

# Modulhandbuch

---

## Master Informatik

---

**Studienordnungsversion: 2021**

**gültig für das Sommersemester 2021**

Erstellt am: 11. Mai 2021  
aus der POS Datenbank der TU Ilmenau  
Herausgeber: Der Präsident der Technischen Universität Ilmenau  
URN: urn:nbn:de:gbv:ilm1-mhb-22058



<b>Wahlbereich Fortgeschrittene Mathematik</b>					FP	5
Graphen & Algorithmen					PL 30min	5
Graphentheorie					PL 30min	10
Kombinatorik					PL 30min	10
Kombinatorische Optimierung					PL 30min	10
Large Networks & Random Graphs					PL 30min	5
Angewandte Optimierung					PL 30min	5
Numerik 1					PL 30min	10
Ordnungs- und Verbandstheorie					PL 30min	5
<b>Nichttechnisches Nebenfach</b>					FP	5
Lehrveranstaltung aus dem aktuellen Katalog					PL	5
<b>Fachpraktikum</b>					FP	30
Fachpraktikum					SL 20	30
<b>Masterarbeit</b>					FP	30
Abschlusskolloquium zur Master-Arbeit					PL 30min	6
Masterarbeit					MA 6	24

## Modul: Effiziente Algorithmen

Modulabschluss: Prüfungsleistung mündlich 25 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Pflichtmodul Turnus: Wintersemester

Modulnummer: 200065

Prüfungsnummer: 2200714

Modulverantwortlich: Prof. Dr. Martin Dietzfelbinger

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2242							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

**Fach- und Methodenkompetenz:** Die Studierenden kennen einige wesentliche fortgeschrittene Algorithmen und die hierfür notwendigen Entwurfs- und Analysetechniken. Sie können mit den erlernten Techniken Algorithmen für abgewandelte Fragestellungen entwerfen und analysieren. Sie können Algorithmen auch auf nicht offensichtliche Anwendungsfragestellungen übertragen. Sie können eine amortisierte Laufzeitanalyse durchführen, wenn die wesentlichen Festlegungen angegeben sind. Die Studierenden kennen die vielfältige Anwendbarkeit von Flussalgorithmen. Sie kennen nichttriviale grundlegende Techniken für die Verarbeitung von Wörtern (Textsuche) und die relevanten Beweistechniken.

In der Vorlesung konnten die notwendigen Kenntnisse und Methoden erworben werden; durch die Übungen sind die Studierenden darin geübt, die Methoden in neuen Aufgabenstellungen selbst anzuwenden und dabei eigene, selbständige, gut begründete Überlegungen anzustellen, im Rahmen der erlernten Methoden und des Standes der Technik.

**Sozialkompetenz:** Die Studierenden haben in den Übungen die Gelegenheit genutzt, eigene Lösungen zu präsentieren und damit der Diskussion in der Gruppe auszusetzen. Wertschätzende Diskussion durch die Gruppe wurde angeleitet, beim Vortrag konnten die Studierenden wertvolle Erfahrung in der Rolle der Präsentierenden machen.

### Vorkenntnisse

Bachelorstudium Informatik, insbesondere:  
 Algorithmen und Programmierung  
 Algorithmen und Datenstrukturen 1 und 2  
 Mathematik 1 und 2  
 Grundlagen und Diskrete Strukturen

### Inhalt

Flussprobleme und -algorithmen: Ford-Fulkerson-Methode, Algorithmus von Edmonds/Karp, Sperrflussmethode (Algorithmus von Diniz).

Matchingprobleme und ihre Algorithmen: Kardinalitätsmatching, Lösung über Flussalgorithmen, Algorithmus von Hopcroft/Karp; gewichtetes Matching: Auktionsalgorithmus, Ungarische Methode; Stabile Paarungen: Satz von Kuhn/Munkres, Algorithmus von Gale/Shapley.

Amortisierte Analyse von Datenstrukturen: Ad-Hoc-Analyse, Bankkontomethode, Potentialmethode.

Implementierung von adressierbaren Priority Queues: Binomialheaps und Fibonacci-Heaps.

Textsuche: Randomisiertes Verfahren; Algorithmus von Knuth/Morris/Pratt, Algorithmus von Aho/Corasick, Algorithmus von Boyer/Moore, Vorverarbeitung für Boyer-Moore-Algorithmus.

### Medienformen

Bereitgestellt: Skript auf der Webseite  
 Tafelvortrag, Presenter-Projektion, Folien

### Literatur

Neben Vorlesungsskript:

- J. Kleinberg, E. Tardos, Algorithm Design, Pearson Education, 2005

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 2nd ed., MIT Press, 2001 (auch auf deutsch bei Oldenbourg)
- M. Dietzfelbinger, K. Mehlhorn, P. Sanders, Algorithmen und Datenstrukturen - Die Grundwerkzeuge, Springer, 2014
- S. Dasgupta, C. Papadimitriou, U. Vazirani, Algorithms, McGraw-Hill, 2007
- V. Heun, Grundlegende Algorithmen, 2. Auflage, Vieweg, 2003

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Hauptseminar IN MSc

Modulabschluss: Studienleistung alternativ Art der Notengebung: Testat / Generierte  
 Sprache: Deutsch/Englisch Pflichtkennz.: Pflichtmodul Turnus: ganzjährig

Modulnummer: 200746 Prüfungsnummer: 2200853

Modulverantwortlich: Silke Eberhardt-Schmidt

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 150 SWS: 0.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2200

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Im Rahmen einer aktuellen Problemstellung mit einer definierten Aufgabe sind die Studierenden in der Lage, sich in eine komplexe fortgeschrittene Fragestellung anhand der Lektüre einzuarbeiten. Sie können sich mit aktuellen wissenschaftlichen Erkenntnissen vertraut machen und diese analysieren und bewerten. Sie sind in der Lage, Erkenntnisse aus der Literatur zusammenzufassen, einzuordnen in einem Seminarbericht schriftlich aufzubereiten darzustellen, in einem Vortrag zu präsentieren und in einer Diskussion zu reflektieren.

### Vorkenntnisse

### Inhalt

Ein Forschungsseminar vermittelt die Arbeit mit wissenschaftlicher Literatur und das Präsentieren von Ergebnissen.

- Erarbeitung eines wissenschaftlich-technischen Themas unter Betreuung
- Dokumentation der Arbeit (Literaturrecherche, Stand des Wissens)
- Verfassen einer schriftlichen Seminararbeit
- Vorstellung der Ergebnisse mit anschließender Diskussion

### Medienformen

### Literatur

Themenspezifische Literatur wird vom Betreuer benannt und ist darüber hinaus selbstständig zu recherchieren.

Empfohlen wird außerdem Literatur zu wissenschaftlichem Arbeiten, Literaturrecherche (beispielsweise Angebote der Bibliothek) und Präsentationstechniken.

### Detailangaben zum Abschluss

- Seminarbericht (schriftliche Arbeit in dem themenangemessenen Umfang)
- Präsentation mit Diskussion

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Komplexe Informationstechnische Systeme

Modulabschluss: mehrere Teilleistungen

Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkennz.: Pflichtmodul

Turnus: Sommersemester

Modulnummer: 200013

Prüfungsnummer: 220429

Modulverantwortlich: Prof. Dr. Armin Zimmermann

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0																		
Fakultät für Informatik und Automatisierung			Fachgebiet: 2236																		
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS											
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																					

### Lernergebnisse / Kompetenzen

Die Studierenden verstehen detailliert Aufbau und Funktionsweise von komplexen informationstechnischen Systemen. Die Studenten verstehen die in eingebetteten Systemen zu beachtenden Echtzeit-, Kommunikations- und softwaretechnischen Aspekte. Die Studierenden sind fähig, Sicherheit, Zuverlässigkeit und Leistungsverbrauch beim Entwurf zu berücksichtigen. Die Studenten haben Kenntnisse in der Entwurfsdomäne Automotive. Methodenkompetenz: Die Studierenden sind in der Lage, Methoden des Systementwurfs, des modellbasierten Entwurfs und des Hardware-Software-Codesigns auf konkrete Problemstellungen anzuwenden. Die Studierenden sind in der Lage, verschiedene Methoden für unterschiedliche Anwendungsgebiete zu bewerten. Systemkompetenz: Die Studierenden entwerfen und validieren auszugsweise komplexe eingebettete Rechnersysteme für konkrete Einsatzszenarien. Sozialkompetenz: Die Studierenden sind in der Lage, praktische Problemstellungen des Entwurfs in der Gruppe zu lösen.

Die Studierenden wenden im Labor-Praktikum Kenntnisse über eingebettete Echtzeitsysteme an. Sie machen praktische Erfahrungen in der Analyse, dem Entwurf, der Programmierung und dem Testen solcher kombinierter Systeme aus Soft- und Hardware. Die theoretischen Kenntnisse werden so real erfahrbar und an einer praktischen Aufgabenstellung geübt.

Die Studierenden erarbeiten im Praktikum in kleinen Teams eigenverantwortlich eine Lösung für ein praktisches Problem. Dabei berücksichtigen sie verschiedene Vorschläge und Einflussfaktoren, diskutieren Lösungsideen und setzen sie gemeinsam um. In der Bewertung der Aufgabe nehmen sie Anmerkungen und konstruktive Kritik auf.

