



Vorstellung der Lehre

SoSe 2022

Ablauf der Vorstellung

13:00 Begrüßung

13:10 AG Gallenbacher

13:15 AG Schömer

13:20 AG Wand

13:25 AG Schuhknecht

13:30 AG Schmidt

13:35 AG Kramer

13:40 AG Hildebrand

13:45 AG Erdweg

13:50 AG Brinkmann

13:55 AG Bouros

14:00 Hoffmann

14:05 Blumenstock

14:10 Ta

14:15 Salkhordehaghghi

14:20 Dziegielewski/Erbes

14:25

14:30 Spichtinger

14:35 Kuntze

14:40 Fischer

Neue Prüfungsordnung B.Sc.

- Zum Sommersemester tritt eine neue Prüfungsordnung in Kraft
- Alle Studierenden, die im jetzigen WS gestartet sind werden automatisch auf die neue PO umgesetzt
- Alle anderen Studierenden haben die Möglichkeit freiwillig auf die neue PO zu wechseln. Weitere Infos folgen in Kürze ...

Hiwis gesucht

- Einführung in die Programmierung
- Einführung in die Softwareentwicklung
- Technische Informatik
- Formale Sprachen und Berechenbarkeit
- Programmiersprachen
- Datenbanken
- Künstliche Intelligenz
- IT-Sicherheit
- Kommunikationsnetze
- Strukturbasierte Bioinformatik
- Machine Learning
- Design Patterns
- Fortgeschrittene Algorithmen
- Praktikum mathematisch Modellierung am Rechner



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

Hiwi fürs SE-Praktikum

- Es wird noch dringend ein Hiwi fürs kommende SE-Praktikum gesucht
- Findet statt vom 21.03. – 01.04.
- Dozent ist Prof. Volker Luckas von der TH-Bingen



**Bewerbung per Mail:
endler@uni-mainz.de**

Betriebspraktikum

- 10-wöchiges Betriebspraktikum in Vollzeit
- Ist eine von drei Optionen im Spezialisierungsmodul
- Anmeldung über Jogustine zu jeder Zeit
- Im entsprechenden Moodle Kurs befinden sich weitere Informationen rund um das Betriebspraktikum und dessen Ablauf



JOHANNES GUTENBERG
UNIVERSITÄT MAINZ



Design Patterns SoSe 2022

Dr. Stefan Endler

endler@uni-mainz.de

Johannes Gutenberg-Universität Mainz

Vorlesung: Donnerstags 12-14, Raum 04-432

Übung: Donnerstags 14-16, ?

Auswahl an Patterns

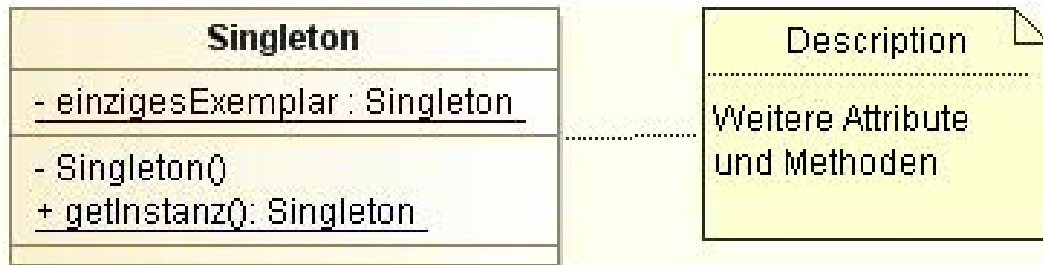


Abbildung: Singleton

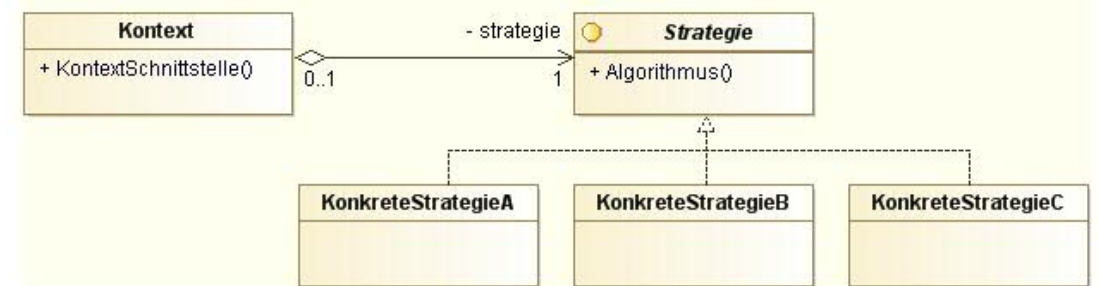


Abbildung: Strategy

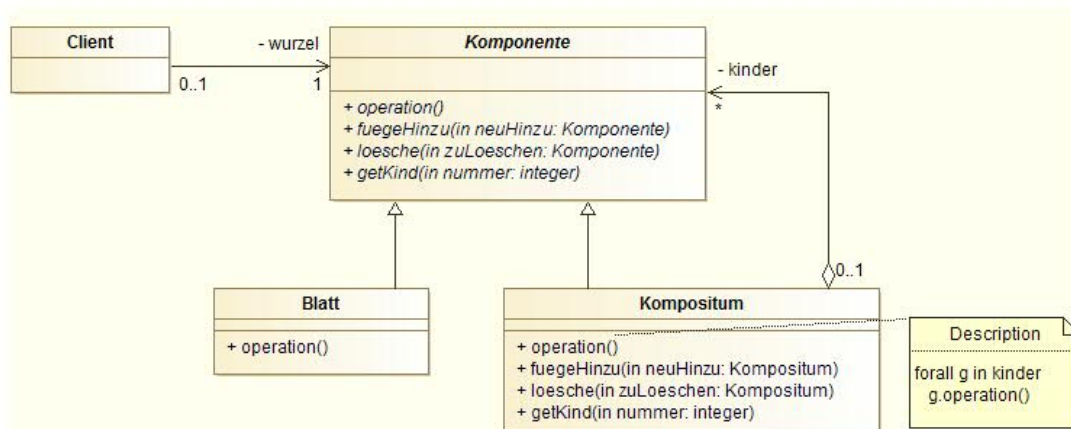


Abbildung: Composite

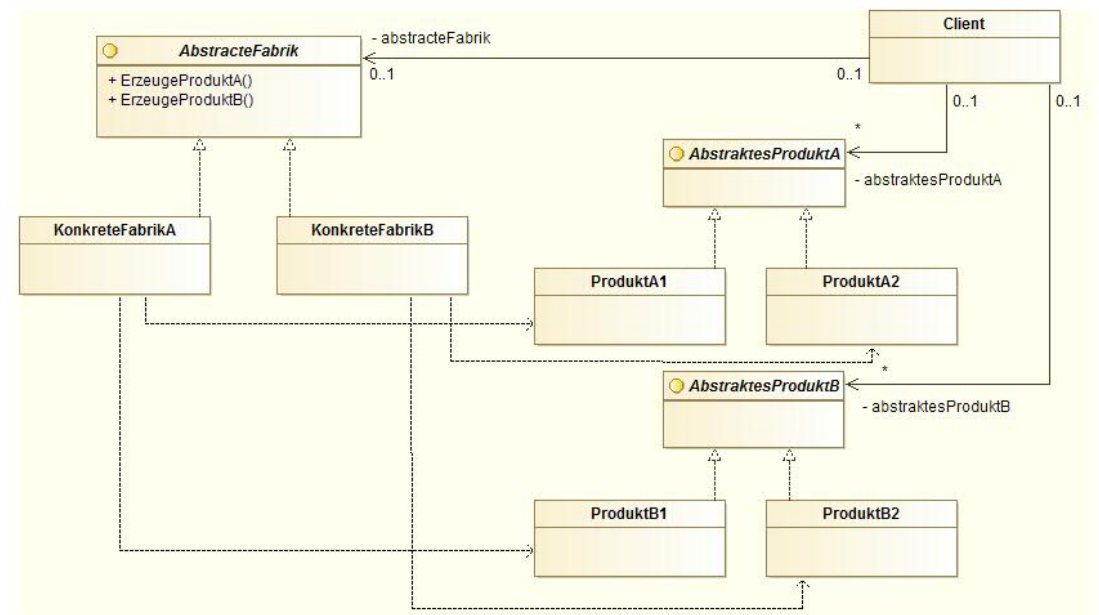


Abbildung: Abstract Factory

Erarbeitung anhand Real-Life Beispielen

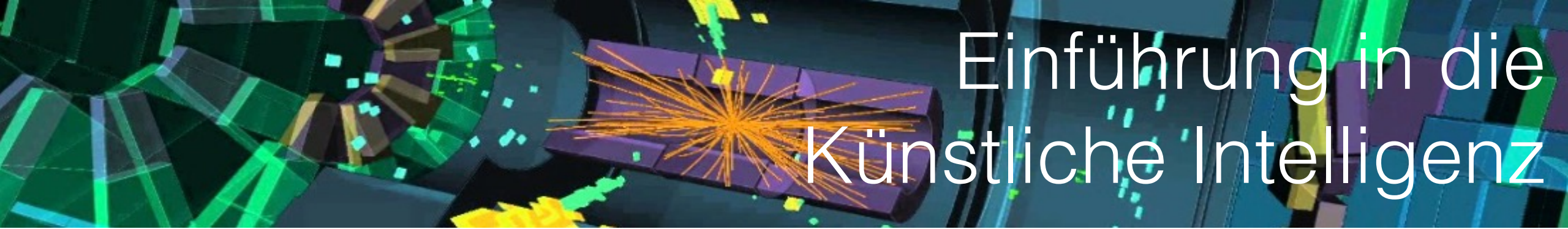


Erarbeitung anhand Real-Life Beispielen



Erarbeitung anhand Real-Life Beispielen





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

3D Vision and Deep Learning

About this course

Prof. Dr. Ulrich Schwanecke



RheinMain University of Applied Sciences

Course Goal and Content

- **Goal**

- Gain an understanding of the theoretical and practical concepts of **3D Computer Vision**, e.g.
 - Camera calibration
 - Epipolar geometry
 - Structure-from-Motion
 - Image rectification
 - Block matching
 - Volumetric fusion
 - ...
- Be able to
 - develop and train computer vision models,
 - reproduce results and
 - conduct original research

- **(Planned) Content (12 Lectures)**

1. Organization, Introduction, History of CV
2. Image Formation
3. 3D Projective Space and 3D Motion
4. Conic Sections and Quadrics
5. Camera Models and Calibration
6. Shape from Shading and Photometric Stereo
7. Structure from Motion
8. Stereo/Multi View Reconstruction, Optical Flow
9. Siamese Networks, End-to-End Learning
10. Volumetric Fusion
11. Coordinate Based Networks
12. Diverse Topics in 3D Computer Vision

Organization

- **SWS 2V + 2Ü, 6 ECTS, Total Workload: 180h**
- **Lecture**
 - Monday, 14:15-15:45, hopefully in presence (03 428)
 - Apr. 25, May. 02/09/16/23/30, Jun. 13/20/27, Jul. 04/11/18
 - All lecture related information at <http://cvmr.info/lectures/3DCVSS22/> (user: 3DCV passwd: ss22)
- **Exercise Sessions**
 - Day/time to be determined (hopefully in presence)
 - Exercises are mandatory
- **Exam**
 - Very likely written (date and time will be announced)
 - To qualify for the exam you have to
 - have $\geq 50\%$ of all achievable points ($\geq 25\%$ for each problem set)
 - present at least one assignment

Course Materials

- **Books**

- Y. Ma, et. al, *An Invitation to 3-D Vision - From Images to Geometric Models*, Springer, 2004, https://www.eecis.udel.edu/~cer/arv/readings/old_mkss.pdf
- R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003, <https://www.robots.ox.ac.uk/~vgg/hzbook/>
- R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2011, <https://szeliski.org/Book>
- I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016, <https://www.deeplearningbook.org>
- J. E. Solem, *Programming Computer Vision with Python*, O'Reilly, 2012
- V. K. Ayyadevara, Y. Reddy, *Modern Computer Vision with PyTorch*, Packt, 2020
- M. P. Deisenroth, A. A. Faisal, C. S. Ong, *Mathematics for Machine Learning*, <https://mml-book.github.io>
- K. B. Petersen, M. S. Pedersen, *The Matrix Cookbook*, http://www.cs.toronto.edu/~bonner/courses/2012s/csc338/matrix_cookbook.pdf

Course Materials

- **Tutorials**

- The Python Tutorial: <https://docs.python.org/3/tutorial>
- Numpy Quickstart: <https://numpy.org/devdocs/user/quickstart.html>
- PyTorch Tutorial: <https://pytorch.org/tutorials>

- **Frameworks, IDEs**

- Visual Studio Code: <https://code.visualstudio.com/>
- Google Colab: <https://colab.research.google.com>

- **Courses**

- Slide deck covering *Szeliski's book* <https://szeliski.org/Book>
- I. Gkioulekas, *Computer Vision* <https://www.cs.cmu.edu/~16385/>
- A. Owens, *Foundations of Computer Vision* <https://web.eecs.umich.edu/~ahowens/eecs504/w20/>

Prerequisites

- Basic math skills
 - Linear Algebra, Calculus, Probability
- Basic computer science skills
 - Variables, functions, loops, classes, algorithms
- Basic Python coding skills
 - <https://docs.python.org/3/tutorial/>
- Basic PyTorch coding skills
 - <https://pytorch.org/tutorials>

Prerequisites

- **Linear Algebra**

- Vectors: $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$
- Matrices: $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$
- Operations:
 - $\mathbf{x}^\top \mathbf{y}, \mathbf{x} \times \mathbf{y}$
 - $\mathbf{A}\mathbf{x}$
 - $\mathbf{A}^\top, \mathbf{A}^{-1}, \text{trace}(\mathbf{A}), \det(\mathbf{A}), \mathbf{A} + \mathbf{B}, \mathbf{A}\mathbf{B}$
- Norms: $\|\mathbf{x}\|_1, \|\mathbf{x}\|_2, \|\mathbf{x}\|_\infty, \|\mathbf{A}\|_F$
- Eigenvalues, Eigenvectors, SVD: $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$

- **Calculus**

- Multivariate functions: $f : \mathbb{R}^n \rightarrow \mathbb{R}$
- Partial derivatives: $\frac{\partial f}{\partial x_i}, i = 1, \dots, n$, Gradient
- Integrals: $\int f(x)dx$

- **Probability**

- Probability distributions: $P(X = x)$
- Expectation: $\mathbb{E}_{x \sim p}[f(x)] = \int_x p(x)f(x)dx$
- Variance: $\text{Var}(f(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$
- Marginal: $p(x) = \int p(x, y)dy$
- Conditional: $p(x, y) = p(x|y)p(y)$
- Bayes rule: $p(x|y) = p(y|x)/p(y)$
- Distributions: Uniform, Gaussian

Time Management

Activity	Times	Total
Attending (watching) the lecture	2h / week	24h
Self-study of lecture materials	2h / week	24h
Participation in exercise	2h / week	24h
Solving the assignments	6h / week	72h
Preparation for the final exam	36h	36h
Total workload		180h

See you on Monday, April 25, 2022



Space-Taste: Zeichenmodus ändern. Linke Maus: Rotieren

Computergrafik II

Elmar Schömer

JOHANNES GUTENBERG
UNIVERSITÄT MAINZ



INSTITUT FÜR INFORMATIK

Sommersemester 2022, Di 10-12 Uhr

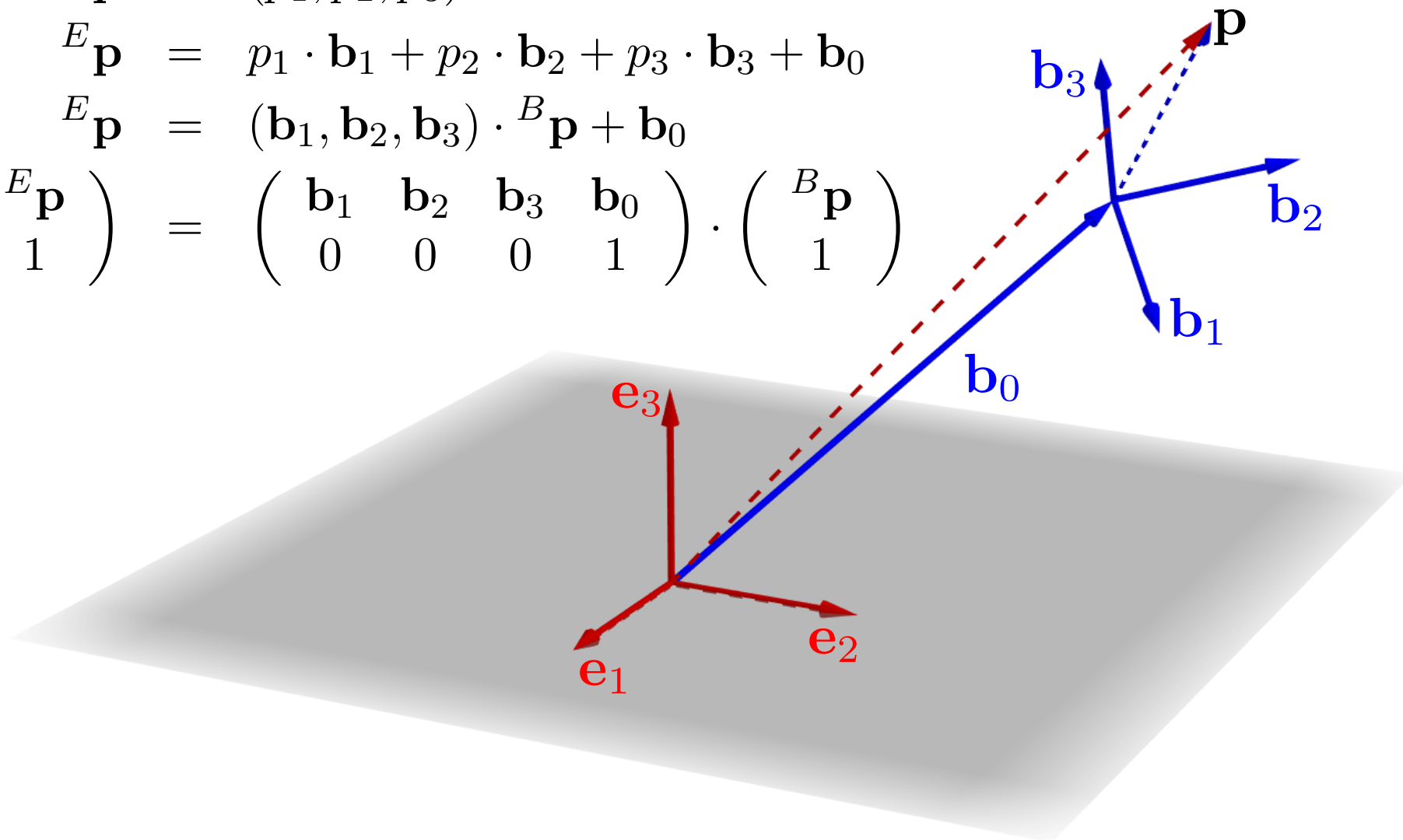
Koordinatentransformation

$${}^B \mathbf{p} = (p_1, p_2, p_3)^T$$

$${}^E \mathbf{p} = p_1 \cdot \mathbf{b}_1 + p_2 \cdot \mathbf{b}_2 + p_3 \cdot \mathbf{b}_3 + \mathbf{b}_0$$

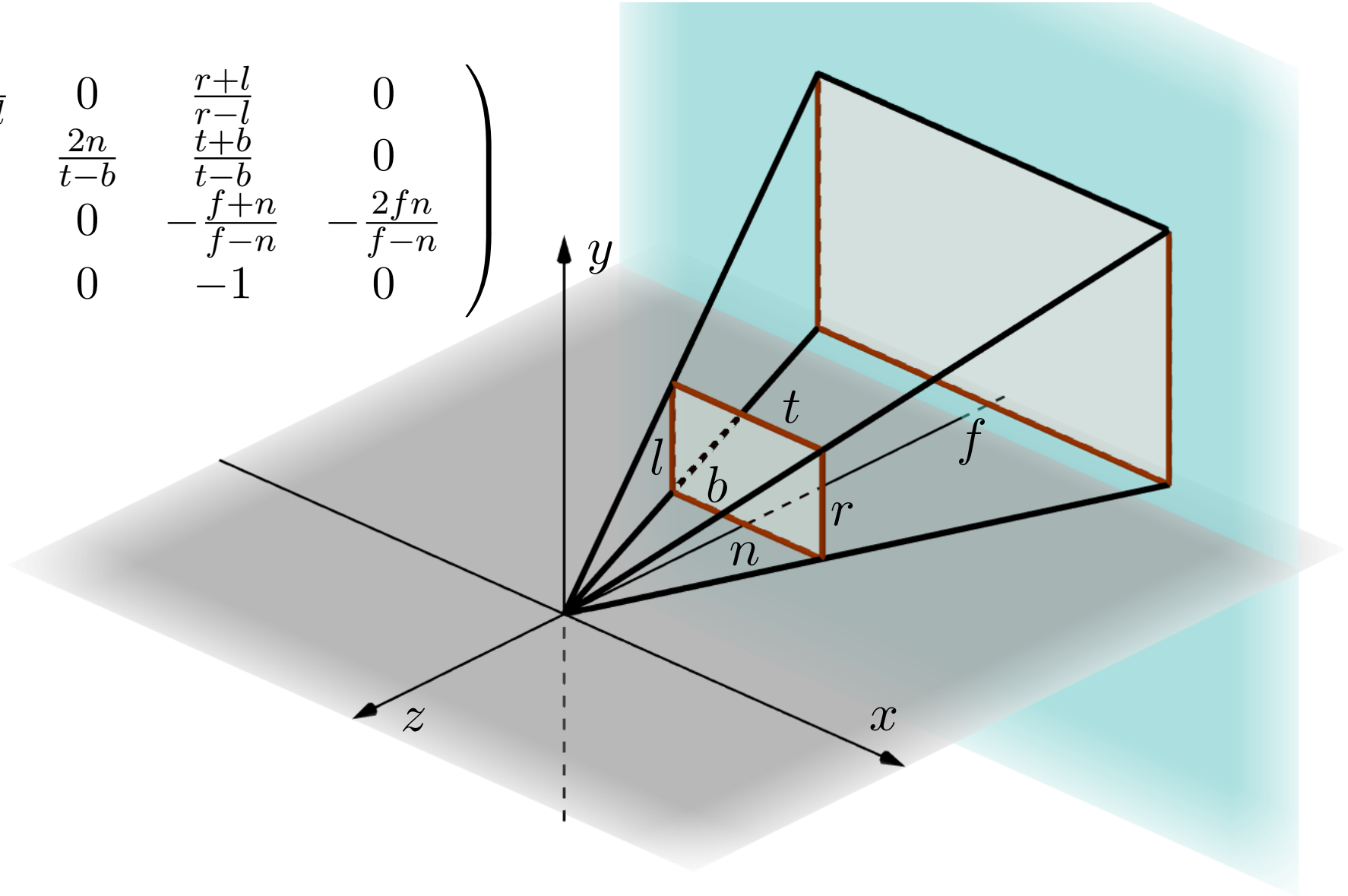
$${}^E \mathbf{p} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3) \cdot {}^B \mathbf{p} + \mathbf{b}_0$$

$$\begin{pmatrix} {}^E \mathbf{p} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \mathbf{b}_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} {}^B \mathbf{p} \\ 1 \end{pmatrix}$$

