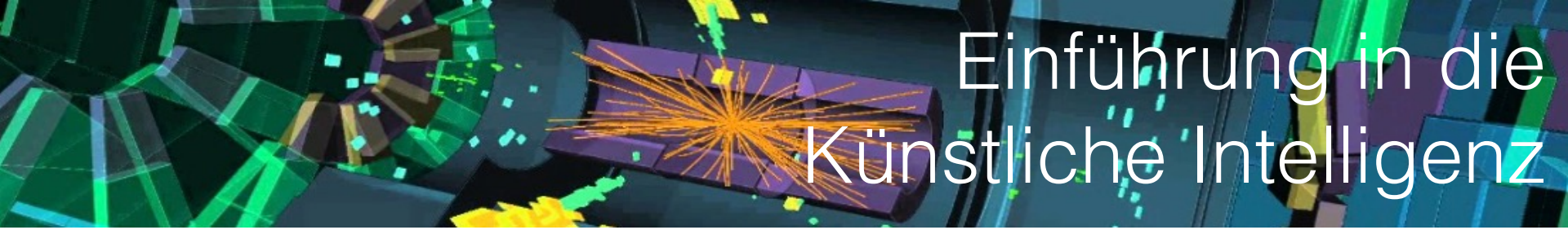
Einführung in die Computationale Logik

- Syntax, Semantik, Beweistheorie, Metatheorie von:
  - Aussagenlogik, Prädikatenlogik erster Stufe, Klausellogiken
  - Beschreibungslogiken (inkl.: Ontologien, OWL, …)
- Inkl.: Theorembeweisen, Resolution, Negation, Answer Set Programming
- Deduktion, Induktion, Abduktion
- Lernen in Logik (Lernen von Klauseln und Klauselmengen)
- Probabilistische Logiken
- Lernen in probabilistischen Logiken

Data Mining Seminar

Ablauf

- Thema aus dem Bereich Data Mining und Machine Learning
- ein Vortrag pro Woche
- nach jeder Präsentation Diskussion und Feedback
- Ausarbeitung
- Feedback zum Schreiben am Ende des Semesters



Einführung in die Künstliche Intelligenz

- Titel: Einführung in die künstliche Intelligenz

- Inhalt: Es werden grundlegende Methoden, Verfahren und Anwendungen der künstlichen Intelligenz behandelt. Im Mittelpunkt der Lehrveranstaltung stehen die Themen Suchen, Planen, Lernen und Schließen.
- Wie jedes Jahr gibt es einen KI-Wettbewerb für alle Studierende mit Urkunden für die bestplatzierten Teams
- Bemerkung: Dies ist **keine** spezielle Vorlesung zu künstlichen neuronalen Netzwerken bzw. Mustererkennung!

- Wer?
  - Prof. Dr. Matthias Schott

- Wann?
  - Dienstags 10:15-11:45 (Vorlesung)
  - Übung: to be defined

- Wo?
  - Raum 03 428

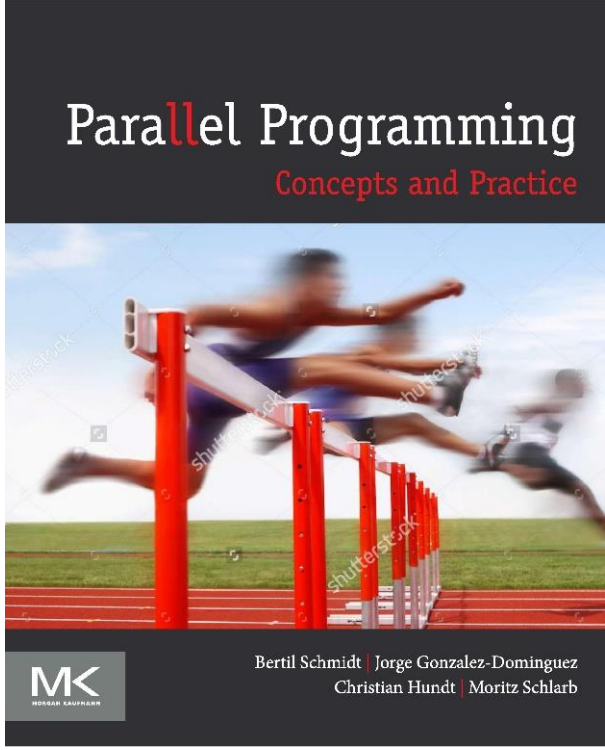
AG Schmidt: Lehrangebot: SoSe 23

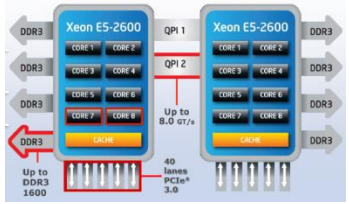
- **Vorlesungen**
  - Technische Informatik (*Grundvorlesung*)
  - High Performance Computing (HPC)
- **Praktikum**
  - Paralleles Programmieren mit CUDA (AccComp, WiSe 22/23)
  - HPC Praktikum (HPC, SoSe 23)
- **Seminar**
  - Paralleles Rechnen

Prof. Bertil Schmidt

High Performance Computing (HPC)

- Background
  - Parallel Hardware & Software
- Shared Memory Programming
  - OpenMP
  - C++ Multithreading
- Distributed Memory Programming
  - MPI
- Parallel Program Development
- Practical Parallel Programming Exercises





Multi-core CPUs → Manycores → Super-computers

Block-Praktikum: Paralleles Programmieren (CUDA)

- **Hinweis:** Veranstaltung im WiSe 22/23
- Paralleles Programmieren mit CUDA
  - **20.3. – 31.3.2023**
  - Voraussetzung: Erfolgreiche Teilnahme an **Accelerated Computing with GPUs**

Seminarvorbesprechung:
High Performance Computing Seminar

Prof. Bertil Schmidt

Seminar: **Paralleles Rechnen – Themen**

1. BWA-MEM-SCALE: Accelerating Genome Sequence Mapping on Commodity Servers
2. Accelerating Lattice QCD Multigrid on GPUs Using Fine-Grained Parallelization
3. Revisiting Huffman Coding: Toward Extreme Performance on Modern GPUs
4. Toward accelerated stencil computation by adapting tensor core unit on GPU
5. Tensor-Accelerated Fourth-Order Epistasis Detection on GPUs
6. Accelerating Random Forest Classification on GPU and FPGA
7. Extreme-Scale Many-against-Many Protein Similarity Search
8. A Programming Model for GPU Load Balancing
9. Stream-K: Work-centric Parallel Decomposition for Dense Matrix-Matrix Multiplication on the GPU
10. Scalable Distributed High-Order Stencil Computations
11. **Selber vorgeschlagenes Thema** (muss aber von mir genehmigt werden)

Themenvergabe

- E-Mail mit **2** bevorzugten Themen (mit Präferenz) an Prof Schmidt **<u>bis spätestens 12.2.2023</u>**
- Ich werde dann versuchen die Themen an Studenten zuzuordnen
- Neue Themen können auch vorgeschlagen werden (müssen dann aber von mir genehmigt werden)