### Vorkenntnisse

Bachelor Informatik / Ingenieurinformatik oder gleichwertiger Abschluss

### Inhalt

1. Einführung, Systementwurf, Modellbasierter Entwurf
2. Echtzeitsysteme, Zuverlässige Systeme, Zuverlässigkeitsbewertung
3. Softwaretechnische Aspekte, Produktlinien
4. Hardware-Software-Codesign, Rechnerarchitektur-aspekte
5. Kommunikation
6. Energieeffizienz

### Medienformen

Folien verfügbar im Moodle-KursTafelanschrieb in der Vorlesung

Aufzeichnung der Vorlesung 2020 als VideoVerweise zu ergänzenden Online-Inhalten (e-Bücher, Vorlesungen anderer Universitäten)

### Literatur

- Blanchard, Fabrycky: Systems Engineering and Analysis  
 Geffroy, Motet: Design of dependable computing systems  
 Wörn, Brinkschulte: Echtzeitsysteme  
 Zöbel: Echtzeitsysteme  
 Wolf: Computers as Components  
 Liu: Real-Time Systems  
 Burns and Wellings: Real-Time Systems and Programming Languages

Cooling: Software Engineering for Real-Time SystemsHinweise in der Lehrveranstaltung und auf den Webseiten.

#### Detailangaben zum Abschluss

**Das Modul Komplexe Informationstechnische Systeme mit der Prüfungsnummer 220429 schließt mit folgenden Leistungen ab:**

- schriftliche Prüfungsleistung über 90 Minuten mit einer Wichtung von 100% (Prüfungsnummer: 2200645)
- Studienleistung mit einer Wichtung von 0% (Prüfungsnummer: 2200646)

Details zum Abschluss Teilleistung 1:  
schriftliche Klausur

Details zum Abschluss Teilleistung 2:  
Praktikumsaufgaben in Gruppen (z.B. Kugelfallversuch), bei guter Lösung können Zusatzpunkte für die Klausur vergeben werden

#### verwendet in folgenden Studiengängen:

Master Informatik 2021



## Modul: Netzalgorithmen

Modulabschluss: Prüfungsleistung mündlich 20 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch      Pflichtkenn.: Pflichtmodul      Turnus: Wintersemester

Modulnummer: 200029      Prüfungsnummer: 2200671

Modulverantwortlich: Prof. Dr. Günter Schäfer

Leistungspunkte: 5      Workload (h): 150      Anteil Selbststudium (h): 116      SWS: 3.0  
 Fakultät für Informatik und Automatisierung      Fachgebiet: 2253

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

- **Fachkompetenz:** Die Studierenden verstehen die gebräuchlichen Routingverfahren kennen die Notwendigkeit für eine bedarfsgerechte Aufteilung des Verkehrsaufkommens in Netzwerken. Sie können die verschiedenen Zielsetzungen beim Netzwerkentwurf voneinander abgrenzen und gegenüberstellen. Auf der Grundlage der Übung sind die Studierenden in der Lage, formale Beweise über die Korrektheit von Routing-Verfahren zu führen.
- **Methodenkompetenz:** Die Studierenden können grundlegende Entwurfs- bzw. Optimierungsprobleme als Multi-Commodity-Flow-Probleme formulieren. Sie sind auf der Grundlage der bearbeiteten Übungsaufgaben in der Lage, Multi-Commodity-Flow-Probleme in Standardformen zu überführen und durch Anwendung mathematischer Standardsoftware zu lösen.
- **Systemkompetenz:** Die Studierenden verstehen die Wechselwirkungen verschiedener Optimierungsziele beim Netzwerkentwurf und -betrieb.
- **Sozialkompetenz:** Auf der Grundlage der in der Vorlesung verwendeten metaphorischen Beispiele sind die Studierenden in der Lage, in den Übungen auch gesellschaftliche Sachzwänge (z.B. der fairen Vergabe von Ressourcen) im Kontext von Optimierungsverfahren durch Setzen entsprechender Zielfunktionen zu durchdenken, zwischen einzelnen Prioritätensetzungen (z.B. Gesamtdurchsatz vs. Fairness) abzuwägen und die vorgenommenen Prioritätensetzungen kritisch zu bewerten.

### Vorkenntnisse

Grundlegende Kenntnisse zu Netzwerken

### Inhalt

1. Einführung: Kommunikation in datagrammorientierten Netzwerken, Routingalgorithmen inklusive Korrektheitsbeweise, Modellierung von Datenverkehr mittels Poisson-Prozess, MM1 Wartesystem, Grundlegende Entwurfsprobleme in Netzwerken
2. Netzwerkmodellierung: Modellierung von Netzwerk-Design-Aufgaben als Multi-Commodity-Flow Probleme, Pure-Allocation-Problem, Shortest-Path-Routing, Fair Networks, Tunnel-Design in MPLS Netzwerken, Multilevel Netzwerke
3. Optimierungsmethoden: Grundlagen der Linearen Optimierung, Simplexalgorithmus, Branch-and-Bound, Gomory-Schnitte, Branch-and-Cut
4. Netzwerkentwurf: Zusammenhang von Netzwerkentwurfsproblemen und mathematischer Modellierung in Standardform, kapazitierte Probleme, Pfaddiversität, Limited-Demand-Split, NP-Vollständigkeit von Single-Path-Allocation, Modular Flows, nichtlineare Zielfunktionen und Nebenbedingungen, Lösung von Problemen mit konvexen und konkaven Zielfunktionen bzw. Nebenbedingungen durch lineare Approximation
5. Network Resilience: Zusammenhangsmaße, Biconnected Components, Algorithmen zur Bestimmung der Blockstruktur von Graphen  
Praktische Probleme und Protokollfunktionen in Kommunikationsnetzen und ihr algorithmischer Hintergrund.

### Medienformen

Folien, Skripte

### Literatur

Michal Pioro, Deepankar Medhi. Routing, Flow, and Capacity Design in Communication and Computer Networks. The Morgan Kaufmann Series in Networking, Elsevier, 2004

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Transaktionale Informationssysteme

Modulabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkenn.: Pflichtmodul

Turnus: Wintersemester

Modulnummer: 200034

Prüfungsnummer: 220439

Modulverantwortlich: Prof. Dr. Kai-Uwe Sattler

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 116	SWS: 3.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2254							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

Nach Besuch dieser Veranstaltung sind die Studierenden mit fortgeschrittenen Methoden zur Sicherstellung transaktionaler Eigenschaften von Informationssystemen vertraut. Sie verstehen Verfahren zur Gewährleistung von Atomarität, Konsistenz, Isolation und Dauerhaftigkeit in datenbankbasierten und verteilten Systemen. Die Studierenden können die verschiedenen Methoden hinsichtlich ihrer Vor- und Nachteile für verschiedene Einsatzzwecke bewerten.

Mit den Übungen können die Studierenden eigene Lösungen zu gestellten Aufgaben präsentieren, sich an themenspezifischen Diskussionen beteiligen und sind bereit, Fragen zu beantworten.

Im Rahmen des Abschlussworkshops konnten die Studierenden ein vorgegebenes Thema selbständig erarbeiten und strukturieren. Durch die Präsentation ihre Ergebnisse in einem Vortrag und die anschließende Diskussion, sind sie in der Lage, andere Meinungen zuzulassen sowie ihre eigene zu hinterfragen.

### Vorkenntnisse

Grundlagen aus den Bereichen Betriebssysteme, Verteilte Systeme und Datenbanksysteme

### Inhalt

Grundbegriffe transaktionaler Verarbeitung, ACID-Eigenschaften; Transaktionssemantiken und -modelle; Serialisierbarkeitsbegriffe; Verfahren zur Herstellung von Serialisierbarkeit, Sperrverfahren, optimistische und semioptimistische Verfahren, Mehrversionensynchronisation; Atomaritätseigenschaften, Verfahren zur Herstellung von Atomarität und Fehlersicherheit, Commit-Protokolle, 2PC, Paxos, Raft, Logging, Shadow Paging; Fallbeispiele: Blockchain, Persistent Memory

### Medienformen

Präsentationen mit Projektor und Tafel, Bücher und Fachaufsätze, Übungsaufgaben und Diskussionsblätter

### Literatur

G. Weikum, G. Vossen. Transactional Information Systems. Morgan Kaufmann, 2001  
 G. Saake, K. Sattler, A. Heuer. Datenbanken - Implementierungstechniken. 4. Auflage, mitp Professional, 2019  
 E. Rahm, G. Saake, K. Sattler. Verteiltes und Paralleles Datenmanagement. Springer Vieweg, 2015

### Detailangaben zum Abschluss

**Das Modul Transaktionale Informationssysteme mit der Prüfungsnummer 220439 schließt mit folgenden Leistungen ab:**

- alternativ semesterbegleitende Prüfungsleistung mit einer Wichtung von 33% (Prüfungsnummer: 2200676)
- mündliche Prüfungsleistung über 20 Minuten mit einer Wichtung von 67% (Prüfungsnummer: 2200677)

Details zum Abschluss Teilleistung 1:  
 Vortrag auf Abschlussworkshop

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Automatische Strukturen

Modulabschluss: Prüfungsleistung mündlich 20 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Wintersemester

Modulnummer: 200053 Prüfungsnummer: 2200698

Modulverantwortlich: Prof. Dr. Dietrich Kuske

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 116 SWS: 3.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2241

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studenten kennen Methoden und Grenzen der algorithmischen Analysierbarkeit unendlicher Strukturen, insbes. am Beispiel der durch endliche Automaten beschriebenen automatischen Strukturen. Sie können diese anwenden, beweisen und fundierte Vermutungen über Erweiterungen aufstellen und begründen.

Die Studierenden können kritische Fragen zum behandelten Stoff, Probleme bei der Erarbeitung des Wissens bzw. bei der Lösung der Aufgaben klar formulieren und in Diskussionen mit Kommilitonen und Lehrenden vertreten.

In den Vorlesungen und im Selbststudium haben sie die genannten Kenntnisse erworben, in den Übungen und im Selbststudium die genannten Fähigkeiten erlernt.