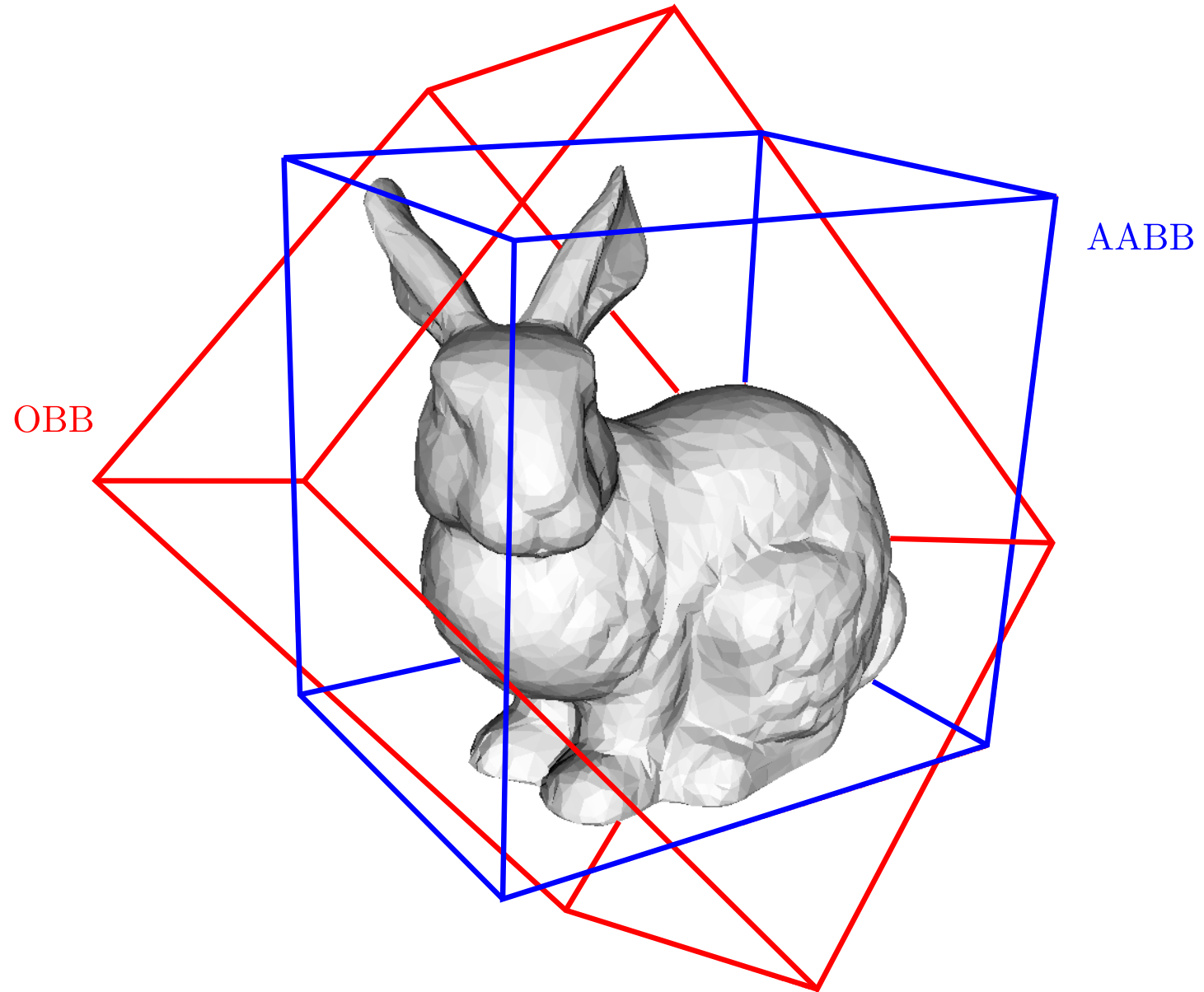


Perspektivische Projektion

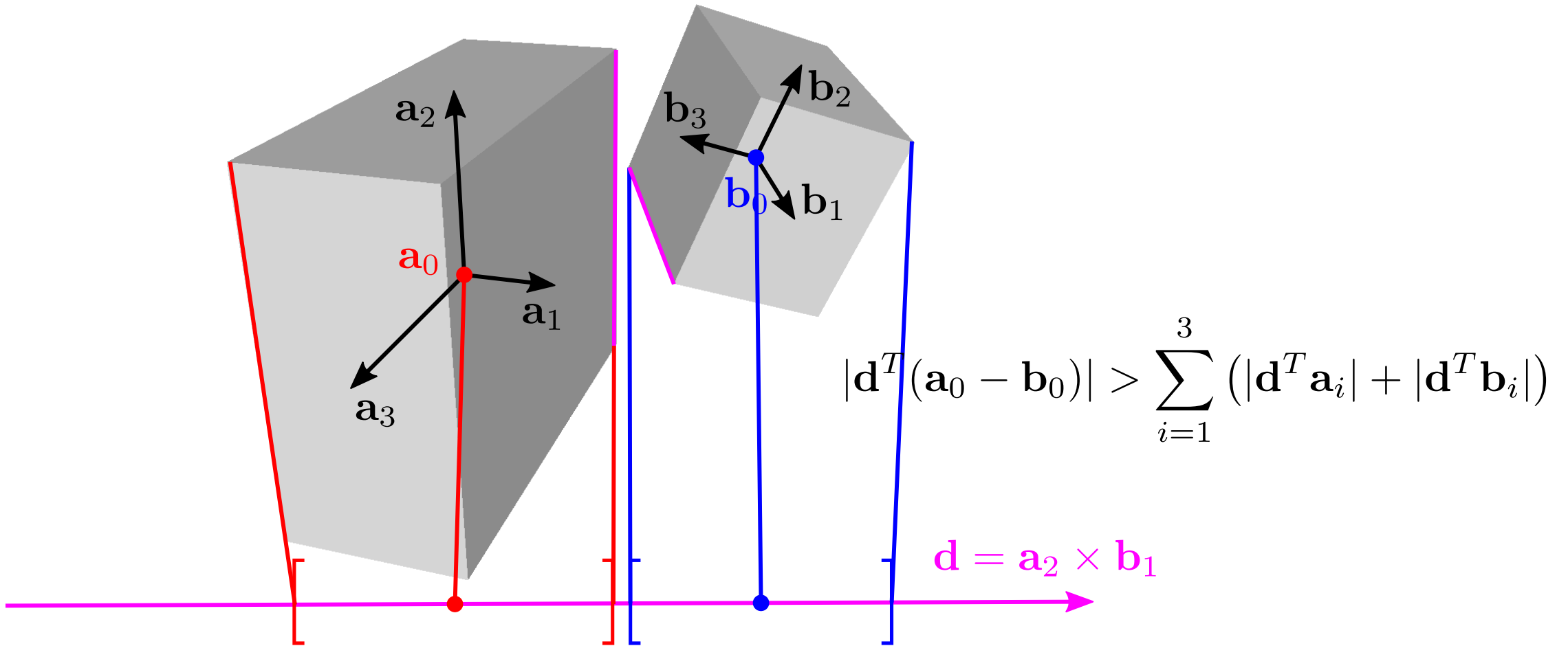
$$\mathbf{P} = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

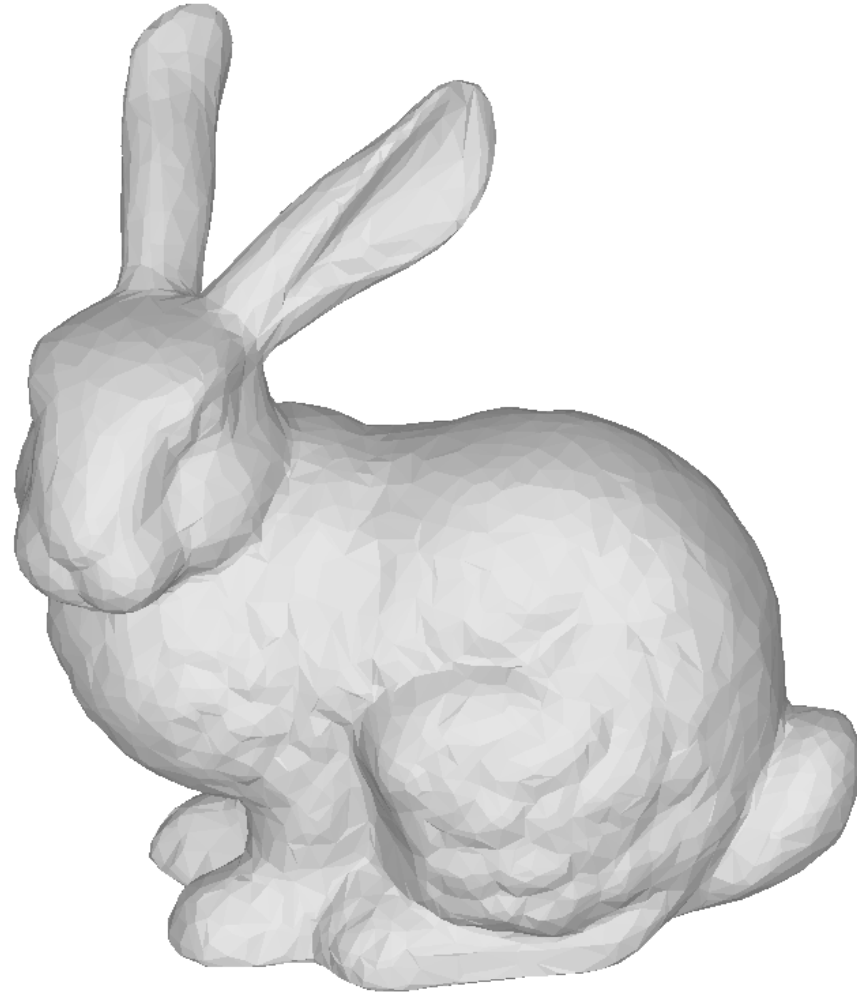


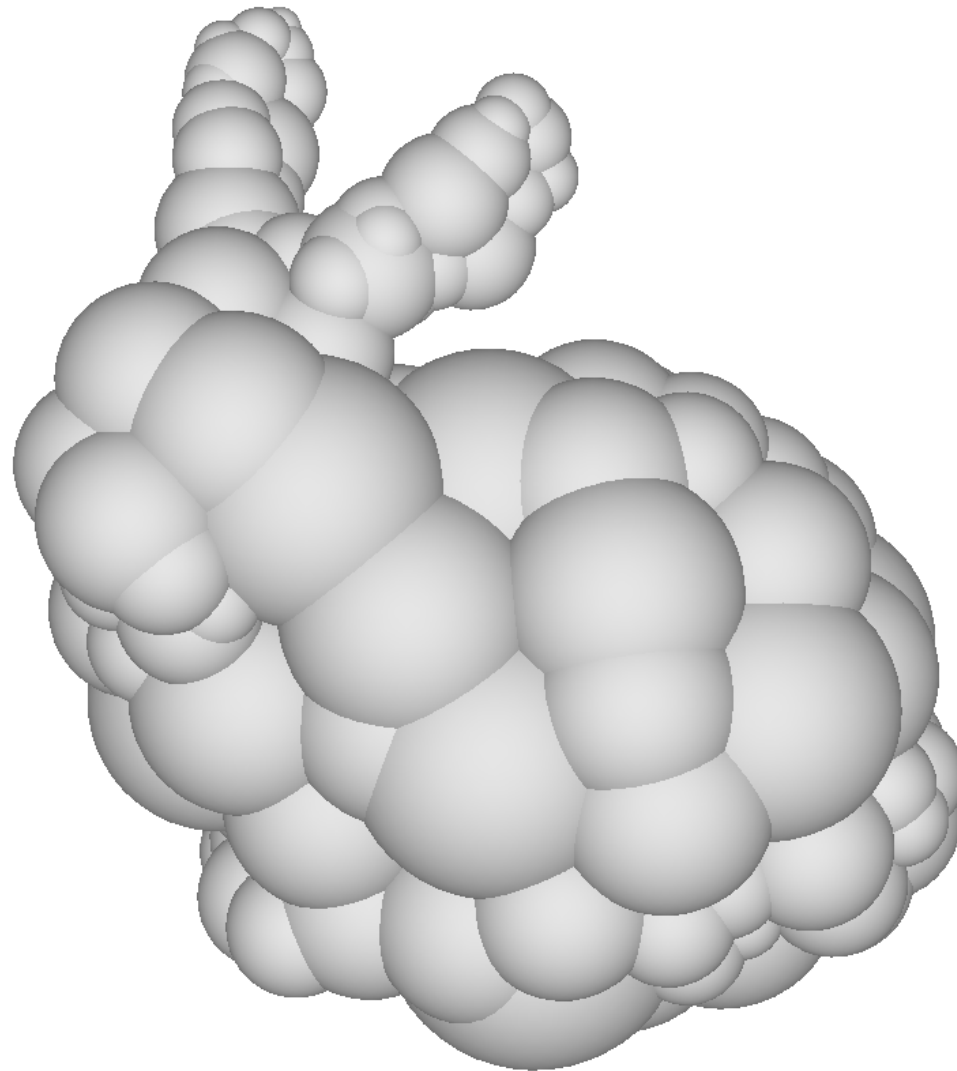
AABB vs OBB



Kollisionstest für OBBs







Kinematik und Dynamik starrer Körper

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

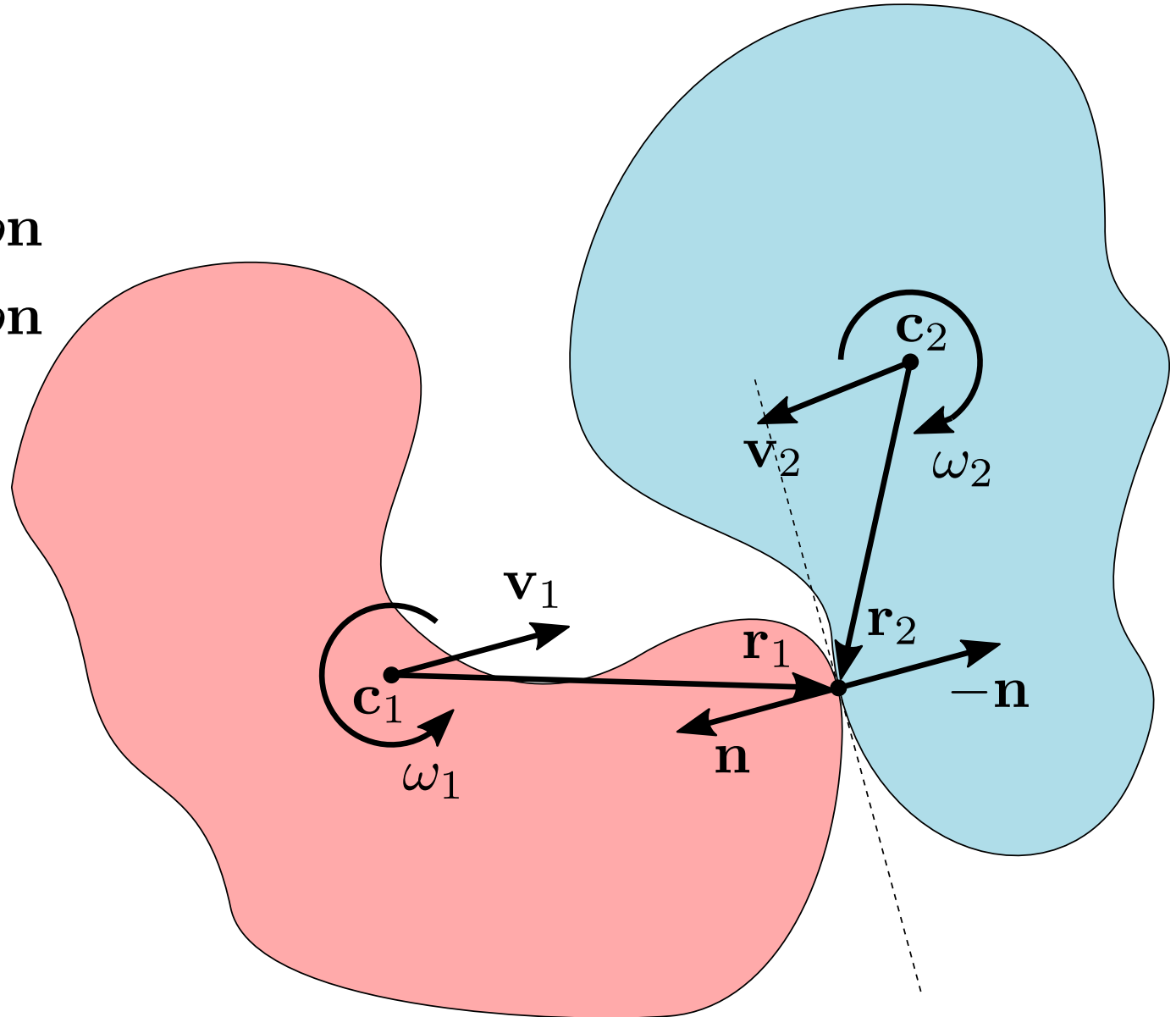
Elastischer/plastischer Stoß

$$m_1 \cdot \mathbf{v}'_1 = m_1 \cdot \mathbf{v}_1 + p\mathbf{n}$$

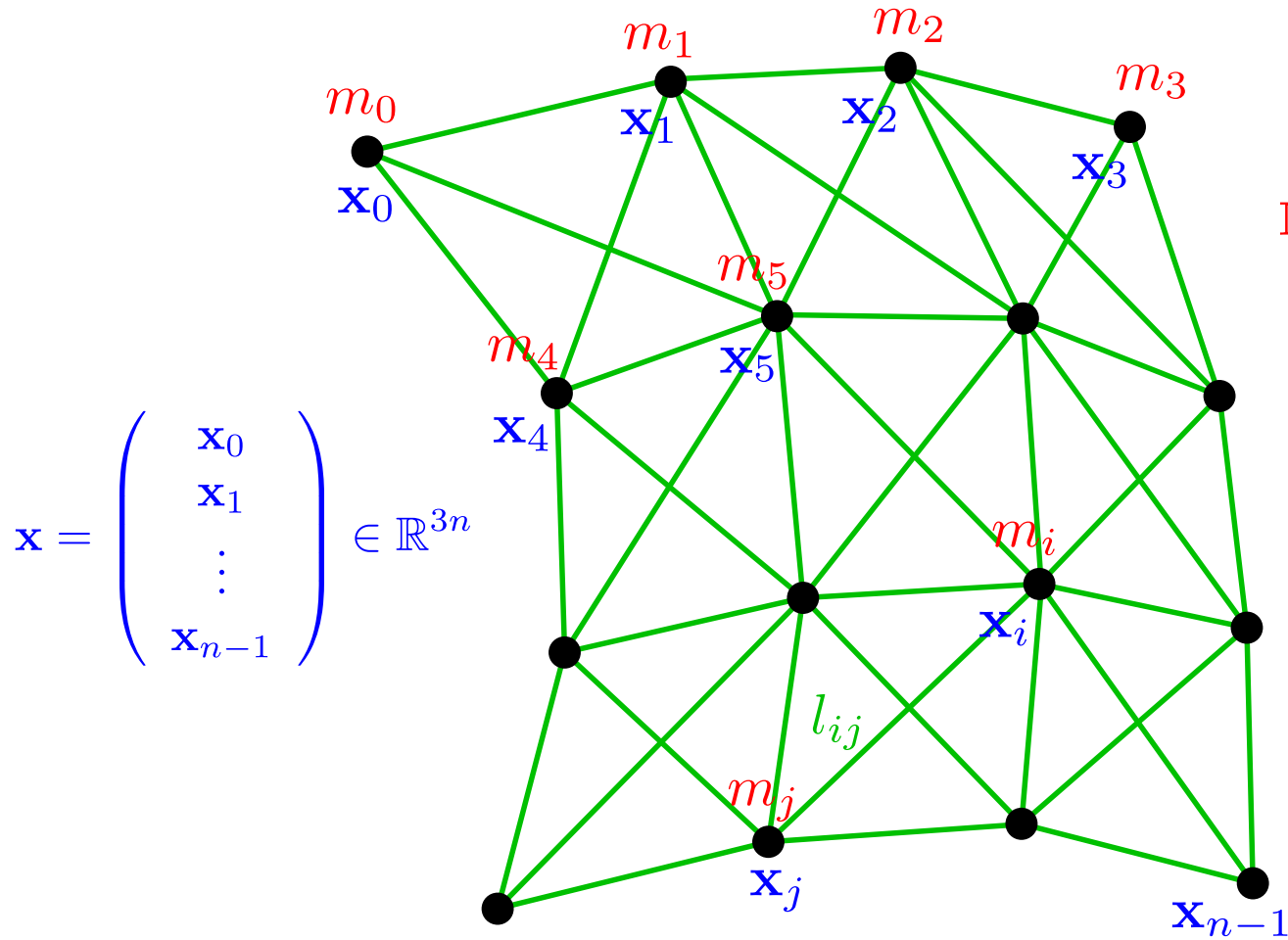
$$m_2 \cdot \mathbf{v}'_2 = m_2 \cdot \mathbf{v}_2 - p\mathbf{n}$$

$$\mathbf{I}_1 \cdot \boldsymbol{\omega}'_1 = \mathbf{I}_1 \cdot \boldsymbol{\omega}_1 + \mathbf{r}_1 \times p\mathbf{n}$$

$$\mathbf{I}_2 \cdot \boldsymbol{\omega}'_2 = \mathbf{I}_2 \cdot \boldsymbol{\omega}_2 - \mathbf{r}_2 \times p\mathbf{n}$$



Feder-Masse-Systeme



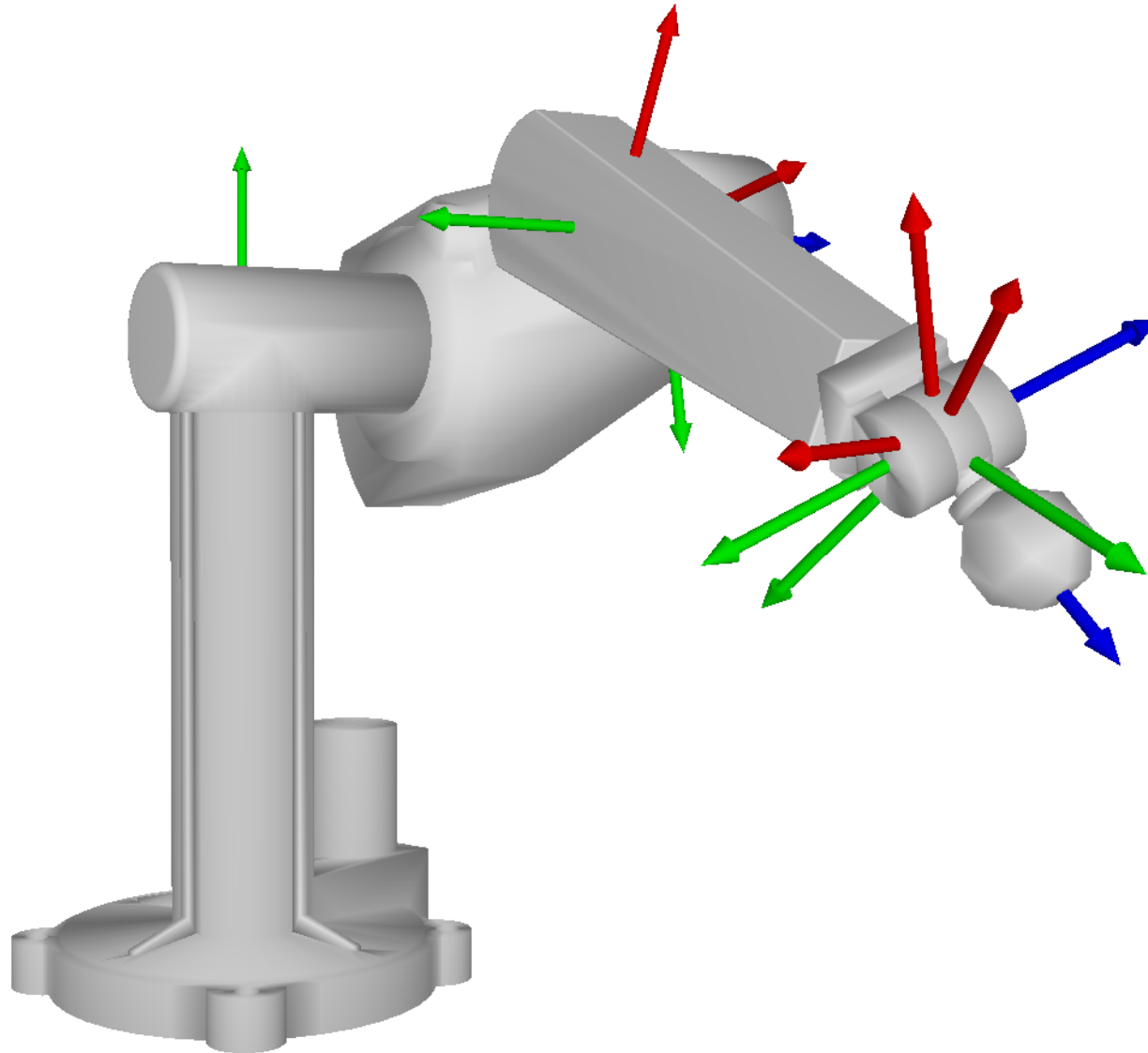
$$\mathbf{M} = \begin{pmatrix} m_0 \mathbf{E} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \\ \vdots & \ddots & m_i \mathbf{E} & \ddots \\ \mathbf{0} & \dots & \mathbf{0} & m_{n-1} \mathbf{E} \end{pmatrix} \in \mathbb{R}^{3n \times 3n}$$

$$\mathcal{E} = \{\{0, 1\}, \{0, 4\}, \{0, 5\}, \dots, \{i, j\}, \dots\}$$

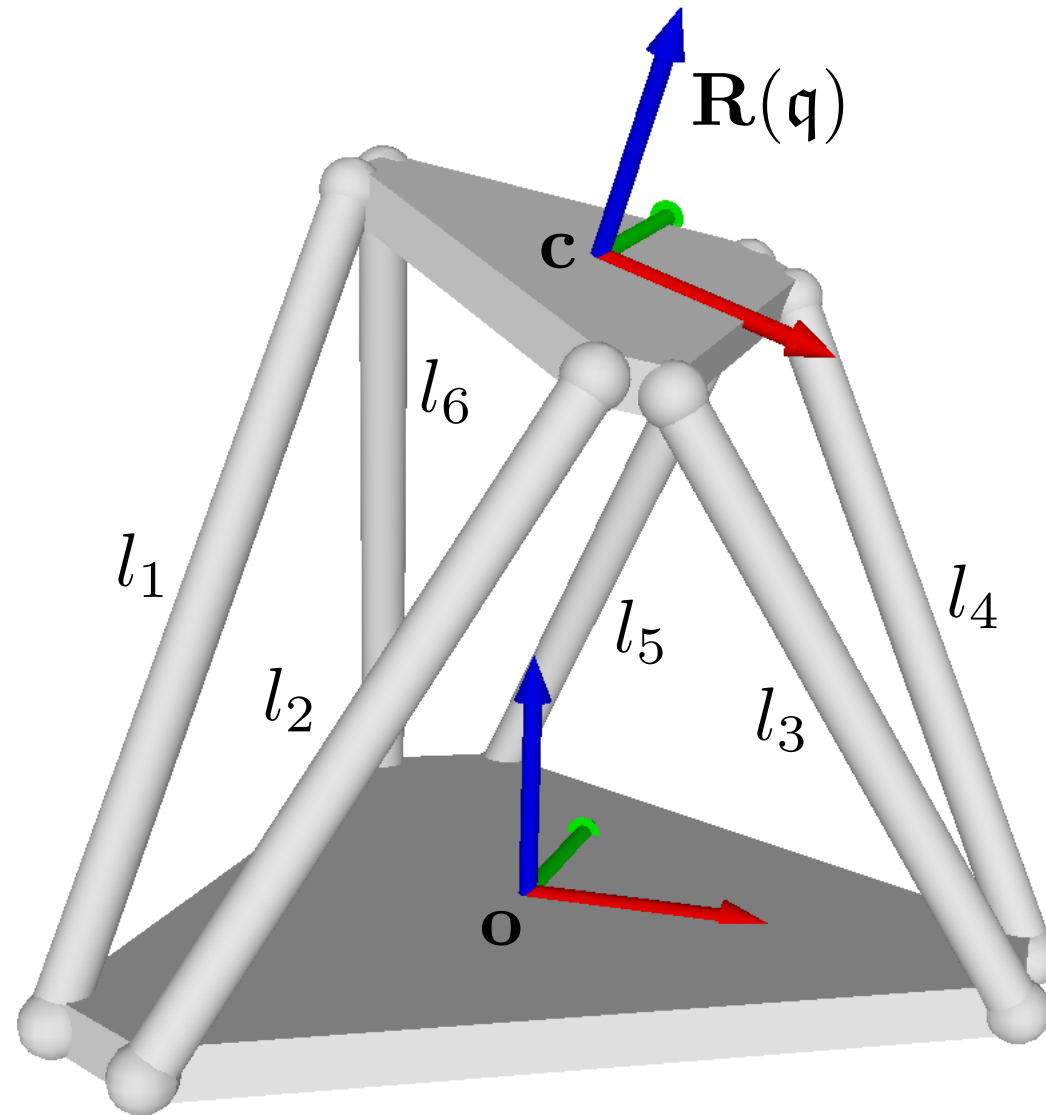
$$|\mathcal{E}| = k$$

$$\mathbf{C} = \begin{pmatrix} \vdots \\ |\mathbf{x}_i - \mathbf{x}_j| - l_{ij} \\ \vdots \end{pmatrix} \in \mathbb{R}^k$$

