Institut für
Informatik

Vorstellung der Lehre

WS 2024/25

Hiwis gesucht

- Einführung in die Programmierung
- Einführung in die Softwareentwicklung
- Software Engineering
- Technische Informatik
- Komplexitätstheorie
- Datenstrukturen und effiziente Algorithmen
- Einführung in die Bioinformatik
- Accelerated Computing with GPU
- Modellierung II (Deep Learning – Theorie und Praxis)
- Deep Learning
- Spieltheorie
- 2D Vision and Deep Learning
- Data Mining
- Computergrafik 1
- Topics in Functional Programming
- Lernwerkstatt
- Studieneingangsprojekt

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

Übungsleiterschulung

- Pflichtveranstaltung für alle neuen Übungsleiter bzw. Übungsleiter, die die Schulung noch nicht besucht haben
- Offen für jeden (1CP Softskills)
- Bei Interesse, direkt bei Stefan Endler melden, da Anmeldung über Jogustine nicht möglich
- Termin:
  - 07. + 08. 10. 2024
  - Ganztägig (9-17 Uhr)
  - Raum: 03-424

Betriebspraktikum

- 10-wöchiges Betriebspraktikum in Vollzeit
- Ist eine von drei Optionen im Spezialisierungsmodul (alte PO) und auch in der neuen PO optional wählbar
- Anmeldung über Jogustine zu jeder Zeit
- Im entsprechenden Moodle Kurs befinden sich weitere Informationen rund um das Betriebspraktikum und dessen Ablauf
- Hinweis zur Zuständigkeit:
  - Bis 30.09. Markus Blumenstock
  - Ab 01.10. Stefan Endler

Generell zu Seminaren

- Anmeldung über Jogustine ab heute 16 Uhr bis zum 17.07., 13 Uhr
- Bei einer Überbelegung (>15 Anmeldungen) werden die Teilnehmer nach den Senatsrichtlinien ausgewählt.

Mensch-Maschine Interaktion

- Dozent:

  Prof. Dr. Volker Luckas (TH Bingen)
- Vorlesung:

  Donnerstags, 10-12 Uhr, Raum 04-432
- Übung:

  Donnerstags, 12-14 Uhr, Raum 04-432

Vorstellung des Lehrangebots – Efficient Computing and Storage Group

WS 2024 / 2025

Prof. Dr.-Ing. André Brinkmann

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Vorlesungen, Seminare, Abschlussarbeiten

- Vorlesung „Technische Informatik“

- Vorlesung „Advanced Topics in Operating Systems“ / „Speichersysteme“
- Seminar „Betriebssysteme“
- Praktika
- Bachelor und Masterarbeiten

Technische Informatik

Lernziel:

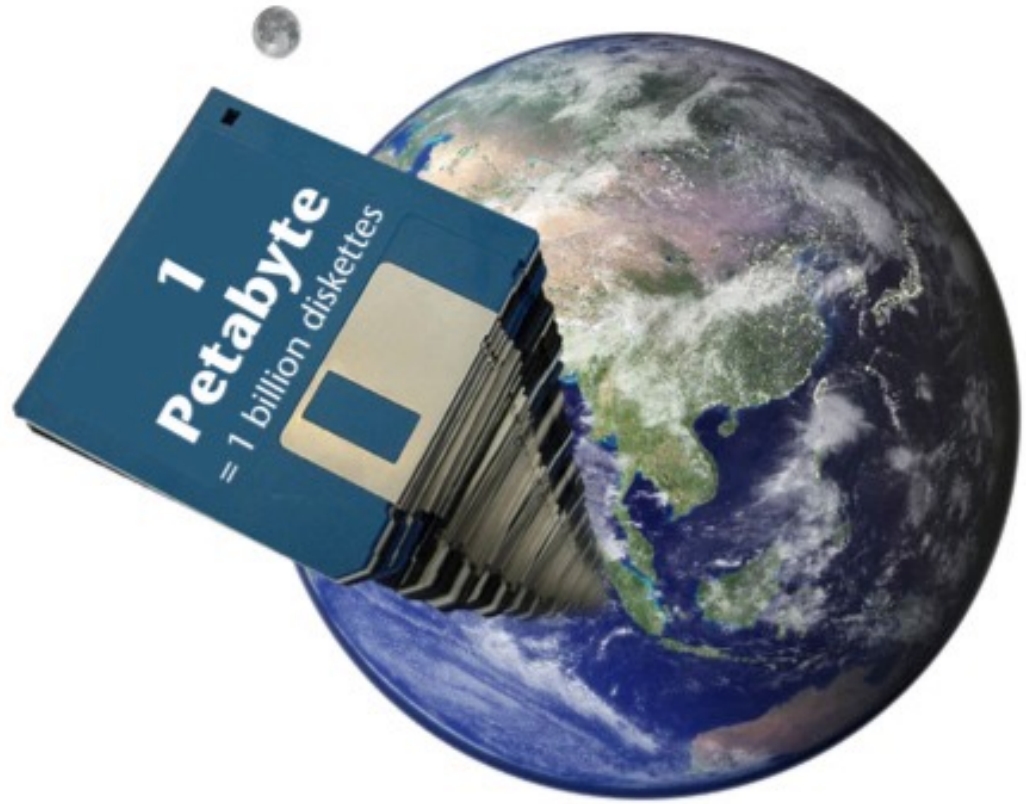
- Wie baue ich einen Rechner?

Themen sind u.a.:

- Logikfunktionen und boolesche Algebra
- Zahlendarstellungen und Rechnerarithmetik
- Schaltwerke und deren systematischer Entwurf
- von-Neumann-Architektur
- Architektur und Maschinenbefehle eines RISC-Prozessors
- Assemblerprogrammierung
- Eigenschaften moderner Rechnerarchitekturen

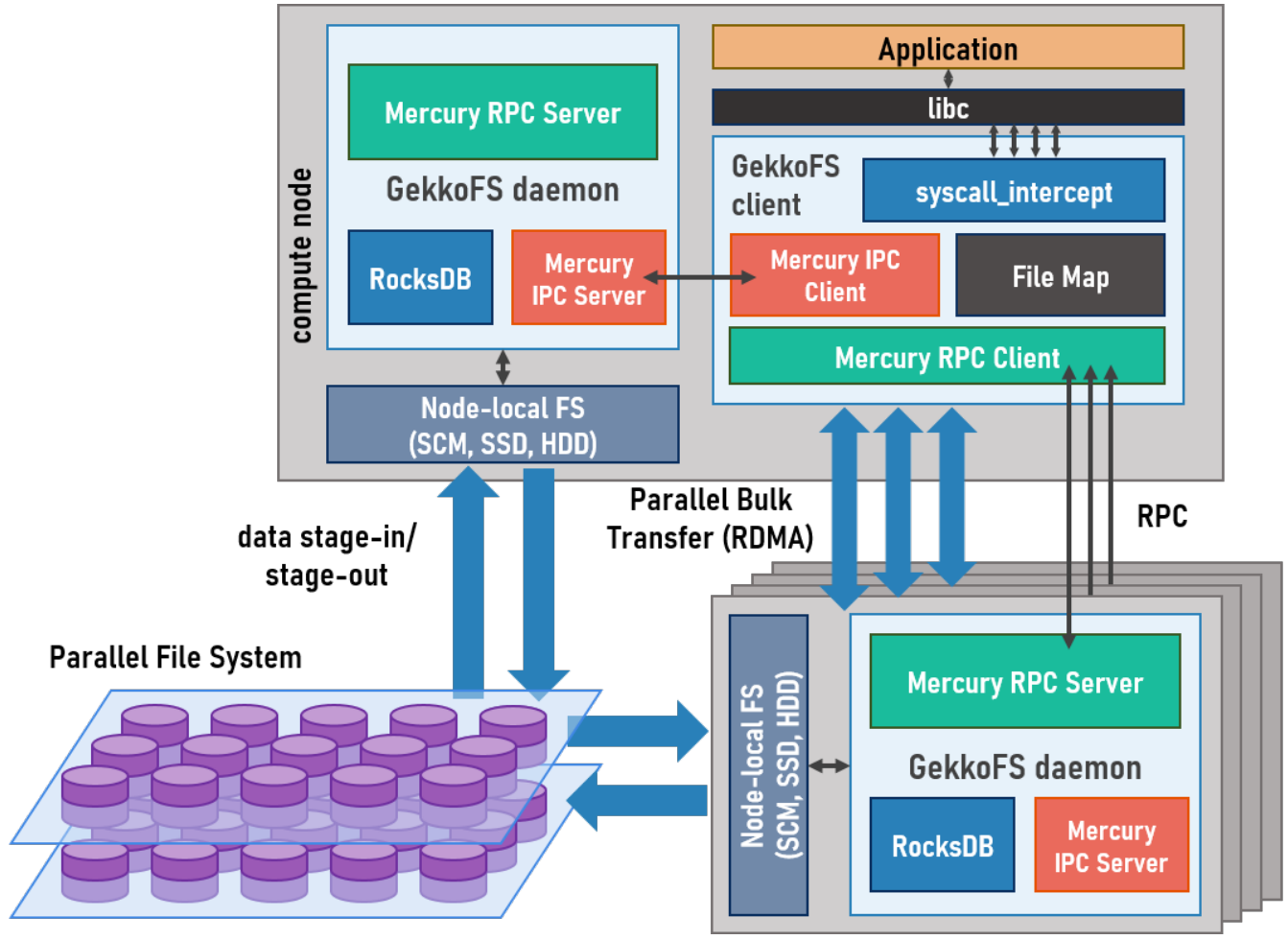
Vorlesung Speichersysteme

- Die Vorlesung wird angefangen bei einzelnen Festplatten / SSDs 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

Seminar Betriebssysteme

- Lernziel:
  In dem Seminar Betriebssysteme erlernen Studierende die Arbeit mit aktueller Primärliteratur, das Vortragen wissenschaftlicher Ergebnisse, den wissenschaftlichen Diskurs sowie das wissenschaftliche Schreiben

- Seminar läuft während der Vorlesungszeit

- Das Seminar wird eine ausgewählte Fragestellung behandeln und die Fortschritte auf diesem Thema über die Zeit vorstellen und diskutieren
- In den letzten Semestern wurden SSDs / Key-Value Stores behandelt

Praktikum Betriebssysteme

- 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

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

Erasmus

- Nutzen sie die Gelegenheit, ein Semester im Ausland zu verbringen
  - Persönliche Weiterentwicklung
  - Berufschancen
  - Tolle und schöne Erfahrung
- Wir möchten ihnen den Prozess so leicht wie möglich machen!
- Bewerbung jeweils bis Ende des Jahres (Nachnominierung möglich)
  - Motivationsschreiben (1-2 Seiten) mit Nennung des Wunschziels und Wunschsemesters
  - Tabellarischer Lebenslauf
  - Aktuelle Leistungsübersicht
- Auch außerhalb von Erasmus gibt es (Förder-) Möglichkeiten
- Bei Fragen: persönliche Beratung nach Terminvereinbarung (`ernst.althaus@uni-mainz.de`)

Austauschmöglichkeiten

- Finnland-Jyväskylän Yliopisto
- Italien-Università degli Studi di Salerno
- Lettland-Latvijas Universitáte
- Norwegen-Universitetet i Agder
- Rumänien-Universitatea Babes-Bolyai din Cluj-Napoca
- Spanien-Universitat Politècnica de València
- Vereinigtes Königreich-University of Glasgow
- Frankreich-Université de Bourgogne
- Italien-Università degli Studi Roma Tre
- Lettland-Liepājas Universitātē
- Polen-Uniwersytet Jagielloński
- Schweden-Högskolan Dalarna
- Spanien-Universitat de Barcelona
- Tschechien-Ceské Vysoké Uceni Technické v Praze
- Italien-Università degli Studi di Palermo
- Kroatien-Visoka škola za informacijske tehnologije
- Norwegen-Høgskolen i Østfold
- Rumänien-Universitatea 'Lucian Blaga' din Sibiu
- Spanien-Universidad de La Coruña
- Spanien-Universitat de València
- Türkei-Üsküdar University
- Ungarn-Pécsi Tudományegyetem

Lehrveranstaltungen der theoretischen Informatik

- Komplexitätstheorie
- Spieltheorie
- Seminar: Algorithmen
- Praktikum Approaching Programming Contests (Domenico Mosca)
- Seminar: Formale Sprachen + Berechenbarkeit (Markus Blumenstock)

Vorlesung: Spielthoerie

Inhalt:

- Spiele in Normalform
- Spiele in Extensivform
- Theorie der Verhandlungen
- Auktionstheorie
- Evolutionäre Spiele
- Wiederholte Spiele

Formales:

- Vorlesung: Mi 10:30-12:00
- Übung: tba
- Organisation der Übung: tdb

Seminar: Algorithmen

- Vorbesprechung: Im Oktober online (genauer Termin wird noch bekannt gegeben)
- Zeit und Ort: Blockseminar im März (genauer Termin wird per Umfrage bestimmt)
- Vorbesprechung des Vortrages: Im Februar online (genauer Termin wird noch bekannt gegeben)
- Voraussetzungen: DSeA oder äquivalent
- Scheinvergabe:
  - Vortrag von ca. 30min
  - Ausarbeitung in LaTeX
  - aktive Teilnehme
- Inhalte:
  - Vorstellung eines Algorithmus ähnlich denen aus DSeA
  - Genauere Themenvorschläge gibt es in der Vorbesprechung
  - Auch Themen zu Spieltheorie und Kryptographie möglich

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.

Seminar: Formale Sprachen und Berechenbarkeit

- Seminar während der Vorlesungszeit
- Termin wird durch Umfrage festgelegt
- Dozent: Markus Blumenstock
- Vorbesprechung: 2. September, 10:00 in BBB (`https://bbb.rlp.net/rooms/v0m-g2u-lsm-cmh/join`)
- Voraussetzung: VL "Formale Sprachen und Berechenbarkeit"
- Kann im Theorie-Bereich eingebracht werden
- Scheinvergabe:
  - Vortrag von 35-40 Minuten (55 % der Note)
  - Ausarbeitung in LaTeX (35 % der Note)
    oder, falls sich das Thema eignet, Implementierung
  - Aktive Teilnahme (10 % der Note)

Seminar: Formale Sprachen und Berechenbarkeit

Die Themen sind u. a.