### Vorkenntnisse

endliche Automaten (vgl. z. B. Modul "Automaten und Formale Sprachen")  
 Prädikatenlogik (vgl. z. B. Modul "Logik und Logikprogrammierung")

### Inhalt

effektive Strukturen  
 Beispiele automatischer Strukturen  
 Beispiel von Strukturen, die nicht automatisch sind  
 Abschlußeigenschaften der Klasse der automatischen Strukturen

Komplexität des Auswertungsproblems in automatischen Strukturen für Erweiterungen der Prädikatenlogik

### Medienformen

Tafel

### Literatur

Rubin: Automata presenting structures - a survey of the finite string case. Bulletin of Symbolic Logic 2008

Barany, Grädel, Rubin: Automata presenting infinite structures. In: Finite and Algorithmic Model Theory, Cambridge University Press 2011

### Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Game Development

Modulabschluss: Prüfungsleistung alternativ

Art der Notengebung: Gestufte Noten

Sprache: Englisch

Pflichtkenn.: Wahlmodul

Turnus: Wintersemester

Modulnummer: 200819

Prüfungsnummer: 2500576

Modulverantwortlich: Prof. Dr. Wolfgang Broll

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0																								
Fakultät für Wirtschaftswissenschaften und Medien			Fachgebiet: 2557																								
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS																	
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																											

### Lernergebnisse / Kompetenzen

Through the lecture the students understand the structure and the essential components of computer games. They know the techniques required for this and can remind them if necessary. Through the seminar they have practical experience in the conception of a computer game and can implement it as part of the practical course within an application context. Through the group work in the seminar as well as in the practical lab course, the students can correctly assess and appreciate the achievements of their fellow students. They take criticism, heed remarks and accept suggestions.

### Vorkenntnisse

Bachelor level knowledge of computer graphics, programming (e.g. course AuP), VR/AR and games (e.g. course VWDS)

### Inhalt

Game development requires deep knowledge from different areas of computer science as well as design experience and artistic skills. Accordingly, the course will start with an introduction to game design. Nevertheless the majority of the course is concerned with technical aspects. Starting with rather general aspects such as choice of the suitable programming language, game entities, factories, etc. it will then highlight individual topics. This refers to game physics including collision detection, animations, special rendering aspects such as deferred shading, networking architectures and synchronization issues for distributed multi-user games, audio in games and various game-related aspects of artificial intelligence, including flocking, pathfinding, state machines, and neural networks. In the seminar and the practical lab course students will design and develop their own computer game.

### Medienformen

- Powerpoint slides
- Lecture content as PDF in Moodle
- Videos
- Live presentations
- Additional material in Moodle

### Literatur

- Introduction to Game Development (Steve Rabin, ed.)
- Game Design (Bob Bates)
- The Art of Game Design (Jesse Schell)

### Detailangaben zum Abschluss

Presentation (25%), demonstration (25%), design and implementation (50%) of a computer game (group work)

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Knowledge Engineering

Modulabschluss: Prüfungsleistung schriftlich 90 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Wintersemester

Modulnummer: 200125 Prüfungsnummer: 2200812

Modulverantwortlich: Prof. Dr. Rainer Knauf

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2238							
SWS nach Fachsemester	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P

### Lernergebnisse / Kompetenzen

Die Studierenden besitzen Kompetenzen auf dem Gebiet der fortschrittlichen Methoden der modernen Wissensverarbeitung. Die Studierenden kennen und verstehen die Strategien der Datenverarbeitung mit evolutionären/genetischen Algorithmen, mit Inferenzmethoden der KI und dem großen Spektrum des Datamining und können diese für informatische/ ingenieurinformatische Problemstellungen anwenden. Die Studierenden sind mit den methodischen Grundlagen vertraut und können die wichtigsten Datenanalyse und -verarbeitungs-Techniken erkennen und bewerten, sowie typische Informatikaufgaben mit ihrer Hilfe analysieren und lösen. Sie sind in der Lage, diese Kompetenzen in den Syntheseprozess komplexer ingenieurtechnischer und informatischer Projekte einfließen zu lassen. Die Studierenden kennen und verstehen die grundlegenden Wirkprinzipien von Produkten und Verfahren, bei deren Entwicklung Methoden der Wissensverarbeitung und des Datamining Anwendung fanden, können die Eignung der vermittelten Technologien für eine gegebene Problemlasse bewerten. In der Nachbereitungsphase der Vorlesung haben die Studierenden das Gelernte geübt und wiederholt und können es auf konkrete Aufgabenstellungen in Form von Übungsaufgaben anwenden. In Diskussionen mit den Mitkommilitonen können sie auch deren Argumentation richtig einschätzen und würdigen, berücksichtigen Kritik und nehmen Hinweise an.

### Vorkenntnisse

Logik und Logikprogrammierung

### Inhalt

- (1) Prädikatenkalkül der ersten Stufe (PK1): Wiederholung und sinnvolle Ergänzungen (Sortenlogik, Prädikatenkalkül der ersten Stufe mit Gleichheit)
- (2) problembezogene Wissensrepräsentationen der KI und Varianten der Implementierung von Inferenzmethoden darüber
- (3) Deduktion: Grundlagen, Deduktionssysteme, Komplexitätsbetrachtungen
- (4) Induktion und maschinelles Lernen: Erlernen von Klassifikationsregeln aus Beispielen, Erlernen eines besten induktiven Schlusses im Prädikatenkalkül der ersten Stufe, Verfahren zur Ermittlung des speziellsten Anti-Unifikators über PK1-Ausdrücken, Klassifikation nach Bayes
- Data Mining:
  - (1) Motivation, typische Aufgabenklassen und Anwendungen, Stufenprozess zur Modellbildung, (2) Ähnlichkeitsmaße für Datenobjekte, (3) Entropie der Information und andere Puritätsmaße, (4) Erlernen von Entscheidungsbäumen: schrittweise Verfeinerung von ID3 zu C 4.5 (numerische Attribute, fehlende Attribute), (5) Entscheidungsbäume über regulären Patterns, (6) Erlernen von Klassifikationsregeln top down and bottom up, (7) kNN-Klassifikation, (8) Klassifikation nach Bayes, (9) Bayesian Belief Networks, (10) Support Vector Machines, (11) Ensemble Methoden, (12) diverse Ansätze zum Umgang mit dem "Class Imbalance Problem"

### Medienformen

PPT, Tafelbild, Übungsaufgaben als PDF

### Literatur

- Inferenzmethoden:
- (1) Luger: Künstliche Intelligenz: Strategien zur Lösung komplexer Probleme. München: Pearson Studium (Übersetzung aus dem Addison-Wesley Verlag), 4. Aufl., 2001



(2) Russel/Norvig: Künstliche Intelligenz: Ein moderner Ansatz, München: Pearson Studium (Übersetzung aus dem Addison-Wesley Verlag), 2004

(3) Knauf: Logische Programmierung und Wissensbasierte Systeme: Eine Einführung. Aachen: Shaker, 1993  
Data Mining:

(1) Tan, Pang-Ning; Steinbach, Michael; Kumar, Vipin: Introduction to Data Mining. ISBN, Pearson Education, 2006.

(2) Markus Lusti: Data Warehousing and Data Mining: Eine Einführung in entscheidungsunterstützende Systeme, ISBN 3-540-42677-9, Springer, 2001.

(3) Petersohn, Helge: Data Mining. Verfahren, Prozesse, Anwendungsarchitektur. ISBN 978-3-486-57715-0, Oldenbourg Verlag, 2005.

(4) Lawrence, Kenneth D.; Kudyba, Stephan, Klimberg, Ronald K.: Data Mining Methods and Applications, ISBN 978-0-8493-8522-3, Boca Raton, FL u.a.: Auerbach, 2008.

#### Detailangaben zum Abschluss

#### verwendet in folgenden Studiengängen:

Master Informatik 2013

Master Informatik 2021

## Modul: Softcomputing

Modulabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkenn.: Wahlmodul

Turnus: Wintersemester

Modulnummer: 200082 Prüfungsnummer: 220452

Modulverantwortlich: Prof. Dr. Horst-Michael Groß

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 105 SWS: 4.0  
Fakultät für Informatik und Automatisierung Fachgebiet: 2233

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Im Modul Softcomputing haben die Studierenden die Begriffswelt der Fuzzy-Logik, der Genetischen Algorithmen (GA) und der evolutionären Strategien (ES) verstanden. Sie verstehen übergreifende Ansätze zur Lösung von Klassifikations- und Regelungs- und Optimierungsproblemen mit Fuzzy- und GA/ES-Methoden. Die Studierenden sind in der Lage, Fragestellungen aus dem o. g. Problembereichen zu analysieren, durch Anwendung des behandelten Methodenspektrums Lösungskonzepte zu entwerfen und diese auf ingenieurtechnische und biomedizinische Fragestellungen zu applizieren, sowie bestehende Lösungskonzepte zu bewerten. Vor- und Nachteile der Verfahren im Kontext praktischer Anwendungen sind den Studierenden bekannt. Nach Abschluss der Lernform "Übung" in Verbindung mit der selbständigen Implementierung einer Python-Fuzzy-Regelung (Teilleistung 2) beherrschen die Studierenden grundlegende mathematische Berechnungen, die Wirkungsweise unterschiedlicher Fuzzy-Operatoren und das Aufstellen von Fuzzy-Regeln. Nach intensiven Diskussionen während der Übungen und zur Auswertung der Python-Implementierung können die Studierenden Leistungen ihrer Mitkommilitonen richtig einschätzen und würdigen. Sie berücksichtigen Kritik, beherzigen Anmerkungen und nehmen Hinweise an.