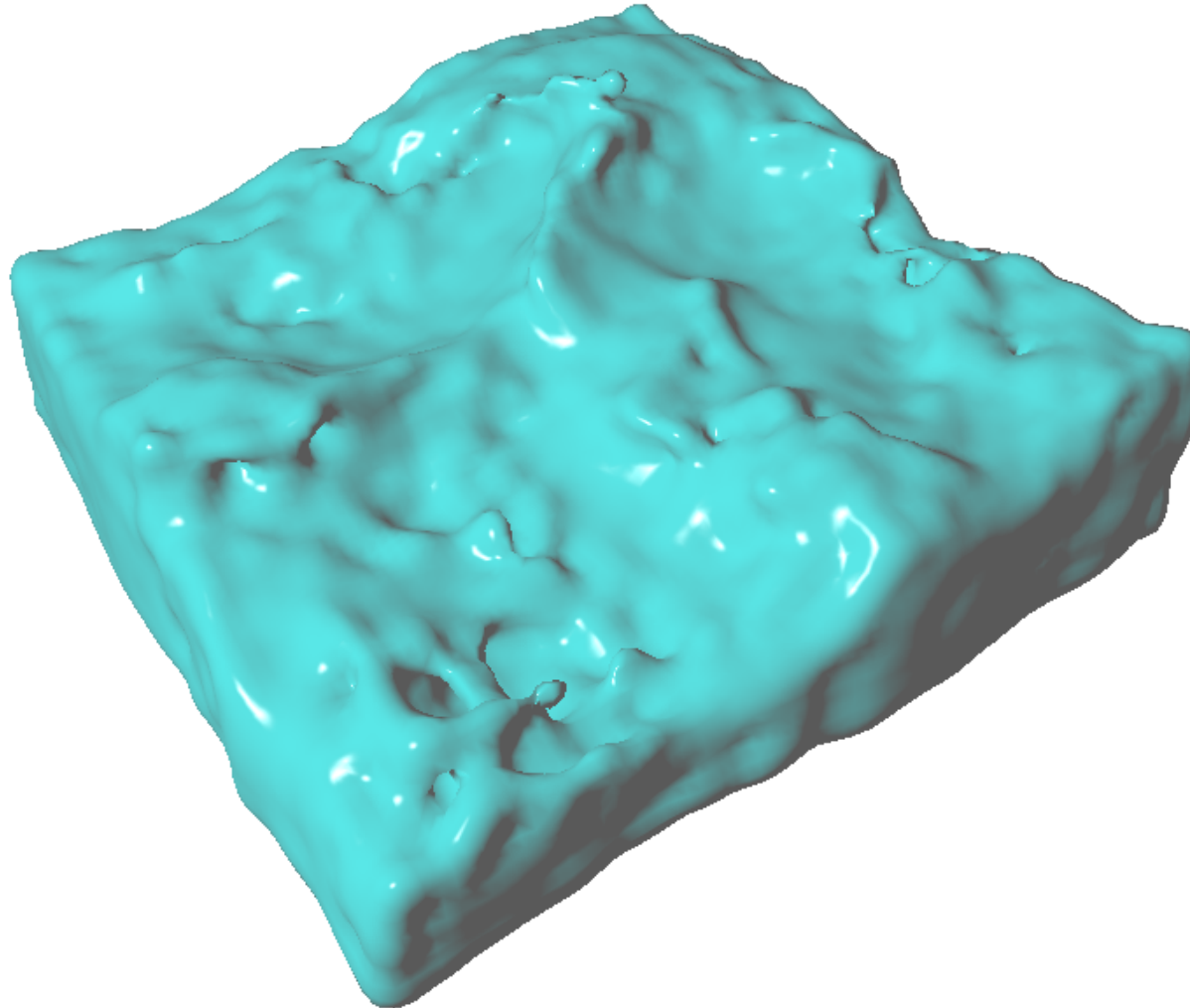
Koordinatensysteme der Gelenke



Stewart-Gough-Plattform

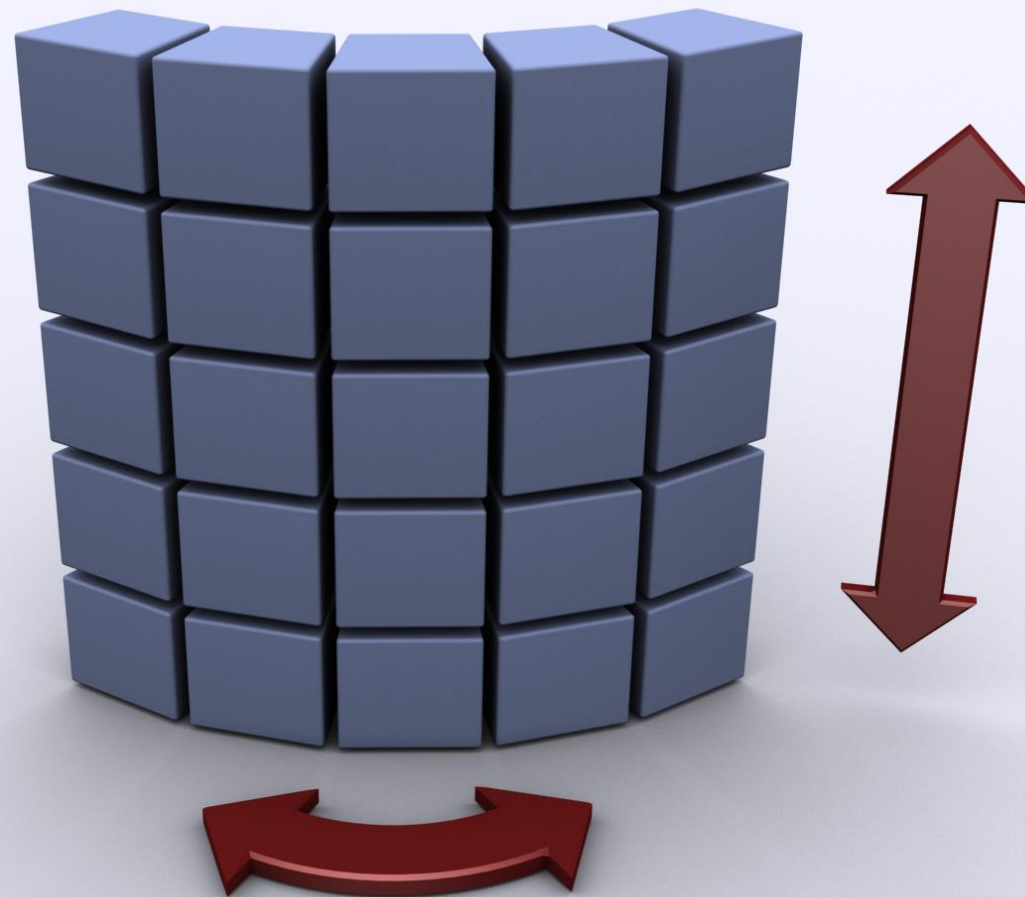


Bachelorarbeit Julius Schellenberg



Lehrangebot Visual Computing

Sommersemester 2022



Veranstaltungen **Visual Computing**

Vorlesungen

- Einführung in die Programmierung
- Modellierung II: Statistische Datenmodellierung

Seminare

- Seminar Visual Computing (zu Mod-2)
 - 1 Woche im Block in den Semesterferien
- Praktikum Visual Computing / Modellierung
 - 2 Wochen im Block in den Semesterferien

Weitere Infos

Lehrveranstaltungen

- Modellierung 2: Blended-learning
- (Demnächst) verlinkt von Jogustine aus

URLs

- Modellierung 2: luna.informatik.uni-mainz.de/mod2-22
- Seminar+Praktikum:
luna.informatik.uni-mainz.de/sem-prak-vc-22

(alles spätestens April 2022 online)

Vorlesungsreihe Modellierung

Reihe Modellierung

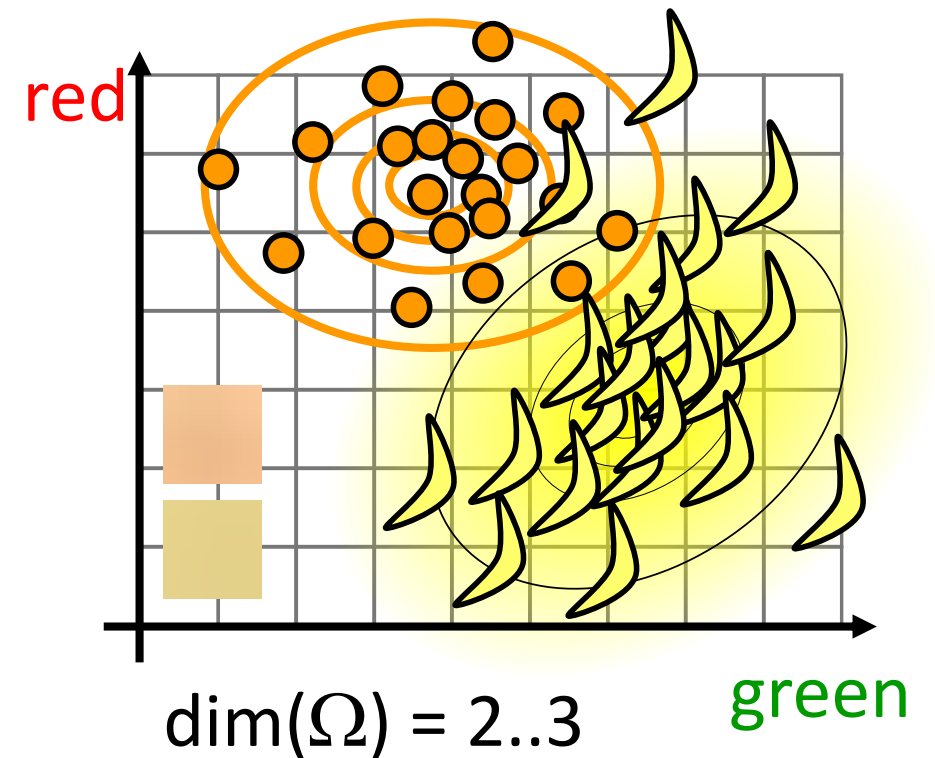


- Modellierung 1: Lineare Modellierung
 - Nächstes mal im Wintersemester 2022/23
- Modellierung 2: Statistische Datenmodellierung
 - Neu seit Sommersemester 2021
 - Voraussetzung: Modellierung 1
- Spezialveranstaltungen
 - Advanced Graphics (Rendering) – zuletzt WS21/22
 - Statistical Geometry Processing – zuletzt SoSem 17

Modellierung II

Statistische Modellierung

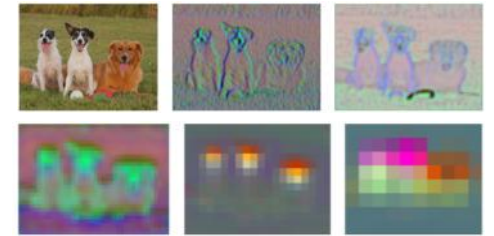
- Grundlagen
 - Kurzübersicht Machine Learning
 - Wiederholung lineare Modelle
 - Einführung Deep Learning
- Klassische Verfahren
 - Markov-Random Fields
 - Matrixfaktorisierung und Einbettungen, Kernels
- Verständnis tiefer Netze
 - Informationstheorie
 - Lineare & spektrale Analyse (alles ist PCA)
 - Approximationen im Limit (NTK etc.)
 - Ideen aus der statistischen Physik (Phasenübergänge, Mean-Field,...)
 - Empirische Ansätze und Ergebnisse (Lottery-Ticket etc.)



Seminar + Praktikum

Seminar Visual Computing

- Aktuelle Themen aus Mustererkennung, maschinellem Lernen und Graphik



Praktikum Visual Computing

- Mustererkennung, CG, ML & Modellierung



Planung

- **Format:** Im Block, nach dem Semester
- **Vorbesprechung:** wird noch bekannt gegeben

Webseite: luna.informatik.uni-mainz.de/sem-prak-vc-22/
(online ab April 2022)

AG Informationssysteme

infosys.informatik.uni-mainz.de

Lehrangebot, Sommer 2022

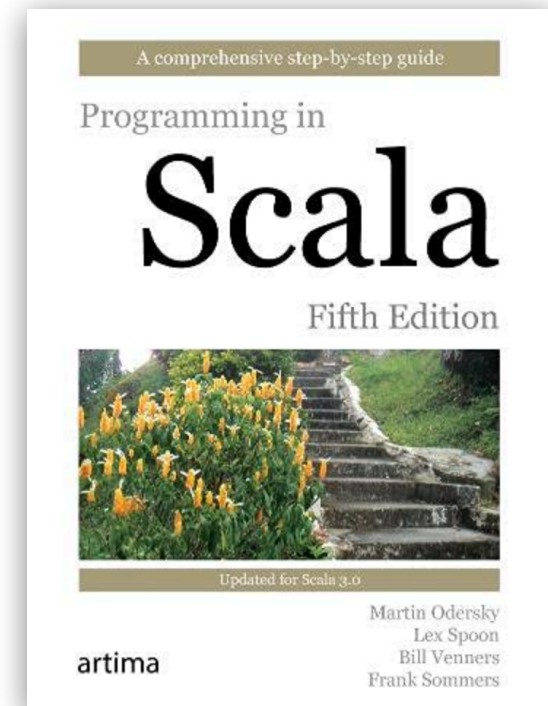


Einführung in die Softwareentwicklung

- Direkte Anknüpfung an "Einführung in die Programmierung".
- **Schwerpunkte:**
 - Entwurf und Umsetzung größerer Softwaresysteme.
 - Objektorientierte Programmierung.
 - Statische Typisierung.
 - (Ein bisschen) funktionale Programmierung.
 - Nebenläufige Programmierung.
 - Performance und Ausführung.
- Wir lernen diese Konzepte anhand von eines Projektes kennen, das wir schrittweise aufbauen, verbessern und erweitern.

Einführung in die Softwareentwicklung

- Wir nutzen die Programmiersprache Scala 3.
- Warum Scala?
 - Kompakte Syntax.
 - Sehr konsequente Objektorientierung.
 - Funktionale Programmierung möglich (und erwünscht).
 - Statische Typisierung mit Typinferenz.
 - Mächtiger 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 (Muschel)
 - Vorlesung/Übung auf Deutsch
- Voraussetzung:
 - Einführung in die Programmierung.
 - Kein Scala-Vorwissen nötig.
- 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)

Blockchain Seminar

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

- Voraussetzung:
 - Datenbanken 1.
 - Einführung in das Thema "Blockchain" zu Beginn der Veranstaltung.

- Kickoff-Meeting:
 - 21.04.2022 um 16:15 Uhr in Raum 04-522 zur Vorstellung der Themen.
 - Anmeldung in Jogustine bis zum 09.02.22 um 13:00 Uhr.

AG Schmidt: Lehrangebot: SoSe 22

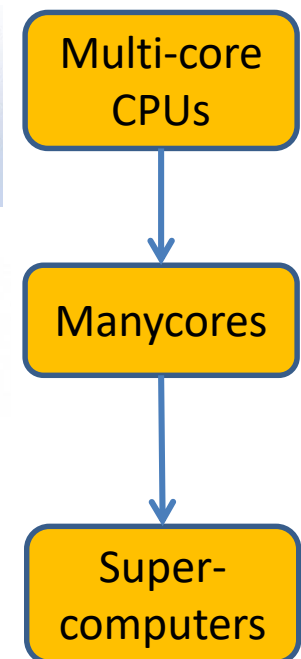
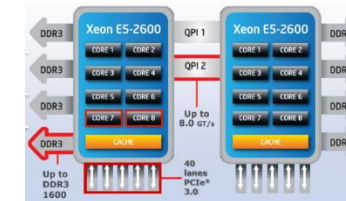
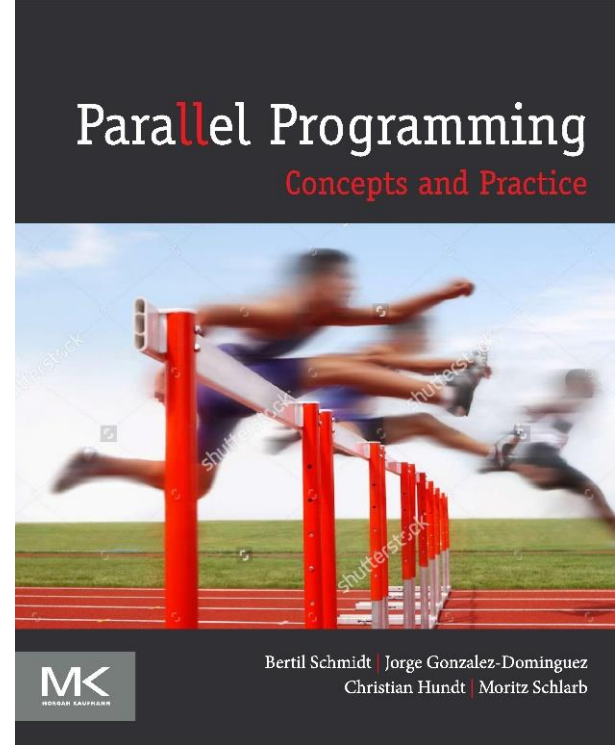
- **Vorlesungen**
 - Technische Informatik (*Grundvorlesung*)
 - High Performance Computing (HPC)
- **Praktikum**
 - Paralleles Programmieren (mit CUDA)
 - Gehört zu WiSe 21/22 (für Teilnehmer der „AccComp with GPUs“ Vorlesung)
- **Seminar**
 - High Performance Computing Seminar



Prof. Bertil
Schmidt

High Performance Computing (HPC)

- Background
 - Parallel Hardware
 - Parallel Software
- Shared Memory Programming
 - OpenMP
 - C++11-Multithreading
- Distributed Memory Programming
 - MPI
- Parallel Program Development
- Practical Parallel Programming Exercises



Block-Praktikum: Paralleles Programmieren (CUDA)

- **Hinweis:** Veranstaltung im WiSe 21/22
- Paralleles Programmieren mit CUDA
 - **28.3. – 8.4.2022 (plus 6.4. for presentations)**
 - Voraussetzung: Erfolgreiche Teilnahme an Accelerated Computing with GPUs (or PAA)



Seminarvorbesprechung: High Performance Computing Seminar

Prof. Bertil Schmidt

Seminar: Paralleles Rechnen – Themen

1. Time Series Mining at Petascale Performance
2. Speeding Up SpMV for Power-Law Graph Analytics by Enhancing Locality & Vect.
3. SparseP: Towards Efficient SpMV on Real Processing-In-Memory Systems
4. Variant Calling Parallelization on Processor-in-Memory Architecture
5. Accelerating Lattice QCD Multigrid on GPUs Using Fine-Grained Parallelization
6. MG-Join: A Scalable Join for Massively Parallel Multi-GPU Architectures
7. Better GPU Hash Tables
8. A SIMD algorithm for the detection of epistatic interactions of any order
9. A Compiler Framework for Optimizing Dynamic Parallelism on GPUs
10. Accelerating Large Scale de novo Metagenome Assembly Using GPUs
11. Harmonia: A High Throughput B+tree for GPUs
12. Systolic Computing on GPUs for Productive Performance
13. A Study of Single and Multi-device Synchronization Methods in Nvidia GPUs
14. Assessment of NVSHMEM for High Performance Computing
- 15. Selber vorgeschlagenes Thema** (muss aber von mir genehmigt werden)

Themenvergabe

- E-Mail mit **2** bevorzugten Themen (mit Präferenz) an Prof Schmidt **bis spätestens 6.2.2022**
- 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. 30-45min (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 5-7 Seiten)
 - **Abgabe bis spätestens vier Wochen nach dem Vortrag!**

Machine Learning

Vorlesung

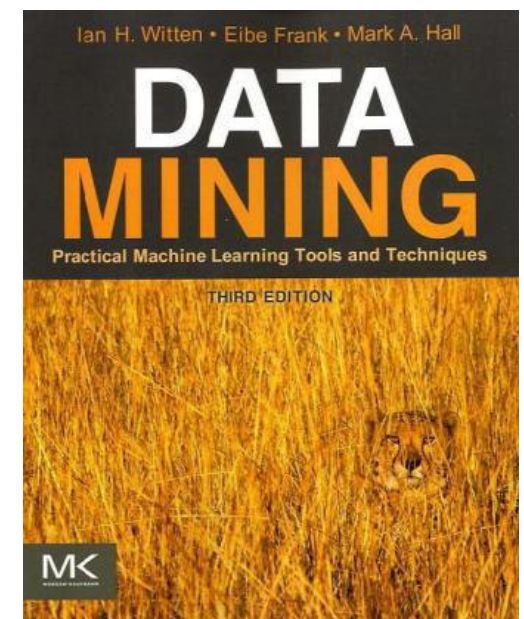
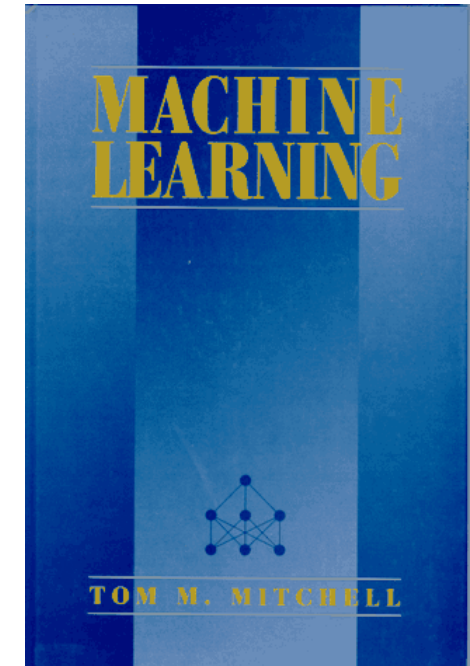
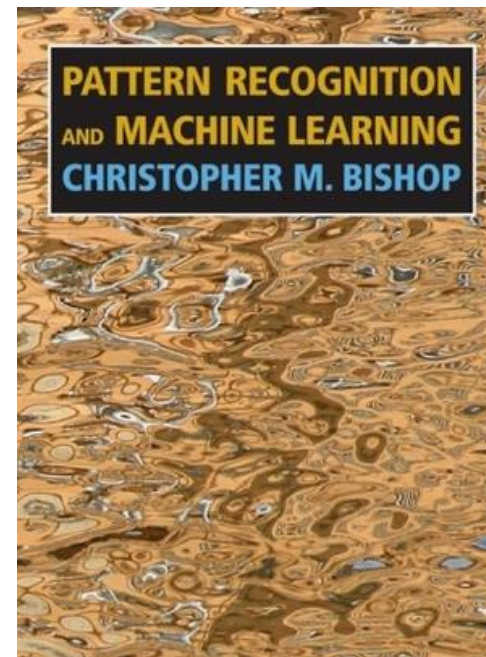
- Donnerstag 14 – 16 Uhr
- Material auf Englisch