- (Nicht)-Kontextfreiheit von natürlichen Sprachen
- Ogdens Lemma (allgemeineres PL für kf. Sprachen)
- Jaffes Pumping Lemma (notwendiges und hinreichendes PL für reguläre Sprachen)
- LR-Parser (deterministisch kf. Sprachen)
- Unentscheidbarkeitsresultate für Grammatiken
- Unentscheidbarkeit von Wang-Parkettierungen ("Affenpuzzle" im Foyer)
- Kleenes Rekursionssatz
- Algorithmische Informationstheorie, Kolmogorov-Komplexität

Lehrangebot der Arbeitsgruppe
„High Performance Computing and its Applications“

Prof. Dr. Sarah Neuwirth

Wintersemester 2024/25

Seminar: Kommunikationsnetze

HPC|A Group

Wann:

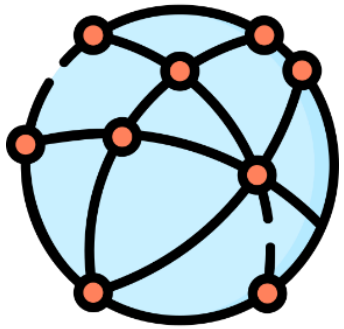
- Einführungsveranstaltung(en), Blockveranstaltung gegen Ende des Semesters

Lernziele:

- Überblick über aktuelle Forschungsthemen auf dem Gebiet der Kommunikationsnetze
- Erlernen grundlegender Fähigkeiten der wissenschaftlichen Arbeit
  - Einübung und Anwendung von eigenständiger Literatursuche und -analyse
  - Wissensaufarbeitung und -vermittlung komplexer Sachverhalte
  - Aneignung erfolgreicher Präsentations- und Kommunikationsmethoden
  - Fachdiskussion, Konstruktives (Peer-)Feedback zu Inhalt und Vortragsstil

Seminarthemen (Auszug):

- Software-Defined Networking (SDN)
- High-Performance Networking Fabrics
- Artificial Intelligence (AI) and Machine Learning (ML) for Networking
- Data Center Network (DCN) Architectures



Advanced Topics in Computer Networks

HPC|A Group

Wann:

- *Vorlesung:* Montag, 14 bis 16 Uhr, Raum 04-512
- *Übung:* voraussichtlich Montag, 16 bis 18 Uhr (to be confirmed…)
- *Praktikum:* Vorlesungsfreie Zeit (TBA) – 2 Wochen

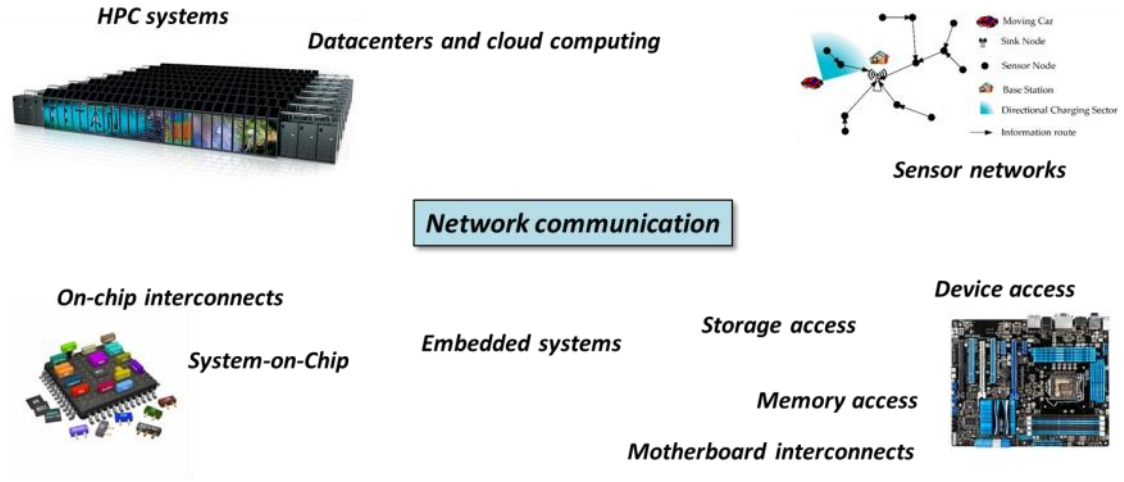
Lernziele:

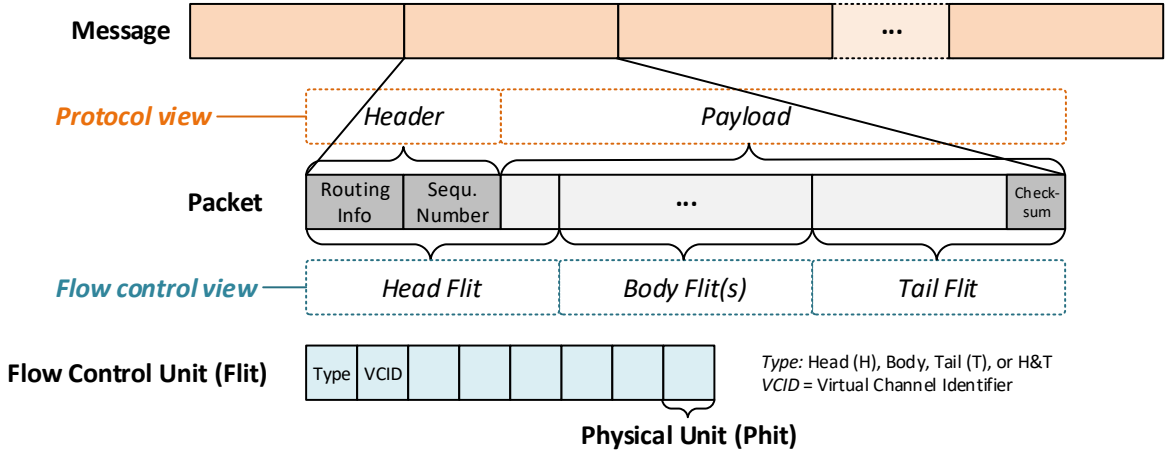
- *Verständnis* und *Anwendung* der erlernten Entwurfskonzepte und -prinzipien von Verbindungsnetzwerken
- *Konfiguration* und *Nutzung* von Verbindungsnetzwerken für gegebene Anforderungen

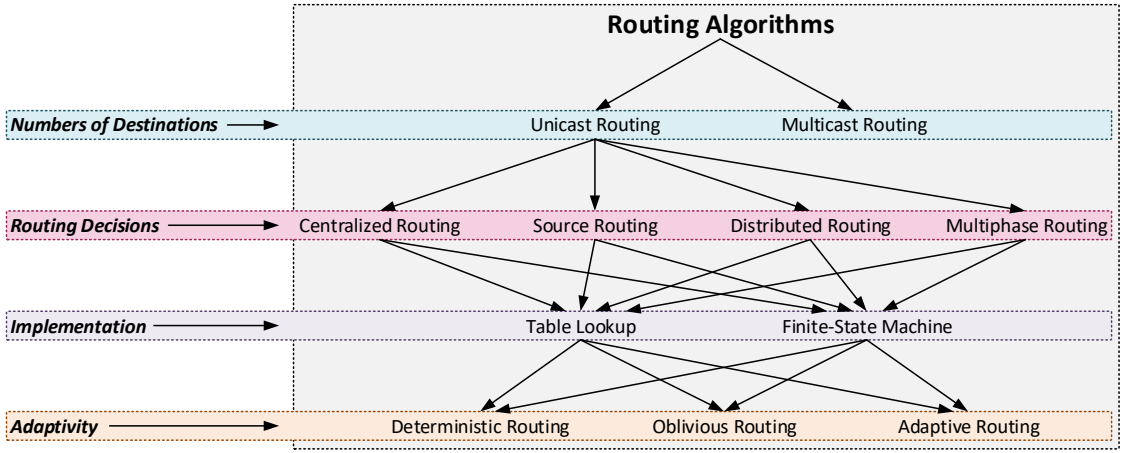
Inhalte (Auszug):

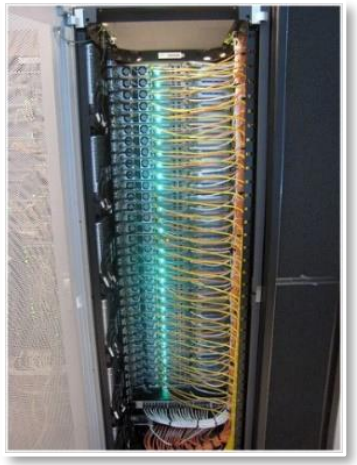
- Vermittlungstechniken, Virtual Channels, Flusskontrolle, Pufferverwaltung
- Fortgeschrittene Routing-Algorithmen und deren Implementierung
- Routing-Deadlocks, Livelocks und Starvation
- Netzwerktechnologie-Standards (on-chip versus off-chip)
- Software-Defined Networking (OpenFlow-Standard)
- Leistungsbewertung und Netzwerk-Simulation (OMNeT++ / Mininet)

Advanced Topics in Computer Networks









Themen der Arbeitsgruppe HPC|A

- **Bachelor- und Masterarbeiten / Hiwi-Tätigkeiten**, mögliche Themen:
  - Monitoring of System Ressources and Workload Behavior in HPC Systems
  - Comprehensive and Automated Performance Engineering and Modeling
  - HPC System Benchmarking – Metrics, Systematization, Comparability, Reproducibility
  - Parallel I/O & Storage Infrastructures for Emerging Workloads (DL/ML, AI, etc.)
  - Simulation of Heterogeneous Network and Storage Infrastructures
  - Modular Supercomputing (i.e., resource disaggregation and virtualization)
- **Weitere Informationen:** https://www.hpca-group.de/
- **Kontakt:** Prof. Dr. Sarah Neuwirth, neuwirth@uni-mainz.de