### Vorkenntnisse

Pflichtmodul "Neuroinformatik und Maschinelles Lernen"

### Inhalt

Das Modul beinhaltet die ausgewählten Teilgebiete "Fuzzy-Logik" und "Evolutionäre/genetische Algorithmen" des Wissenschaftsgebietes Softcomputing. Beide Teilgebiete sollen ergänzend zu den Neuroinformatik-Modulen die Grundlagen für alternative Verfahren der Informations- und Wissensverarbeitung für Ingenieure und Informatiker legen. Damit würde der Absolvent über breite methodische und anwendungsorientierte Grundlagen der "Computational Intelligence" verfügen. Das Modul vermittelt sowohl Faktenwissen als auch begriffliches Wissen aus den folgenden Kernbereichen:

Fuzzy-Set-Theorie: Überblick, Einordnung und Historie, Grundlagen der Fuzzy-Logik (Basisvariablen, Linguistische Variablen, Terme, Zugehörigkeitsfunktionen, Fuzzifizierung, Fuzzy-Operatoren, unscharfe Zahlen und Relationen), Fuzzy-Regeln, unscharfes und plausibles Schließen, Fuzzy-Inferenz, Defuzzifizierungsmethoden, ausgewählte Anwendungsbeispiele aus dem technischen und nichttechnischen Bereich.

Genetische Algorithmen (GA) und Evolutionäre Strategien (ES): Einführung, Historie, philosophische Einordnung, Grundlagen und Begriffe, einführende Beispiele, prinzipielle Struktur eines GA/EA, Operatoren (Mutation, Crossover), Kodierungsvarianten und -probleme, Auswahl von Selektionsmechanismen bei GA/EA, Vor- und Nachteile, Ergänzende Beispiele und Anwendungen, Hinweis auf genetische und evolutionäre Programmierung. Die Studierenden haben die Möglichkeit, eine Problemstellung mit GA/ES softwaretechnisch umzusetzen und vorzustellen.

### Medienformen

Powerpoint, Demo-Applikationen in Python, Matlab, Java, Moodle-Kurs

### Literatur

Fuzzy-Logik:  
Zimmermann, H.-J.: Fuzzy Set Theorie - and its Applications. Kluwer in Boston, 1991  
Kosko, B.: Neural Networks and Fuzzy-Systems. Prentice Hall, New Jersey, 1992  
Böhme, G.: Fuzzy-Logik. Springer-Verlag, Berlin..., 1993

Bothe, H.-H.: Fuzzy-Logik - Einführung in Theorie und Anwendungen. Springer-Vlg., Berlin, Heidelberg, 1995  
Bothe, H.-H.: Neuro-Fuzzy-Methoden. Springer-Vlg., Berlin, Heidelberg, 1998  
Fuller, R.: Introduction to Neuro-Fuzzy Systems. Physica-Verlag, Heidelberg, 2000  
Tizhoosh, H. R.: Fuzzy-Bildverarbeitung. Springer-Vlg., Berlin, Heidelberg, 1998  
Höppner, F., Klawonn, F., Kruse, R.: Fuzzy-Clusteranalyse. Vieweg-Vlg., Braunschweig, 1997

Genetische Algorithmen und Evolutionäre Strategien :

Nissen, V.: Einführung in Evolutionäre Algorithmen. Vieweg-Vlg. Braunschweig, 1997  
Jacob, Ch.: Principia Evoliva. dpunkt.verlag, Heidelberg, 1997  
Gerdes, I., Klawonn, F., Kruse, R.: Evolutionäre Algorithmen. Vieweg-Vlg. Wiesbaden, 2004  
Heistermann, J.: Genetische Algorithmen. B.G. Teubner Verlagsgesellschaft, Stuttgart, Leipzig, 1994  
Lippe, W.-M.: Soft-Computing. Springer-Verlag, Berlin, Heidelberg, 2006  
Rechenberg, I.: Evolutionsstrategie 94, frommann-holzboog Vlg., Stuttgart, 1994

#### Detailangaben zum Abschluss

**Das Modul Softcomputing mit der Prüfungsnummer 220452 schließt mit folgenden Leistungen ab:**

- schriftliche Prüfungsleistung über 90 Minuten mit einer Wichtung von 100% (Prüfungsnummer: 2200737)
- alternativ semesterbegleitende Studienleistung mit einer Wichtung von 0% (Prüfungsnummer: 2200738)

Details zum Abschluss Teilleistung 2:

eigene Python-Implementierungen von vorgegebenen Algorithmen und Übungsaufgaben

verwendet in folgenden Studiengängen:

Master Informatik 2013

Master Informatik 2021

## Modul: Softwarearchitekturen

Modulabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkennz.: Wahlmodul

Turnus: Wintersemester

Modulnummer: 200067

Prüfungsnummer: 220446

Modulverantwortlich: Dr. Detlef Streitferdt

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0																								
Fakultät für Informatik und Automatisierung			Fachgebiet: 223																								
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS																	
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																											

### Lernergebnisse / Kompetenzen

#### Fachkompetenz:

Die Studierenden verfügen über Kenntnisse und Überblickswissen über Methoden und Werkzeuge des Software Engineering. Sie können dieses Wissen im Projektkontext anwenden und die Aussagekraft / Qualität der jeweiligen Ergebnisse bewerten. Die Studierenden sind fähig Softwareentwicklungsprozesse zu analysieren und auf die jeweiligen Gegebenheiten eines Projektes anzupassen. Sie verstehen Architekturmuster / -stile und können diese im Projektkontext einsetzen.

#### Methodenkompetenz:

Die Studierenden sind fähig die vorgestellten Entwicklungsmethoden anzuwenden und deren Ergebnisse früh im Entwicklungsprozess abzuschätzen. Sie sind fähig aus den vermittelten Methoden die für ein gegebenes Projekt passfähigen auszuwählen und anzuwenden.

#### Systemkompetenz:

Die Studierenden verstehen das grundsätzliche Zusammenwirken der Methoden und Prozessschritte der Entwicklung einer Softwarearchitektur.

#### Sozialkompetenz:

Die Studierenden können im Rahmen der semesterbegleitenden Gruppenaufgabe die Architekturdokumentation eines Open Source Projektes erarbeiten. Sie können die Auswirkungen der "weichen" Faktoren innerhalb von Softwareentwicklungsprozessen durch die eigene Gruppenarbeit richtig abschätzen.

Über ein studentisches Projekt während des Semesters, mit einer Abschlusspräsentation, wird das anwendungsnahe Implementierungswissen geprüft. In der Prüfung wird der Fokus nur noch auf das integrative Verständnis gelegt, sodass hier eine Reduktion der Prüfungsbelastung während des Prüfungszeitraumes erreicht wird.

### Vorkenntnisse

- Kenntnisse über Softwareentwicklungsprozesse
- Objektorientierte Modellierung
- Objektorientierte Programmierung

### Inhalt

Diese Vorlesung vermittelt Studenten der Informatik und Ingenieurinformatik Methoden und Techniken des Software Engineering. Über die Einbettung der Aktivitäten in den Softwareentwicklungsprozess werden die einzelnen Schritte und in den Übungen vertieft. Die Veranstaltung enthält die Erarbeitung von Softwarearchitekturzielen, Beschreibungsansätze der verschiedenen Modelle und Dokumente, Vorgehen bei der Entwicklung (Prozesse), Entscheidungsfindung, Architekturstile / -muster und ihre Qualitätseigenschaften, sowie die Prüfung/Bewertung von Architekturen.

(Die Vorlesung wird in Deutsch gehalten, einige der Materialien sind jedoch nur in Englisch verfügbar - was allerdings im Hinblick auf die spätere Arbeitswelt nur von Vorteil ist!)

## Medienformen

- Vorlesungsfolien
- PDF Dokumente (auch wissenschaftliche Beiträge)
- Prozessbeschreibungen (HTML), Templates

## Literatur

### Umfassende Werke

- [Balz 1996] Helmut Balzert, "Lehrbuch der Software-Technik", Spektrum Akademischer Verlag, 1996.  
[Fowl 1999] Martin Fowler, "Refactoring - Improving the Design of Existing Code", Addison Wesley, 1999.  
[Gamm 1995] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns - Elements of Reusable Object-Oriented Software", Addison Wesley, 1995.  
[Mart 2009] Robert C. Martin, "Clean Code", Prentice Hall, 2009.  
[McCo 2004] Steve McConnell, "Code Complete 2nd Edition", Microsoft Press, 2004.  
[RooC 2004] Stefan Roock, Martin Lippert, "Refactorings in großen Softwareprojekten", dpunkt.Verlag GmbH, 2004.  
[Somm 2007] Ian Sommerville, "Software Engineering", Pearson Studium, 2007.  
[Mens 2008] T. Mens and S. Demeyer, Eds., Software Evolution. Springer-Verlag New York Inc, 2008.

### Spezielle Themen ...

#### Entwicklungsprozesse

- [Beck 2000] Kent Beck, "eXtreme Programming eXplained", Addison Wesley, 2000.  
[Buns2002] C. Bunse and A. von Knechten, Vorgehensmodelle kompakt. Fraunhofer Publica [<http://publica.fraunhofer.de/oai.har>] (Germany), 2002.  
[Carr 1993] Marvin J. Carr, Suresh L. Konda, Ira Monarch, F. Carol Ulrich, Clay F. Walker, "Taxonomy-Based Risk Identification", Carnegie Mellon University, Technical Report CMU/SEI-93-TR-6, ESC-TR-93-183, 1993.  
[Open 2011] Eclipse Process Framework, "Open Unified Process, OpenUP", content retrieved 2011-10-01, 2011.