Scheinkriterien und Organisation

- **Voraussetzung:** Teilnahme an „HPC“ oder „Accelerated Computing with GPUs“(mindestens Klausurzulassung)
- Vortrag von ca. 30-35min (inkl. Q&A)
  - Termine: werden noch bekanntgegeben (voraussichtlich Freitags, 12-14Uhr)
- Abgabe der Vortragsfolien
  - **zwei Wochen vor dem Vortrag einzureichen per Email**
  - Danach persönliche Vorbesprechung
- Abgabe einer ausführlichen Ausarbeitung als Basis für die Bewertung des schriftlichen Teils
  - Ausarbeitung im IEEE CS Format (Umfang mindestens 5-6 Seiten)
  - **Abgabe bis spätestens vier Wochen nach dem Vortrag!**

Vorstellung des Lehrangebots – Efficient Computing and Storage Group

Prof. Dr.-Ing. André Brinkmann

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Vorlesungen, Seminare, Abschlussarbeiten

- Kommunikationsnetze

- Vorlesung Speichersysteme
- Praktikum Betriebssysteme
- Bachelor und Masterarbeiten

Kommunikationsnetze

Lernziel:

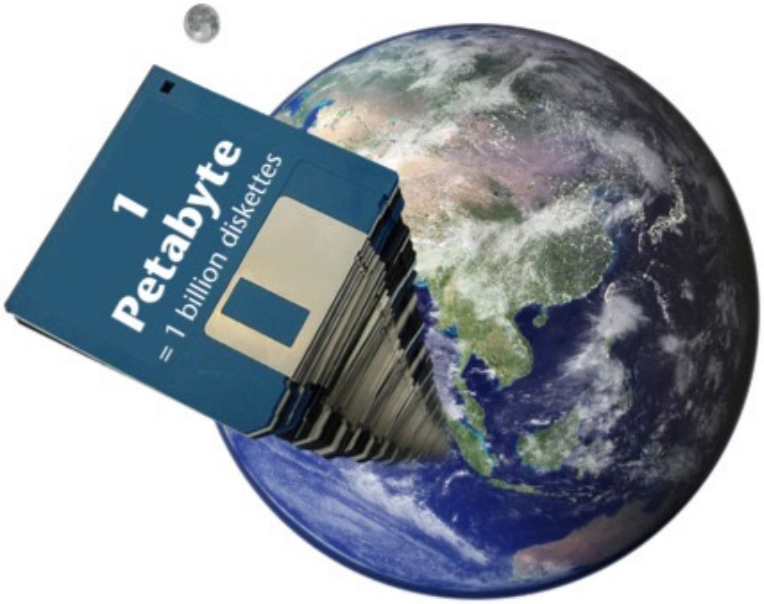
- Prinzipien vernetzter und sicherer Kommunikation in Rechnernetzen (insbesondere im Internet) verstehen und bewerten können

Themen:

- Theoretische Grundlagen der Datenübermittlung und Kodierungstheorie
- Dienste und Protokolle
- Switching und Routing
- Weitverkehrsnetze und lokale Netze
- Risiken, Sicherheitsprobleme, Angriffsszenarien,
- Netzmanagement

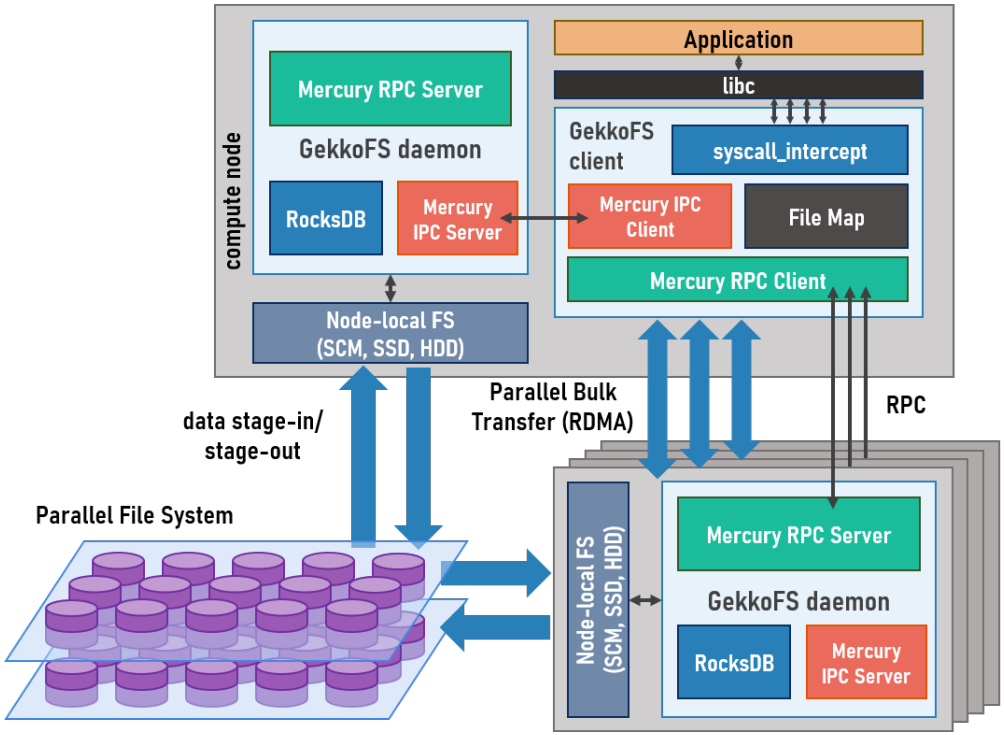
Vorlesung Speichersysteme

- Die Vorlesung wird angefangen bei einzelnen Festplatten bis hin zu global verteilten Speicher- und Peer-to-Peer-Systemen eine Einführung in moderne Speichertechnologien gegeben
- Schwerpunkte sind der Aufbau und die Entwicklung von Speichersystemen, RAID-Systemen und Speichernetzwerken sowie zugehörige Protokolle



Vorlesung Speichersysteme

- Aufbauend auf den Architekturen werden **lokale und verteilte Dateisysteme** sowie Speichermanagement-Konzepte diskutiert
- Den Abschluss der Vorlesung bildet ein Ausblick auf neue Speicherkonzepte auf Basis von Anwendungen des Cloud Computings



Vorlesung Speichersysteme

- Die theoretischen Grundlagen der Vorlesung werden in den Übungen durch Modellierungs- und Simulationsphasen praktisch vertieft
- Aufbauend auf diesen Übungen werden einzelne Aspekte der Speichersystemanbindung in Betriebssysteme am Beispiel von Linux implementiert

JG|U

Praktika Betriebssysteme und Kommunikationsnetze

- In den Praktika wird jeweils eine Teilaufgabe aus den Vorlesungsinhalten innerhalb einer 2-wöchigen Präsenzphase vertiefend behandelt
- Nach einer theoretischen Einarbeitung in die Thematik werden die Umsetzungen innerhalb des Linux-Systems und auf Basis eingebetteter Hardware vorgenommen

- Mögliche Themen:
  - Programmierung eines SSD-Controllers
  - Anbindung von „intelligenten“ SSDs
  - Einbindung intelligenter Netzwerkkarten (DPUs)
  - Aufbau von VLANs und Routing unter Linux

JG|U

Abschlussarbeiten

Bachelor- und Masterarbeiten auf den folgenden Themen:

- Entwicklung von parallelen Datei- und Speichersystemen
- Nutzung von DPUs und smart Switchen
- High-Performance Computing (Molecular Dynamics, Atmosphärenphysik, Quantum Computing)
- Scale-up und Scale-out Indexstrukturen
- Optimierung von SSDs