Übung

- Zeit: TBA
- Auch: „Live Übungen“

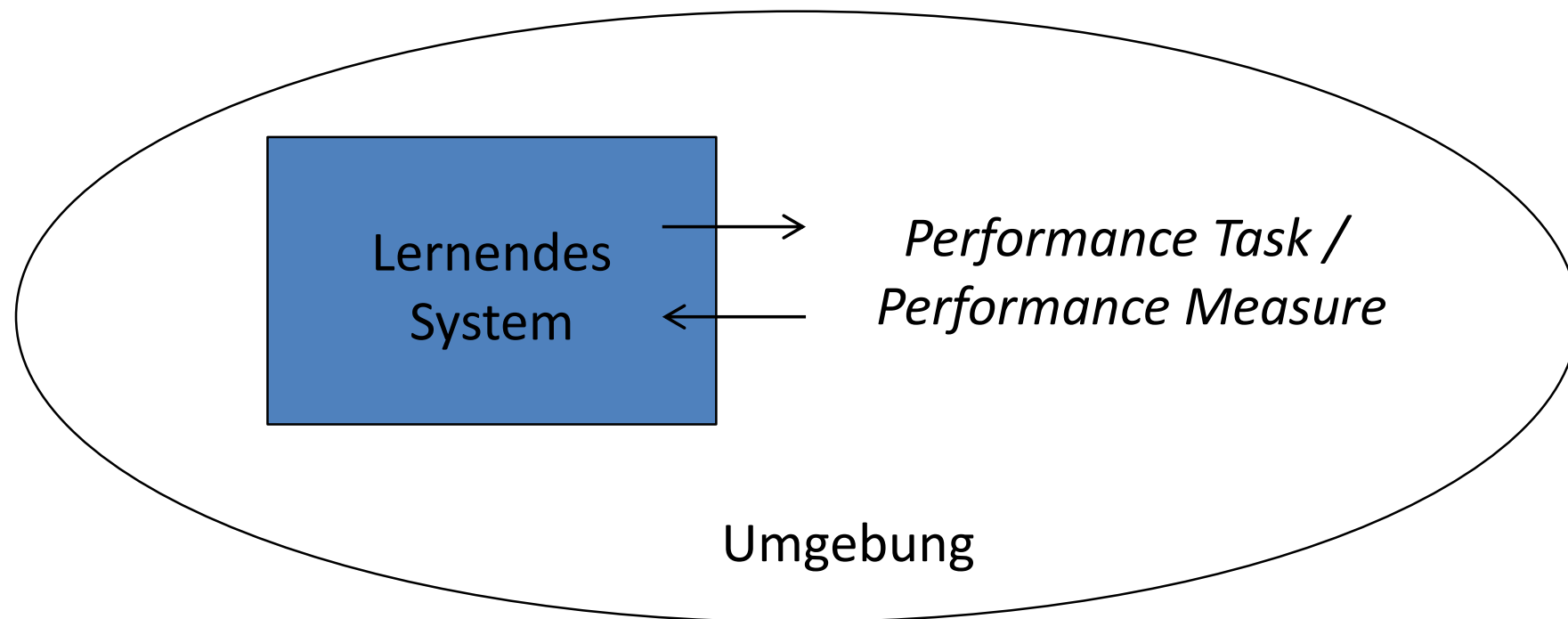
Prüfung

- schriftlich



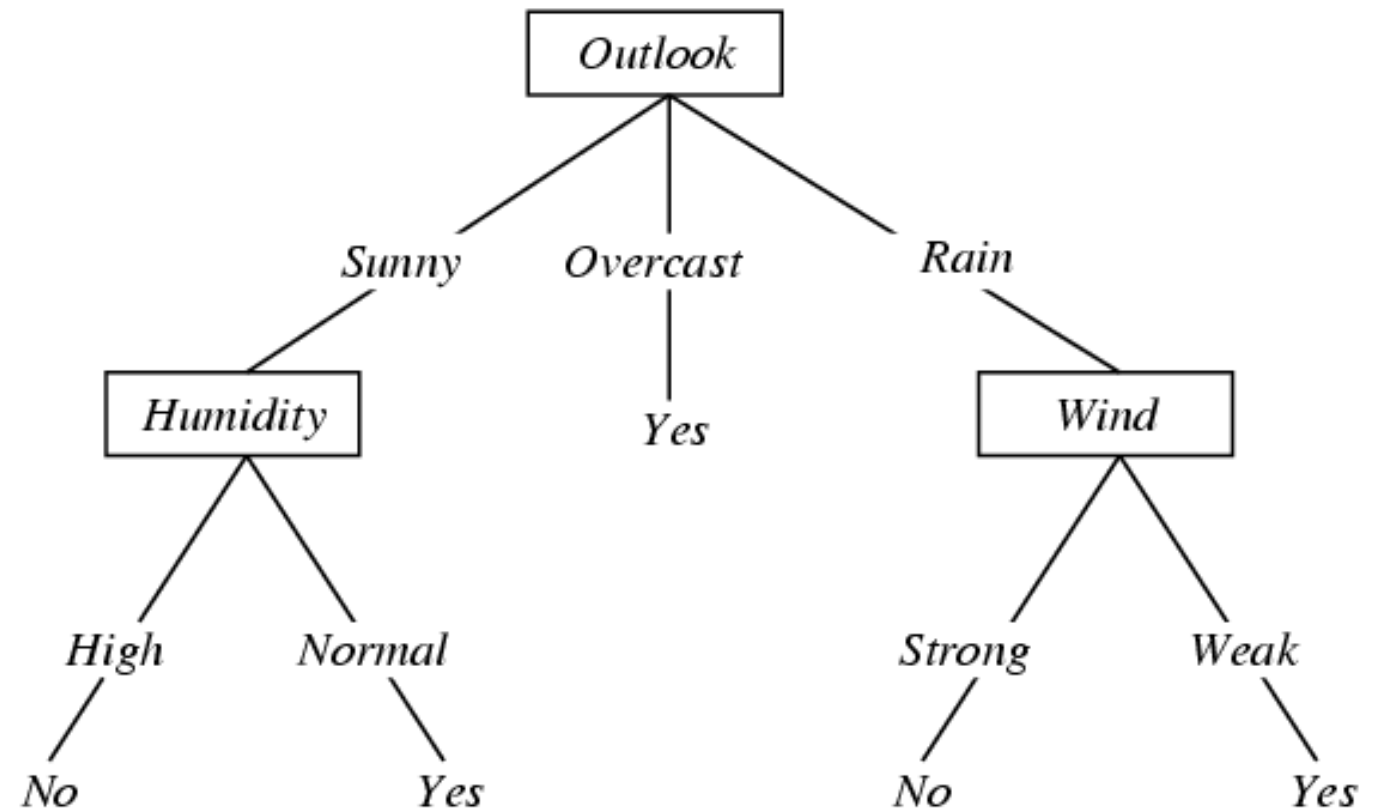
Machine Learning

- 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

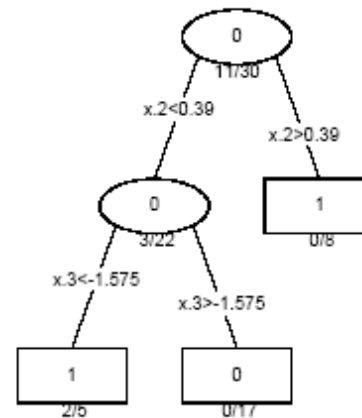
- Decision trees
- Ensemble methods
- Bayesian learning
- Linear models
- Neural networks and Deep Learning
- Instance-based learning
- SVMs and kernels



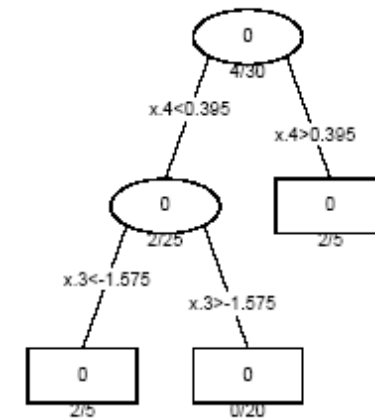
Machine Learning Course Overview

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

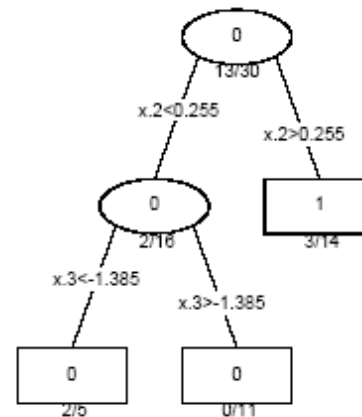
Bootstrap Tree 2



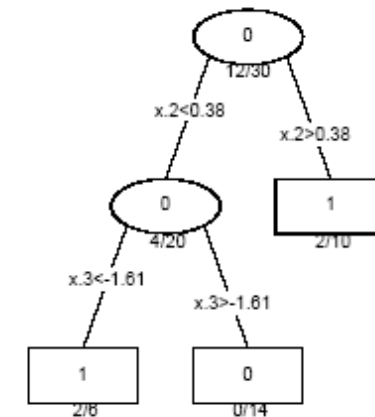
Bootstrap Tree 3



Bootstrap Tree 4

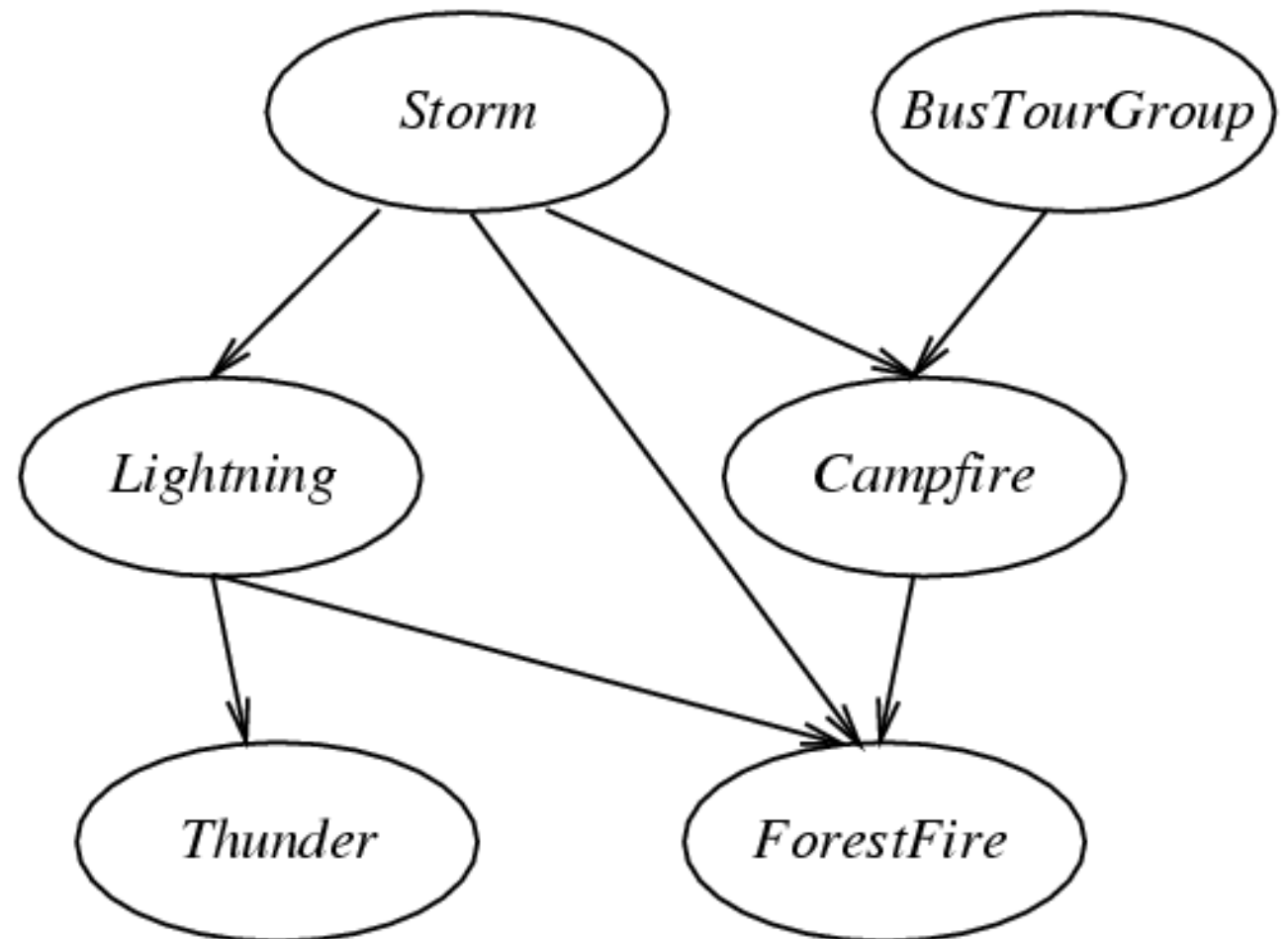


Bootstrap Tree 5



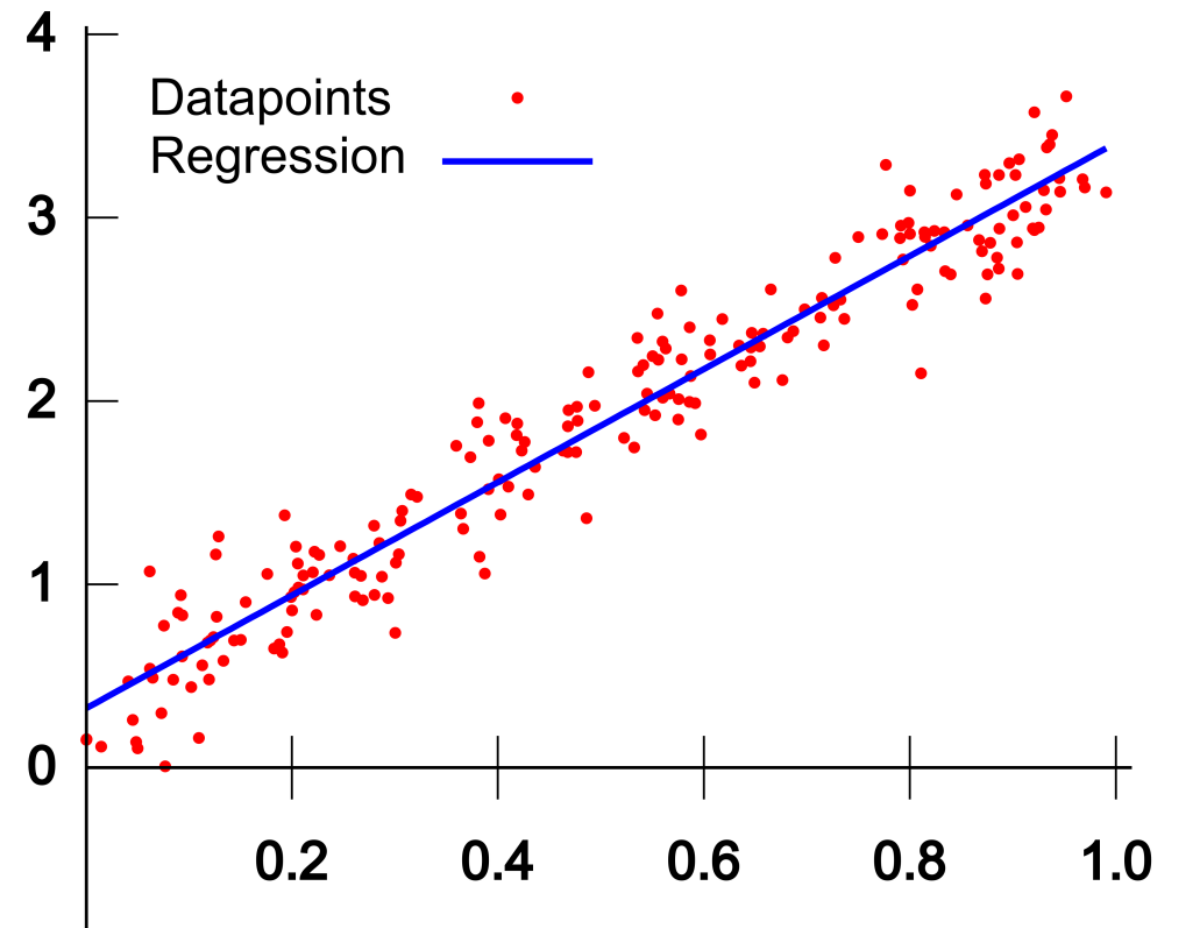
Machine Learning Course Overview

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



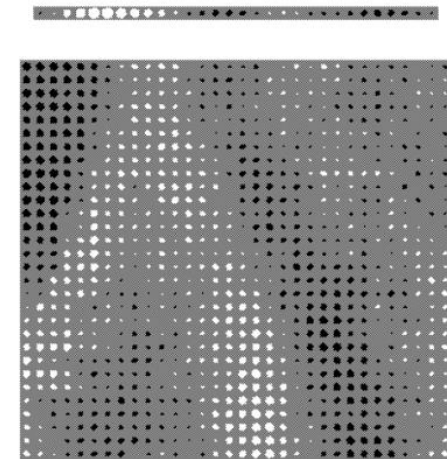
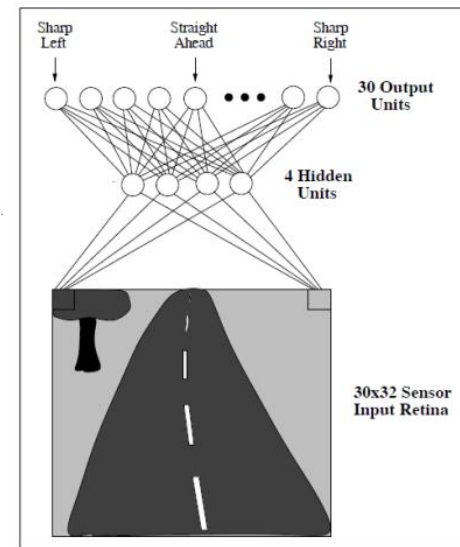
Machine Learning Course Overview

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



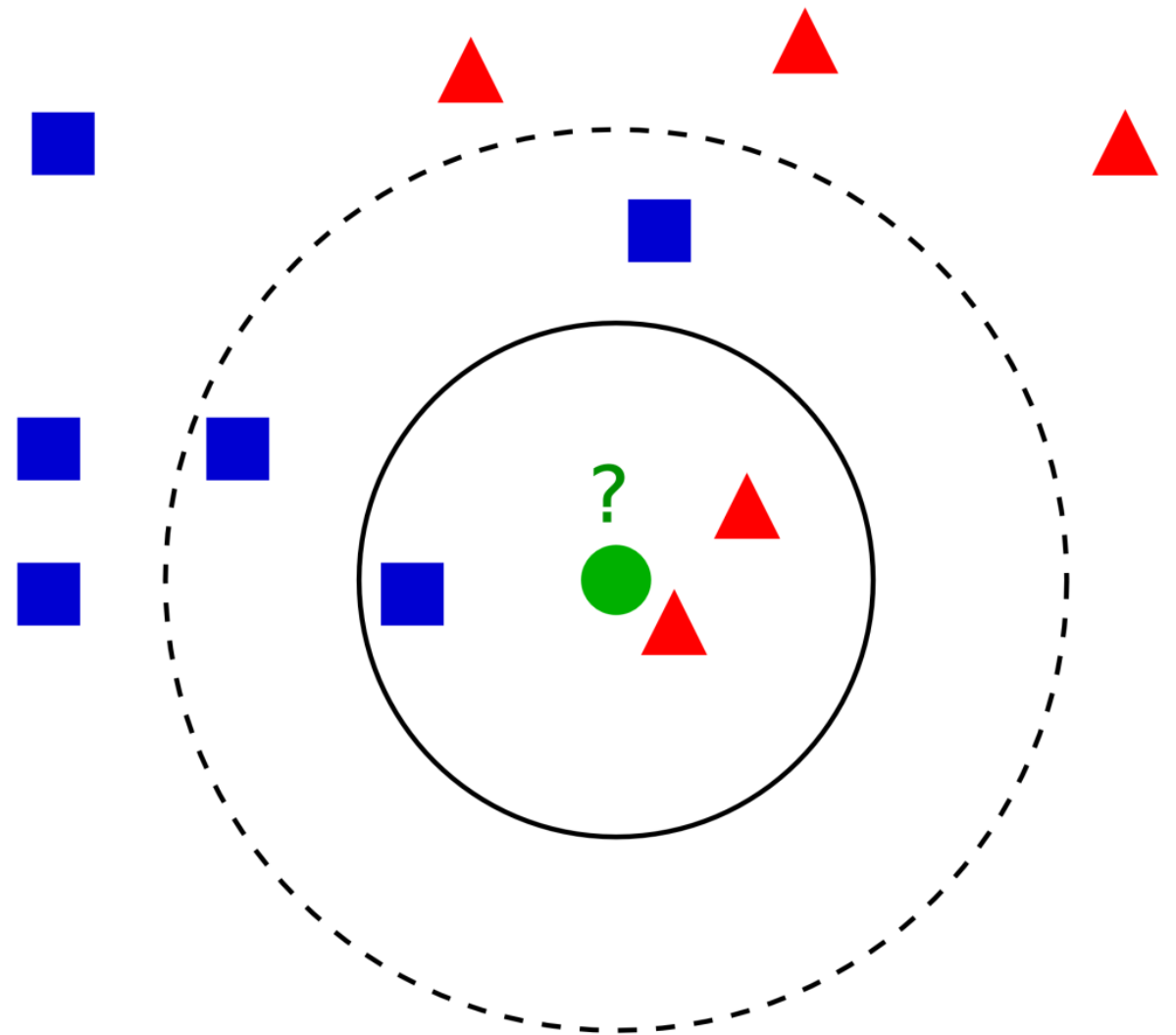
Machine Learning Course Overview

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



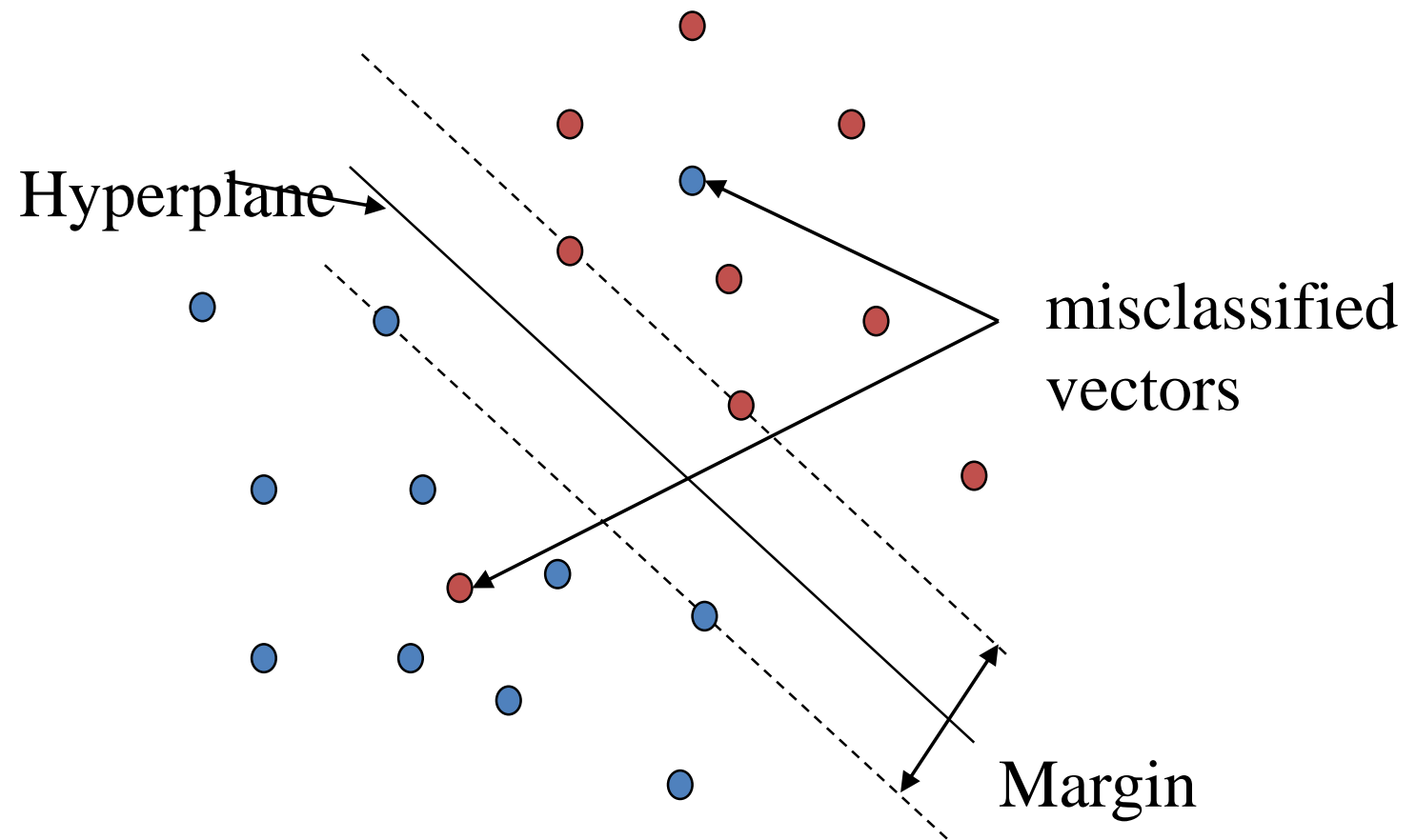
Machine Learning Course Overview

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



Machine Learning Course Overview

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



Einführung in die Computationale Logik

Vorlesung

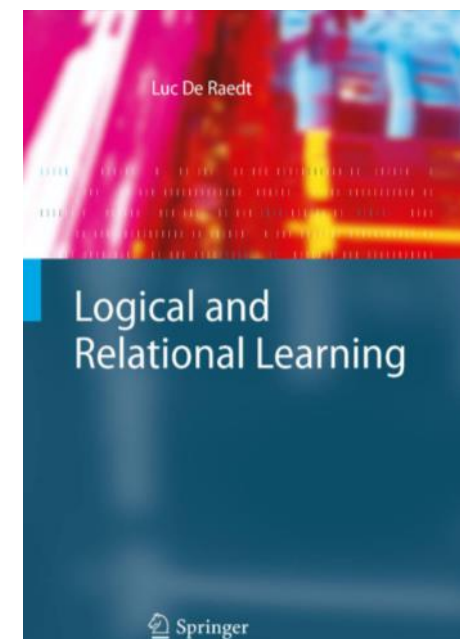
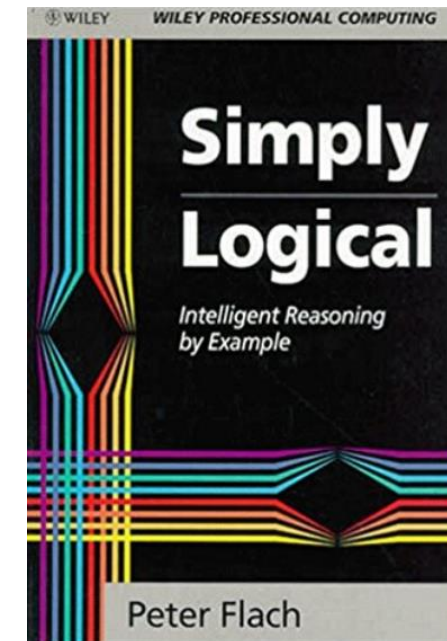
- Montag 16 – 18 Uhr
- Material auf Deutsch und Englisch