#### Requirements

- [Bere 2009] Brian Berenbach, Daniel J. Paulish, Juergen Kazmeier, Arnold Rudorfer, "Software & Systems Requirements Engineering In Practice", Mc Graw Hill, 2009.  
[Haya 1990] S. I. Hayakawa, "Language in Thought and Action", Harvest Books, 1990.  
[KoSo 1998] Gerald Kotonya, Ian Sommerville, "Requirements Engineering - Processes and Techniques", John Wiley & Sons, 1998.  
[Kula 2000] Daryl Kulak, Eamonn Guiney, "Use Cases - Requirements in Context", Addison-Wesley, 2000.  
[Lams 2001] Axel van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour", in Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE 2001), 27-31 August 2001, Toronto, Canada, 2001.  
[Lams 2009] Axel van Lamsweerde, "Requirements Engineering: From System Goals to UML Models to Software Specifications", John Wiley & Sons, 2009.  
[McCo 2006] Steve McConnell, "Software Estimation", Microsoft Press, 2006.  
[Pohl 2008] Klaus Pohl, "Requirements Engineering: Grundlagen, Prinzipien, Techniken", dpunkt.Verlag GmbH, 2008.  
[Robe 1999] Suzanne Robertson, James Robertson, "Mastering the Requirements Process", Addison-Wesley, 1999.  
[Rupp 2002] Chris Rupp, "Requirements-Engineering und -Management", Hanser Verlag, 2002.  
[Schu 2000] G. Gordon Schulmeyer, Garth R. Mackenzie, "Verification & Validation of Modern Software-Intensive Systems", Prentice Hall, 2000.  
[SoSa 1997] Ian Sommerville, Pete Sawyer, "Requirements Engineering: A Good Practice Guide", John Wiley & Sons, 1997.  
[Wieg 1999] Karl E. Wiegers, "Software Requirements", Microsoft Press, 1999.  
[With 2007] Stephen Withall, "Software Requirement Patterns", Microsoft Press, 2007.

#### Architektur, Produktlinien

- [Boec 2004] Günter Böckle, Peter Knauber, Klaus Pohl, Klaus Schmid, "Software-Produktlinien: Methoden, Einführung und Praxis", dpunkt.Verlag GmbH, 2004.  
[Clem 2002] Paul Clements, Rick Kazman, Mark Klein, "Evaluating Software Architectures", Addison Wesley, 2002.

- [Hrus 2012] P. Hruschka and G. Starke, Architektur-Knigge für Softwarearchitekten-Der Verschätzer. 2012.
- [Kang 1990] K. Kang, S. Cohen, J. Hess, W. Novak, and A. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study", SEI Institute, Carnegie Mellon University, USA, CMU/SEI-90-TR-021, 1990.
- [Kazm 2000] Rick Kazman, Mark Klein, Paul Clements, "ATAM: Method for Architecture Evaluation", TECHNICAL REPORT, CMU/SEI-2000-TR-004, ESC-TR-2000-004, 2000.
- [Lind 2007] F. J. van der Linden, K. Schmid, and E. Rommes, Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Berlin: Springer, 2007.
- [Love 2005] Robert Love, "Linux Kernel Development (2nd Edition)", Novell Press, 2005.
- [Masa 2007] Dieter Masak, "SOA? Serviceorientierung in Business und Software", Springer Verlag, 2007.
- [Pohl 2005] Klaus Pohl, Günter Böckle, Frank van der Linden, "Software Product Line Engineering - Foundations, Principles, and Techniques", Springer, Heidelberg 2005.
- [Posc 2007] Torsten Posch, Klaus Birken, Michael Gerdorn, "Basiswissen Softwarearchitektur", d.punkt Verlag, 2004 oder 2007.
- [Spin 2009] D. Spinellis and G. Gousios, Beautiful Architecture: Leading Thinkers Reveal the Hidden Beauty in Software Design. O'Reilly Media, 2009.

#### Detailangaben zum Abschluss

**Das Modul Softwarearchitekturen mit der Prüfungsnummer 220446 schließt mit folgenden Leistungen ab:**

- schriftliche Prüfungsleistung über 90 Minuten mit einer Wichtung von 50% (Prüfungsnummer: 2200716)
- alternativ semesterbegleitende Prüfungsleistung mit einer Wichtung von 50% (Prüfungsnummer: 2200717)

Details zum Abschluss Teilleistung 2: Im Verlauf der Veranstaltung sollen bestehende Projekte (Open Source) analysiert und vorgestellt werden. Es ist eine Ausarbeitungen mit Präsentation zu erstellen.

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Softwaretechnik für sicherheitskritische Systeme

Modulabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten  
 Sprache: Deutsch/Englisch Pflichtkennz.: Wahlmodul Turnus: Wintersemester

Modulnummer: 200002 Prüfungsnummer: 220423

Modulverantwortlich: Prof. Dr. Patrick Mäder

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0																		
Fakultät für Informatik und Automatisierung			Fachgebiet: 2234																		
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS											
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																					

### Lernergebnisse / Kompetenzen

Fachkompetenzen hauptsächlich erlangt in Vorlesungen und geprüft durch die abschließende mPI:

- Die Studierenden kennen die Konzepte und Terminologie abhängiger und sicherheitskritischer Systeme (dependability und safety).
- Die Studierenden verfügen über Kenntnisse wesentlicher Entwicklungsstandards sicherheitskritischer Systeme und deren Anforderungen an den Entwicklungsprozess von Systemen.
- Die Studierenden wissen, welche zusätzlichen Maßnahmen in allen wesentlichen Phasen eines Software- und Systementwicklungsprozesses im Kontext sicherheitskritischer Entwicklungen, je nach Kritikalität der Anwendung, ergriffen werden sollten und wie diese umzusetzen sind.
- Die Studierenden verfügen über Kenntnis zur qualifizierten Auswahl von Programmiersprachen, Werkzeugen, Code Analyse Techniken für sicherheitskritische Systeme.

Methodenkompetenzen hauptsächlich erlangt in den Seminaren und geprüft durch die begleitende aPI (Assignments):

- Die Studierende sind in der Lage Spezifikationen für sicherheitskritische Systeme zu erstellen.
- Die Studierenden sind in der Lage Sicherheitsanalysen und Safety Cases zu erstellen.
- Die Studierenden sind in der Lage Architekturen und Entwürfe für sicherheitskritische Systeme zu erstellen.
- Die Studierenden sind in der Lage ausgewählte Programmiersprachen für sicherheitskritische Systeme anzuwenden.

Sozialkompetenzen erlangt in Seminaren und Vorlesungen:

- Die Studierenden sind in den Seminaren fähig, die in der Vorlesung gelehrt Methodiken in Gruppenarbeit anzuwenden (z.B. gemeinsam eine FMEA durchzuführen), Lösungsstrategien zu diskutieren und Lösungen zu entwickeln.
- Studierende können die Risiken und Risikoerwägungen sicherheitskritischer Entwicklungen (z.B. tollerierbares Risiko) und die damit verbundene Verantwortung der Entwicklungsbeteiligten zum Beispiel anhand schwerer Unfälle mit ihren Lehrenden diskutieren. Sie kennen moralische Erwägungen und länderspezifische Ansätze zum Umgang mit Restrisiko.

### Vorkenntnisse

- Grundkenntnisse in Softwaretechnik (Software Engineering) vorteilhaft

### Inhalt

Sicherheitskritische Systeme sind solche, deren Versagen oder unzureichende Funktionalität katastrophale Folgen für Menschen, die Umwelt und die Wirtschaft haben kann. Diese Systeme werden kontinuierlich

komplexer in ihren Funktionalitäten, aber auch in ihren Interaktionen mit der Umgebung. Die Veranstaltung widmet sich dem Thema Softwareentwicklung für sicherheitskritische Systeme und stellt Techniken von den eingehenden Sicherheitsanalysen, über Spezifikation und Entwicklung bis zur Verifikation vor. In umfangreichen Übungen werden diese Techniken an Beispielen erlernt und unterstützende Applikationen vorgestellt.  
Schwerpunkte:

- System Safety
- Safety Standards und Safety Case
- Requirements Engineering und Modellierung\*
- Requirements Management, Verifikation und Validierung\*
- Architektur und Design Entwicklung, Verifikation und Validierung\*
- Safety und Risiko Analyse
- Programmiersprachen, Programmierung, Metriken\*
- Testen, Verifikation und Validierung auf Code-Ebene\*
- Qualitätssicherung und -management\*

\*) im Kontext sicherheitskritischer Software- und Systementwicklungen

#### Medienformen

- Vorlesungsfolien verfügbar über Moodle
- Tutorials, White-Paper und wissenschaftliche Beiträge
- Entwicklungswerkzeuge
- Auszüge aus Entwicklungsprojekten
- Aufgabenblätter als PDF
- Management der Assignments über Moodle

#### Literatur

- C. Hobbs: Embedded Software Development for Safety-critical Systems. CRC Press (2015)
- K. E. Wiegers and J. Beatty: Software Requirements. Microsoft Press (2013)
- C. Carlson: Effective FMEAs: Achieving safe, reliable, and economical products and processes using failure mode and effects analysis. John Wiley & Sons (2012)
- B. P. Douglass: Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems. Addison Wesley (2002)
- E. Hull and K. Jackson and J. Dick: Requirements engineering. Springer (2011)
- Van Lamsweerde: Requirements engineering: from system goals to UML models to software specifications. Wiley Publishing (2009)
- J. Barnes: Safe and secure software: An invitation to Ada 2012. AdaCore (2013)
- J. W. Vincoli: Basic guide to system safety. John Wiley & Sons (2006)
- J.-L. Boulanger: Static analysis of software: The abstract interpretation. John Wiley & Sons (2013)
- J. Schäuffele and T. Zurawka: Automotive software engineering-principles, processes, methods and tools. SAE International (2005)