Computergrafik I

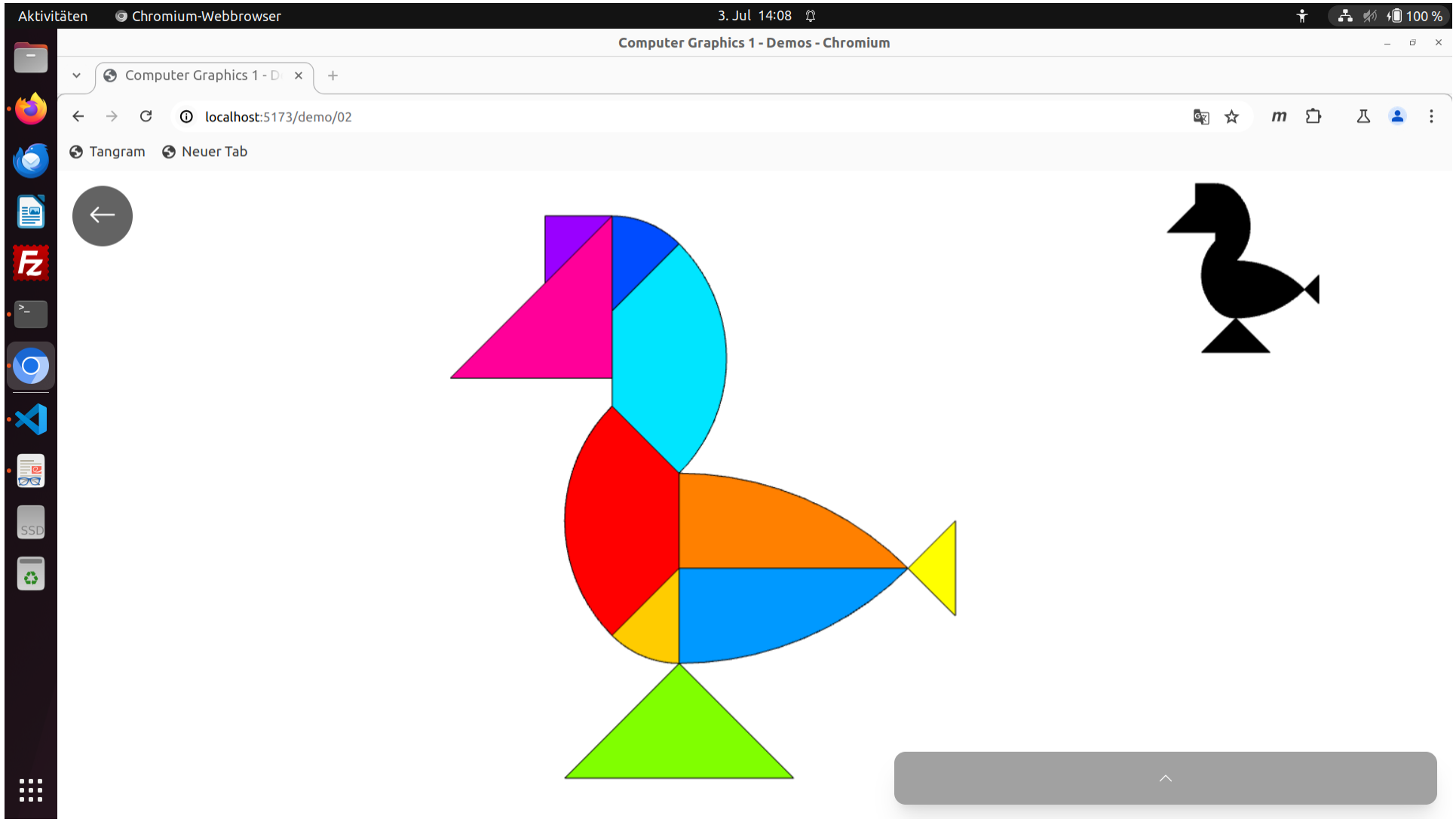
Elmar Schömer

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

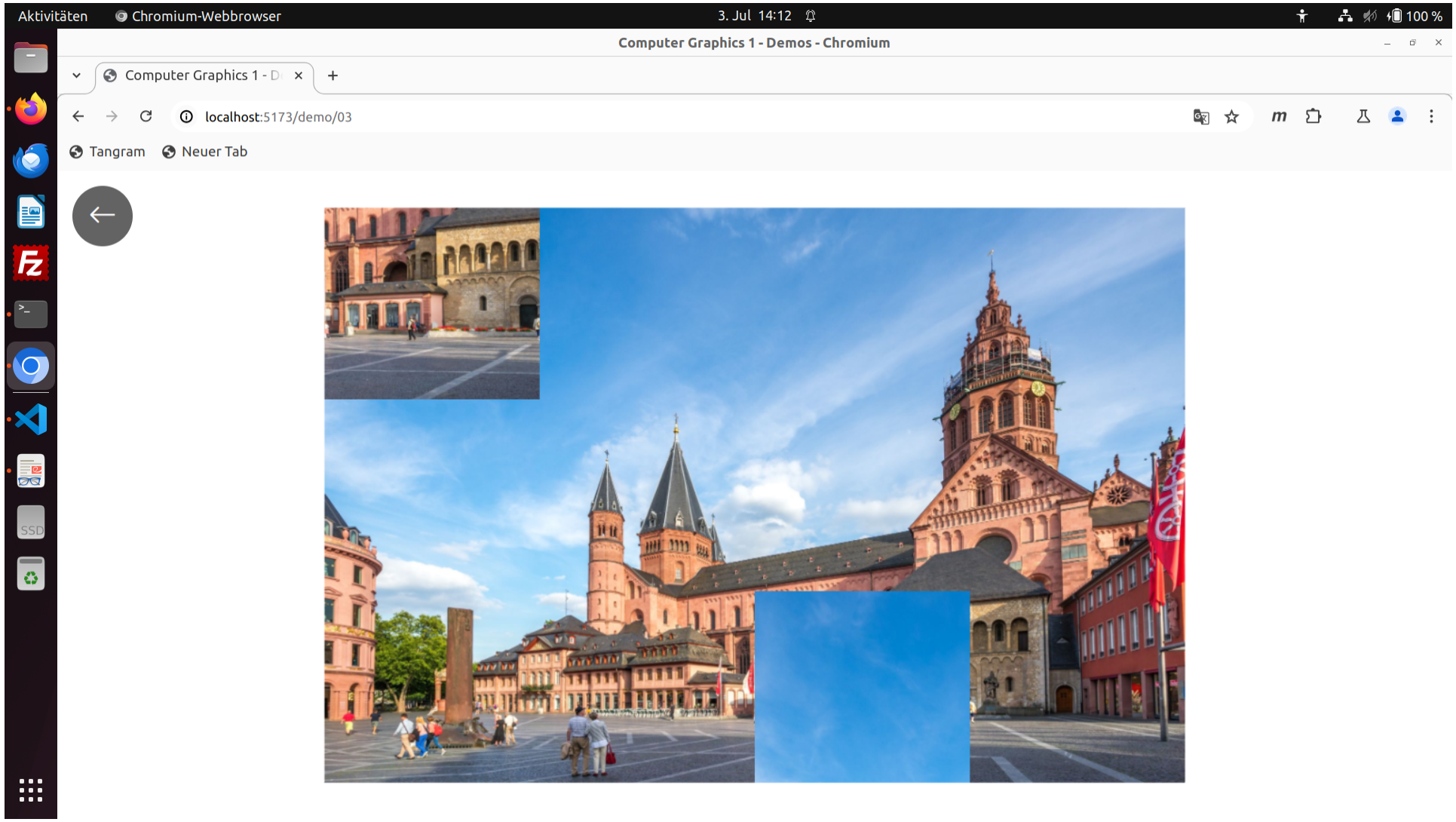
INSTITUT FÜR INFORMATIK

Wintersemester 24/25, Di 10-12 Uhr

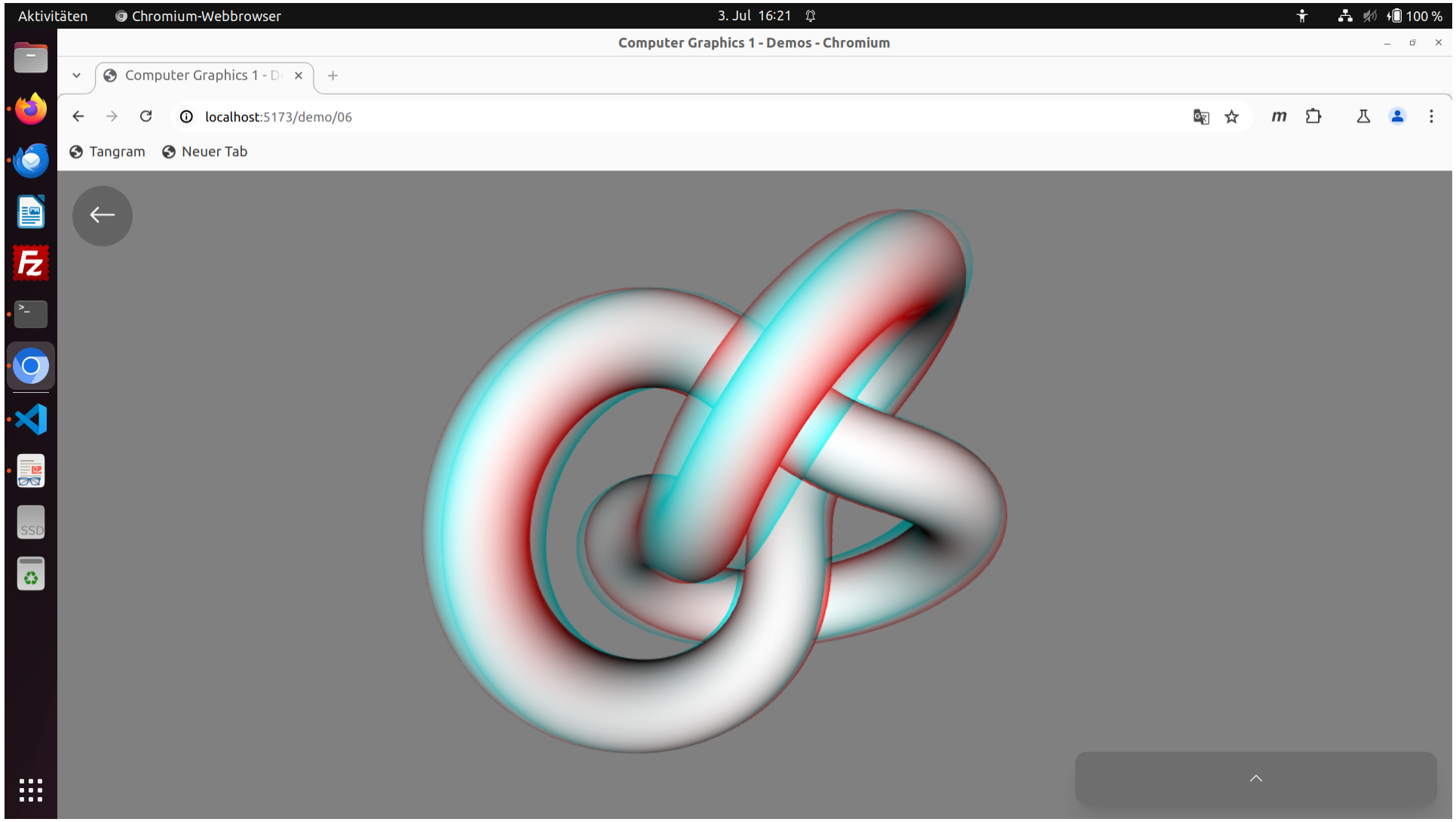
2D-Graphik - Legespiele



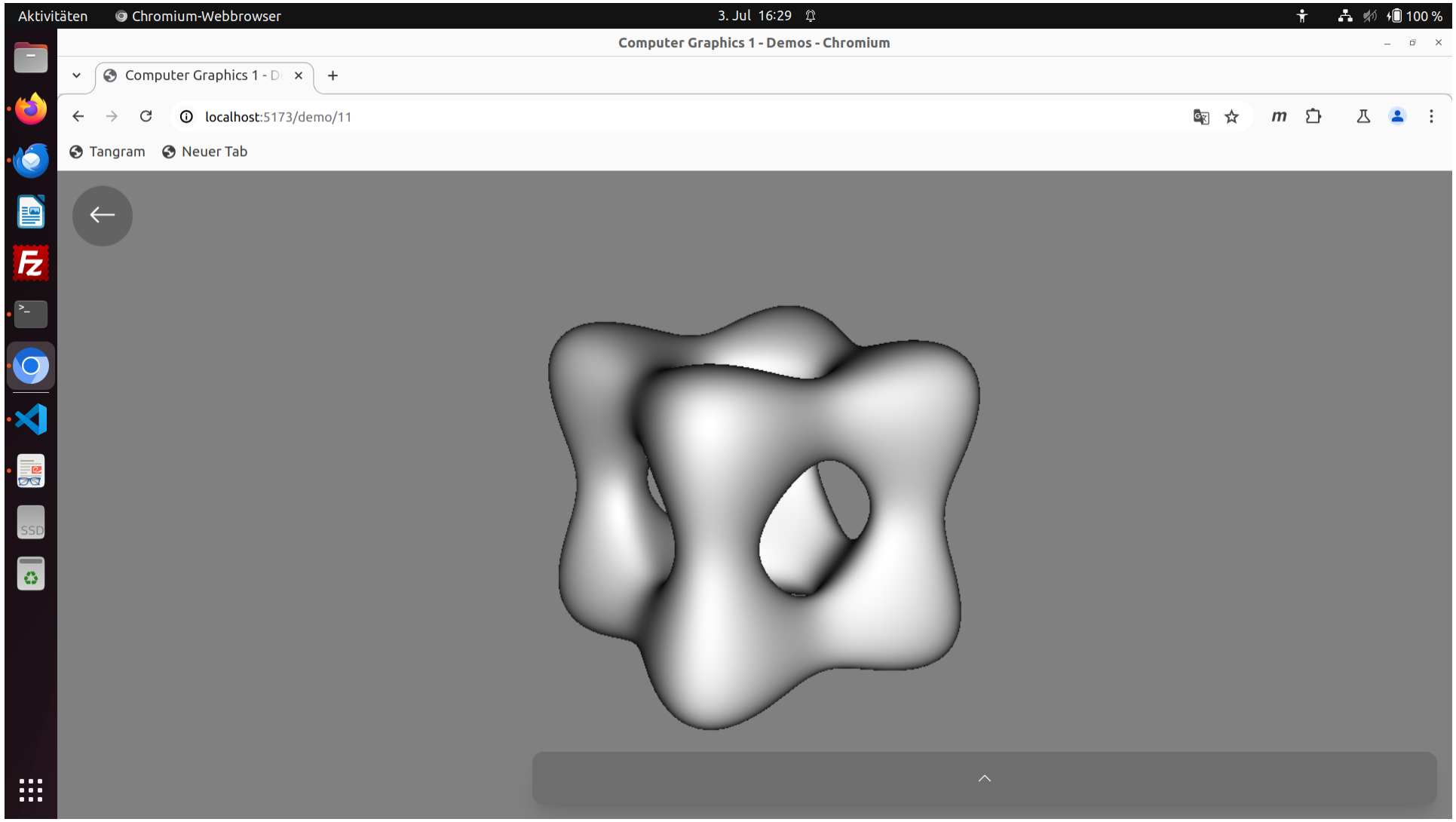
2D-Graphik - Schiebepuzzle



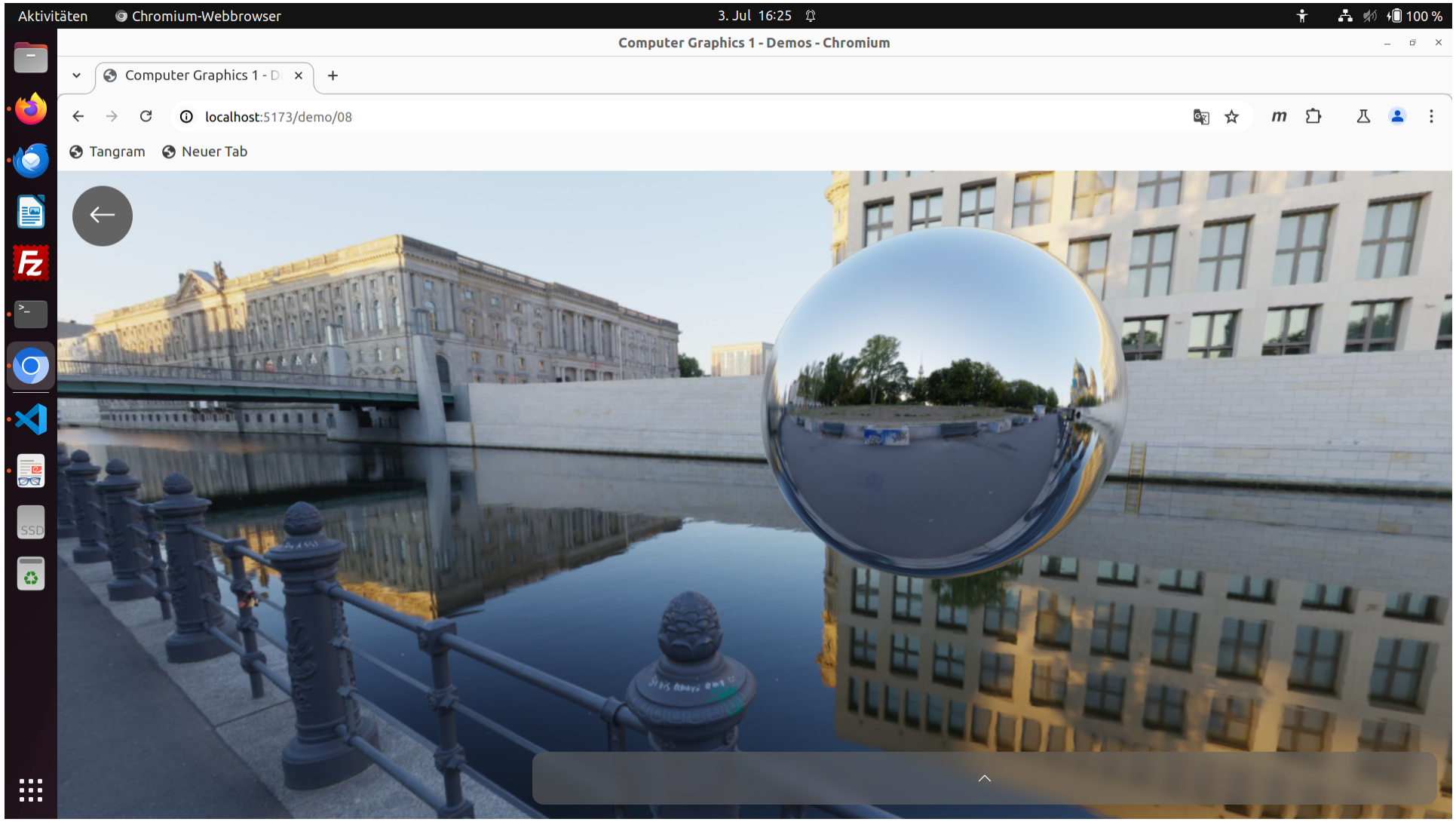
3D-Graphik + Stereo



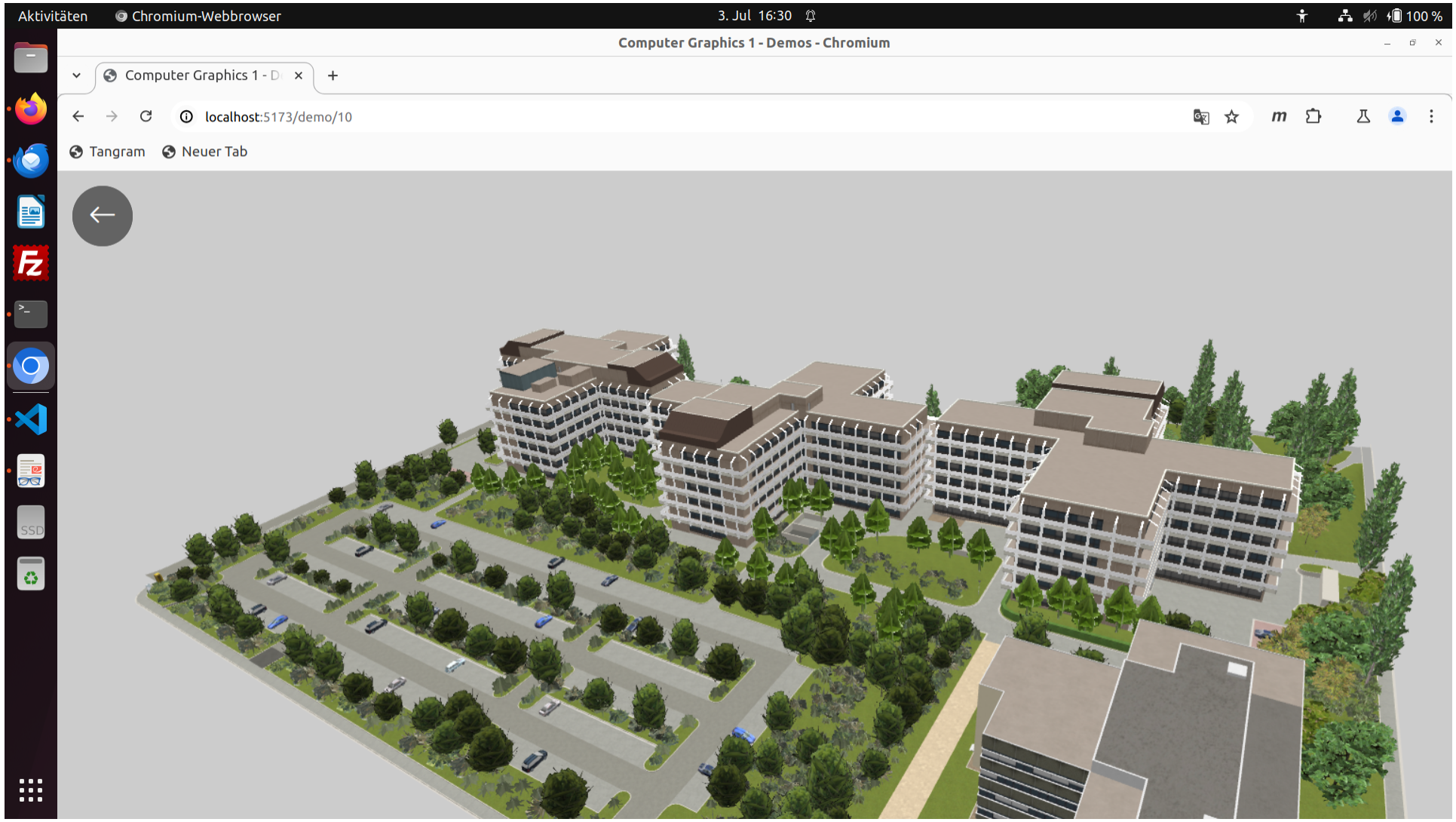
Implizite Flächen



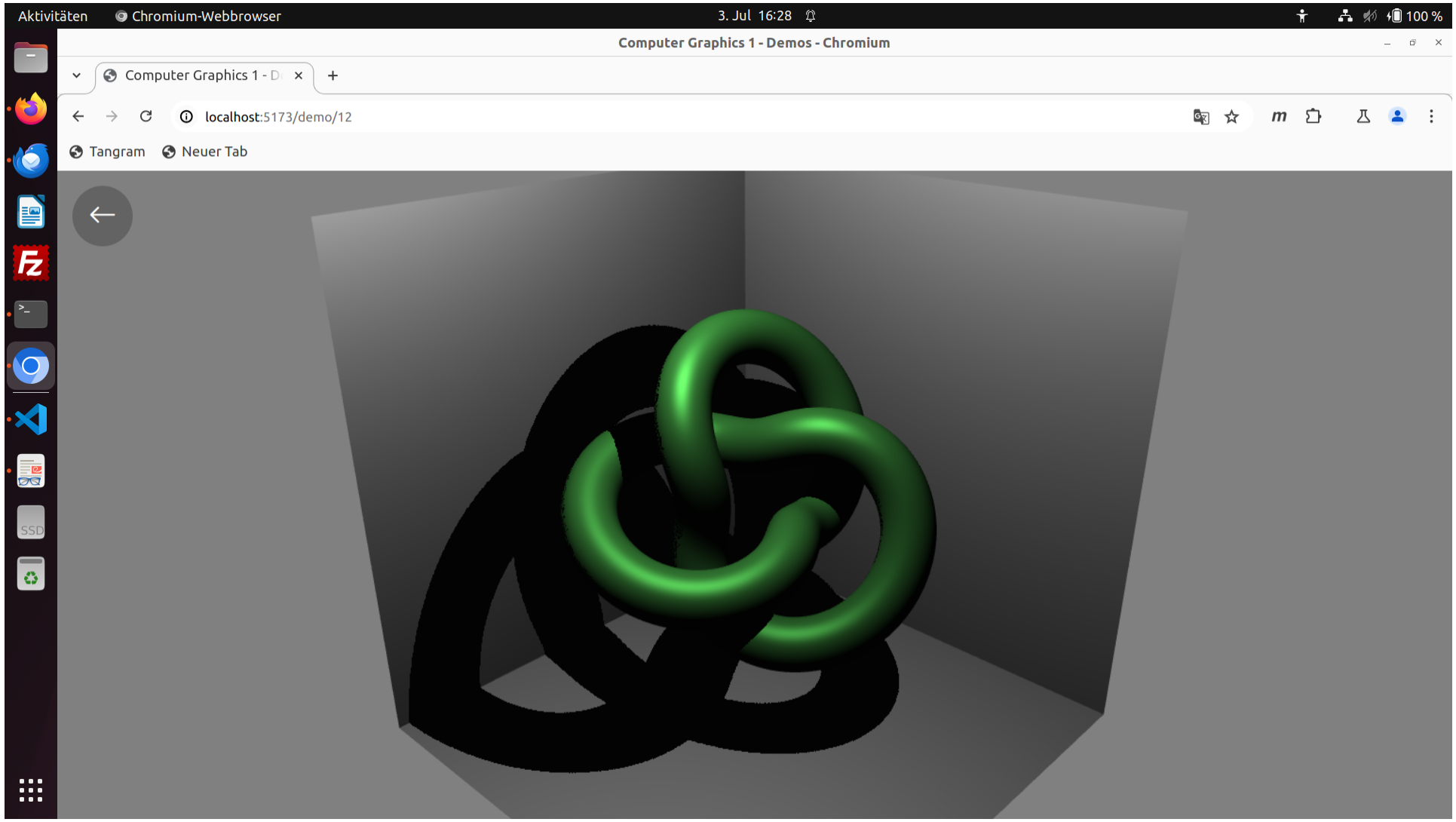
Environment Mapping



Szenegraphen



Schattenberechnung

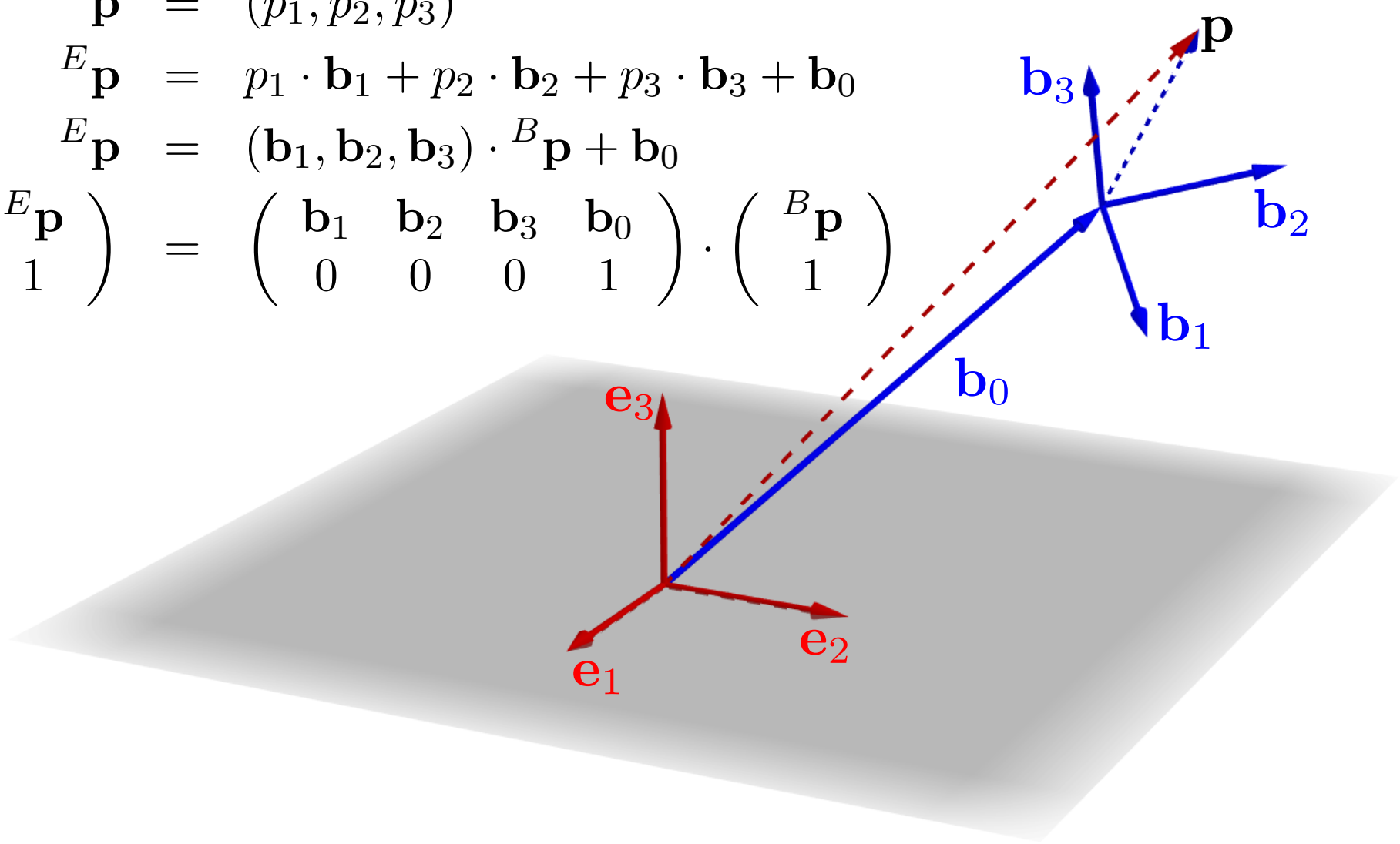


Rechenregeln für Vektoren - Kreuzprodukt

$$
\begin{aligned}
\mathbf{a} \times \mathbf{b} &= -\mathbf{b} \times \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix} \\
(\alpha\, \mathbf{a}) \times \mathbf{b} &= \alpha\, (\mathbf{a} \times \mathbf{b}) \\
\mathbf{a} \times (\mathbf{b} + \mathbf{c}) &= \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c} && \text{Distributivität} \\
|\mathbf{a} \times \mathbf{b}| &= \sin \sphericalangle(\mathbf{a}, \mathbf{b})\, |\mathbf{a}|\, |\mathbf{b}| \\
\mathbf{a}^T (\mathbf{b} \times \mathbf{c}) &= \det(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} && \text{Spatprodukt} \\