Übung

- Zeit: TBA

Prüfung

- 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

Data Mining Praktikum

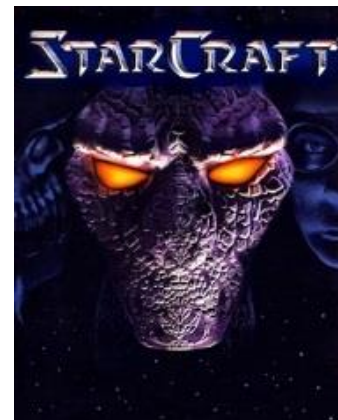


Ablauf

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



BUNDESLIGA



Data Mining Seminar und Praktikum

Vorbesprechung

- 2.02.2022 10:30-11:30
- Information wird noch verschickt

AG Scientific Computing and Bioinformatics

Lehrveranstaltungen im Sommersemester 2022

Prof. Dr. Andreas Hildebrandt

JOHANNES GUTENBERG
UNIVERSITÄT MAINZ



Strukturbasierte Bioinformatik

- Dozent: Andreas Hildebrandt
- Modus: Präsenzlehre
- 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

Vertiefungsseminar Bioinformatik

- Dozent: Andreas Hildebrandt
- Modus: Präsenzlehre
- Termin: Dienstags, 10–12 Uhr
- 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-Praktikum

- Dozent: Thomas Kemmer
- Modus: In Absprache mit den Teilnehmenden
- Umfang: 2 Wochen Vollzeit
- Termin: In Absprache mit den Teilnehmenden
- Sprache: Deutsch
- Inhalte: Projekte aus der Vorlesung im Wintersemester 2021/22
- Hinweise:
 - Vorheriger Besuch der Vorlesung „Big Data“ wird vorausgesetzt!

Kurse des Lehrstuhls Programmiersprachen

– Sommersemester 2022 –

Sebastian Erdweg

Lehrstuhl Programmiersprachen

<https://www.pl.informatik.uni-mainz.de/>

Source code



Compile

Binary



Deploy

Customer /
Production

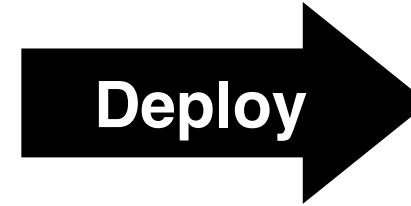


Failure

Source code



Binary



Customer /
Production



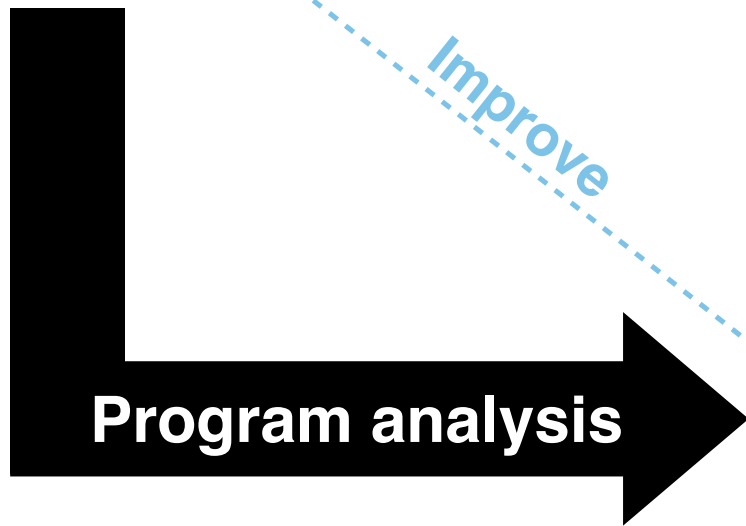
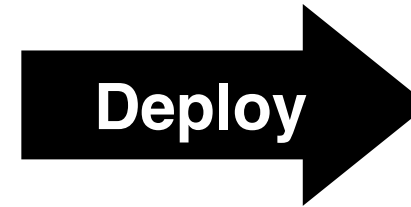
Research goal

Protect developers from security vulnerabilities, unsafe code, performance bottlenecks, and specification violations

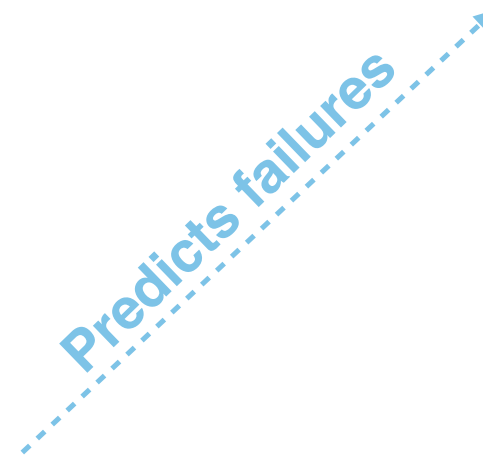
Source code

Binary

Customer /
Production



Automated
feedback



Improve

Predicts failures

Failure

British Airways

Criminal Theft of Customer Data, more information

To: Sebastian Erdweg,

Reply-To: British Airways



As you may be aware, from 22:58 BST 21 August 2018 until 21:45 BST 5 September 2018 inclusive, the personal and financial details of customers making or changing bookings at [ba.com](https://www.britishairways.com), and on our app were compromised. We're truly sorry, but you may have been affected.

The personal information compromised includes full name, billing address, email address and payment card information. This includes your card number, expiry date and CVV. Unfortunately this information could be used to conduct fraudulent transactions using your account. We recommend that you contact your bank or credit card provider immediately and follow their advice.

Kurs Programmanalysen:

- **Structural Operation Semantics**
- **Type systems and soundness**
- **Control-flow graphs**
- **Data-flow analysis**
- **Abstract interpretation**

Grundstudium:

- Programmiersprachen

Hauptstudium:

- **Programmanalyse: Vorlesung, Praktikum
Seminar im nächsten Semester**
- **Sprach- und Compilerbau: Seminar
Voraussetzung: Vorlesung**

Achtung: Forschungssemester im WS 2022/23

Sebastian Erdweg

Lehrstuhl Programmiersprachen

<https://www.pl.informatik.uni-mainz.de/>

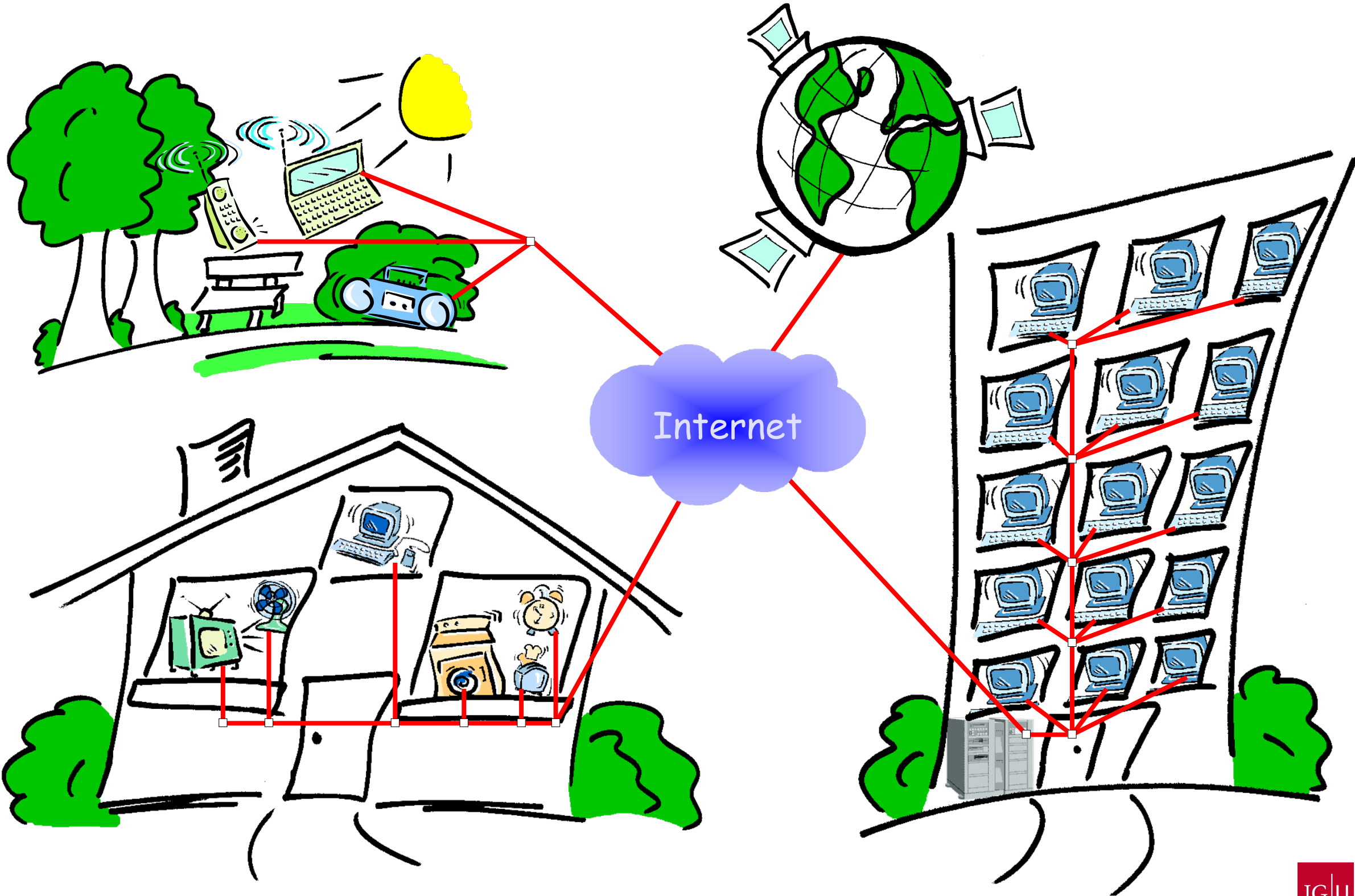
Lehrangebot Sommersemester 2022

André Brinkmann

Institut für Informatik

Johannes Gutenberg-Universität Mainz

Kommunikationsnetze



AG Brinkmann: Lehrangebot SS 2022

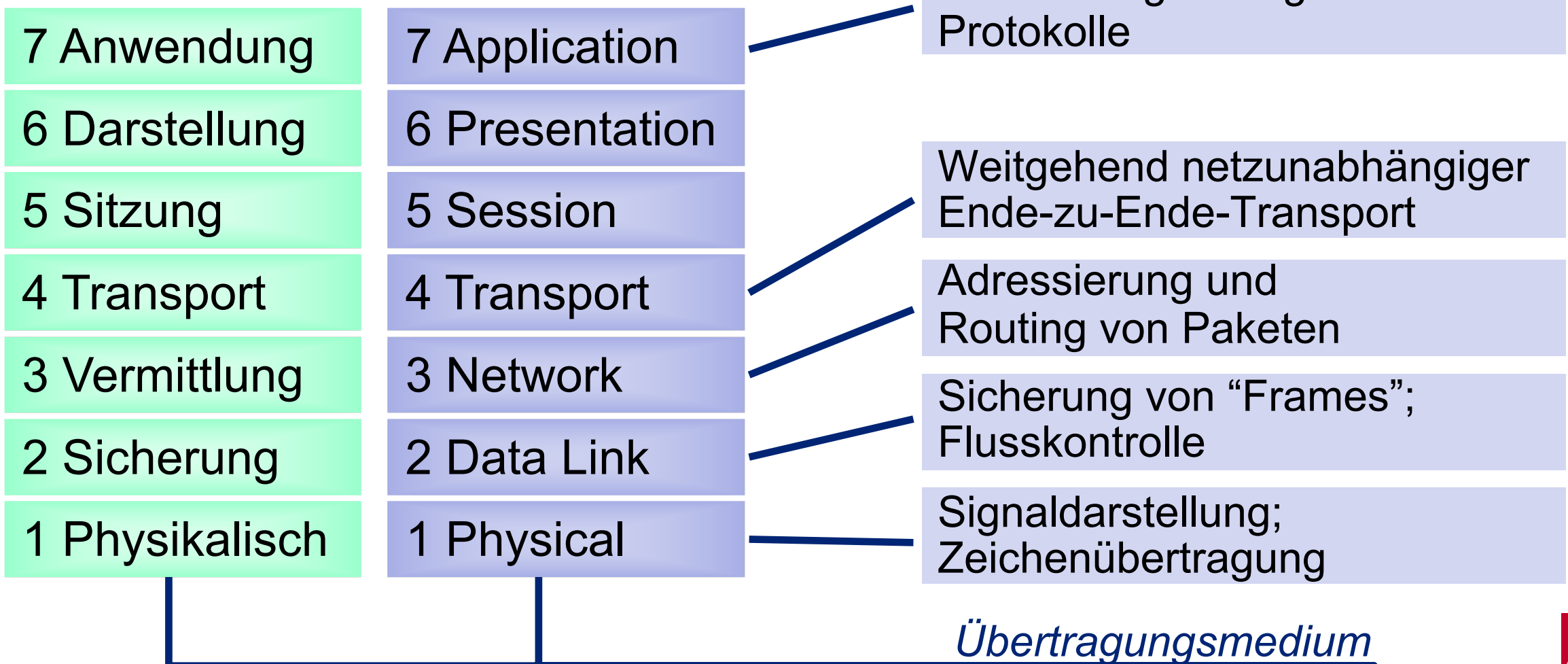
- Vorlesung:
 - Kommunikationsnetze (Bachelor)
- Seminar
 - Betriebssysteme (Bachelor und Master)
- Praktikum
 - Advanced Topics in Operating Systems (Master)

Das ISO/OSI-Referenzmodell

Prinzip:

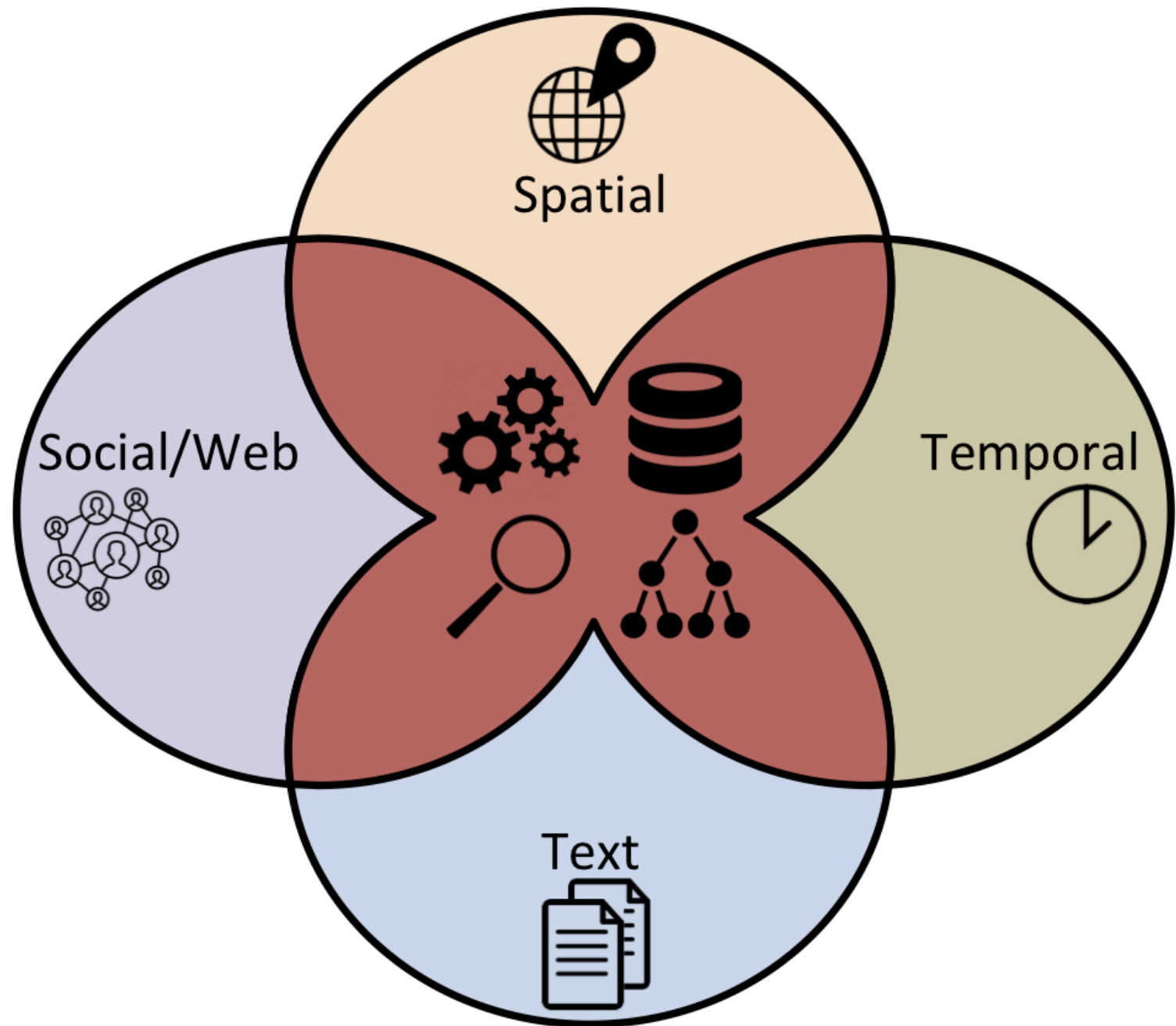
Bewältigung der Komplexität einer Kommunikation (also der Einzelheiten, auf die man achten muss) durch Schichten

7 Schichten (Ebenen, Layer)



Welcome to SoSe 2022

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



Datenbanken I



- **Vorlesung**
 - Physical
 - Tuesdays, 14:00 – 16:00, Rm 0-512 (N2)
 - Digital
 - Panopto pre-recorded videos
 - MS Teams live chat on Tuesdays, 14:00 – 16:00
- **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.
- **Website**
 - TBA

Datenbanken I

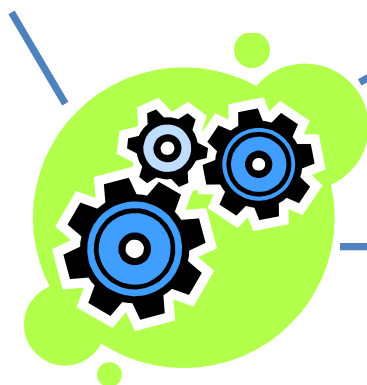


- **Course objectives**

- 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



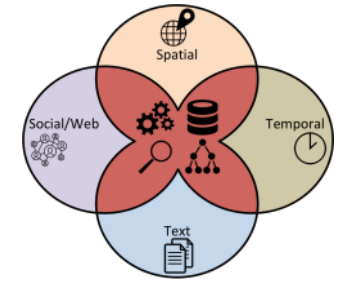
- **Termin**
 - TBD
 - Two or three weeks before the start of WiSe 2022/23
- **Inhalte**
 - Hands-on experience, practical aspects
 - Building a functional and practical UI on top of a DBMS
 - Web programming
- **Unterrichtssprache**
 - English/German
- **Studiengang**
 - B. Sc. / M. Ed.
- **Website**
 - TBA

Datenbanken II - Seminar



- **Termin**
 - TBD
 - Meetings and chat 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.
- **Website**
 - TBA

AG Data Management



For more information about the group, visit
<https://datamanagement.cs.uni-mainz.de/>



Projektmanagement

Dr. Sebastian Hoffmann, PMP®

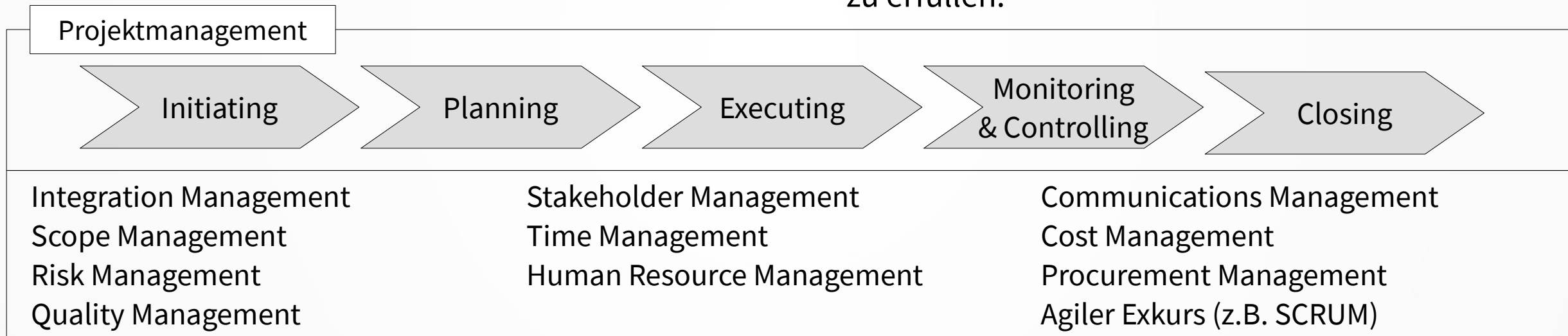
Freitags 9:00 – 12:00, MS Teams

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.