#### Detailangaben zum Abschluss

**Das Modul Softwaretechnik für sicherheitskritische Systeme mit der Prüfungsnummer 220423 schließt mit folgenden Leistungen ab:**

- mündliche Prüfungsleistung über 20 Minuten mit einer Wichtung von 60% (Prüfungsnummer: 2200628)
- alternativ semesterbegleitende Prüfungsleistung mit einer Wichtung von 40% (Prüfungsnummer: 2200629)

Details zum Abschluss Teilleistung 1:

- es handelt sich um eine mündliche Prüfung nach Vorlesungsende, Terminabstimmung während der Veranstaltung

Details zum Abschluss Teilleistung 2:

- bewertete Seminaraufgaben und Hausaufgaben mit Abgabetermin, einzureichen über Moodle
- es werden, wenn in den ersten Vorlesungen nicht anders bekannt gegeben, sechs Aufgaben mit gleicher Punktzahl gestellt und bewertet, von denen die besten fünf zur Berechnung der 40%-Teilnote herangezogen werden



- die verbindliche Anmeldung zur aPI erfolgt über thoska ca. in Semesterwoche 3 und 4

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Verifikation

Modulabschluss: Prüfungsleistung mündlich 20 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch      Pflichtkennz.: Wahlmodul      Turnus: Wintersemester

Modulnummer: 200048      Prüfungsnummer: 2200693

Modulverantwortlich: Prof. Dr. Dietrich Kuske

Leistungspunkte: 5      Workload (h): 150      Anteil Selbststudium (h): 116      SWS: 3.0  
 Fakultät für Informatik und Automatisierung      Fachgebiet: 2241

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studenten kennen die Systemmodelle endliche Kripkestruktur, Kellersystem, (vergeßliches) Kanalsystem und wohlstrukturiertes Transitionssystem. Sie kennen die grundlegenden Verifikationsprobleme Erreichbarkeit, wiederholte Erreichbarkeit und das Auswertungsproblem temporaler Logiken. Sie sind vertraut mit den algorithmischen Möglichkeiten und Beschränkungen der Behandlung dieser Probleme sowie die Ausdrucksstärke temporaler Logiken. Sie können ähnliche temporale Logiken bzgl. dieser Kriterien bewerten und die Methoden auf ähnliche Systemmodelle adaptieren.

Sozialkompetenz: Die Studierenden können kritische Fragen zum behandelten Stoff, Probleme bei der Erarbeitung des Wissens bzw. bei der Lösung der Aufgaben klar formulieren und in Diskussionen mit Kommilitonen und Lehrenden vertreten.

### Vorkenntnisse

endliche Automaten: NFAs, DFAs, Konstruktionen hierzu (vgl. z. B. Modul "Automaten und Formale Sprachen")

### Inhalt

Erreichbarkeitsproblem, wiederholtes Erreichbarkeitsproblem, Auswertungsproblem temporaler Logiken  
 endliche Kripkestrukturen, Kellersysteme, (vergeßliche) Mehrkanalsysteme, wohlstrukturierte Transitionssysteme

### Medienformen

Tafel, Übungsblätter

### Literatur

Clark, Grumberg, Peled: Model Checking, MIT Press 2000

Gabbay, Hodkinson, Reynolds: Temporal Logic, Ox. Univ. Press 1994

Emerson: Temporal und Modal Logic. In: J. van Leeuwen (Ed.): Handbook of Theoretical Computer Science, Chapter 16, Amsterdam 1990

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Advanced Computer Graphics

Modulabschluss: Prüfungsleistung schriftlich 60 min Art der Notengebung: Gestufte Noten  
 Sprache: Englisch Pflichtkenn.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200058 Prüfungsnummer: 2200706

Modulverantwortlich: Prof. Dr. Beat Brüderlin

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 105 SWS: 4.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2252

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Students of this class have expert knowledge in real time rendering of massive 3D data and real time global illumination using modern graphics hardware. They know how to exploit the power of modern graphics processing units (GPUs) by awareness of their massively parallel architectures. Students understand methods for rendering massive 3D data, massive amounts of light sources and massive texture maps in real time, using culling methods, virtual texture mapping, deferred shading and post processing. After the exercises they know how to apply suitable graphic effects and real time approaches for global illumination, such as screen space ambient occlusion (SSAO), precomputed radial transfer functions, spherical harmonics, voxel cone tracing, light skin, and others. Students are experienced in shader development using OpenGL and shader programming languages.

### Vorkenntnisse

Students are expected to know and understand fundamental computer graphics concepts such as light propagation, color space, radiance, irradiance, illumination models (Phong, Cook-Torrance), raster conversion, texture mapping, ray tracing, signal and image processing (Nyquist sampling theorem, digital filters) as presented in the Foundations of Computer Graphics Class in the Bachelor Program of the TU-Ilmenau - or equivalent!

### Inhalt

Physically-based rendering (photon vs. electromagnetic wave models), Photon Tracing as Ground Truth for Real Time Approaches  
 Processor and Memory Architecture of modern GPUs. The challenges of SIMD processor, uniform shader models, programming: Latency, latency hiding, pipeline stalls, branching.  
 Massive Data visualization in real time (visibility-guided Rendering, Virtual Texture Mapping)  
 Deferred Shading and Post Processing, Screen Space Ambient Occlusion (SSAO). Global Illumination effects via real time approximation of indirect illumination effects. Precomputed Radiance Transfer Functions / Spherical Harmonics representation of light and light transfer, Voxel Cone Tracing, Light Skin.  
 In the practical seminar efficient shader programming with OpenGL will be implemented and performance will be analyzed

### Medienformen

Video / audio presentations of slides and as .pdf  
 see homepage of Computer Graphics Group

### Literatur

Visual Studio 2015 and above is recommended  
 See homepage Computer Graphics Group  
 or obtain directly from Microsoft

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Advanced Mobile Communication Networks

Modulabschluss: Prüfungsleistung schriftlich 120 min Art der Notengebung: Gestufte Noten  
 Sprache: Englisch Pflichtkenn.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200068 Prüfungsnummer: 2200718

Modulverantwortlich: Prof. Dr. Andreas Mitschele-Thiel

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 105 SWS: 4.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2235

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

. Fachkompetenz: Die Studierenden verfügen nach der Vorlesung über Kenntnisse und Wissen zu Aufbau und Funktionsweise von Mobilkommunikationsnetzen, insbesondere IP-basierter mobiler drahtloser Systeme und deren Protokolle, sowie Kenntnisse des Zusammenspiels verschiedener Funktionen.

. Methodenkompetenz: Die Studierenden sind in der Lage, komplexe Fragestellungen IP-basierter Mobilkommunikationssysteme und ihrer Funktionen zu verstehen und dieses Verständnis selbständig zu vertiefen.

. Systemkompetenz: Durch die Kombination aus Vorlesung und der Bearbeitung umfangreicher Testfragen zur Vertiefung des Stoffes verstehen die Studierenden im Anschluss das Zusammenwirken der verschiedenen Komponenten und Protokollfunktionen des Systems und können den Einfluss von Entwurfsentscheidungen bei der Realisierung von Protokollfunktionen auf andere Funktionen und das System als Ganzes einschätzen.

. Sozialkompetenz: Die Studierenden sind in der Lage, Problemstellungen der Mobilkommunikation selbständig zu lösen und darzustellen. Durch Diskussionen der Antworten zu unserem umfangreichen Fragekatalog haben Sie gelernt, Meinungen anderer Studierender zu beachten und diese kritisch zu hinterfragen. Das für die Lösung der Aufgaben benötigte Wissen konnten sie sich selbständig bzw. in Zusammenarbeit mit anderen aus verfügbaren Quellen erarbeiten, wurden sich durch die Präsentation der verschiedenen Möglichkeiten der Herangehensweise bei der Problemlösung bewusst und sind in der Lage die Leistungen Anderer entsprechend zu würdigen.

### Vorkenntnisse

Bachelor degree, basics of communication networks

### Inhalt

- Introduction to mobile communications with focus on protocols and systems
- Basics of wireless transmission
- Media access schemes
- Mobility management
- Transport protocols
- Quality-of-Service
- Security
- Communication systems (802.11, GSM/GPRS, UMTS)

### Medienformen

Presentations

### Literatur

Jochen Schiller Mobilkommunikationsnetze (for details see intro-slide) and further literature

### Detailangaben zum Abschluss

- Written examination during the official university examination period (registration via Moodle).
- Binding registration for the exam (using Thoska or the registration form provided by the examination office IA) is required at the beginning of each semester (check the registration time window which is defined each semester) in order to participate in the final exam. As your course grade is a result of the final exam, only

formally registered students are eligible for participation in the final exam at the end of the semester and may receive credits for it.

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Advanced Networking Technologies

Modulabschluss: Prüfungsleistung mündlich 20 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200044 Prüfungsnummer: 2200689

Modulverantwortlich: Prof. Dr. Günter Schäfer

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 116 SWS: 3.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2253

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

**Fachkompetenz:** Die Studierenden verfügen über Kenntnisse und Überblickswissen zu aktuellen, fortgeschrittenen Entwicklungen in der Netzwerktechnologie. Sie erkennen die besonderen Anforderungen an effiziente und flexible Kommunikationssysteme in bei einer Realisierung in Hard- und/oder Software und können diese im Kontext konkreter drahtgebundener Szenarien einschätzen. Die Studierenden kennen die grundsätzlichen Ansätze, wie der Datentransport in großen Netzen organisiert werden kann. Sie verstehen die unterschiedlichen Protokollkonzepte hierfür und können diese bewerten.