&= \mathbf{b}^T (\mathbf{c} \times \mathbf{a}) = \mathbf{c}^T (\mathbf{a} \times \mathbf{b}) \\
\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) &= \mathbf{b}(\mathbf{a}^T \mathbf{c}) - \mathbf{c}(\mathbf{a}^T \mathbf{b}) && \text{Graßmann-Identität} \\
(\mathbf{a} \times \mathbf{b})^T (\mathbf{c} \times \mathbf{d}) &= (\mathbf{a}^T \mathbf{c})(\mathbf{b}^T \mathbf{d}) - (\mathbf{a}^T \mathbf{d})(\mathbf{b}^T \mathbf{c}) && \text{Lagrange-Identität}
\end{aligned}
$$

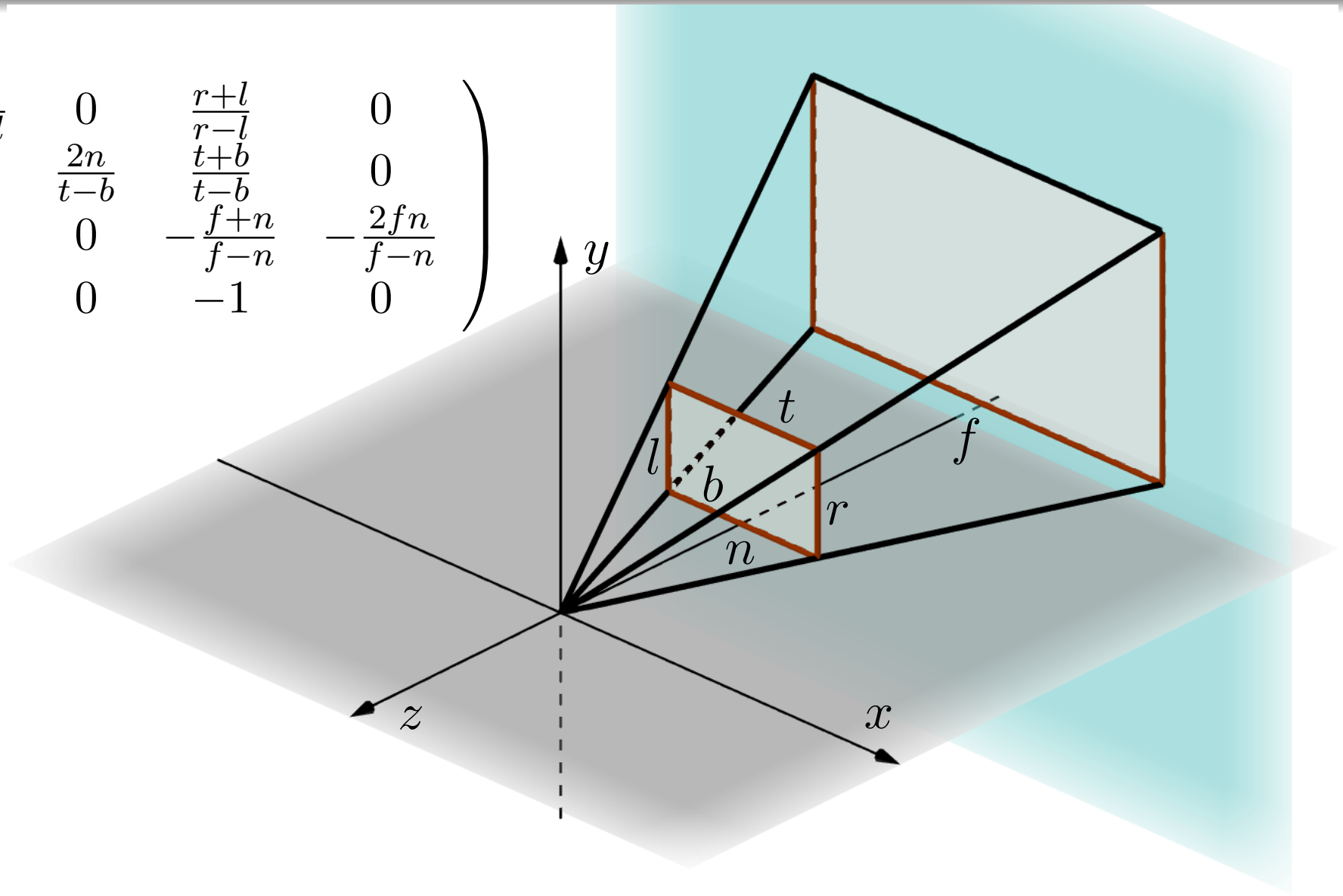
Koordinatentransformation

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



Perspektivische Projektion

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

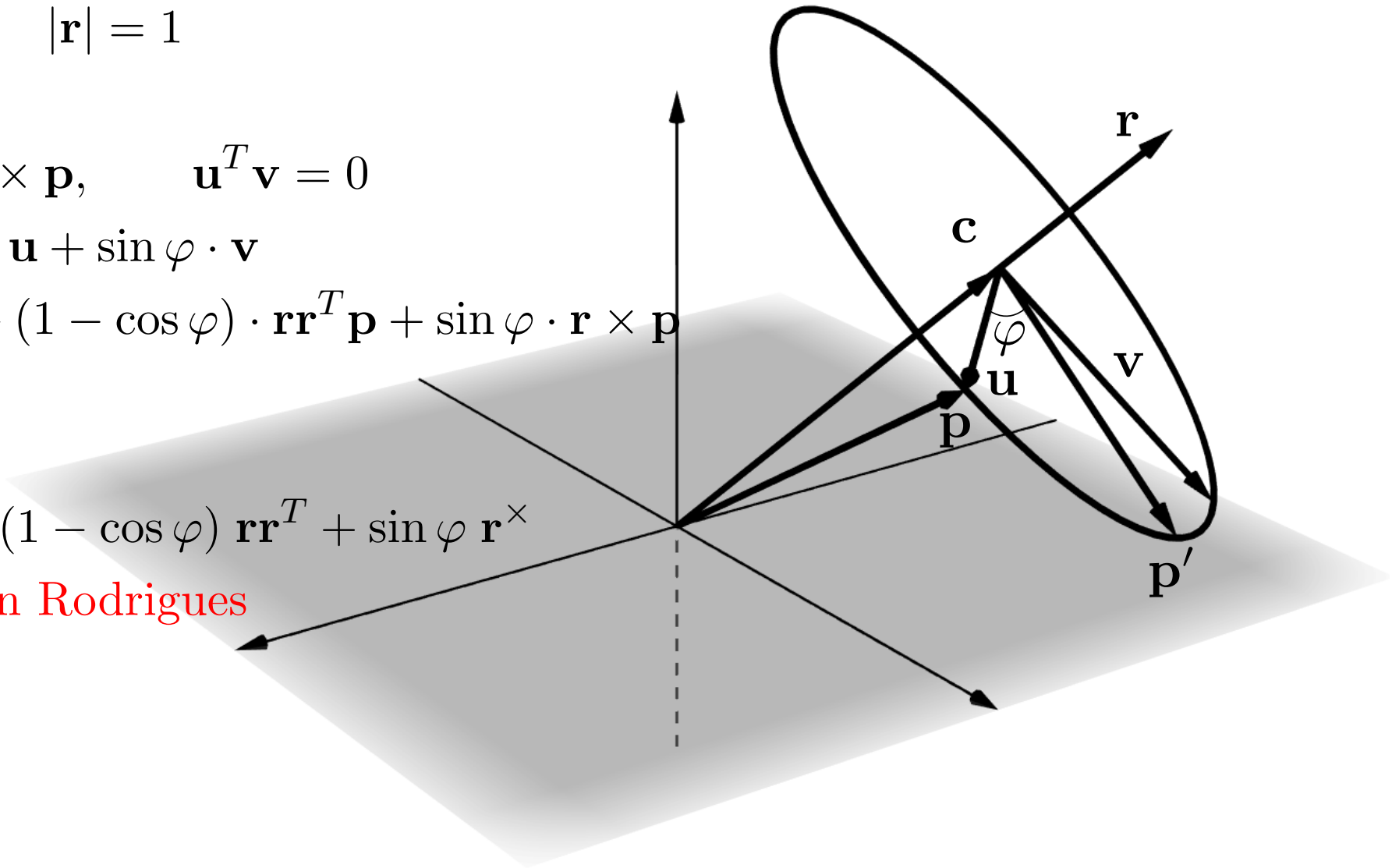


Drehung um eine Achse

$$
\begin{aligned}
\mathbf{c} &= \mathbf{r}(\mathbf{r}^T\mathbf{p}), \qquad |\mathbf{r}| = 1 \\
\mathbf{u} &= \mathbf{p} - \mathbf{c} \\
\mathbf{v} &= \mathbf{r} \times \mathbf{u} = \mathbf{r} \times \mathbf{p}, \qquad \mathbf{u}^T\mathbf{v} = 0 \\
\mathbf{p}' &= \mathbf{c} + \cos\varphi \cdot \mathbf{u} + \sin\varphi \cdot \mathbf{v} \\
\mathbf{p}' &= \cos\varphi \cdot \mathbf{p} + (1 - \cos\varphi) \cdot \mathbf{r}\mathbf{r}^T\mathbf{p} + \sin\varphi \cdot \mathbf{r} \times \mathbf{p} \\
\mathbf{p}' &= \mathbf{R} \cdot \mathbf{p}
\end{aligned}
$$