In vielen Fällen bieten wir die Möglichkeit, als studentische Hilfskraft in Themen einzusteigen

JG|U

AG Scientific Computing and Bioinformatics

Lehrveranstaltungen im Sommersemester 2023

Prof. Dr. Andreas Hildebrandt

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Strukturbasierte Bioinformatik

- Dozent: Andreas Hildebrandt
- Termin: Dienstags, 14–16 Uhr
- Sprache: Deutsch oder Englisch
- Inhalte:
  - Grundlegende Begriffe der Bioinformatik
  - Einführung in die Proteinphysik
  - Simulation von Molekülbewegungen
  - Proteinstrukturminimierung
  - Docking und Wirkstoffdesign

Bioinformatik-Seminar

- Dozent: Andreas Hildebrandt
- Termin: TBA (semesterbegleitend)
- Sprache: Englisch
- Inhalte: Aktuelle Forschungsthemen der Bioinformatik
- Hinweise:
  - Themenwahl erfolgt im Anschluss an die Hauptanmeldephase
  - Vorheriger Besuch der Vorlesung „Strukturbasierte Bioinformatik“ dringend empfohlen!

Big-Data-Seminar

- Dozent: Thomas Kemmer
- Termin: TBA (semesterbegleitend)
- Sprache: Deutsch oder Englisch
- Inhalte: Methoden und Forschungsthemen aus der Big-Data-Welt
- Hinweise:
  - Themenwahl erfolgt im Anschluss an die Hauptanmeldephase
  - Vorheriger Besuch der Vorlesung „Big Data“ wird vorausgesetzt!

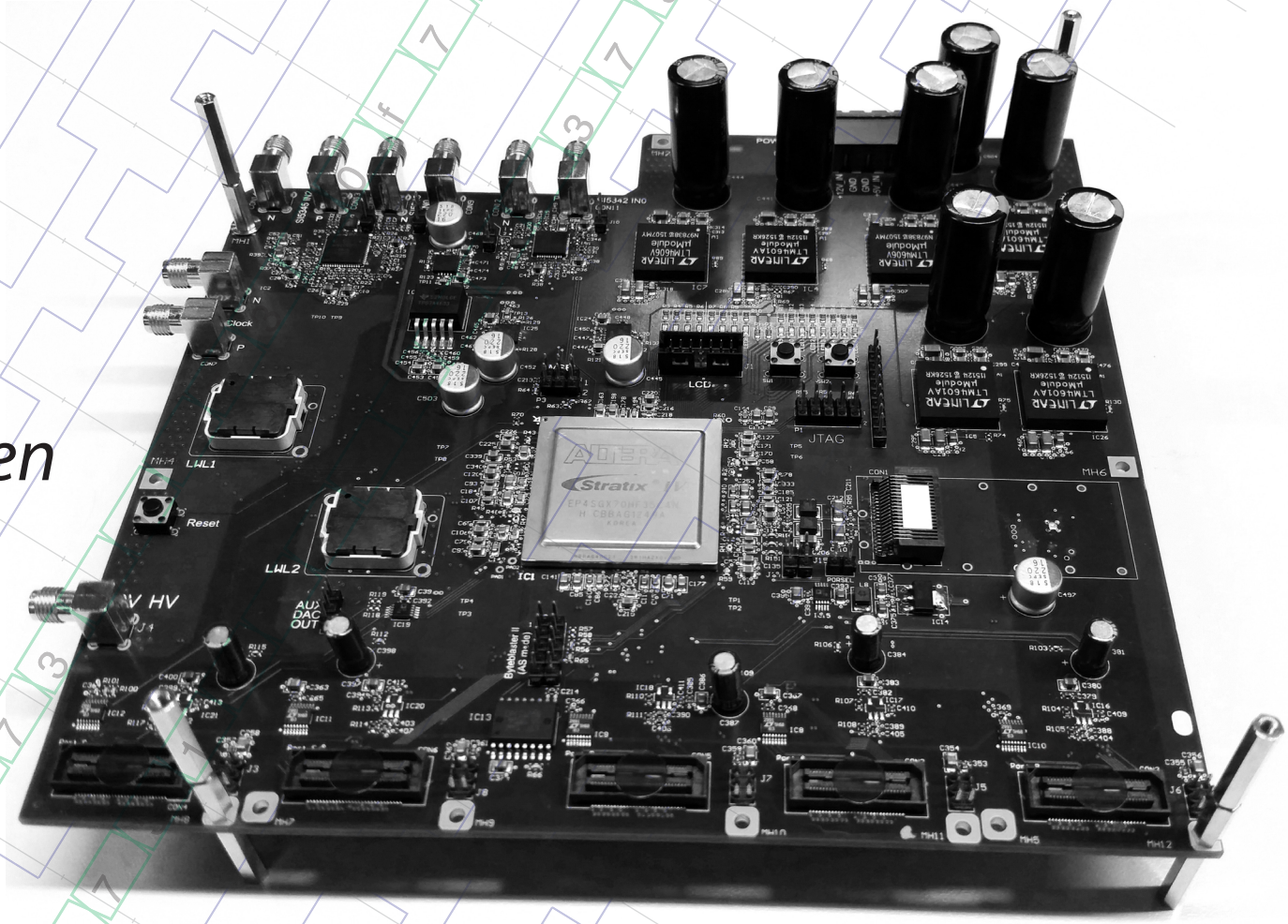
Big-Data-Praktikum

- Dozent: Thomas Kemmer
- Umfang: 2 Wochen Vollzeit
- Termin: In Absprache mit den Teilnehmenden
- Sprache: Deutsch
- Inhalte: Projekte aus der Vorlesung im Wintersemester 2022/23
- Hinweise:
  - Vorheriger Besuch der Vorlesung „Big Data“ wird vorausgesetzt!

FPGA-Programmierung

Niklaus Berger,
Sommersemester 2023
Vorlesung mit praktischen Übungen

Mittwoch 10-12
+ Übungen



FPGAs

- Field Programmable Gate Array
- Programmierbare Logikbausteine
- Massiv parallele Datenverarbeitung
- Low-level: Manipuliere einzelne Bits

- Werden in einer Hardware-Beschreibungssprache programmiert: Hier VHDL

Kursziele

- Praktisch: Sie können in VHDL einen FPGA programmieren: Übungen mit FPGA-Boards - von LED blinken lassen bis zu einem RISC-V Prozessor

- Technische Informatik: Wie funktioniert ein FPGA? Einblick in digitale Schaltungen