**Methodenkompetenz:** Die Studierenden sind in der Lage, für einzelne Teilaufgaben der Systemoptimierung geeignete Zielfunktionen zu identifizieren. Weiterhin können sie Optimierungen durchführen und bei der Verwendung mehrerer Zielfunktionen auftretende Zielkonflikte erkennen und gegeneinander abwägen.

**Sozialkompetenz:** Die Studierenden können alternative Gestaltungsmöglichkeiten für moderne Netzwerkarchitekturen erkennen und sind sich dabei der Konsequenzen spezifischer Entwurfsentscheidungen gewahr. In kritischer Diskussion der jeweiligen Vor- und Nachteile alternativer Lösungsvorschläge haben sie gelernt, einzelne Zielsetzungen miteinander in Beziehung zu setzen und dabei von unterschiedlichen Parteien eingebrachte Prioritäten gegeneinander abzuwägen und im Konsens in Einklang zu bringen. Die hierbei gewonnenen Erkenntnisse ermöglichen es den Studierenden, auf der Grundlage eines vertieften Verständnisses der jeweiligen Sachzwänge auch nicht optimal gestaltete technologische Lösungen zu akzeptieren und anzuerkennen.

### Vorkenntnisse

Bachelorstudium Informatik,  
 Bei Studium in Ilmenau: Vorlesung "Telematik 1"; vorteilhaft ist die vorherige Belegung der Vorlesungen "Telematik 2" und "Leistungsbewertung" bzw. die kombinierte Variante "Telematik 2 / Leistungsbewertung" (letztere mit PO 2013 eingeführt)

### Inhalt

Der Fokus der Vorlesung liegt auf modernen Netzwerktechnologien. Momentan sind die Hauptthemen Hardware-Router, Software-Defined Networking und Network Functions Virtualization:

- 01 Routers and Switches
- 02 Input Buffering in Routers
- 03 Size and Organization of Router Buffers
- 04 Interfacing NICs
- 05 Software Defined Networking
- 06 Network Functions Virtualization

### Medienformen

Folien, Skripte

### Literatur

- . H. Karl, A. Willig. Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons, 2005.
- . M. Hofmann, L. R. Beaumont. Content Networking Architecture, Protocols, and Practice. Morgan Kaufmann Publishers, 2005.

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Data Science: Methoden und Techniken

Modulabschluss: Prüfungsleistung mündlich 30 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200042 Prüfungsnummer: 2200687

Modulverantwortlich: Prof. Dr. Kai-Uwe Sattler

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0
Fakultät für Informatik und Automatisierung			Fachgebiet: 2254

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Nach Besuch dieser Veranstaltung sind die Studierenden mit fortgeschrittenen Methoden zur Auswertung und Analyse großer Datenbestände vertraut. Sie verstehen Data-Mining-/Machine Learning-Verfahren zur Analyse klassischer relationaler Geschäftsdaten als auch von raum- bzw. zeitbezogenen Daten, Graph- und Textdaten. Weiterhin kennen sie Prinzipien verteilter und paralleler Architekturen inkl. Data Warehouses und moderner Big-Data-Plattformen zur Verwaltung und Analyse sehr großer Datenbestände. Die Studierenden können die zugrundeliegenden Methoden sowie die technischen Aspekte erklären und hinsichtlich ihrer Vor- und Nachteile für verschiedene Einsatzzwecke bewerten.

Mit den Übungen können die Studierenden Standardwerkzeuge (Datenbanken, Data Warehouses, interaktive Notebooks) anhand konkreter Aufgabenstellungen zur Datenanalyse praktisch anwenden. Sie können eigene Lösungen entwickeln, bewerten und diese präsentieren, können sich an themenspezifischen Diskussionen beteiligen und sind bereit, Fragen zu beantworten.

### Vorkenntnisse

Datenbanksysteme, Statistik, Programmierkenntnisse

### Inhalt

Datenanalysepipeline; Big-Data-Architekturen; Data Warehousing und OLAP; Data-Mining-Techniken: Clustering, Frequent Itemset Mining; Analyse von Graph-Daten (Mustersuche in Graphen, Erkennen von Communities, Erkennung häufiger Subgraphen), Mining raum-zeitbezogener Daten (Sequential Pattern Mining, Trajectory Mining); NLP und Text Mining: Relationship-Extraktion, Word Sense Disambiguation, Named Entity Recognition; Sentiment Analyse; Parallelisierung und Verteilung: Partitionierungstechniken, datenparallele Verarbeitung

### Medienformen

Vorlesung mit Präsentationen und Tafel, Handouts,  
 Link zum Moodle-Kurs:  
<https://www.tu-ilmeneau.de/modultafeln/?fnq=200042>

### Literatur

Köppen, Saake, Sattler: Data Warehouse Technologien: Technische Grundlagen, mitp-Verlag, 2012.  
 Kumar, Steinbach, Tan: Introduction to Data Mining, Addison Wesley, 2005.  
 Lehner, Sattler: Web-Scale Data Management for the Cloud, Springer, 2013.  
 Rahm, Saake, Sattler: Verteiltes und Paralleles Datenmanagement, Springer, 2015.

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2013  
 Master Informatik 2021



## Modul: Effiziente geometrische Algorithmen und Datenstrukturen

Modulabschluss: Prüfungsleistung schriftlich 60 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200069 Prüfungsnummer: 2200719

Modulverantwortlich: Prof. Dr. Beat Brüderlin

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 116 SWS: 3.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2252

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

**Fachkompetenz:** Die Studierenden verfügen nach der Vorlesung über Kenntnisse und Überblickswissen zu effizienten geometrischen Algorithmen zur Verarbeitung großer Mengen geometrischer Daten in 2D und 3D. Für ausgewählte Problemstellungen kennen die Studierenden Erweiterungen der Ansätze auf höhere Dimensionen. Die Studierenden kennen unterschiedliche Anwendungen effizienter geometrischer Algorithmen in der Robotik, der Computergrafik, in der virtuellen Realität oder für Computerspiele, zum geometrischen Modellieren / CAD oder zur Verarbeitung von Punktwolken.

**Methodenkompetenz:** Die Studierenden sind in der Lage geometrische Aufgabenstellungen mit geometrischen Algorithmen und Datenstrukturen effizient, d.h. mit niedriger asymptotischer Laufzeitkomplexität zu lösen und deren Laufzeit für den average case, best case und worst case zu bestimmen. Für Problemstellungen wie die Berechnung konvexer Hüllen, Delaunay Triangulierung, Schnitte von Liniensegmenten und Dreiecken, Voronoi-Diagramme und räumlicher Indexstrukturen sind den Studierenden jeweils unterschiedliche Ansätze bekannt, z. B. iterative Ansätze, Divide und Conquer, Plane Sweep-Algorithmen, probabilistische, sowie problemspezifische und output-sensitive Ansätze. Sie kennen den Unterschied zwischen algorithmischer Komplexität und Problemkomplexität. Auch der Einfluss der Hardwarearchitektur und deren Eigenschaften (SIMD, Pipeline, Zugriffe auf RAM, externer Langzeitspeicher mit Latenzen) sind den Studierenden bekannt. Sie sind in der Lage unter alternativen Ansätzen die für die selbe Aufgabe, je nach Daten- und Hardware-Voraussetzung, die geeignetsten Ansätze auszuwählen und zu optimieren.

### Vorkenntnisse

Algorithmen und Datenstrukturen, Laufzeitkomplexität - O-Notation, Bachelor in Informatik oder Äquivalent

### Inhalt

Algorithmen: Einführung, asymptotische Laufzeit-Komplexität (O-Notation), effiziente Verarbeitung von 3d-Punktendaten, räumliche (mehrdimensionale) Suchstrukturen: Grid, Voxel, Octree, K-d-Bäume, Grid-file, hierarchische AABB, OBB, k-DOP, R\* -Bäume-

Punktsuche, Bereichsuche, körperhafte Objekte als hochdimensionale Punkte, Loose Trees, Hüllkörperhierarchie mit Überlappung, Nachbarschaftssuche, Anwendungsbsp. Surface-Area-Heuristik (SAH) für Ray Tracing, Kollisionserkennung (Physiksimulation, Boolean)

Effiziente geometrische Datenstrukturen & Algorithmen: Konvexe Hüllen. Definition und Konstruktion. Methode mit Stützgeraden. Erweiterung auf höhere Dimensionen.

Konvexe Hülle mit der Fächermethode nach Graham, Divide & Conquer, Quick-Hull.

Schneiden von Liniensegmenten, sowie Closest-Pair-Bestimmung mit dem Plane Sweep Verfahren.

Voronoi-Zellen, Delaunay Triangulierung, Skelette.

Output-Sensitivität, Hardwareabhängige Laufzeit, Temporale Kohärenz, probabilistische Ansätze.

Im angegliederten Seminar werden ausgewählte Verfahren in C++ programmiert.

## Medienformen

Skripte als PDF, Vorlesung mit Präsentation auch als Video mit Audiokommentar, interaktive Beispiele zur Arbeitsweise von Algorithmen/Datenstrukturen, Beispieldatensätze für Programmierübungen

## Literatur

Beat Brüderlin, Andreas Meier:  
Computergrafik und Geometrisches Modellieren.  
Teubner, Wiesbaden, Germany, 2001.

Rolf Klein:  
Algorithmische Geometrie.  
Addison-Wesley, Bonn, Germany, 1997.

Skripte zur Vorlesung Effiziente Geometrische Algorithmen von Prof. Brüderlin in der Vorlesung ausgegeben.