$$
\mathbf{R} = \cos\varphi\, \mathbf{E} + (1 - \cos\varphi)\, \mathbf{r}\mathbf{r}^T + \sin\varphi\, \mathbf{r}^\times
$$

Formel von Rodrigues



AG Scientific Computing and Bioinformatics

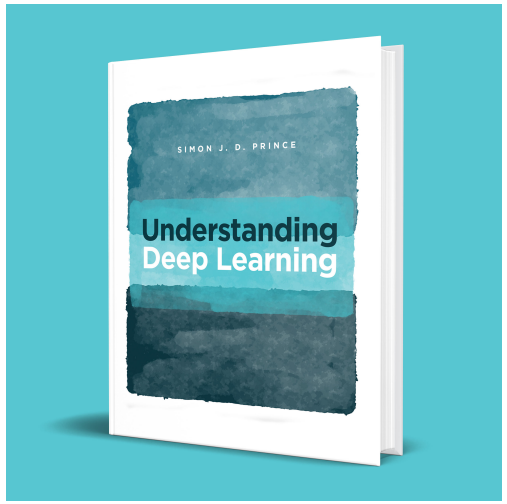
Lehrveranstaltungen im WiSe 2024/25

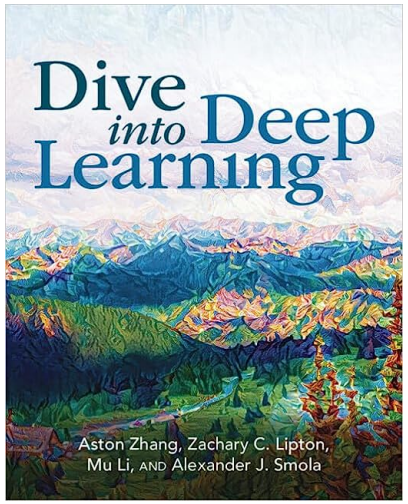
Prof. Dr. Andreas Hildebrandt

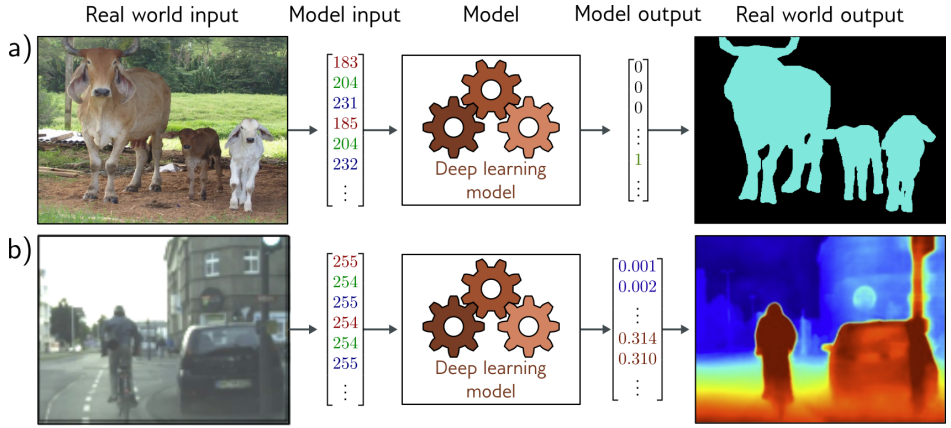
JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Deep Learning

- Dozent: Andreas Hildebrandt
- Termin: Donnerstags, 12-14 Uhr
- Sprache: Deutsch
- Inhalte:
  - (Deep) Neural Networks
  - Loss Functions
  - Fitting models (Gradient descent, training algorithm hyperparameters)
  - Backpropagation algorithm
  - Regularization
  - Convolutional networks
  - Transformers
  - Reinforcement learning

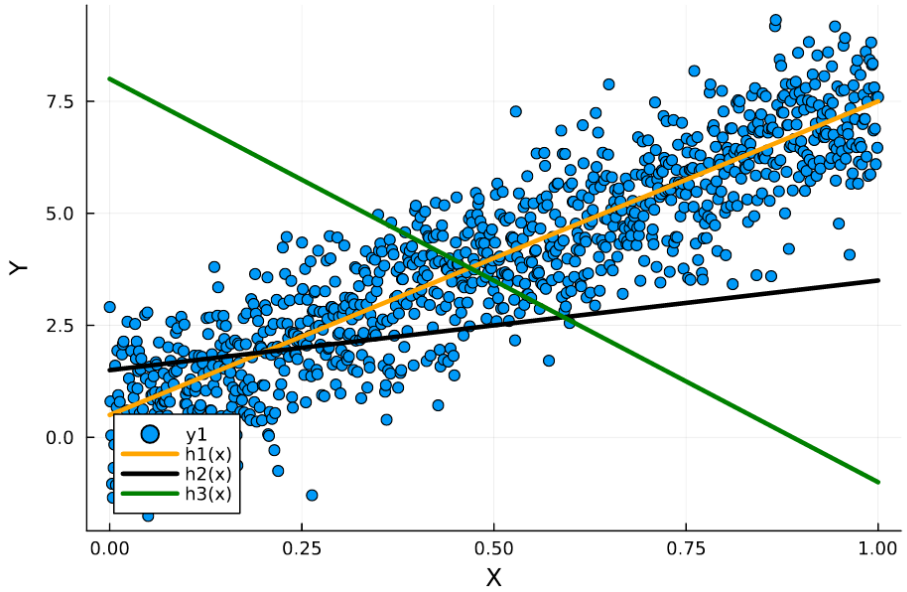


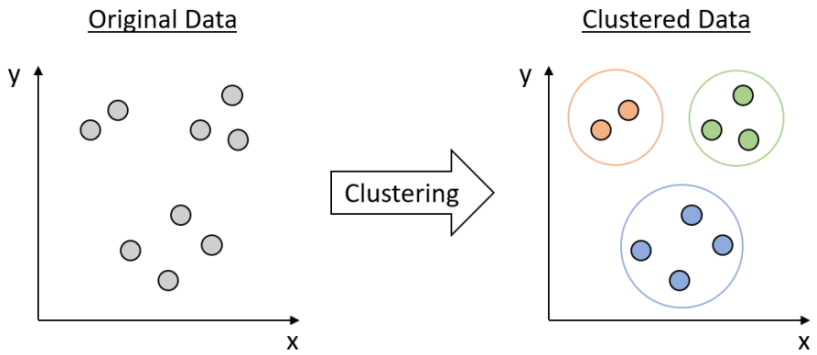




Bioinformatik-Praktikum

- Modus: Blockpraktikum (2 Wochen)
- Termin: TBA (vorlesungsfreie Zeit)
- Sprache: Deutsch, Materialien auf Englisch
- Inhalte:
  - Einführung in die Programmiersprache Julia
  - Arbeiten mit DataFrames
  - Datenanalyse
    - Datenaufbereitung
    - Lineare Modelle, Regression/Klassifikation, Clustering, ...
    - Visualisierung
  - Bearbeitung von Case Studies in Kleingruppen





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!

Einführung in die Bioinformatik

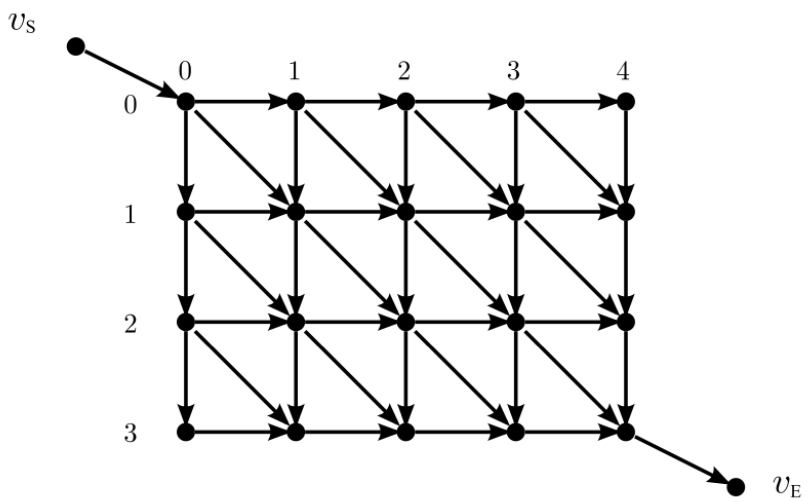
WiSe 2024/25

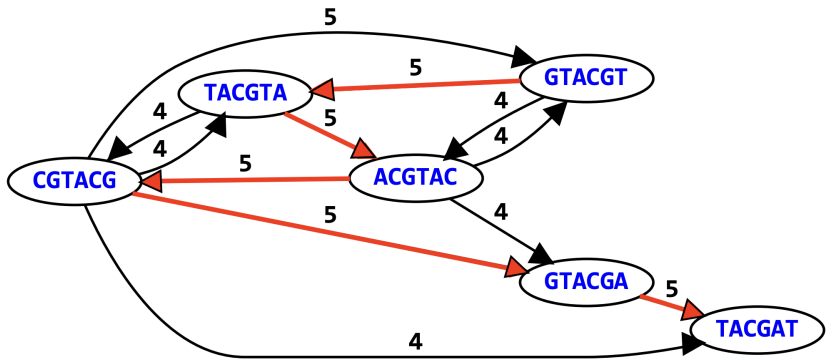
Dr. Nina Luhmann

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Einführung in die Bioinformatik

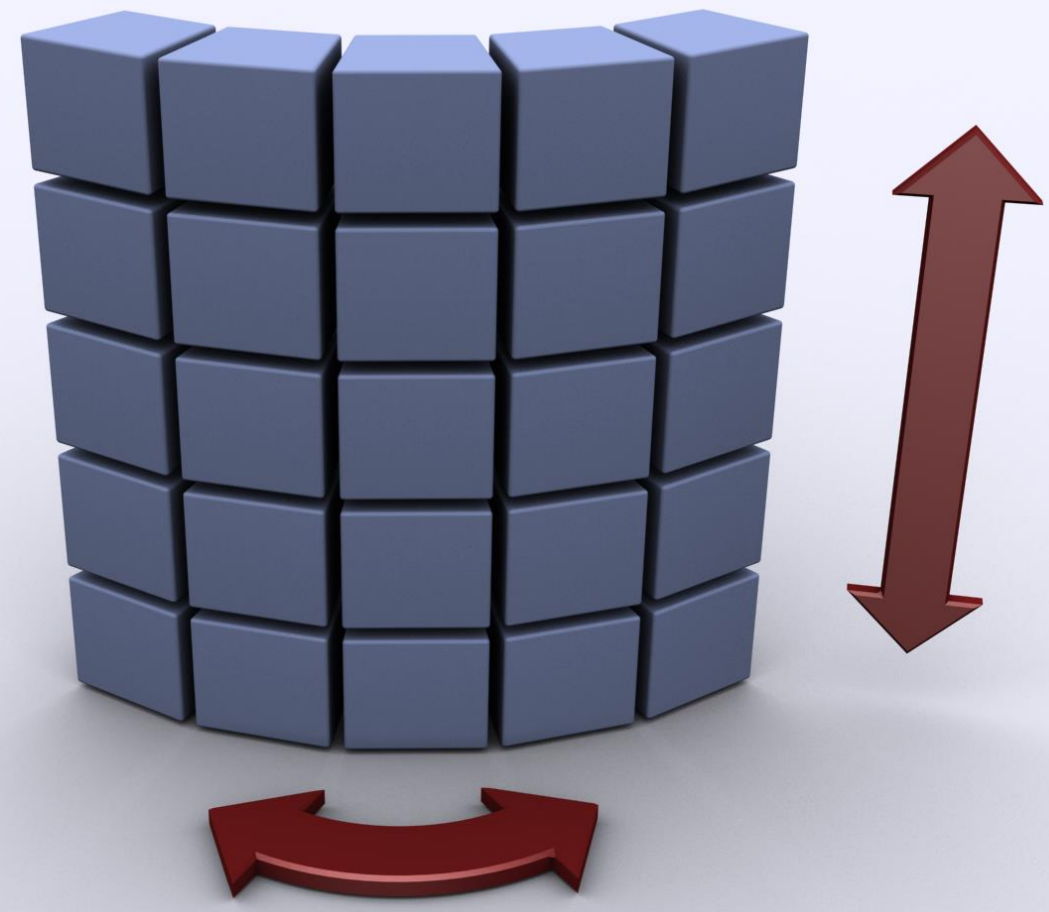
- Termin: Montags, 12-14 Uhr
- Sprache: Deutsch
- Inhalte:
  - Grundbegriffe der Bioinformatik
  - Komplexität von Problemen und Algorithmen
  - Algorithmen der sequenzbasierten Bioinfo
    - Paarweise Sequenzalignments
    - Multiple Sequenzalignments
    - Genomassemblierung
    - Phylogenetische Bäume





Lehrangebot Visual Computing

Wintersemester 2024/25



JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Institut für Informatik

Veranstaltungen Visual Computing

Vorlesungen

- Einführung in die Programmierung
- Deep Learning – Theorie und Praxis

Seminar & Praktikum

- Seminar Modellierung, Computer Vision & Machine Learning
  - Eine Woche im Block in den Semesterferien
- Praktikum Visual Computing
  - Zwei Wochen im Block in den Semesterferien
- Kombination mit Modellierung / DL / Computer Vision