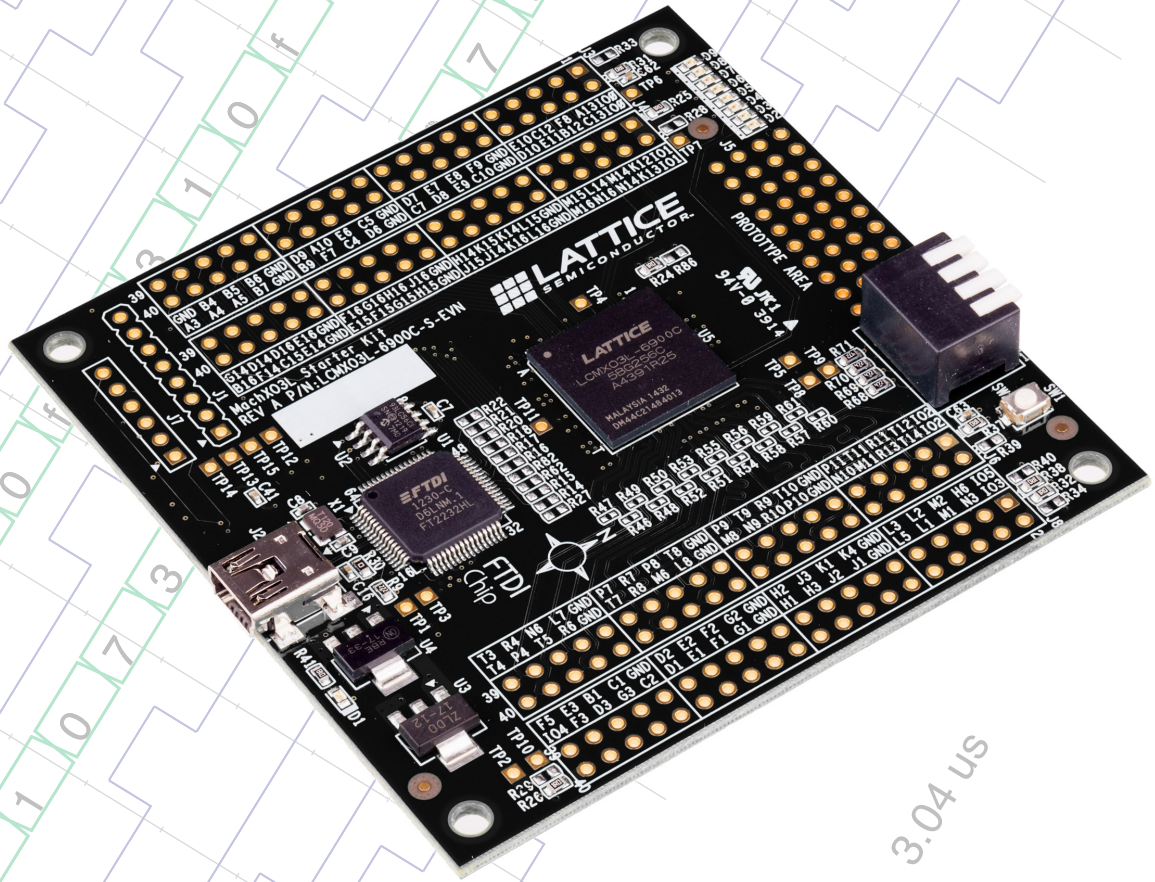
- Algorithmen: Wie kommt mein Design in den FPGA? Logikvereinfachung, Technology Mapping, Place & Route - alles exponentiell schwierige Probleme, für die elegante Heuristiken existieren

Vorläufiges Vorlesungsprogramm

- Weshalb FPGAs?
  Weshalb VHDL?
- CMOS, Gatter, kombinatorische Logik
- Register, Takt, Pipelines
- Simulation
- Synthese
- Place & Route
- Schnelle Schaltungen
- VHDL Design Patterns
- Synchronisation
- Speicher und vordefinierte Logikblöcke (IP)
- Busse und andere Interfaces
- Anwendungsbeispiele

FPGA@Home

- Kleines FPGA Board für zu Hause
- Probieren Sie Schaltungen in Hardware aus



Projektmanagement

Dr. Sebastian Hoffmann, PMP®

Freitags 9:00 – 12:00, tlw. MS Teams

Ein **Projekt** ist …

- [PMI] … ein zeitlich begrenztes Vorhaben zur Schaffung eines einmaligen Produktes, einer Dienstleistung oder eines Ereignisses.

Projektmanagement ist …

- [PMI] … die Anwendung von Wissen, Fertigkeiten, Werkzeugen und Methoden auf Projektvorgänge, um die Projektanforderungen zu erfüllen.

Projektmanagement

Initiating → Planning → Executing → Monitoring & Controlling → Closing

Integration Management	Time Management	Procurement Management
Stakeholder Management	Cost Management	Quality Management
Communications Management	Risk Management	Agiler Exkurs (z.B. SCRUM)
Scope Management	Human Resource Management	

- Tlw. digitale Lehre in MS Teams, s. JOGU-StINe
- Vorlesung und Übung (2+2 SWS)
- Schriftliche Prüfung am Ende der Veranstaltung
  - 50% der erreichbaren Übungspunkte
  - Aktive Teilnahme an den Übungen

- Übungsbearbeitung in
  - max. 5 Projektgruppen
  - je 4-5 Personen
- Alltagsprojekte, keine IT-Projekte

- Theoretische Aufgaben:
  - Abgabe bis Mittwoch 20:00 Uhr
- Praktische Aufgaben:
  - Versand zum Review bis Mittwoch 16:00 Uhr

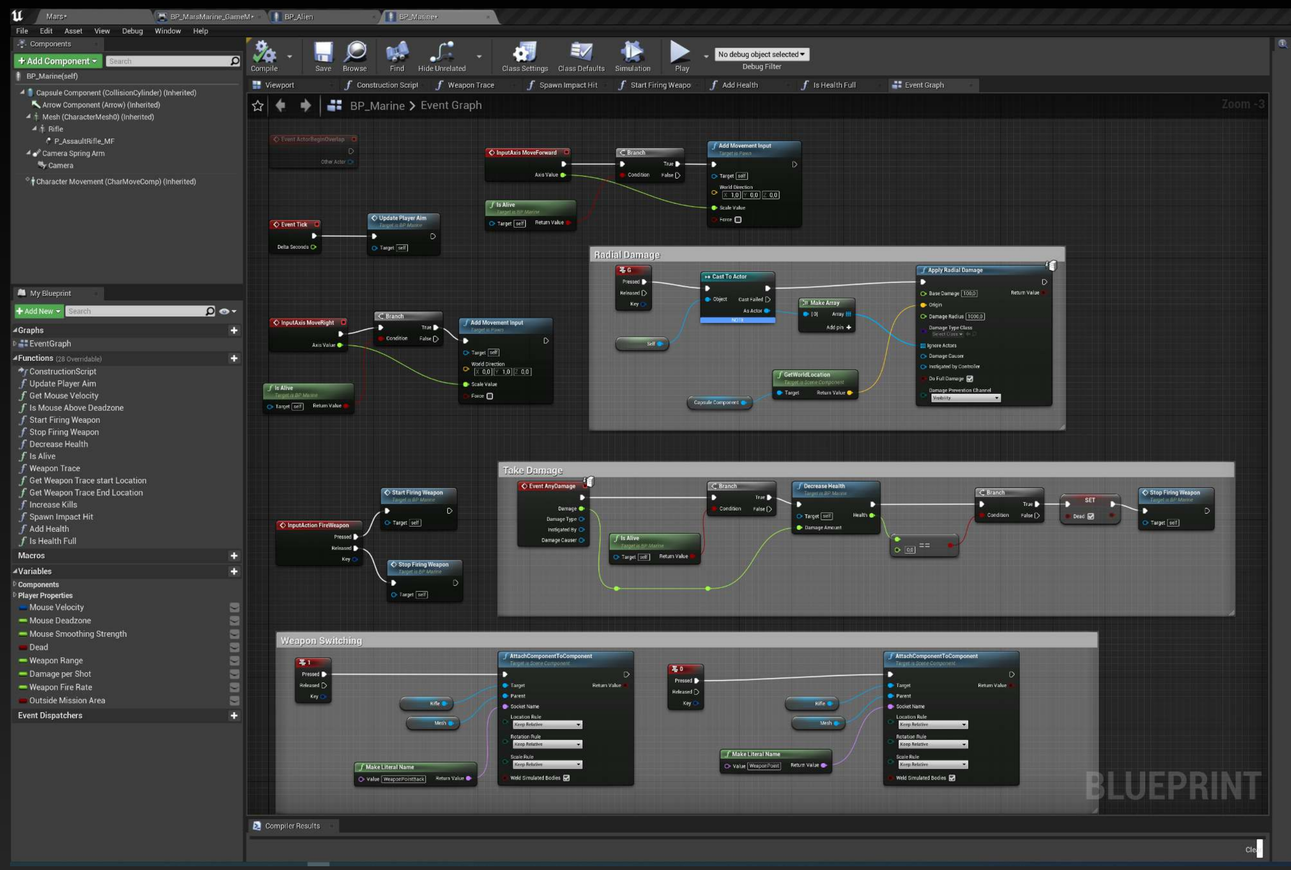
Einführung in die Computergrafik mit **Game Engines**