- 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



- Vorlesung: Formale Sprachen und Berechenbarkeit
F. Schmid, Di 10-12
- Vorlesung: Algorithmen und Techniken der Optimierung
F. Fischer, Di 12-14
- Vorlesung: Fortgeschrittene Algorithmen (inverted classroom)
M. Blumenstock, Mi 10-12
- Seminar: Algorithmen, semesterbegleitend
M. Blumenstock, Mo 10-12
- Praktikum: Approaching Programming Contests, semesterbegleit.
D. Mosca

Vorlesung: Fortgeschrittene Algorithmen

Formales:

- Inverted Classroom in Panopto mit Quizzes zum Video
- Nachbesprechungsstunde: Mi 10-12, M. Blumenstock
- Ort: 03-428, Panopto, Infos in Teams (Teamcode: pm8hx9j)
- Organisation des Übungsbetriebs: M. Blumenstock
- Übungstermin(e): vorläufig Montag 12-14 Uhr geplant

Inhalt:

- Fortgeschrittene MST- und Flussalgorithmen
- Online-Algorithmen
- Randomisierte Algorithmen
- Approximationsalgorithmen
- Lineare und ganzzahlige lineare Programmierung
- Verteilte Algorithmen

- Zeit und Ort: semesterbegleitend, Montag 10-12, 04-426
- Betreuung: Markus Blumenstock
- Voraussetzungen: mindestens DSEA, idealerweise FA
- Teams-Beitrittscode: zy5gkor
- Scheinvergabe:
 - Vortrag von ca. 40 min
 - Ausarbeitung in LaTeX
- Inhalte:
 - Vorstellung eines Algorithmus/Datenstruktur, ähnlich denen aus DSEA
 - Es können auch einige Themen mit Bezug zu Spieltheorie, Berechenbarkeitstheorie und Optimierung angeboten werden

Approaching Programming Contests (Praktikum)

- Zeit und Ort: wöchentlich, wird mit den Teilnehmern vereinbart
- Betreuung: Domenico Mosca (vermutlich)
- 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

Erwünscht:

- Bereitschaft, am NWERC teilzunehmen

FPGA-Programmierung

Duc Bao Ta, Sommersemester 2022

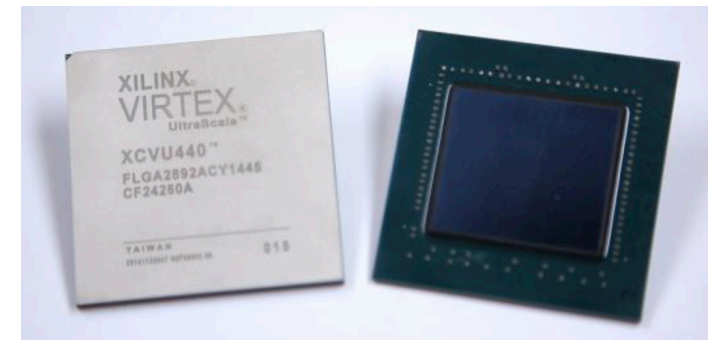
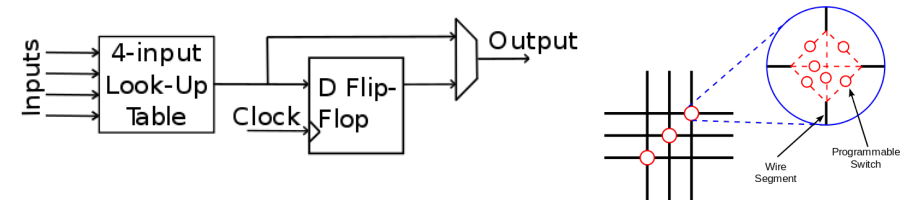
Vorlesung mit praktischen Übungen: MI 10-12 plus Übungen



26 Jan 2022, Vorstellung des Lehrangebots Informatik

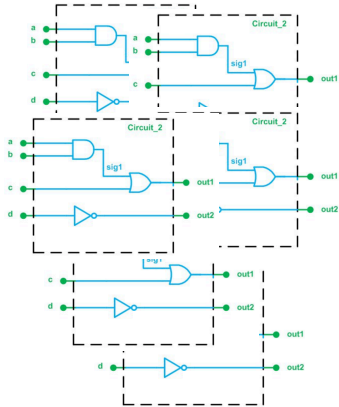
FPGAs

- FPGA = **Field Programmable Gate Array** = Programmierbare Logikbausteine
 - Feld aus **Basislogikblock** ($\sim O(1M)$) mit Verknüpfungsmatrix
 - I/O-Blöcke, Taktgeneratoren, RAM, DSP etc.
- Einsatz in **schnellen elektronischen Schaltungen**: flexibel konfigurierbar (“programmierbar”)
- **Kostengünstig**, u.A. als Entwicklungsplattform für ASICs (“Computer-Chips” mit fester Funktion)
- Programmierbar in **Hardwarebeschreibungssprache**: Notationen, die hardwarespezifisch sind (z.B. Netze, state machines etc.), Übersetzung in Verknüpfungsmatrix



FPGAs in a nutshell

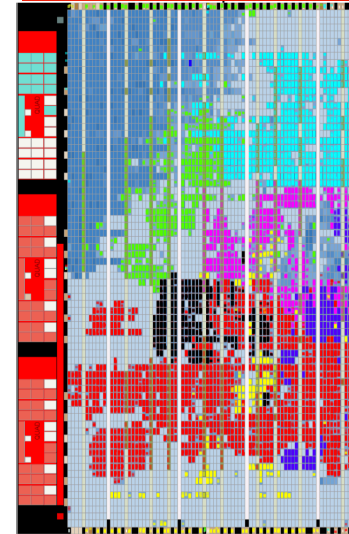
Elektronische Schaltung



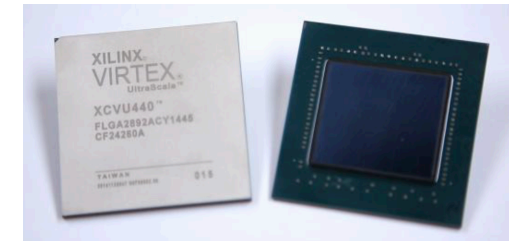
Programmierung in VHDL

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity circuit_2 is
5      Port ( a : in  STD_LOGIC;
6            b : in  STD_LOGIC;
7            c : in  STD_LOGIC;
8            d : in  STD_LOGIC;
9            out1 : out STD_LOGIC;
10           out2 : out STD_LOGIC);
11 end circuit_2;
12
13 architecture Behavioral of circuit_2 is
14     signal sig1: std_logic;
15 begin
16     sig1 <= ( a and b );
17     out1 <= ( sig1 or c );
18     out2 <= (not d);
19 end Behavioral;
```

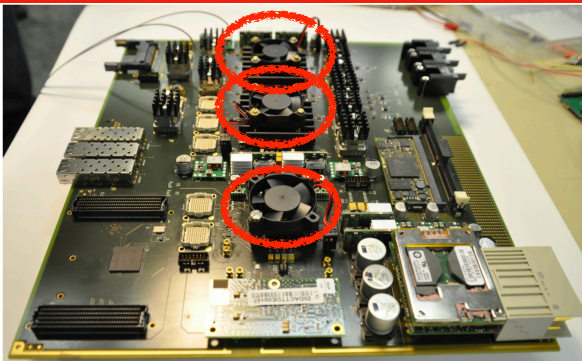
Place & Routing



Konfiguration FPGA firmware



Einsatz in elektronischen Geräten



ATLAS L1Calo Trigger, ETAP Mainz

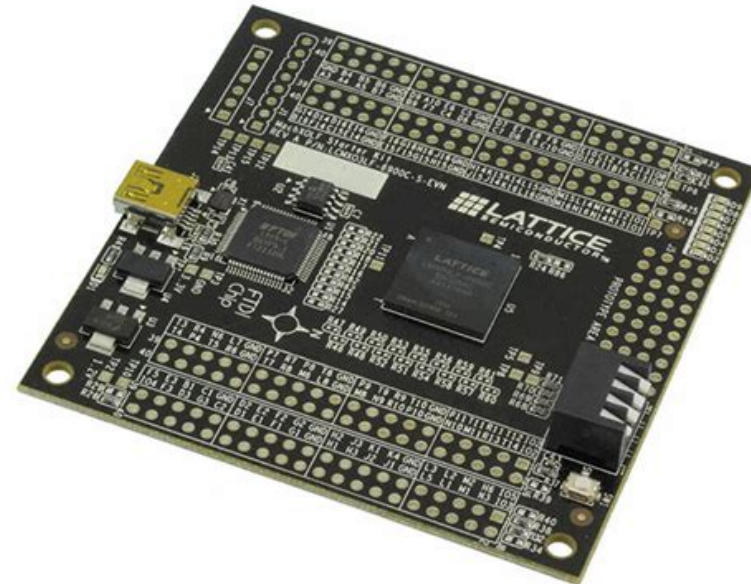
• Einsatzbereiche

- Forschung (Physik: Teilchenphysikexperimente CERN/Fermilab/KeK)
- Automobilindustrie/Telekommunikation (update-/anpassungsfähig)
- Günstige FPGA für den Hobbybereich (~O(10€)/Stück)

Was erwartet Sie in der Vorlesung?

- **Handwerk der Firmware-Programmierung**
 - **Verständnis** kombinatorischer und getakteter digitaler Schaltungen
 - **VHDL schreiben** und **lesen** können
 - Ein Projekt **simulieren** und im **FPGA implizieren** können
 - Idiomatisches VHDL schreiben können
 - Schaltungen in Bezug auf Geschwindigkeit und Größe **optimieren**
 - **Kommunikation** in der Außenwelt
- **Praktischer Teil (Übungen)**
 - Schriftliche **Programmieraufgaben** können vor Ort (Zuhause) auf **echten FPGAs** ausprobiert werden

- **Verständnis der Algorithmen**
 - Algorithmen zur **Logikvereinfachung und Synthese**
 - **Place & Routing** Algorithmen
 - **Simulation:** Anwendung und Grenzen
 - **Massiv parallele** Implementation



Vorlesungsprogramm

- **Was sind FPGA? Was ist VHDL?**
- **Elektronik**
 - CMOS, Gatter, kombinatorische Logik
 - Register, Takt, Pipelines
- **Programmierung in VHDL**
 - Design patterns
 - Synchronisation
 - Speicher und vordefinierte Logikblöcke
- **Praxis**
 - Simulation -> Synthese -> Place&Route
 - Busse und andere Interfaces
 - Anwendungsbeispiele

Computergrafik I Seminar

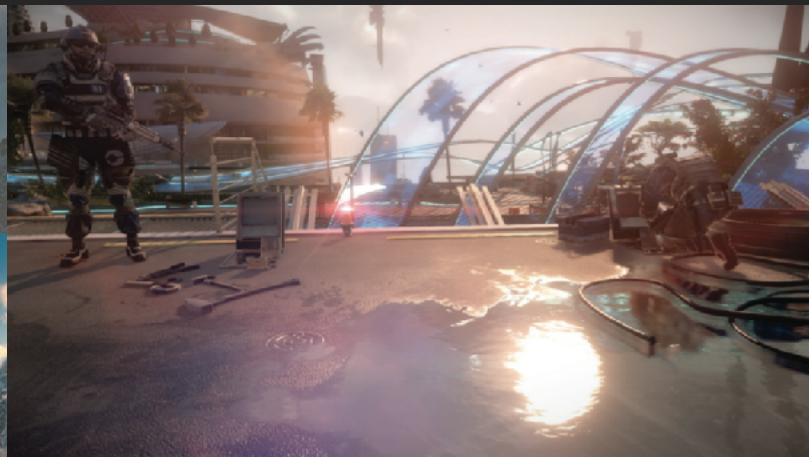
Arbeitsgruppe: Computational Geometry (Prof. Schömer)

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:



Computergrafik I Praktikum

Arbeitsgruppe: Computational Geometry (Prof. Schömer)

in der vorlesungsfreien Zeit nach dem Semester, online

Thema aus Computergrafik II

Voraussetzung: erfolgreiche Teilnahme an Computergrafik I
oder Computergrafik II

Einführung in die Computergrafik mit Game Engines Arbeitsgruppe: Computational Geometry (Prof. Schömer)

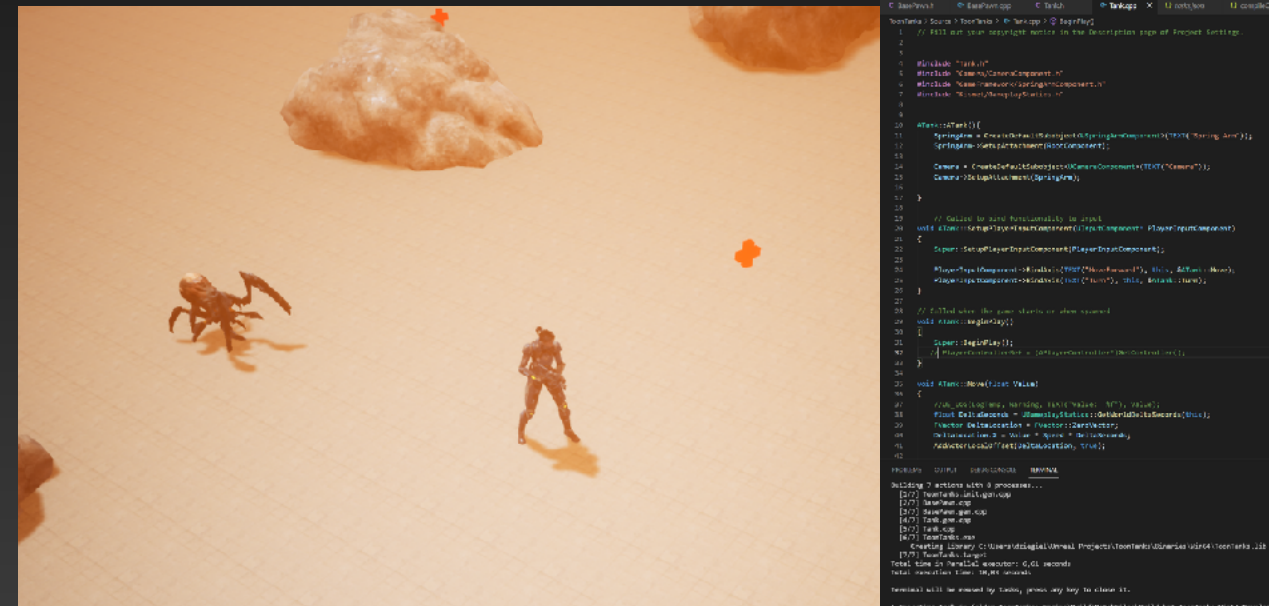
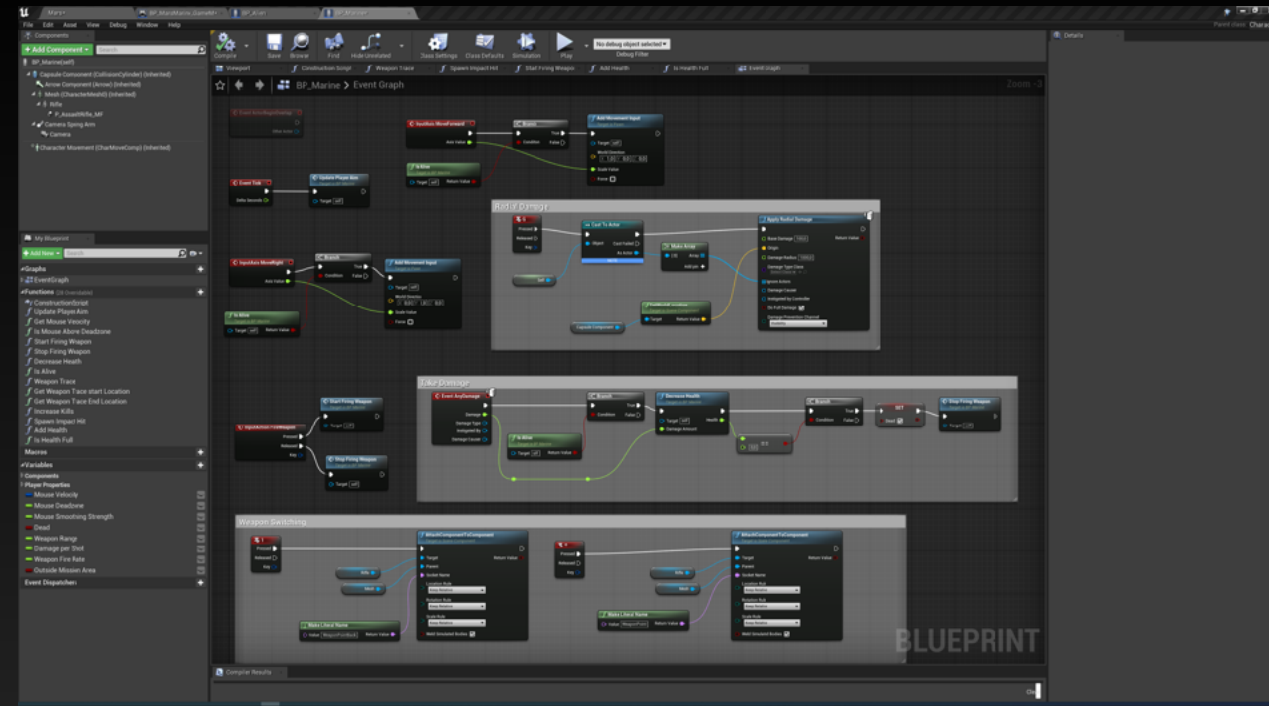
Dozenten: Rainer Erbes, Andreas von Dziegielewski

Do 10:15-11:45, online

Praxis: 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) und (falls zeitlich möglich) UE4 C++ API

Theorie: Ausgewählte Themen aus Computergrafik, Software Patterns und AI, die Anwendung in der UE4 finden.

Organisation: Bachelor-Veranstaltung, Online-Veranstaltung, kombinierbar als Modul Computergrafik mit Computergrafik Seminar (WS22/23) und Computergrafik-Praktikum (vorlesungsfreie Zeit nach SS22)



Modellierung mit partiellen Differentialgleichungen

Peter Spichtinger

Institut für Physik der Atmosphäre, JGU Mainz

26. Januar 2022

3 Säulen der Naturwissenschaften

- ▶ Theorie/Modell
- ▶ Experiment
- ▶ (Numerische) Simulation

3 Säulen der Naturwissenschaften

- ▶ Theorie/Modell
- ▶ Experiment
- ▶ (Numerische) Simulation

Differentialgleichungen zur Modellbildung

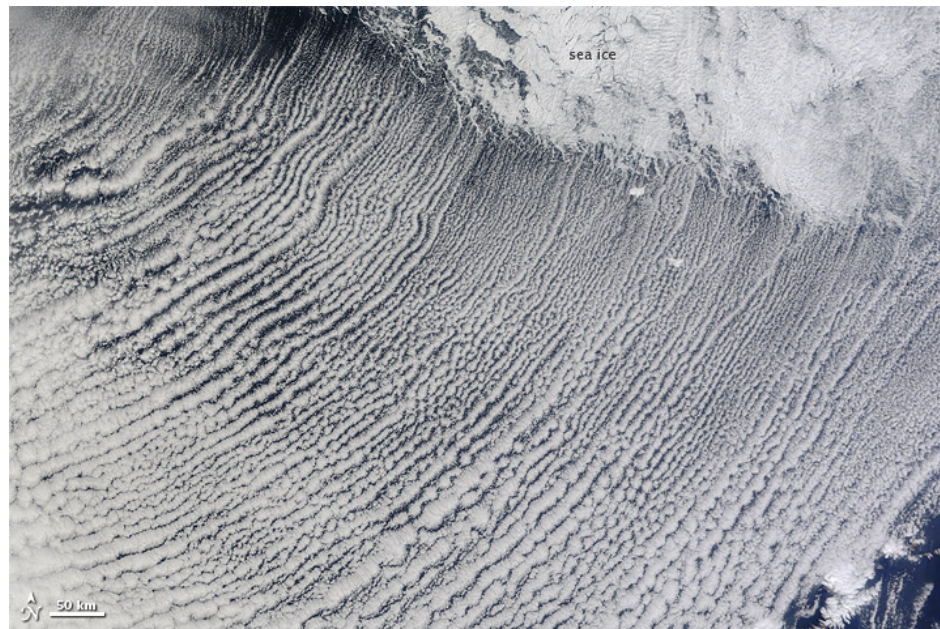
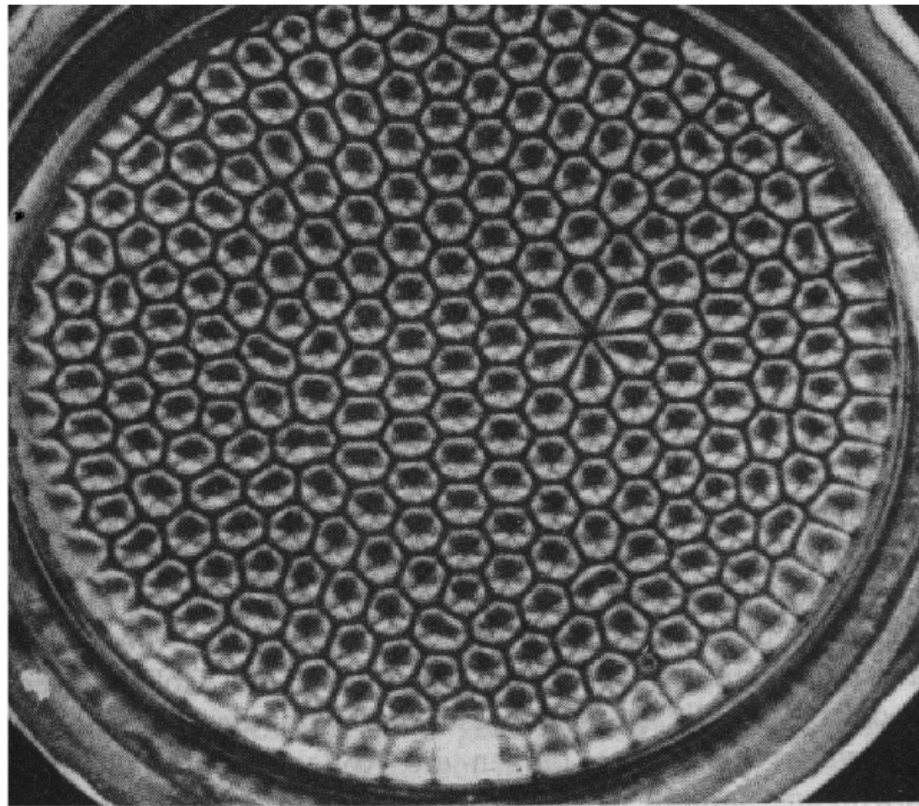
Viele physikalische Prozesse können mit Differentialgleichungen beschrieben werden.

Einfaches Beispiel:

Wärmeleitung in einem Stab

$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2} \quad (1)$$

$T(t, x)$ Temperaturverteilung mit zeitlicher/räumlicher Entwicklung



Wärmeleitungs-/Diffusionsgleichung

$$\frac{\partial u(t, x)}{\partial t} = \frac{\partial}{\partial x} \left(D(x) \frac{\partial u(t, x)}{\partial x} \right) \quad (2)$$

Wellengleichung

$$\frac{1}{c^2} \frac{\partial^2 u(t, x)}{\partial t^2} = \frac{\partial^2 u(t, x)}{\partial x^2} \quad (3)$$

Strukturgleichung (Swift-Hohenberg-Gleichung)

$$\frac{\partial u(t, x)}{\partial t} = r \cdot u(t, x) - u(t, x)^3 - \left(\frac{\partial^2}{\partial x^2} + 1 \right)^2 u(t, x) \quad (4)$$

Inhalt

- ▶ (Ein bisschen) Theorie von PDEs
- ▶ Lineare PDEs in Beispielen
- ▶ Numerische Methoden für PDEs
- ▶ Nichtlineare PDEs (Navier-Stokes Gleichung für Strömungen, Reaktions-Diffusions-Gleichungen für Strukturbildung)

Voraussetzungen

- ▶ Mathematikvorlesungen aus dem Bachelor
- ▶ Von Vorteil: Modellierung mit gewöhnlichen Differentialgleichungen
- ▶ Interesse an naturwissenschaftlicher Modellierung

Termine

Mo: 10:15 – 11:45, Di: 8:30 – 10:00 (vierzehntägig)

Übung wird nach Absprache festgelegt

Algorithmen und Techniken der Optimierung

Frank Fischer

JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

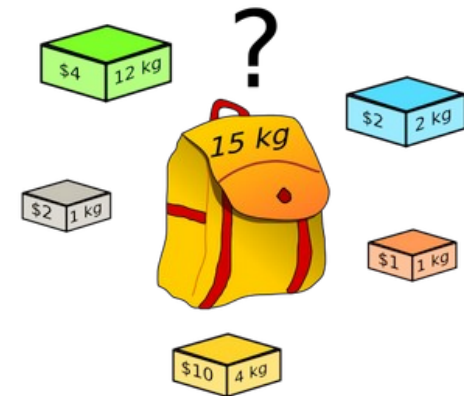


Worum geht's?

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

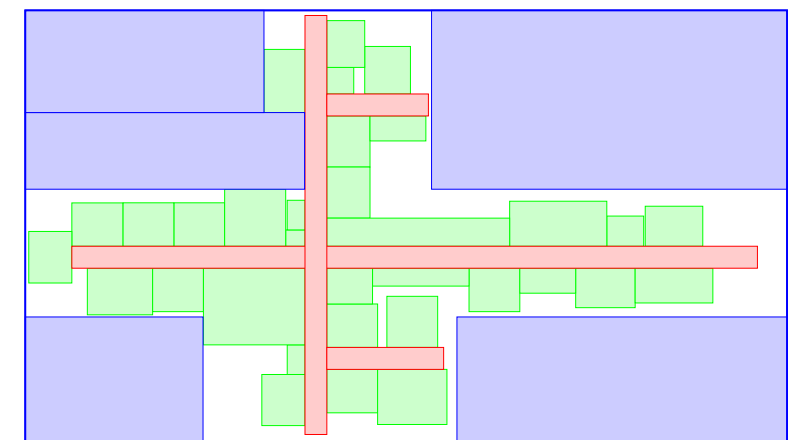
- Rucksackproblem:

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



Quelle: Wikipedia

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



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

Ziele

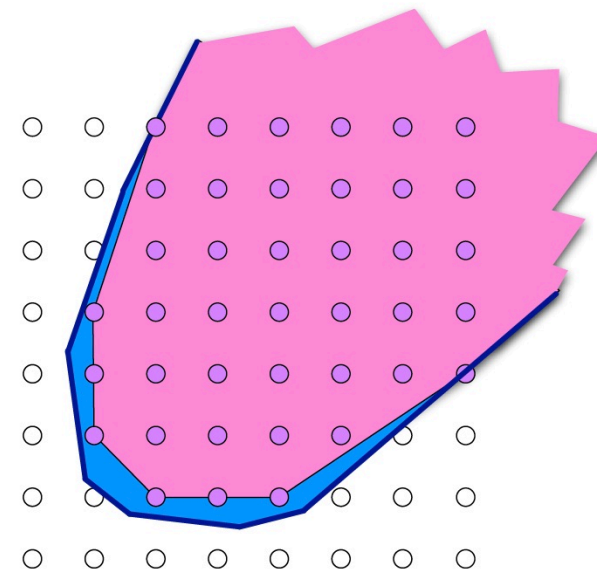
- Kenntnis der *Möglichkeiten* und *Grenzen* mathematischer Optimierung

„*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



Zielgruppe

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

Sommersemester 2022

- 2 SWS Vorlesung + Übung
 - Theoretische Grundlagen

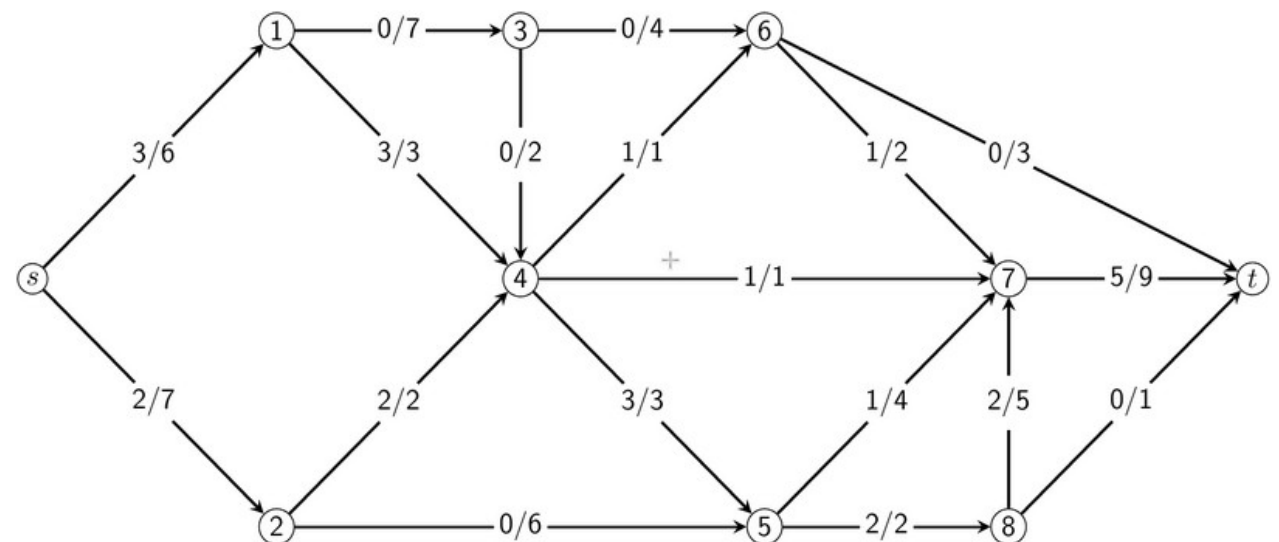
Wintersemester 2022/2023

- 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

Vorlesung

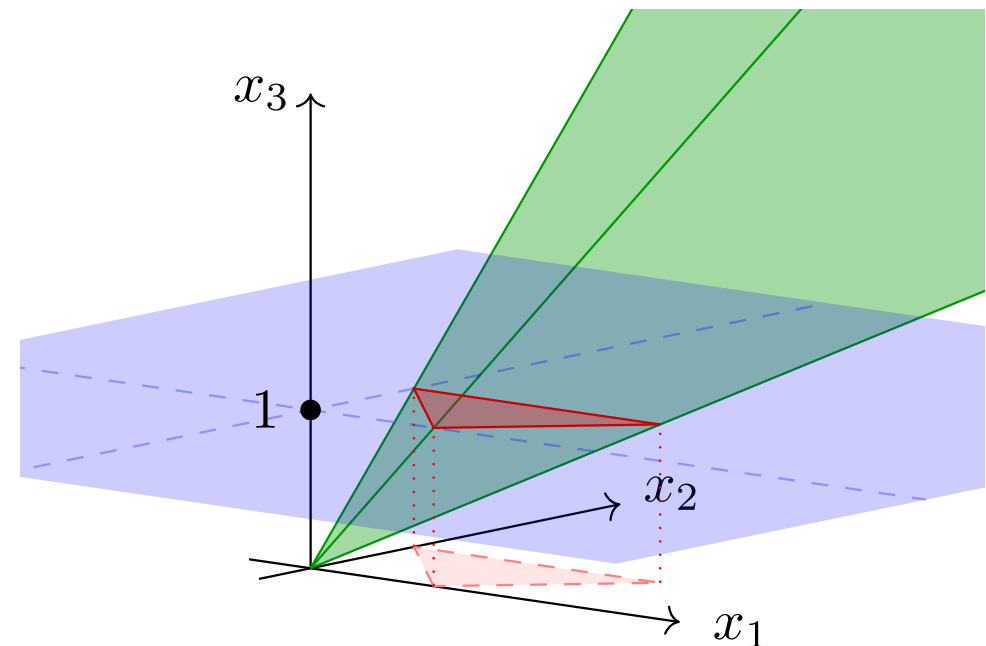
- Dienstag, 12-14 Uhr, 03-428

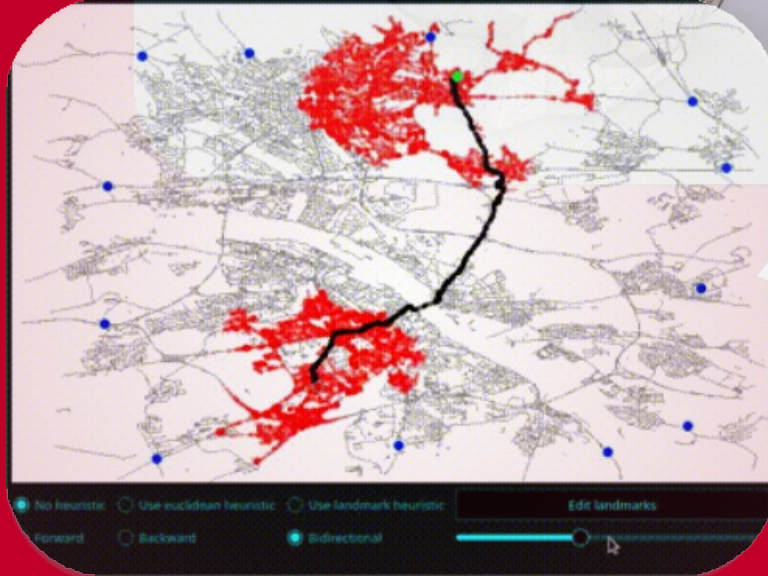
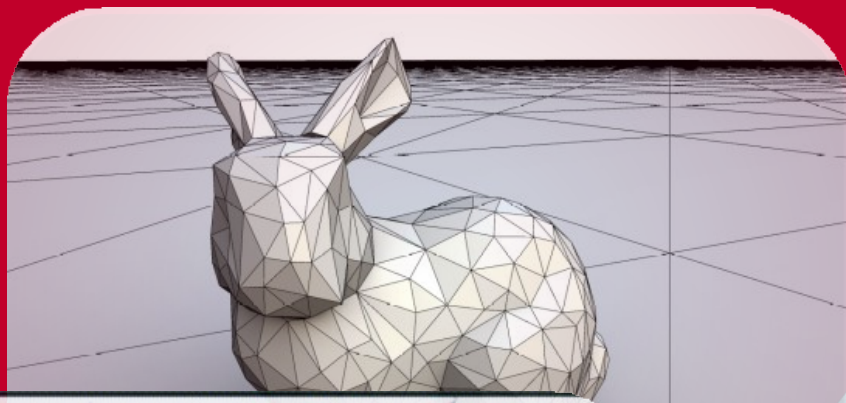
Übung

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

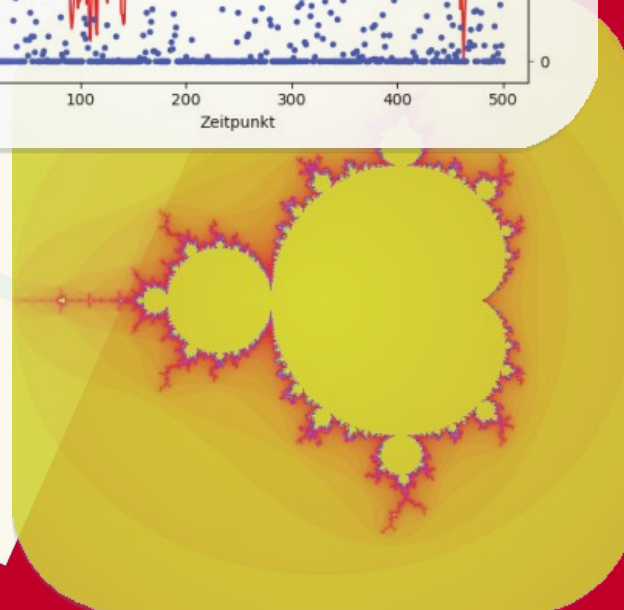
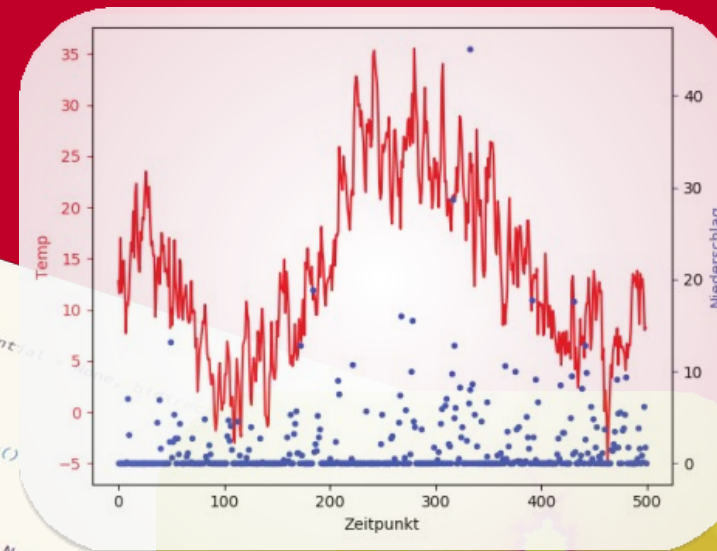
Bei Fragen

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