Weitere Infos

URLs

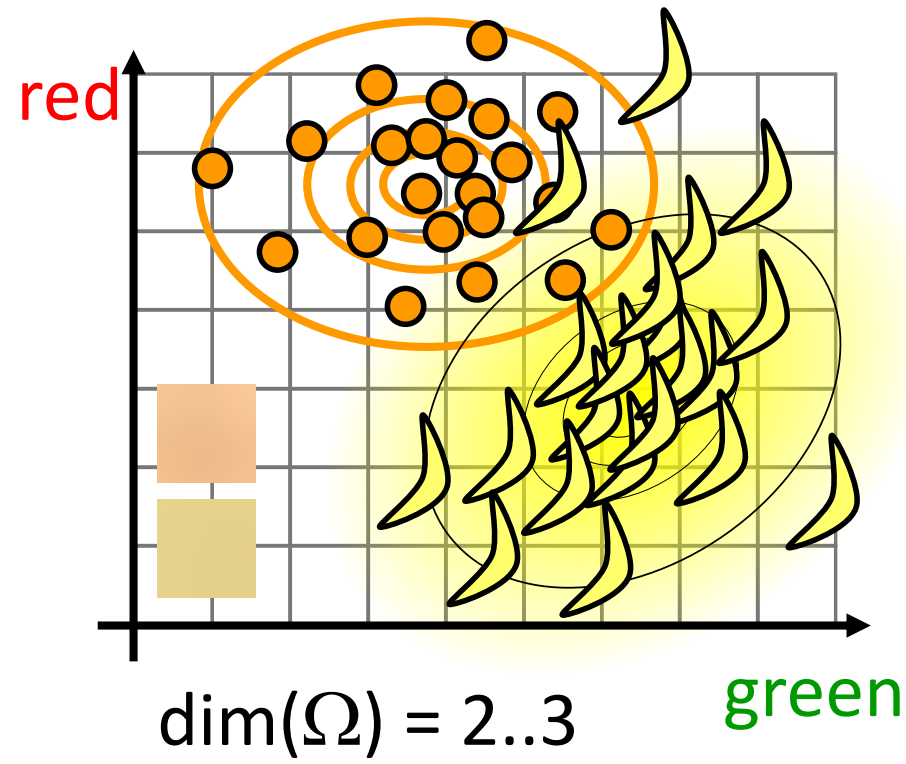
- Einführung in die Programmierung:
  luna.informatik.uni-mainz.de/eip-2425
- Deep Learning Theorie & Praxis:
  luna.informatik.uni-mainz.de/dl-2425
- Seminar+Praktikum:
  luna.informatik.uni-mainz.de/sem-prak-vc-2425

(spätestens Oktober 2024 online)

Deep Learning Theorie & Praxis

DL Theorie + Praxis

- Statistische Lerntheorie und Generalisierung
- Deep Learning Methoden
- Verständnis tiefer Netze
  - Informationstheorie
  - Geometrische Modelle von Information
  - Kernels + Gauß'sche Prozesse
  - Linearisierungen und Taylornäherungen
  - Ideen aus der statistischen Physik
  - Aktuelle Forschungsfragen und Ergebnisse



Abgrenzung

Deep Learning Praxis und Theorie (A. Hildebrandt)

- Wir machen vor allem Analyse + Theorie

Modellierung 2 „Statistische Datenanalyse“

- Umbenennung & Weiterentwicklung

Modellierung 1

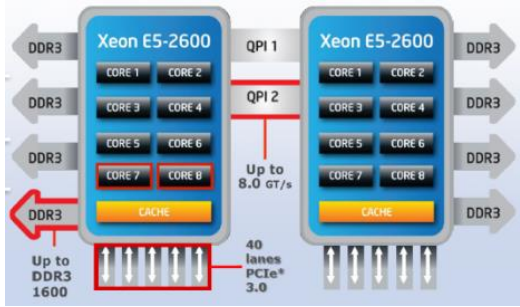
- Als Voraussetzung empfohlen
- Vorkenntnisse in ML (m.E. CV) gleichwertig

AG Schmidt: Lehrangebot: WiSe 24-25

- **Vorlesungen**
  - Accelerated Computing with GPUs (vormals PAA)
- **Praktika**
  - HPC Praktikum (HPC, SoSe24)
  - Paralleles Programmieren mit CUDA (AccComp, WiSe 24-25)

Vorlesung: Accelerated Computing with GPUs (*PAA*)

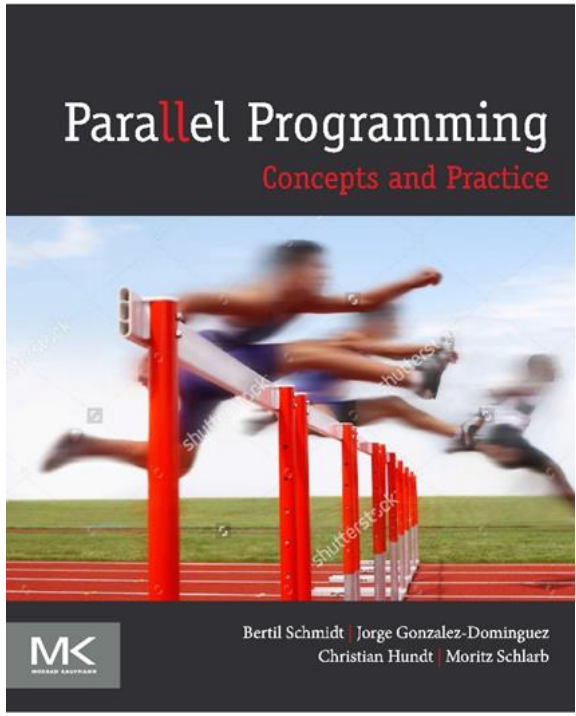
- <u>Ehemals:</u> “Parallel Algorithms and Architectures” (PAA)
- Architekturen
  - **GPUs** und **GPU-Cluster**
  - PRAM
- Programmierung
  - **<u>CUDA</u>**
  - OpenACC
  - MPI + CUDA
  - Viele praktische Übungen
- Algorithmen
  - Matrix Multiplikation, Sortieren, Reduktion, SpMV, Convolution, Data Mining, etc.



CPUs

GPUs

Clusters



Vorlesung: Accelerated Computing with GPUs (*PAA*)

- <u>1. Anmeldephase</u>: 1.7.24 13 Uhr – 18.7.24 13 Uhr

- Umfrage zur Findung von Übungsterminen
  - Nach erster Anmeldephase für alle bereits Angemeldeten
  - Genauere Infos folgen in Moodle

Block-Praktikum: HPC Praktikum (gehört zu SoSe 24)

- Paralleles Programmieren mit MPI, OpenMP, C++ multi-threading
  - **19.-30.08.2024** (Anmeldung bis 2.8.24 möglich)
  - **<u>Voraussetzung:</u>** Erfolgreiche Teilnahme an HPC

Block-Praktikum: Paralleles Programmieren (mit CUDA)

- **Hinweis:** Veranstaltung Ende WiSe 24/25
- Paralleles Programmieren mit CUDA
  - **Ende März / Anfang April 2025 (TBC)** – 2 Wochen
  - **<u>Voraussetzung:</u>** Erfolgreiche Teilnahme an *„Accelerated Computing with GPUs“*

Topics in Functional Programming

JG|U

People

Prof. Dr. Sebastian Erdweg and Dr. Prashant Kumar

Why Functional Programming?

- **New paradigm** – Recursion, higher order functions, and function composition vs loops and statefulness.

```
quickSort :: Ord a => [a] -> [a]
quickSort [] = []
quickSort (x:xs) = quickSort ltxs ++ [x] ++ quickSort gtxs
    where ltxs = filter (<= x) xs
          gtxs = filter (> x) xs
```

- **Better Paradigm for Programming**
  - Strong typing => less debugging time

  - function composition allows modular program development.

    f = sum . map square. filter odd

Why Functional Programming?

```
quickSort :: Ord a => [a] -> [a]
quickSort [] = []
quickSort (x:xs) = quickSort ltxs ++ [x] ++ quickSort gtxs
    where ltxs = filter (<= x) xs
          gtxs = filter (> x) xs
```

- **Prove properties about Program –** Induction proofs

  `length xs = length (quicksort xs)`

Use these properties for testing – QuickCheck library in Haskell

Why Functional Programming?

- Succinct Code

Haskell vs. Ada vs. C++ vs. Awk vs. ...
An Experiment in Software Prototyping Productivity*

Paul Hudak
Mark P. Jones

Yale University
Department of Computer Science
New Haven, CT 06518
{hudak-paul,jones-mark}@cs.yale.edu

July 4, 1994

Abstract

We describe the results of an experiment in which several conventional programming languages, together with the functional language Haskell, were used to prototype a Naval Surface Warfare Center (NSWC) requirement for a *Geometric Region Server*. The resulting programs and development metrics were reviewed by a committee chosen by the Navy. The results indicate that the Haskell prototype took significantly less time to develop and was considerably more concise and easier to understand than the corresponding prototypes written in several different imperative languages, including Ada and C++.

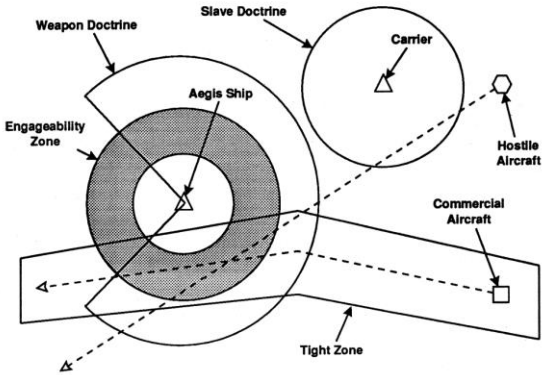


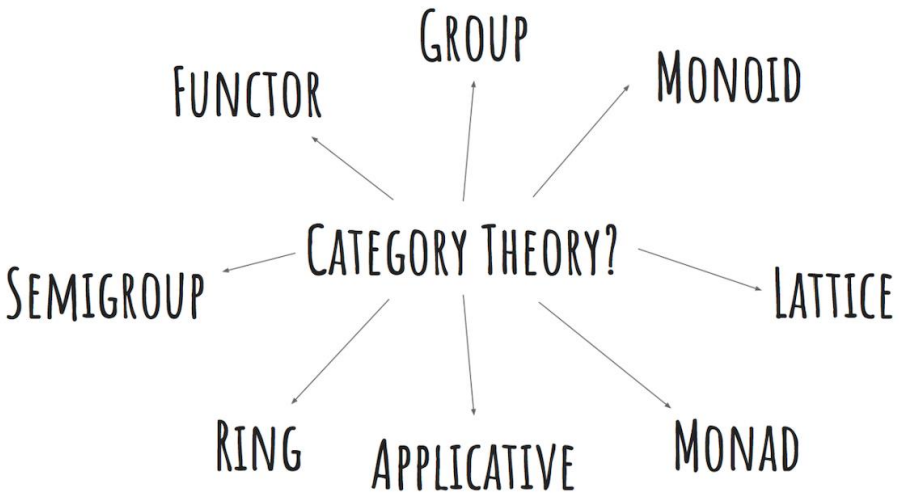
Figure 2: Geo-Server Input Data

Language	Lines of code	Lines of documentation	Development time (hours)
(1) Haskell	85	465	10
(2) Ada	767	714	23
(3) Ada9X	800	200	28
(4) C++	1105	130	–
(5) Awk/Nawk	250	150	–
(6) Rapide	157	0	54
(7) Griffin	251	0	34
(8) Proteus	293	79	26
(9) Relational Lisp	274	12	3
(10) Haskell	156	112	8

Figure 3: Summary of Prototype Software Development Metrics

Why Functional Programming?

- **Better Programming Abstractions –** use Maths to craft better programs



Organisation of the course

3 Parts:

- Functional programming and design of better programming abstractions
- Functional data structures
- Compilers of functional languages

Trustworthy AI

Fairness, Interpretability and Privacy

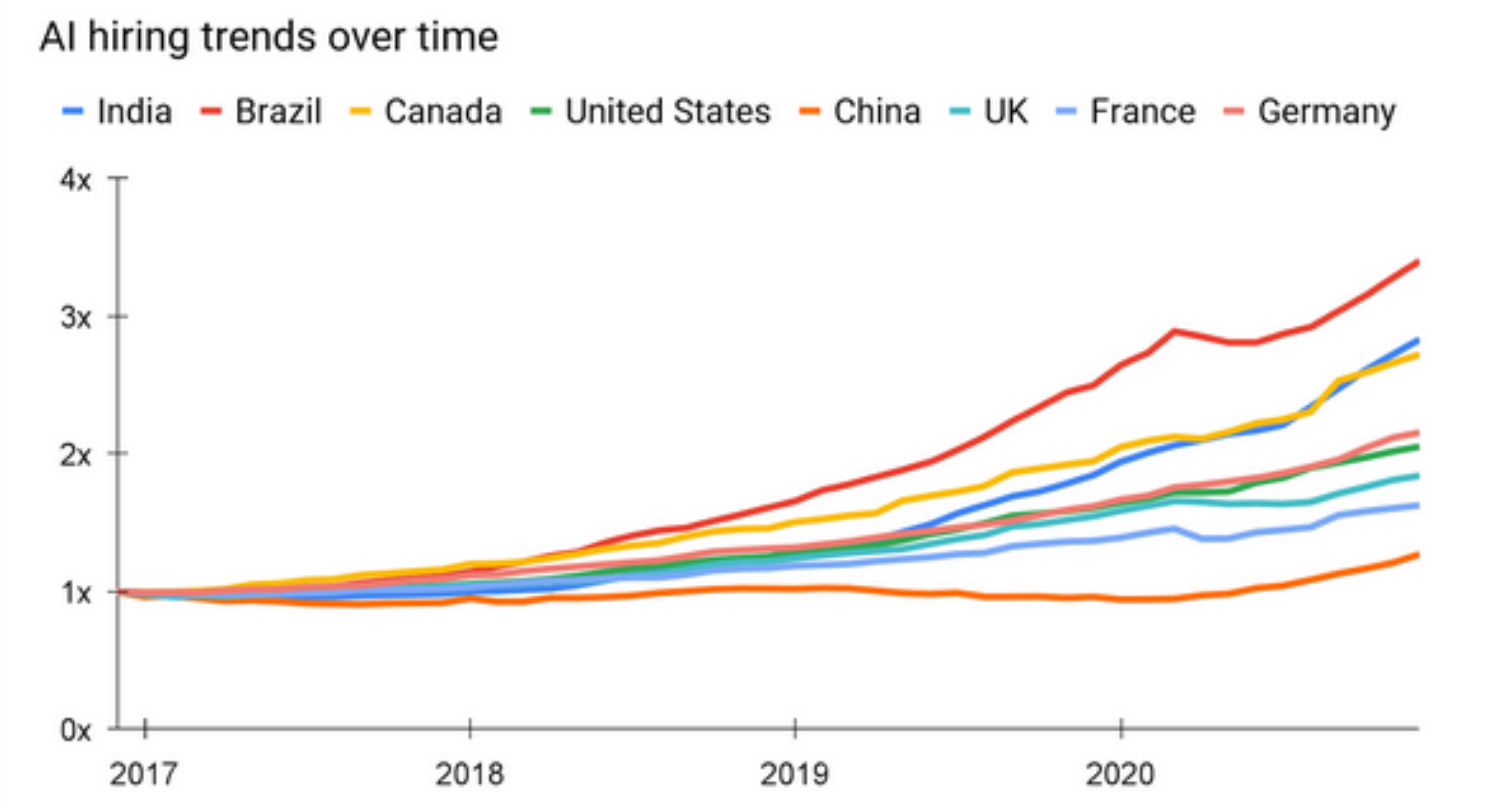
Mattia Cerrato, PhD.
mcerrato@uni-mainz.de
pibborn.github.io/trust-class