Arbeitsgruppe: Computational Geometry (Prof. Schömer)

Dozenten: Andreas von Dziegielewski, Ronja Schnur

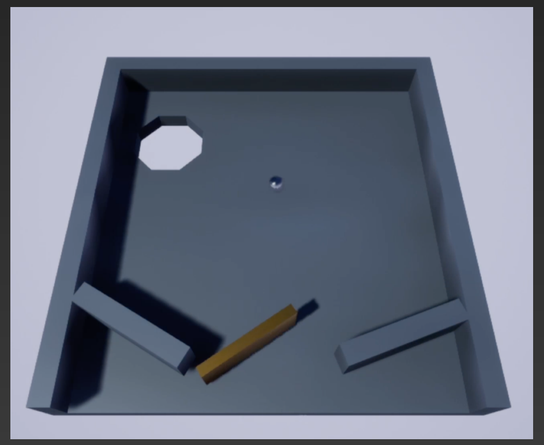
Veranstaltungstyp: Hybrid (Teile sind online, Teile sind in Präsenz)

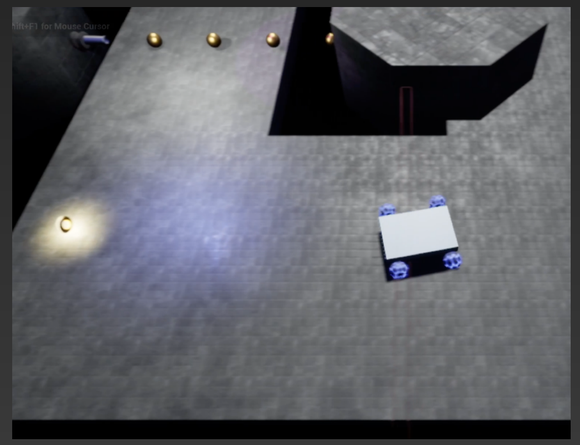
Präsenztermine: Mo 10:15-11:45



Inhalt: Unreal Engine 4 (UE4) Editor, Assets importieren oder erstellen, Spielmechanik (Gameplay) implementieren anhand von 3-4 Spielen/Demos, unter Verwendung von Blueprint (UE4 Visual Scripting System)

Organisation: Bachelor-Veranstaltung, kombinierbar als Modul Computergrafik mit Computergrafik Seminar (WS23/24) und Computergrafik-Praktikum (vorlesungsfreie Zeit nach SS23)

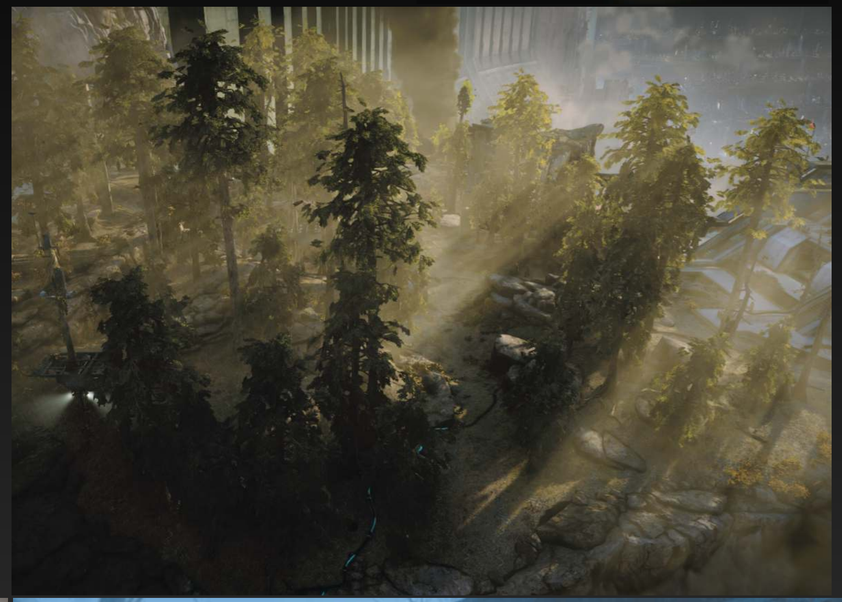




UNREAL ENGINE

Computergrafik I Seminar
Arbeitsgruppe: Computational Geometry (Prof. Schömer)

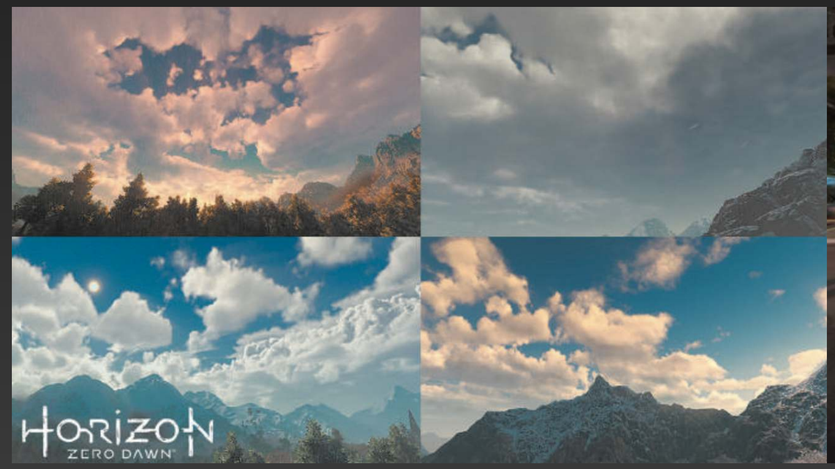
TBA, online

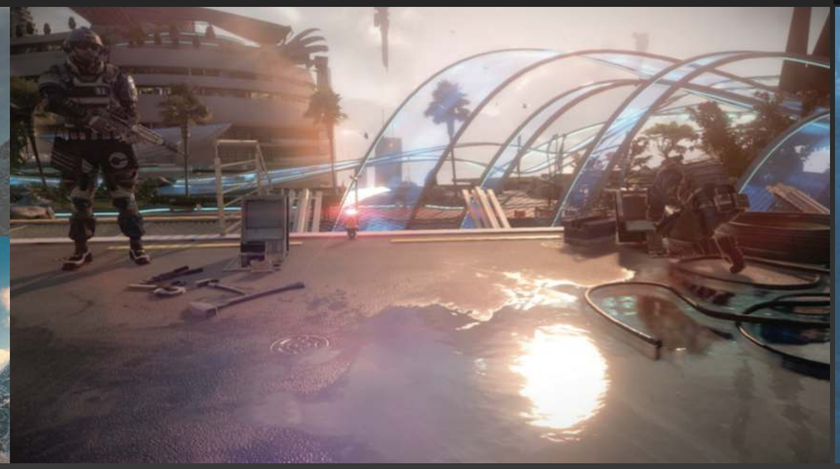


Voraussetzung: erfolgreiche Teilnahme an Computergrafik I

Besprechung und Themenvergabe erfolgt in den ersten Wochen des Semesters, nach Einladung in MS Teams Channel.