```
noHeuristic = 0  
euclideanHeuristic = 1  
landmarkHeuristic = 2  
  
def dijkstra(g, s, t, potential):  
    grey = 1  
    black = 2  
    visited = []  
    fwd_color = {}  
    fwd_queue = PQQueueBinaryHeap()  
    bwd_color = {}  
    bwd_queue = PQQueueBinaryHeap()  
    assert not bidirectional or t != None  
    if potential is None:  
        potential = lambda u: 0  
    initialize data for node 's'  
    fwd_dist = 0  
    incoming_edge = None  
    fwd_queue.push(s, 0)  
    fwd_color[s] = Grey  
    if bidirectional:  
        t.bwd_dist = 0  
        t.outgoing_edge = None  
        bwd_queue.push(t, 0)  
        bwd_color[t] = Grey
```



Mathematische Modellierung am Rechner I & II

Frank Fischer

Di. 16-18 Uhr, 03-428

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

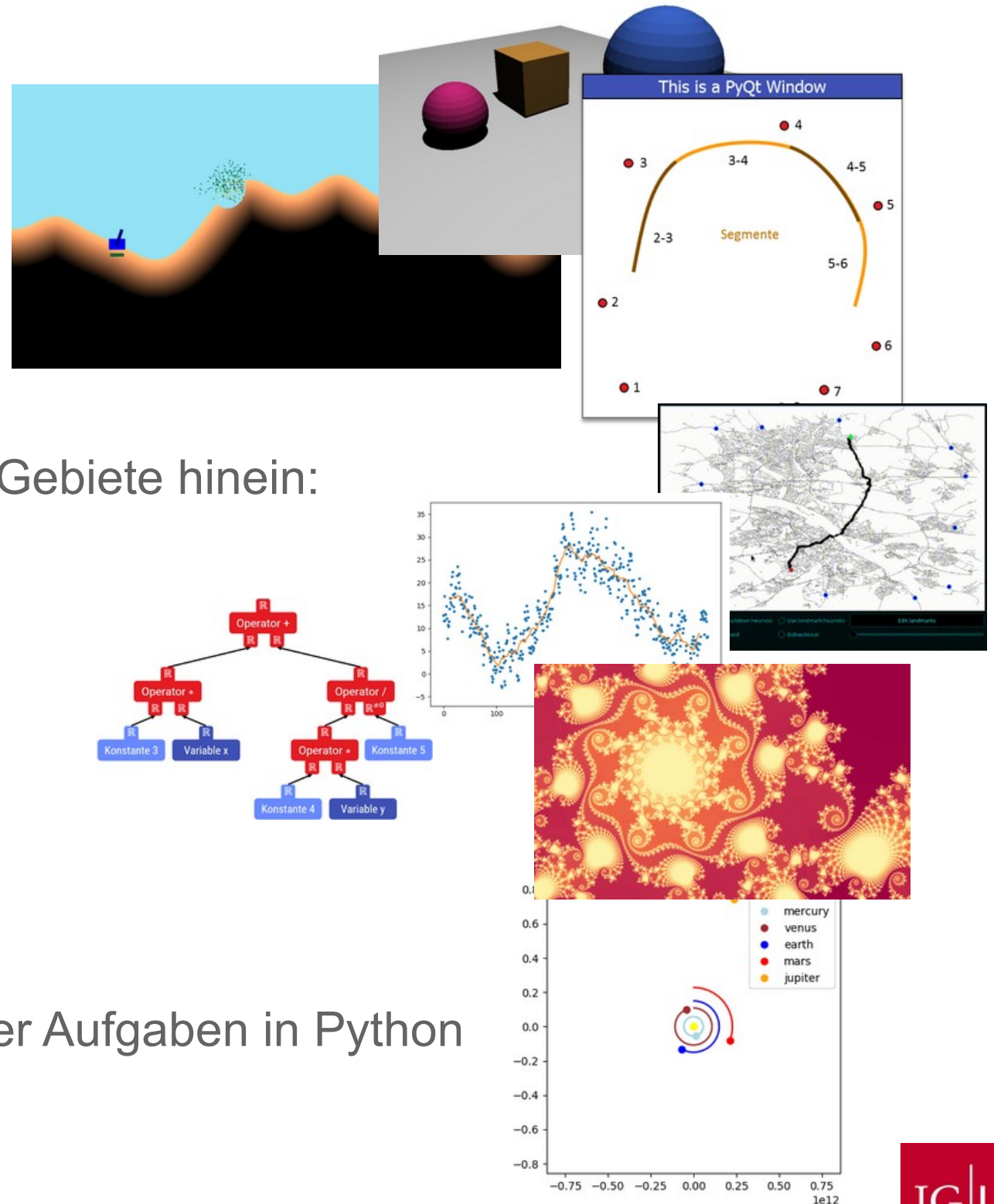
Inhalte und Ziele

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



Zielgruppe

- B.Sc. Informatik ab dem 2. Semester

Aufbau

- MMR 1 im Sommersemester 2022
- MMR 2 im Wintersemester 2022/2023

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

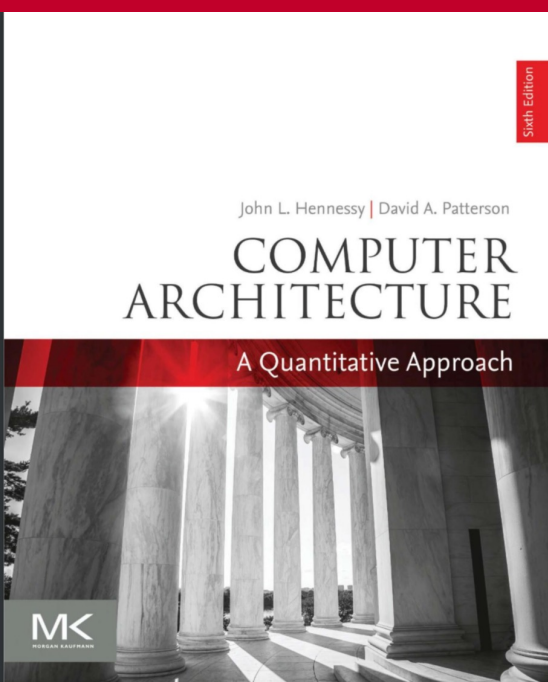
Bei Fragen

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

Advanced Computer Architecture

Course Introduction

Hossein Asadi (hasadi@uni-mainz.de)
Efficient Computing and Storage Group
Department of Computer Science
Johannes Gutenberg-Universität Mainz

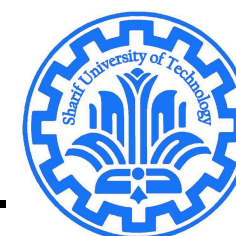


JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

JG|U

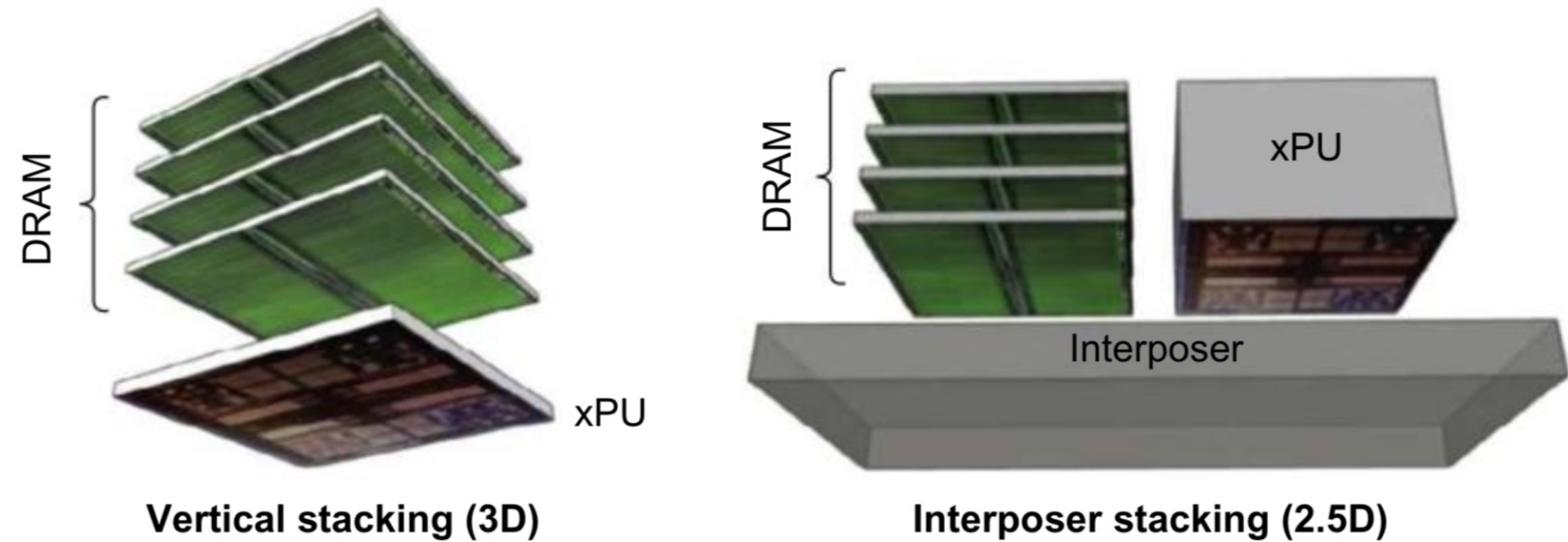
My Quick Background

- Ph.D. from **Northeastern Univ.**, Boston, MA (2007)
- BSc & MSc from **Sharif Univ.** of Tech, Tehran (2000/2002)
- Visiting Professor at **JGU Mainz** (July 2021 ~ July 2022)
- Visiting Professor at **EPFL** (2015)
- Full Professor, Dept. of CE, **Sharif Univ.** of Tech.
- Director of **Sharif HPC Center** (2015~2021)
- Director of **Sharif IT Service Center** (2017~2019)
- Founder of **DSN Lab** at Sharif Univ. of Tech. (2009)
- Co-Founder of **HPDS Corp.** (SAN storage manufacturer)
- Senior Hardware Engineer at **EMC**, Hopkinton, MA (2006~2009)

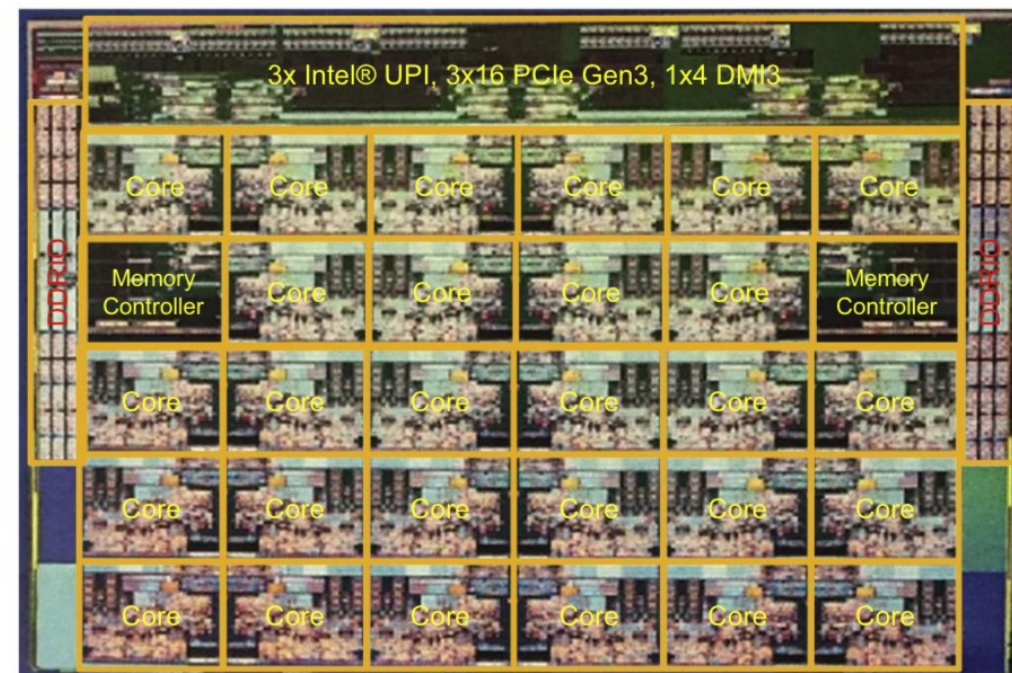


Topics Covered in This Course (I)

- **Memory Internals** in Advanced Computer Systems

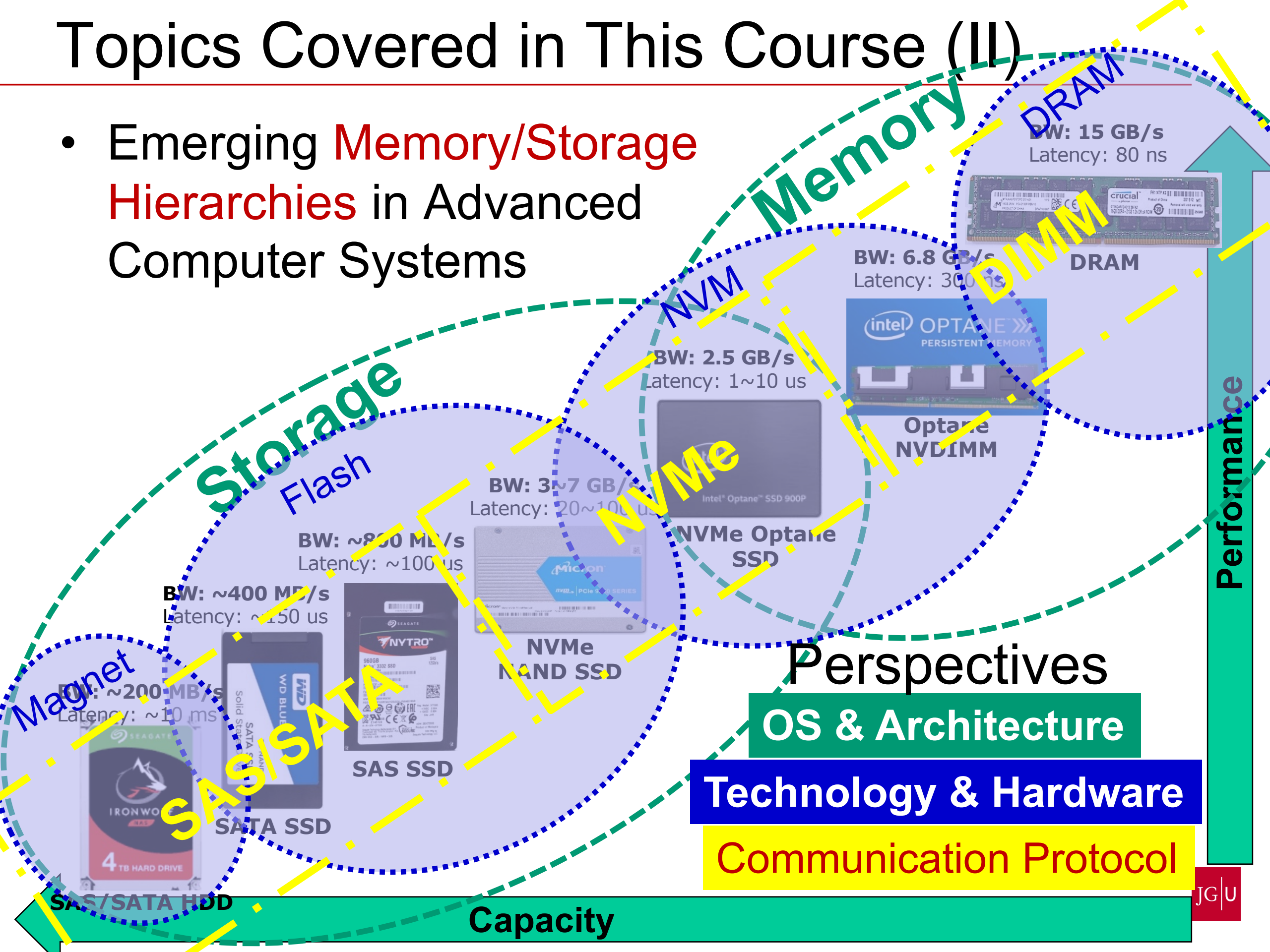


- **Major Functional Units** in Advanced Processors



Topics Covered in This Course (II)