Machine Learning

Machine Learning-powered applications are having a huge impact in the economy and our everyday lives

ML powers: social media, search engines, healthcare

Machine Learning



Source: state of AI report 2021

A few ML success stories

Automatic classification of images in your "Gallery" app via **object detection**

A few ML success stories

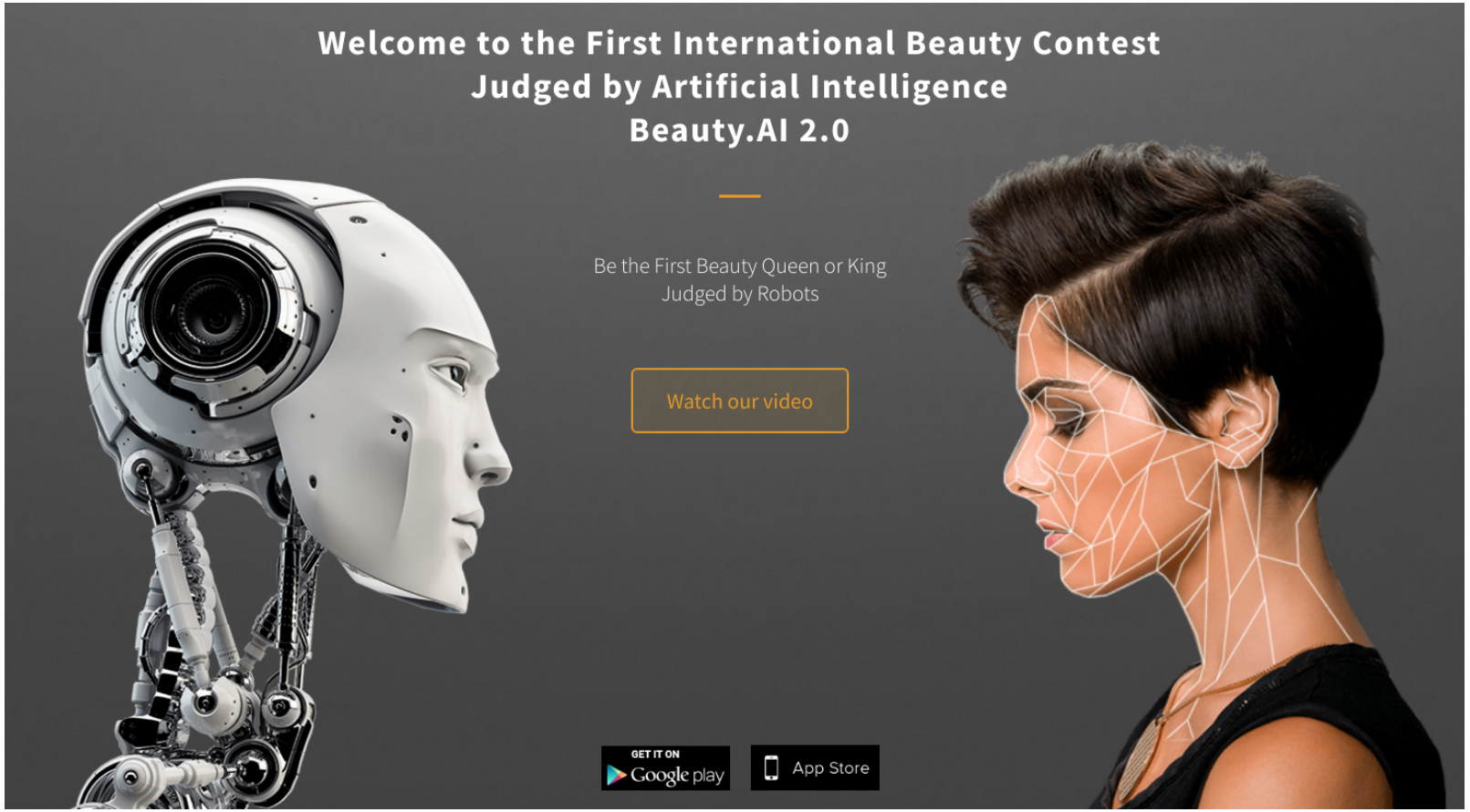
Forbes , black individuals misclassified

Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software

Maggie Zhang Forbes Staff

Tech

I write about technology, innovation, and startups.



A few ML success stories

A "beauty pageant algorithm"

This article is more than **4 years old**

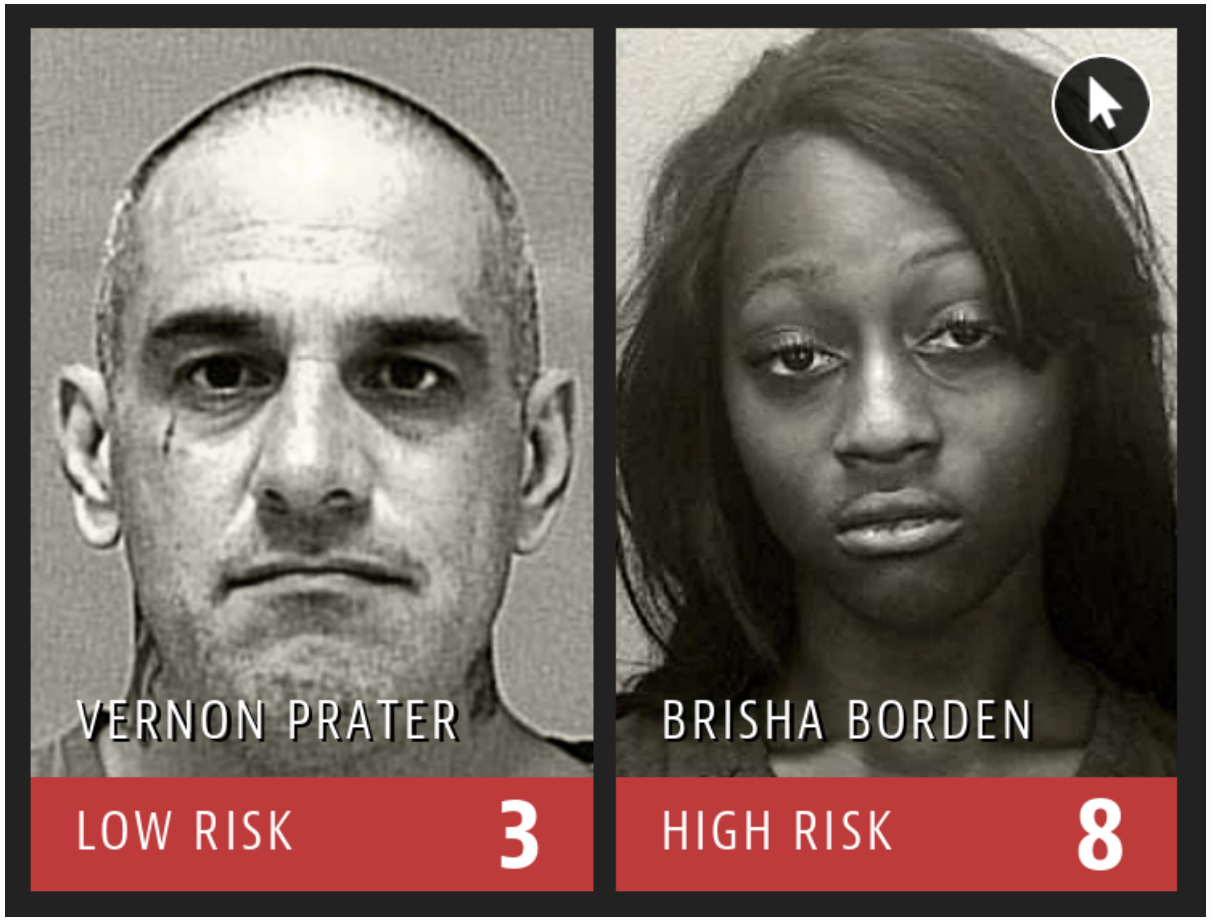
A beauty contest was judged by AI and the robots didn't like dark skin

The first international beauty contest decided by an algorithm has sparked controversy after the results revealed one glaring factor linking the winners

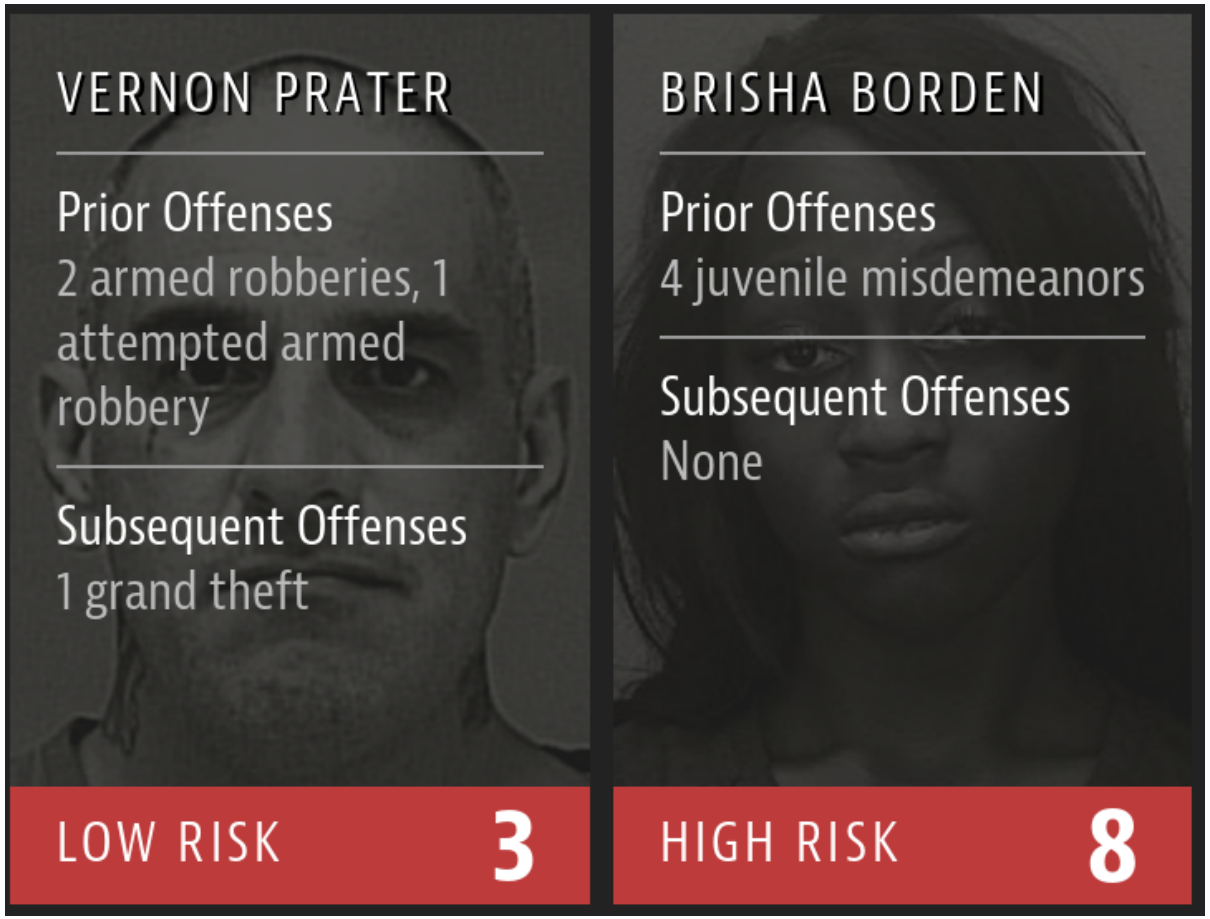
A few ML success stories

Guardian, mostly white winners in the AI Beauty Pageant

A few ML success stories



A few ML success stories

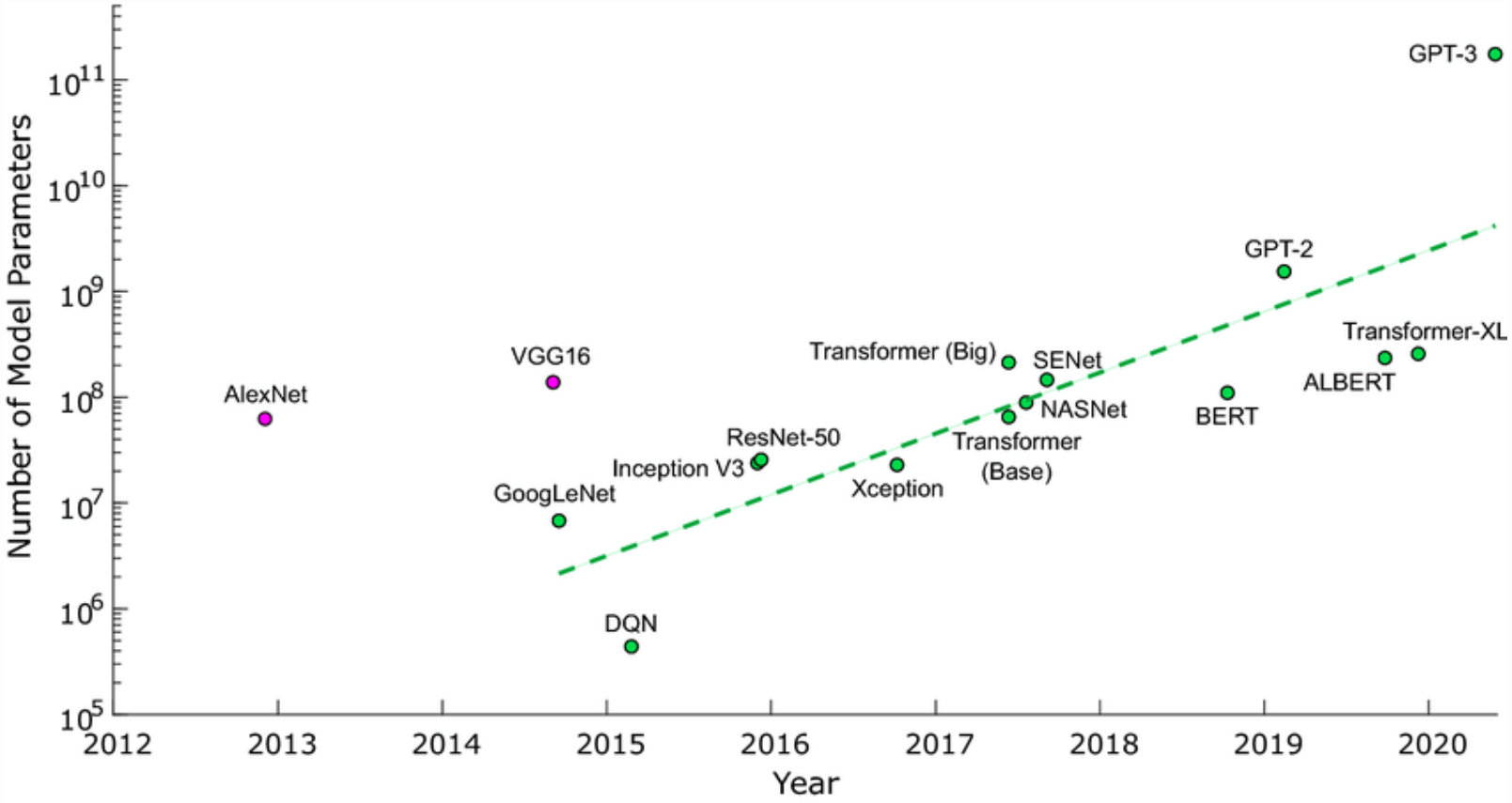


How does this happen?