Beispiele aus letzten Semestern:





Lehrveranstaltungen der theoretischen Informatik

- Formale Sprachen und Berechenbarkeit
- Graphenalgorithmen
- Seminar: Algorithmen
- Praktikum Approaching Programming Contests (Domenico Mosca)

Graphenalgorithmen

Inhalt: Direkte Weiterführung von DSeA

- Matchings
- Lowest Common Ancestors
- Randomisierte Graphenalgorithmen
- Parametrisierte Algorithmen

Formales:

- Vorlesung: Mi 10-12
- Übungsleitung: Jonas Blochwitz
- Übung: tba

Seminar: Algorithmen

- Zeit und Ort: Blockseminar im September (genauer Termin wird per Umfrage bestimmt)
- Vorbesprechung: Anfang April online (genauer Termin wird noch bekannt gegeben)
- Voraussetzungen: DSeA oder äquivalent
- Scheinvergabe:
  - Vortrag von ca. 30min
  - Ausarbeitung in LaTeX
- Inhalte:
  - Vorstellung eines Algorithmus ähnlich denen aus DSeA
  - Genauere Themenvorschläge gibt es in der Vorbesprechung

Approaching Programming Contests (Praktikum)

- Zeit und Ort: wird mit den Teilnehmern vereinbart
- Betreuung: Domenico Mosca
- Voraussetzungen: EiP, EiS, DSEA

Inhalt:

- Lösen von Problemen, wie sie typischerweise in Programmierwettbewerben auftreten
- Eingesetzte Algorithmen: z.B. Breitensuche, Berechnen von minimalen Spannbäumen, Max-Flow-Algorithmen, dynamische Programmierung, Divide-And-Conquer
- Teilnahme am GCPC (Termin noch unbekannt)

Erwünscht:

- Bereitschaft, am NWERC teilzunehmen

NWERC 2016 und 2017 in Bath, UK

NWERC 2016 und 2017 in Bath, UK

GCPC und NWERC

Die Mainzer Teams erreichten beim German Collegiate Programming Contest die Plätze 18, 48 und 55 bei 80 angetretenen Teams.

Insgesamt nehmen in Nordwesteuropa etwa 700 Teams bei nationalen Contests teil.

Beim Northwestern European Contest 2017 erreichte Mainz Platz 60 von 118 Teams.

Numerische Algorithmen

Apl.Prof. Dr. Georg von Hippel

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Sommersemester 2023

Was ist das?

- Computer = Rechner
- Rechnen mit (i.d.R. Gleitkomma-)Zahlen = Numerisches Rechnen
- Numerische Algorithmen finden konkrete numerische Antworten:
  - Lösen von algebraischen und Differentialgleichungen
  - Berechnen von Integralen und Ableitungen
  - Interpolation und Approximation
  - ...

Warum ist das wichtig?

- In den (Natur/Ingenieur/Wirtschafts-)Wissenschaften wird mit Zahlen gerechnet
- Analytische Lösungen selten für realistische Probleme
- Numerische Verfahren von größter Relevanz
- Ohne numerische Algorithmen keine
  - Wettervorhersage
  - Klimamodellierung
  - Computertomographie
  - Raumfahrt
  - ...

Was machen wir in dieser Vorlesung?

- Interpolation und Extrapolation
- Numerisches Lösen linearer Systeme (Gauss-Methode, LU und Cholesky-Zerlegung)
- Numerisches Lösen nicht-linearer algebraischer Gleichungen
- Numerisches Ableiten und Integrieren
- Numerisches Lösen gewöhnlicher Differentialgleichungen

Eine Auswahl aus

- Fast Fourier Transform (FFT) und Anwendungen
- Numerisches Lösen partieller Differentialgleichungen
- Iterative Lösungsmethoden für lineare Systeme
- Modellierung von Daten

Was lernen wir dabei?

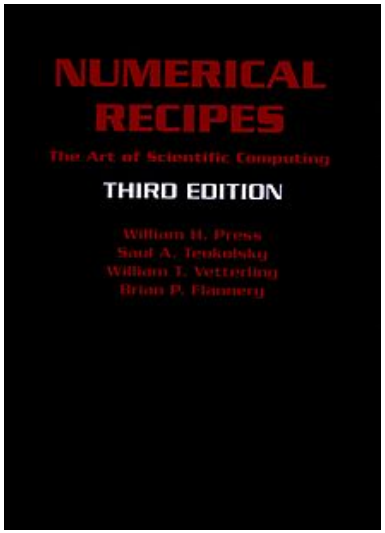
- Arbeiten mit Gleitkommazahlen
- Verständnis für besondere Eigenschaften (Genauigkeit, Stabilität) numerischer Algorithmen
- Implementation numerischer Algorithmen
- Kenntnis wichtiger numerischer Algorithmen

Organisatorisches

- 2V+2Ü
- Vorlesung Montags, 10:00-12:00
- Übung n.V.

Empfohlene Literatur

W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery,
Numerical Recipes: The Art of Scientific Computing (Third edition),
Cambridge University Press 2007.



http://numerical.recipes/

Algorithmen und Techniken der Optimierung

Frank Fischer

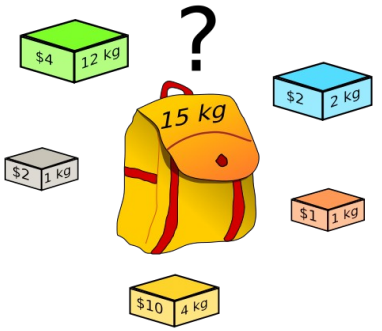
JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Überblick

Worum geht‘s?

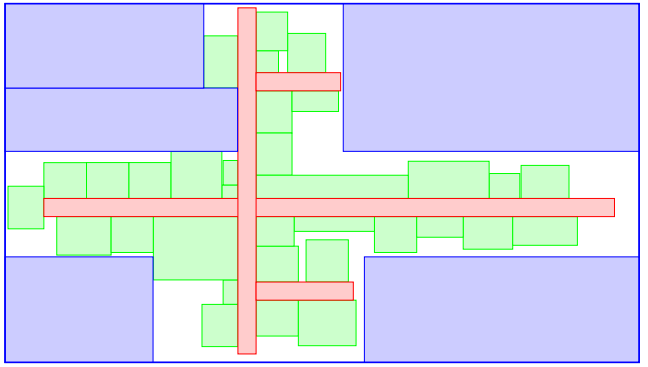
- Viele (kombinatorische) Optimierungsprobleme lassen sich als *mathematische Optimierungsaufgabe* auffassen:
  - Rucksackproblem:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} w_i x_i \\ \text{subject to} \quad & \sum_{i=1}^{n} g_i x_i \leq G, \\ & x \in \{0,1\}^n \end{aligned}$$



Quelle: Wikipedia

  - Probleme aus Logistik, Fabrikplanung, Lagerhaltung, Transport, Verkehr, Telekommunikation, …



- *Exakte* Lösungsverfahren mit *Gütegarantie*
- Alternative zu heuristischen Verfahren (Genetische Algorithmen, Simulated Annealing, Tabu-Search, ...)