- Emerging **Memory/Storage Hierarchies** in Advanced Computer Systems



Topics Covered in This Course (III)

- Advanced **Optimizations in Memory Hierarchy**

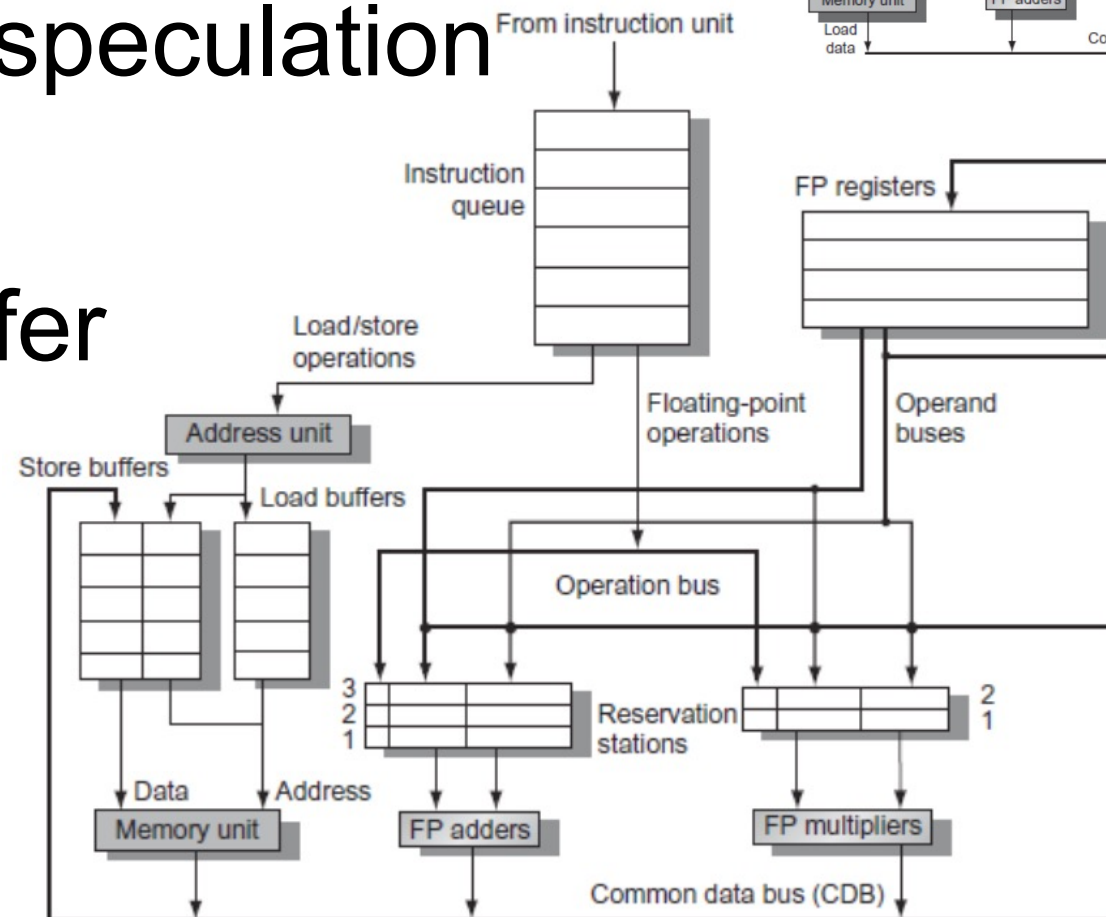
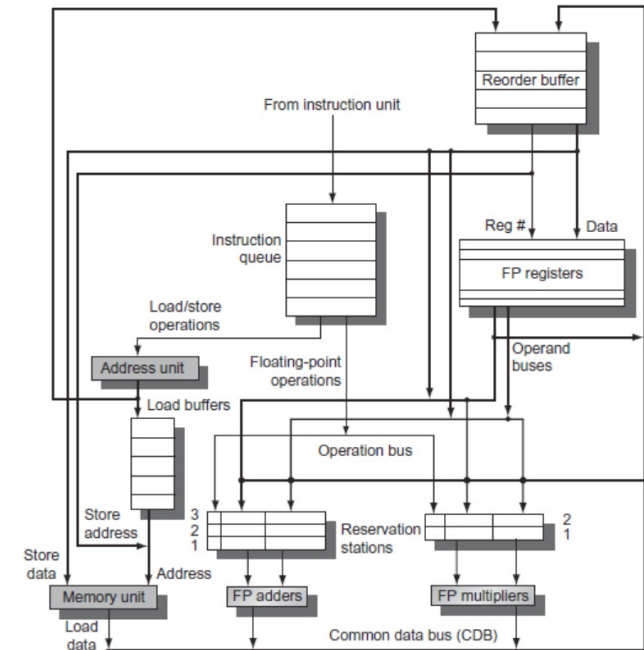
- Pipelined caches
- Multibanked caches
- Non-blocking caches
- Write buffers
- Compiler prefetching
- Hardware prefetching

Write address	V	V	V	V		
100	1	Mem[100]	0	0	0	0
108	1	Mem[108]	0	0	0	0
116	1	Mem[116]	0	0	0	0
124	1	Mem[124]	0	0	0	0

Write address	V	V	V	V				
100	1	Mem[100]	1	Mem[108]	1	Mem[116]	1	Mem[124]
	0		0		0		0	
	0		0		0		0	
	0		0		0		0	

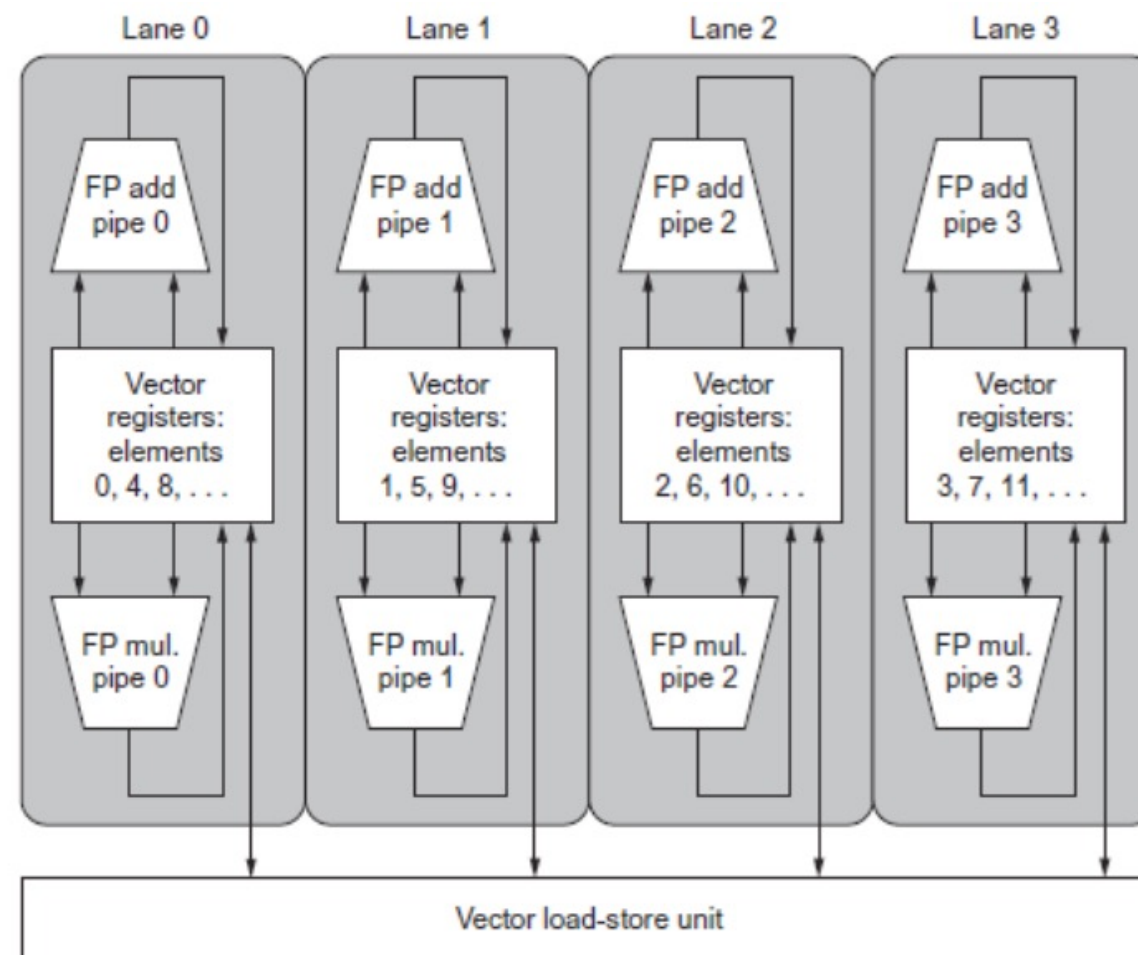
Topics Covered in This Course (IV)

- **Instruction-Level Parallelism (ILP)**
 - Branch prediction
 - Dynamic scheduling
 - Register renaming
 - Tomasulo's Algorithm
 - Hardware-based speculation
 - Reorder buffer
 - Branch target buffer
 - VLIW



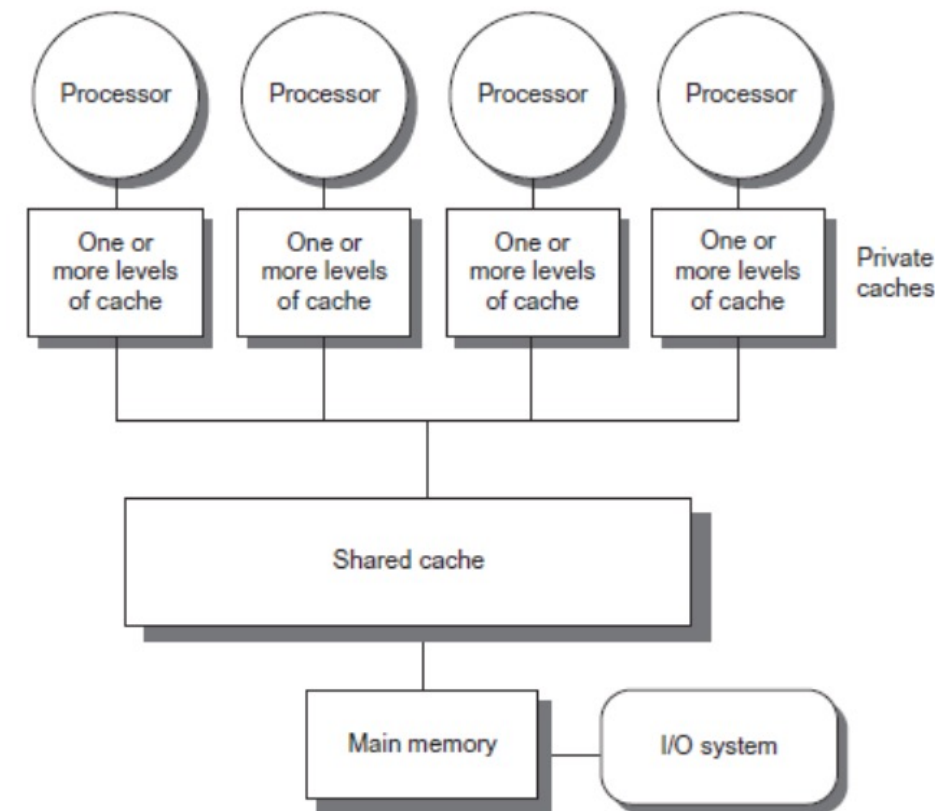
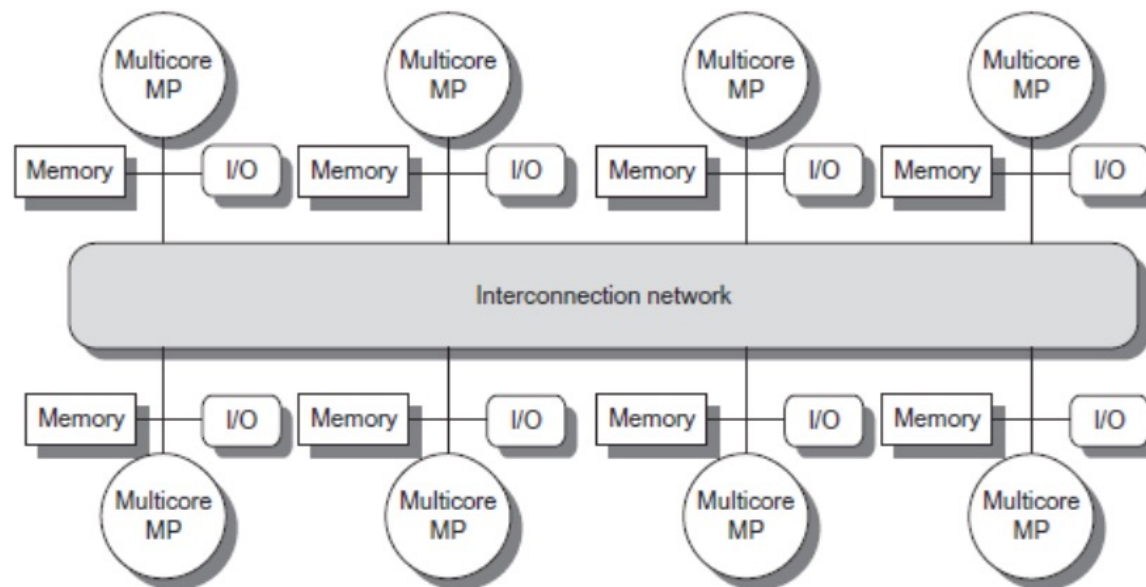
Topics Covered in This Course (V)

- **Data-Level Parallelism**
 - **Vector** architectures
 - **SIMD** extensions
 - **Graphics Processor Units (GPUs)**



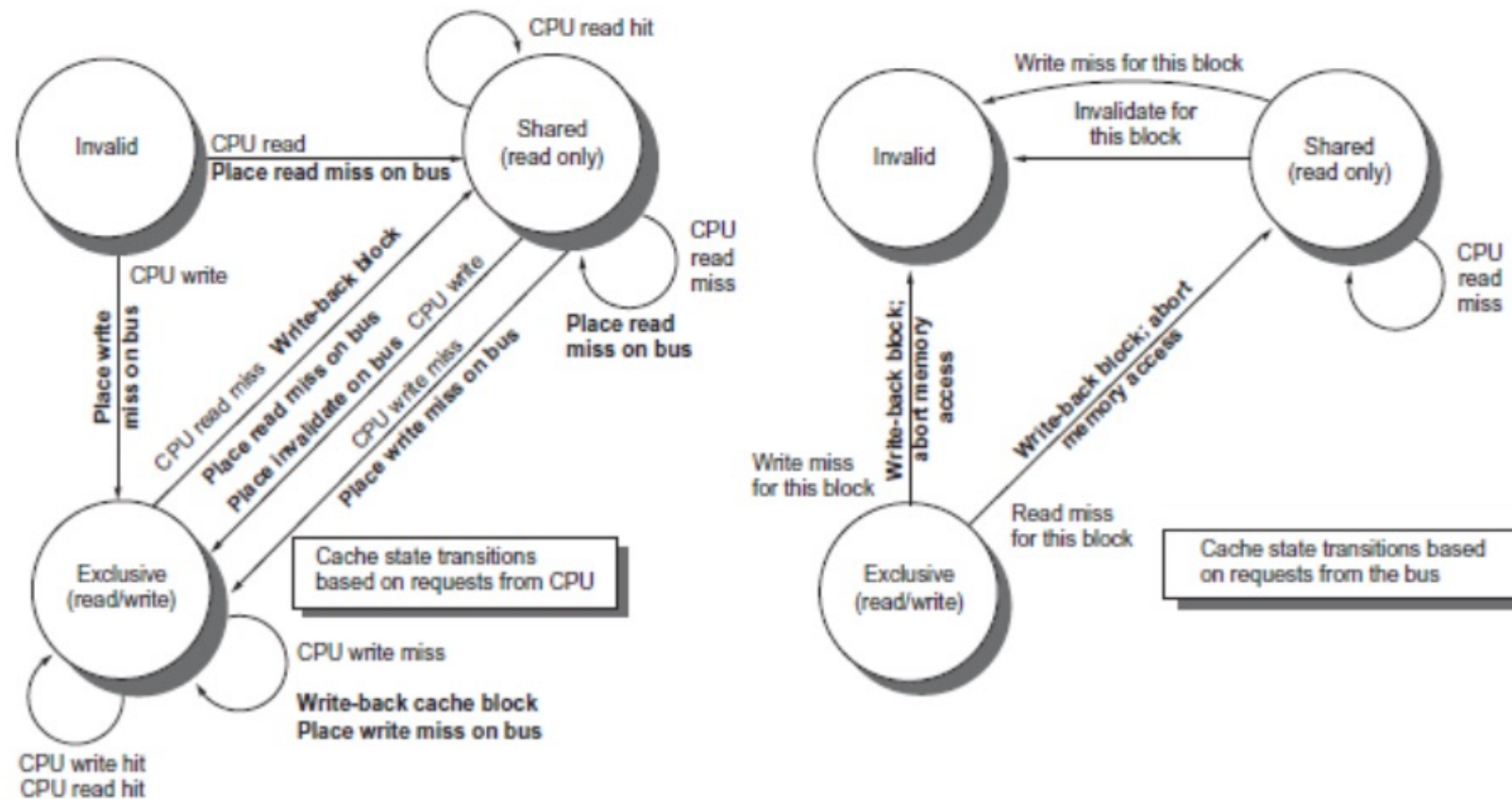
Topics Covered in This Course (VI)

- **Thread-Level** Parallelism
 - Symmetric multiprocessors (SMP)
 - Distributed shared memory (DSM)



Topics Covered in This Course (VII)

- Cache Coherency
 - Snoopy coherence protocols

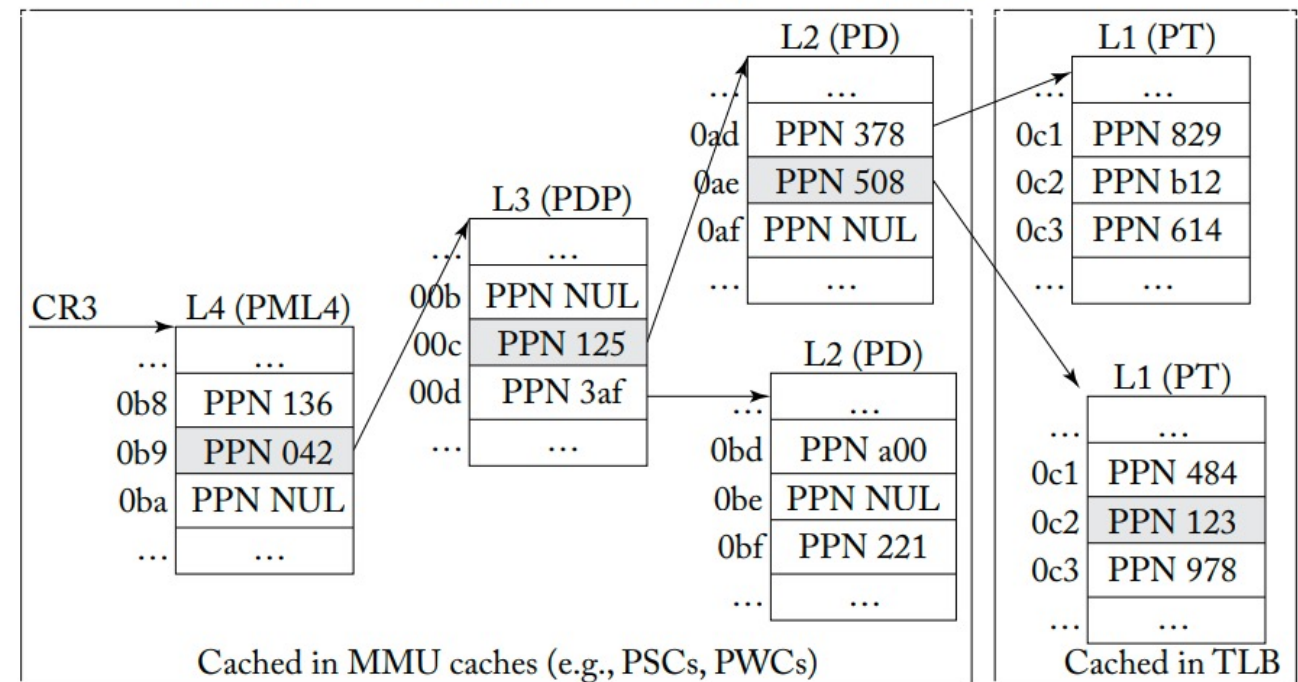


Topics Covered in This Course (VIII)

- **Architecture Support for Operating Systems**
 - Virtual Memory (VM) basics
 - Translation-Lookaside Buffers (TLBs)
 - Multi-level TLBs
 - HW Support for **VM conherency & consistency**
 - HW stack for VM

VA = 0x

5	c	8	3	1	5	c	c	2	0	1	6
0101	1100	1000	0011	0001	0101	1100	1100	0010	0000	0001	0110
0b9			00c		0ae		0c2		016		
L1 offset			L2 offset		L3 offset		L4 offset				



Textbooks

- Main Textbooks
 - J. Hennessy and D. Patterson, “[Computer Architecture: A Quantitative Approach](#)”, 6th Edition, Nov. 2017.
 - A. Bhattacharjee and D. Lustig, “Architectural and Operating System Support for Virtual Memory”, 2017.
- Further Readings
 - Will be announced in the class

Assignments & My Schedule

- **Assignments & Project**
 - Three to four series of assignments
 - Course project will be discussed in Week 8.
- **Class Sessions**
 - TBD shortly
- **Feel free to reach me at:**
 - Email: hasadi@uni-mainz.de
 - Skype: [hossein_asadi](https://www.skype.com/people/hossein_asadi)