A very short list of reasons:

- **Biases** in the training data
- **Opacity** in the models

Free parameters



From Bernstein et al. 2021, Nature Scientific Reports

Trustworthy AI

A broad term for AI/ML applications which output we can **trust**.

Three possible, non-exhaustive desiderata:

- **Fairness**: how can we learn non-discriminating models from biased data?
- **Interpretability**: how can we make sense of a model's decisions?
- **Privacy**: how do we make sure that a model trained with our personal data does not leak our data to third parties?

Trustworthy AI

During this class, you will learn about the current state of ML research and discussion on the topics of interpretability, fairness and privacy.

Objective: preparing students for the challenges of a ML/Data Science job beyond optimizing performance metrics

Class details

6 Credits

Lectures: Tuesdays 10-12

Tutorials: TBD, usually maximizing students availability

Best (but not compulsory) to have taken Machine Learning or Data Mining

Class details

Final Grade: 75% Oral Examination + 25% Exercises

Exercises are a mix of programming assignments, essay writing and live discussion

Trustworthy AI

TOPML: Trading Off Non-Functional Properties of Machine Learning

Trustworthy AI

Fairness, Interpretability and Privacy

Mattia Cerrato, PhD.
mcerrato@uni-mainz.de
pibborn.github.io/trust-class
pibborn.github.io/projects

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Modellierung mit Differentialgleichungen

(Gewöhnliche Differentialgleichungen, 3V+2Ü)

Peter Spichtinger

Institut für Physik der Atmosphäre, JGU Mainz

3. Juli 2024

Differentialgleichungen zur Modellbildung

Viele physikalische Prozesse können mit Differentialgleichungen beschrieben werden.

Einfache Beispiele:

- Schwingendes Pendel

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} + \kappa \frac{\mathrm{d}x}{\mathrm{d}t} + \omega^2 x = 0 \tag{1}$$

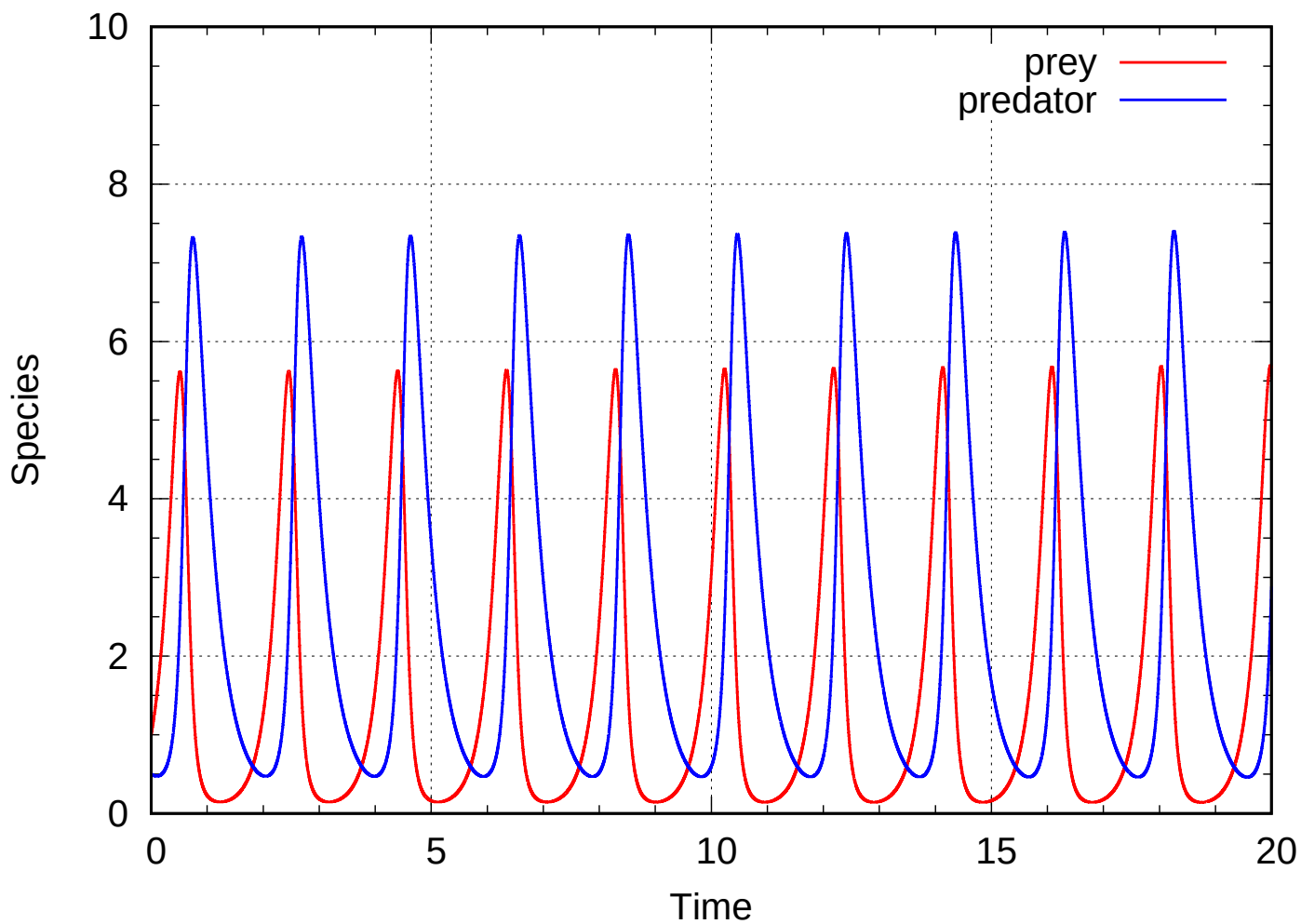
- Wärmeleitung in einem Stab

$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2} \tag{2}$$

Räuber-Beute-Modell

$$\frac{\mathrm{d}x}{\mathrm{d}t} = ax - bxy \qquad \text{Beute} \tag{3}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -cy + bxy \qquad \text{Räuber} \tag{4}$$

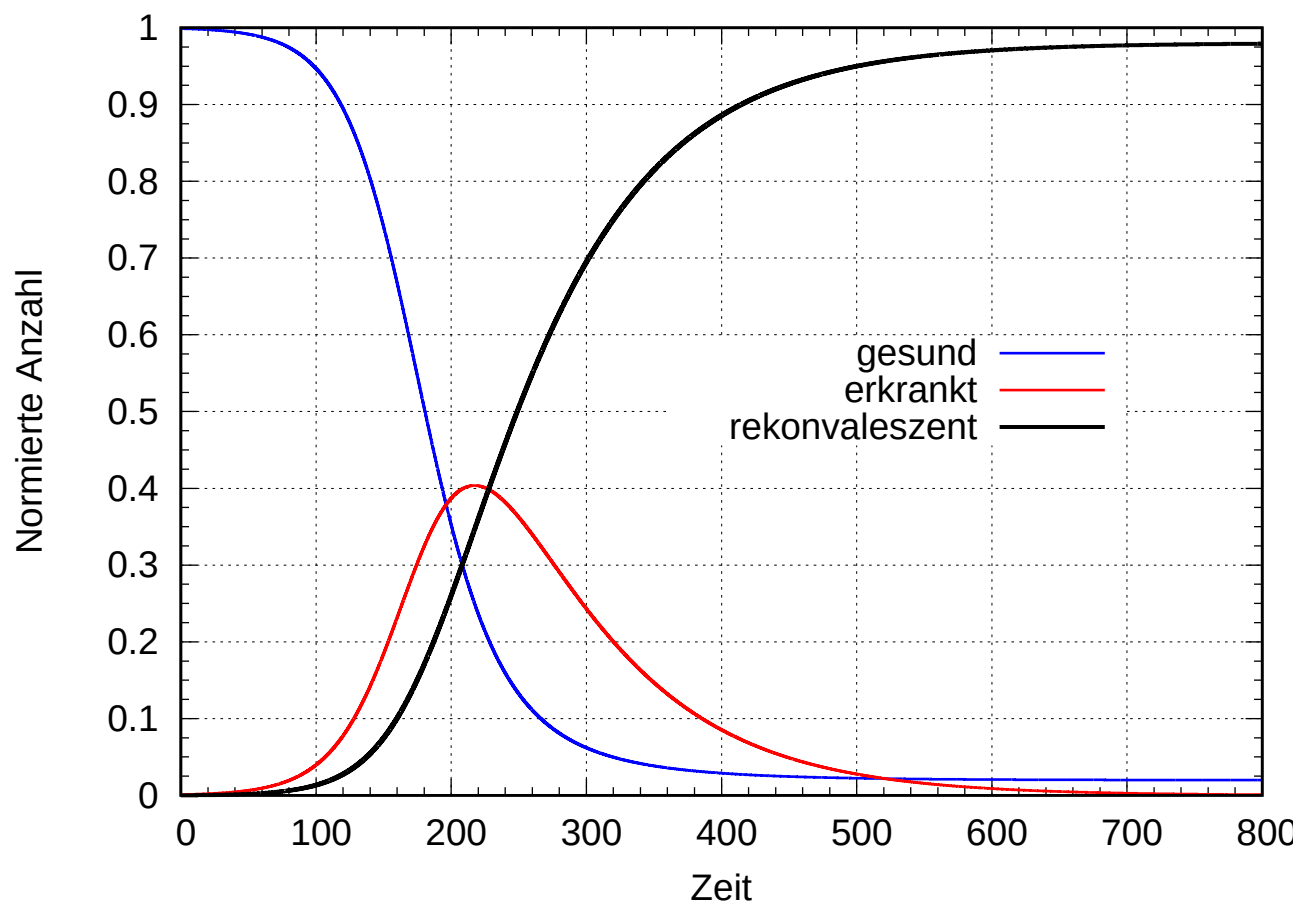


Epidemiemodell

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -\lambda x y \qquad \text{gesund} \tag{5}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \lambda x y - \gamma y \qquad \text{erkrankt} \tag{6}$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \gamma y \qquad \text{rekonvaleszent} \tag{7}$$

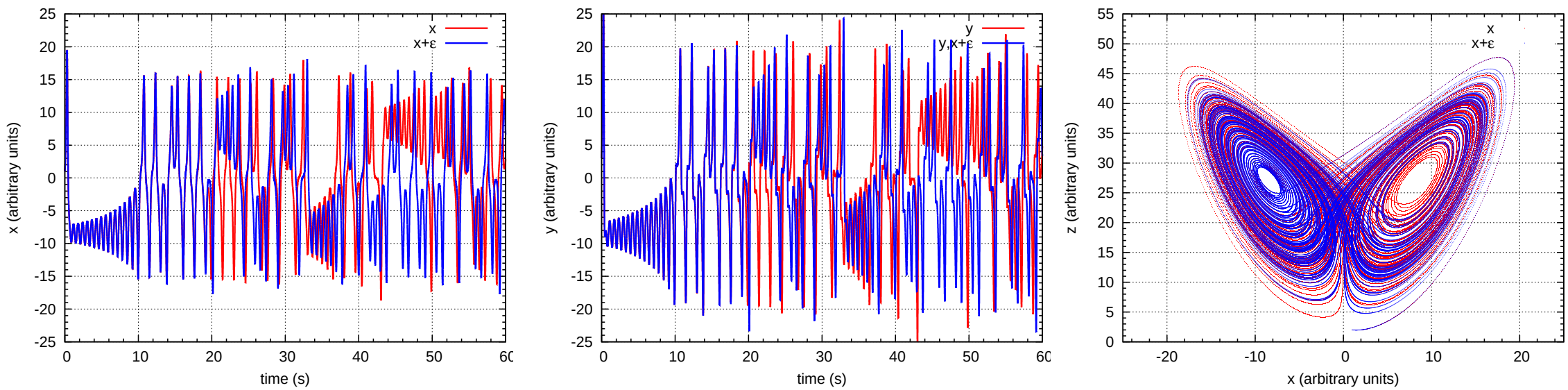


Lorenz-System (chaotisches Atmosphärenmodell)

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x) \tag{8}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = (r - z)x - y \tag{9}$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - bz \tag{10}$$



Spezialisierung auf gewöhnliche Differentialgleichungen (ODEs)

Themen

- Nutzen und Grenzen von Modellen
- Qualitative Theorie und Analyse von ODEs
- Asymptotische Analyse
- Numerische Methoden und Simulationen

Organisatorisches

Vorlesungstermine (in Präsenz, + Übung, noch festzulegen)

- Mittwoch 12:30 -14:00 Uhr (14-tägig)
- Freitag 10:15 - 11:45 Uhr

Vorlesung findet in der NatFak (Becherweg 21) statt

Geeignet für Studierende (Ende) Bachelor oder Master

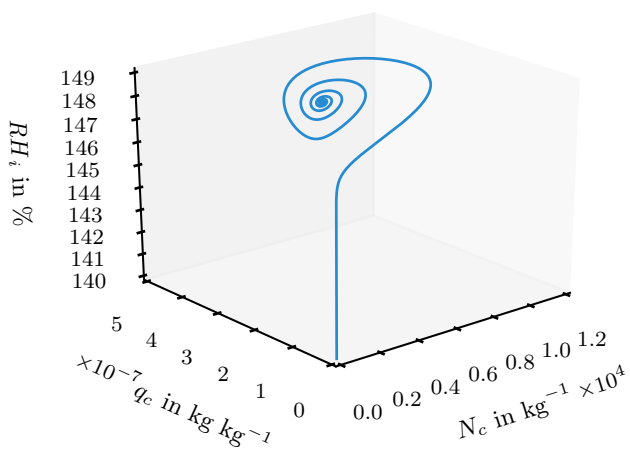
bei weiterem Interesse Fortsetzung im Sommersemester durch Modellierung mit partiellen Differentialgleichungen

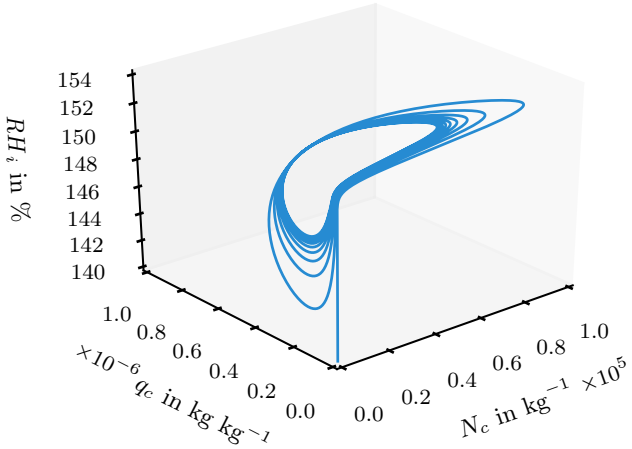
Viele Anwendungsbeispiele aus Physik/Chemie/Biologie

- Räuber-Beute-Modelle
- Epidemiemodelle
- Chemische Reaktionen
- Einfache Atmosphärenmodelle
- Nichtlineare Oszillatoren (Synchronisation)

Analysetechniken aus Theorie dynamischer Systeme

- Fixpunkte
- Grenzzyklen
- Übergang zum Chaos





Vorlesung Modellierung - Themen

Approximationstechniken mit Asymptotik

- Reguläre/singuläre Störungstheorie
- Mehrskalenasymptotik
- Matched asymptotics

Numerische Methoden

- Explizite/implizite Verfahren
- Konvergenz
- Stabilität
- Simulation von einfachen Modellsystemen

Vorlesung Modellierung - Themen

Approximationstechniken mit Asymptotik

- Reguläre/singuläre Störungstheorie
- Mehrskalenasymptotik
- Matched asymptotics

Numerische Methoden

- Explizite/implizite Verfahren
- Konvergenz
- Stabilität
- Simulation von einfachen Modellsystemen

Fragen?

peter.spichtinger@uni-mainz.de