Inhalte und Ziele

Ziele

- Kenntnis der *Möglichkeiten* und *Grenzen* mathematischer Optimierung

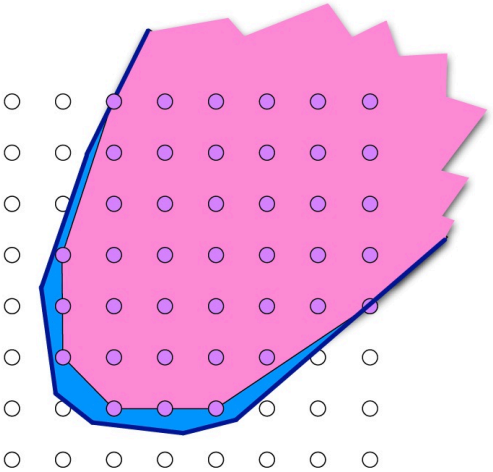
  „know your tools“

- Modellierung praktischer Problemstellungen als *„mathematische Programme“*
- Lösungs- und Näherungsverfahren, Gütegarantien, Komplexität
- *Praktikum:* Umsetzung mit State-of-the-Art Optimierungssoftware

Inhalte

- Lineare Optimierung
- Lineare Ganzzahlige Optimierung
  - Komplexität
  - Exakte Verfahren für Spezialfälle (Unimodularität)

    Schnittebenenverfahren
- Dekompositionsverfahren



Modulinhalt

Zielgruppe

- B.Sc. und M.Sc., theoretische Informatik

Sommersemester 2023

- 2 SWS Vorlesung + Übung
  - Theoretische Grundlagen

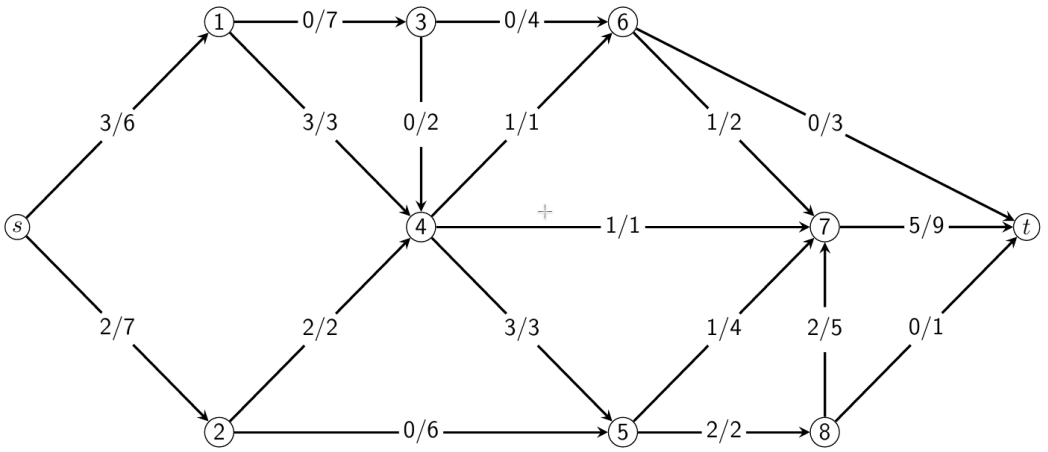
Wintersemester 2023/2024

- Seminar
  - Selbständige Bearbeitung vertiefender theoretischer Konzepte
- Praktikum
  - Einführung in State-of-the-Art Optimierungssoftware
  - Eigenständige Modellierung und Implementierung
  - Praktische Umsetzung fortgeschrittener algorithmischer Konzepte

Voraussetzungen

Notwendig

- Solide Grundkenntnisse Lineare Algebra
- Geometrisches Vorstellungsvermögen
- Praktikum: Programmierung

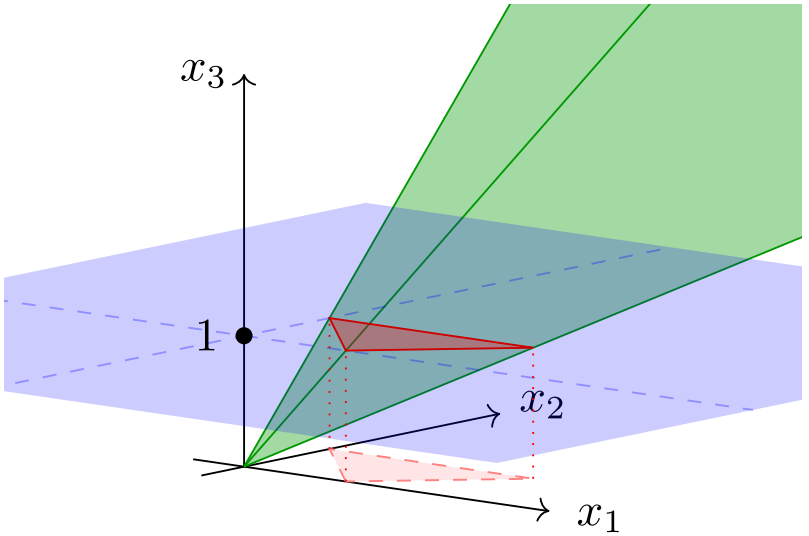


Empfohlen

- Graphentheorie
- Grundkenntnisse Kombinatorische Optimierung (Flussprobleme, Färbeprobleme, …)

Es geht um *Mathematische* Optimierung

Termine



Vorlesung

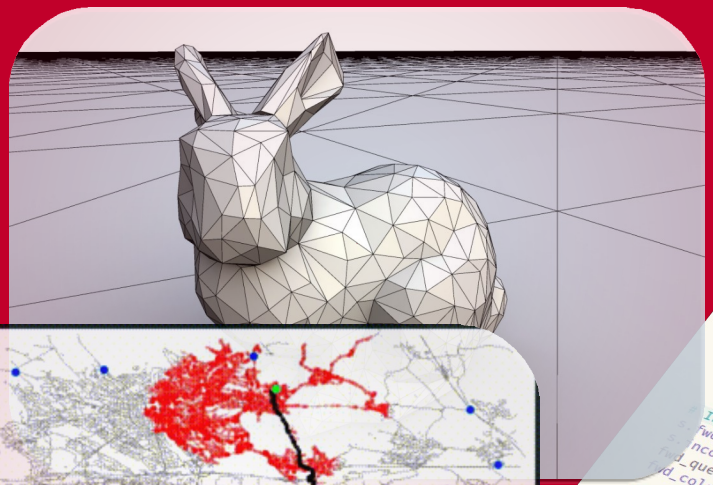
- Dienstag, 12-14 Uhr, 03-428

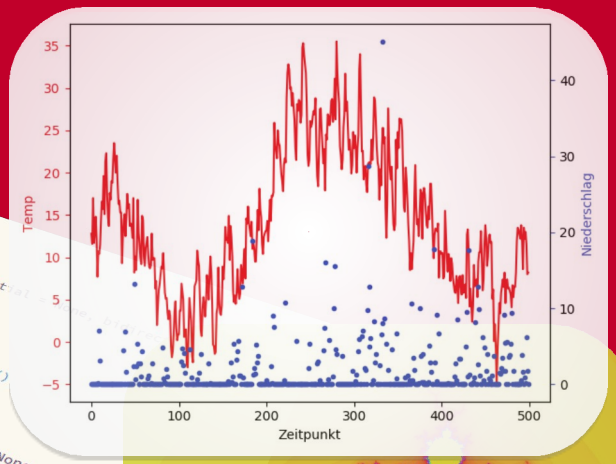
Übung

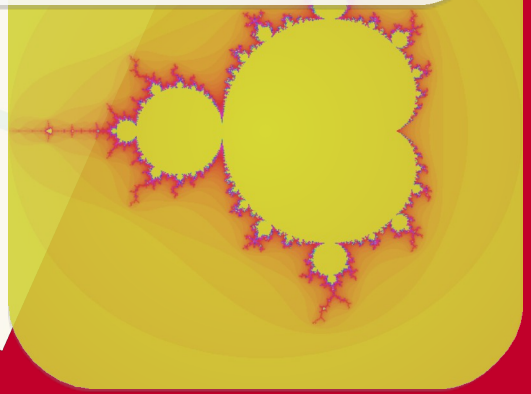
- Steht noch nicht fest (vermutlich Freitag, 8-10 Uhr)

Bei Fragen

- Frank Fischer <frank.fischer@uni-mainz.de>







Mathematische Modellierung am Rechner I & II

Frank Fischer

Di. 16-18 Uhr, 03-428

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Überblick

Worum geht‘s?

- Mathematik sehr abstrakt
- Wichtig für viele Bereiche der Informatik (und für‘s Studium ;))
- Man merkt erst viel später, wozu das alles gut ist

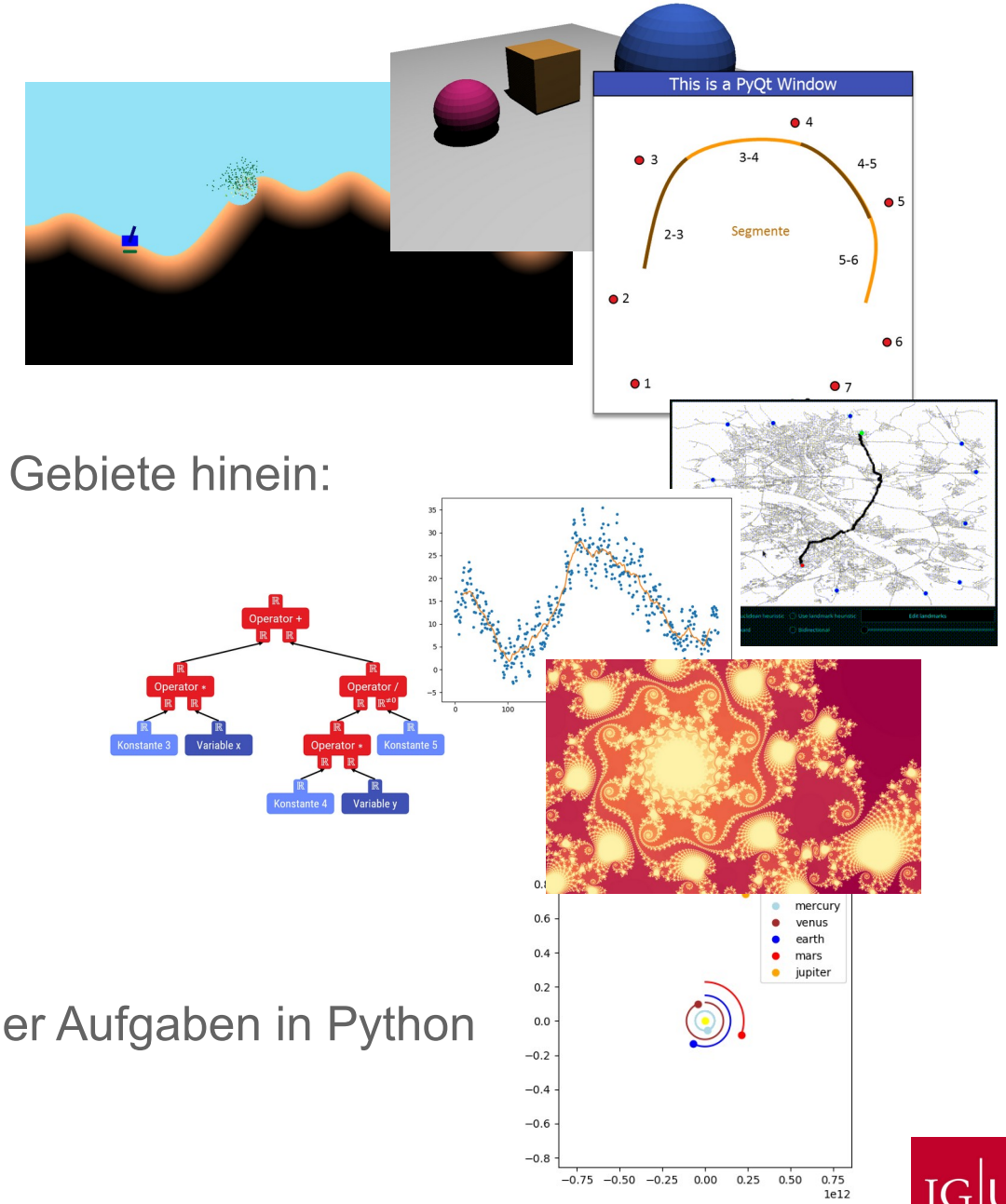
MMR bietet einen angewandten Einstieg

- Frühzeitig gemeinsam Beispiele erarbeiten, wo und wie Mathematik angewandt werden kann
- Praktisch, das heißt, wir werden Mathematik *am Rechner* anwenden
- Ergänzung zu Grundvorlesungen

Inhalte und Ziele

Ziele

- Mathe ist cool
- Informatik ist cool
- Mathe + Informatik = …



Inhalte

- Wir schnuppern in verschiedene Gebiete hinein:
  - Mathematische Modellierung
  - Analysis, Physik
  - Algebra
  - Geometrie/3D
  - Datenanalyse
  - Optimierung
  - Komplexität
- Umsetzung kleiner algorithmischer Aufgaben in Python

Modulinhalt

Zielgruppe

- B.Sc. Informatik ab dem 2. Semester

Aufbau

- MMR 1 im Sommersemester 2023
- MMR 2 im Wintersemester 2023/2024

Nur wer beide besucht, kann das Modul abschließen!

Ablauf

- Praktikum, Gruppenarbeit in Gruppen zu je drei Personen
- Je zwei Wochen zur Bearbeitung einer Aufgabe
- Dazwischen Diskussion über das Thema in Form einer Vorlesung/eines Tutoriums

Voraussetzungen und Termin

Notwendig

- Grundkenntnisse mathematische Sprache und Notation
- Programmierung (in Python)

Empfohlen

- Interesse oder gar Spaß an angewandter Mathematik

Termine

- Dienstag 16-18 Uhr, 03-428

Bei Fragen

- Frank Fischer <frank.fischer@uni-mainz.de>