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Geometrische Modellierung

Modulabschluss: Prüfungsleistung schriftlich 60 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200057 Prüfungsnummer: 2200705

Modulverantwortlich: Prof. Dr. Beat Brüderlin

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 116 SWS: 3.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2252

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

**Fachkompetenz:** Die Studierenden verfügen nach der Vorlesung über Kenntnisse der fachspezifischen mathematischen Theorien geometrischer Modelle, wie auch deren informationstechnischer Umsetzung durch Datenstrukturen, wie sie in Computer-Aided-Design / Modellier-Systemen angewendet werden.

**Methodenkompetenz:** Die Studierenden kennen die Zusammenhänge aber auch die Abgrenzung zwischen a) der mathematischer Theorie (metrische Räume, topologische Räume, graphbasierte Polyedertheorie und viele mehr), b) der Informatischen Umsetzung der Datenstrukturen auf dem Computer (Datenrepräsentierung mittels Boundary-Representation, Winged-Edge, etc., sowie Polynom-Kurven und Flächen: Bézier, B-Spline, Rationale Polynome, NURB) und c) der physischen Realisierbarkeit eines modellierten Artefakts. Die Studenten sind sich der grundsätzlichen Problematik einer Abbildung der drei Welten untereinander bewusst. Die Studierenden sind in der Lage Eigenschaften von geometrischen Datenstrukturen als abstrakte Datentypen mathematisch herzuleiten und in Form von Computerprogrammen robust umzusetzen (insbesondere reguläre Boole'sche Mengenoperationen mittels der Boundary-Repräsentation für 2-mannigfaltige Begrenzungen auch bei Inzidenz-Situationen). Den Studierenden sind die potentiell auftretenden Fehler durch unzulässige Verwendung von Floating-Point-Darstellung Objekten, welche mathematisch als Teil- (Mengen) von mehrdimensionalen reellen Zahlenräumen definiert wurden, bekannt. Sie kennen Lösungsansätze zur Vermeidung von solchen Fehlern. Studierende können Approximationsansätze zur Darstellung von parametrischen (stückweisen) Polynomkurven und Flächen nach De Casteljau zur effizienten Darstellung anwenden.

### Vorkenntnisse

Computergrafik (Grundlagen) wird empfohlen, ist aber nicht zwingend notwendig.

### Inhalt

Mathematische Grundlagen, Datenrepräsentierungen, geometrische Operationen:

- Metrik, metrische Räume, Metriken (L-2, L-1, L-unendlich-Norm), Epsilon-Umgebung, offene (abgeschlossene) Mengen, Nachbarschaft, Operatoren: Abschluss, Inneres, Komplement, Rand, Boolesche Mengenoperationen (Vereinigung, Durchschnitt, Differenz)
- Abstandsfunktionen für Mengen, Hausdorff-Metrik, Problematik nichtmetrischer Abstandsfunktionen.
- Topologie, topologische Räume, stetige Abbildungen, Homöomorphismen.
- Einbettung, topologische Dimension, reguläre Körper, reguläre Mengenoperationen (praktische Bedeutung)
- d-Simplexe, simpliziale Komplexe. Orientierung, Orientierbarkeit
- Mannigfaltigkeiten (3-, 2-Mannigfaltigkeit mit, bzw. ohne Rand) 2-Mannigfaltigkeit als simplizialer Komplex, Pseudo 2-Mannigfaltigkeit.
  - Polyedertehorie: Polyedersatz, Eulercharakteristik, Platonische Körper (Hinweise: Kristalle, Dreiecksnetze / Speicherbedarf. geometriebasierte Datenkompression.)
  - Euleroperatoren, Euler-Poincaré-Charakteristik.
  - Euler Operatoren auf simplizialen Komplexen, abstrakte Polyeder. Beispiele für Euler-Poincaré

## Charakteristik

- Zusammenhänge der mathematischen Definitionen, Datenrepräsentierung als funktionale Abbildung (Vollständigkeit, Eindeutigkeit, Genauigkeit, Effizienz, etc.) B-Rep, CSG, Winged-Edge, Drahtmodelle, Voxel, Simplex.

- Algorithmische Umsetzung von regularisierten Mengenoperationen auf Polyedern.

- Robustheit geometrischer Algorithmen. Intuitionistische Inzidenzrelation.

- Implizite vs. explizite (parametrische) Kurven, Ferguson- Darstellung, Bézier-Darstellung. De Casteljau-Beziehung. Konvexe-Hüllen-Eigenschaft.

- De Casteljau-Zerlegung: Flatnesstest, adaptive Zerlegung /Approximation nach DeCasteljau.

Eigenschaften: Positive Definiteness, Variation-Diminishing-Eigenschaft

- Bezier Flächen. Zerlegung in Zeilen- und Spaltenkurven. Adaptive, rekursive Zerlegung v. Bezierflächen nach de Casteljau.

- Computer Algebra Methoden (Gröbner Basen, Resultante) Polynomgrad von Flächen und Trimmkurven sowie Flächenschnitten.

- Rationale Bezierkurven

- B-Spline-Kurven (Stückweise Polynomkurven)

- Freiformflächen (Trimmkurven, Komposition, B-Spline, NURBS, Tessellierung)

## Medienformen

Skripte als PDF, Vorlesung mit Präsentation, Video mit Audiokommentaren und Tafel, interaktive Beispiele zu Algorithmen/Datenstrukturen

## Literatur

Beat Brüderlin, Andreas Meier:

Computergrafik und Geometrisches Modellieren.

Teubner, Wiesbaden, Germany, 2001.

Christoph M. Hoffmann:

Geometric and Solid Modeling. An Introduction.

Morgan Kaufmann, San Mateo, CA, USA, 1989.

Martti Mäntylä:

An Introduction to Solid Modeling.

Computer Science Press, College Park, MD, USA, 1988.

David F. Rogers, J. Alan Adams:

Mathematical Elements for Computer Graphics.

2nd edition, WCB/McGraw-Hill, New York, NY, USA, 1990.

Dieter Roller:

CAD. Effiziente Anpassungs- und Variantenkonstruktion.

Springer, Berlin, Germany, 1995.

Skripte zur Vorlesung Geometrische Modellierung von Prof. Brüderlin in der Vorlesung ausgegeben.

## Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021



- Einleitung
  - Historische und wahrnehmungsphysiologische Aspekte der 3D-Erfassung
  - Überblick zu technischen Grundansätzen zur 3D-Erfassung
- Grundlagen
  - Algebraische Beschreibung von geometrischen Transformationen, Abbildungen und Messanordnungen
  - Optische Grundlagen
  - Modellierung und Kalibrierung von Messkameras (Tsai-Modellierung)
- Binokularer / polynokularer inkohärent optischer Ansatz zur 3D-Erfassung
  - Grundlagen der Stereobildverarbeitung (Korrespondenzsuche in Bildern: Constraints und Algorithmen)
  - Polynokulare Messanordnungen / Photogrammetrie
  - Verfahren der Musterprojektion (Streifenmuster, statistische Muster, Musterfolgen, Farbmuster)
- Prinzipien und Randbedingungen der praktischen Anwendung
- Monokular inkohärent optische Verfahren zur 3D-Erfassung
  - Depth from -Motion, -Shading, -Texture, -Fokus
  - Tiefenerfassung mit dem Laufzeitverfahren (Time-of-flight-Prinzip)
- Randbedingungen der praktischen Anwendung
- Anwendung der Sensoren in der Multisensor-Koordinatenmesstechnik
- Praxisrelevante weitere Aspekte der 3D-Erfassung
  - Prozesskette der Auswertung von 3D-Daten
  - Abnahme- und Überwachung von 3D-Sensorsystemen
  - Kalibrierverfahren für 3D-Sensoren

#### Medienformen

elektronisches Vorlesungsskript "Grundlagen der 3D-Bildverarbeitung", Übungs- / Praktikumsunterlagen  
 Bitte für das Fach unter folgendem Link einschreiben:  
 Einschreibung der Fächer für das Fachgebiet Qualitätssicherung und industrielle Bildverarbeitung

#### Literatur

- R. Hartley, A. Zisserman: Multiple View Geometry in computer vision. Cambridge University Press, 2010, ISBN 987-0-521-54051-3
- G. Hauske, Systemtheorie der visuellen Wahrnehmung. Shaker Verlag 2003, ISBN 978-3832212933
- R. Klette, A. Koschan, K. Schlüns: Computer Vision - Räumliche Information aus digitalen Bildern. Vieweg Verlag, Braunschweig/Wiesbaden, 1996, ISBN 3-528-06625-3
- W. Richter: Grundlagen der Technischen Optik, Vorlesungsskripte, Technische Universität Ilmenau, Institut für Lichttechnik und Technische Optik, Fachgebiet Technische Optik
- R. Zhang et.al.: Shape from Shading: A Survey. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol. 21, Nr. 8, S. 690-706, 1999
- O. Schreer: Stereoanalyse und Bildsynthese, Springer, 2005, ISBN 3-540-23439-X
- Middlebury Stereo Vision Page: Taxonomy and comparison of many two-frame stereo correspondence algorithms. <http://vision.middlebury.edu/stereo/>
- sowie die Vorlesungsunterlagen zu den Fächern Grundlagen der Bildverarbeitung und Mustererkennung bzw. Grundlagen der Farbbildverarbeitung

#### Detailangaben zum Abschluss

**Das Modul Grundlagen der 3D-Bildverarbeitung mit der Prüfungsnummer 230480 schließt mit folgenden Leistungen ab:**

- schriftliche Prüfungsleistung über 60 Minuten mit einer Wichtung von 100% (Prüfungsnummer: 2300670)
- Studienleistung mit einer Wichtung von 0% (Prüfungsnummer: 2300671)

Details zum Abschluss Teilleistung 2:

Praktika gemäß Testatkarte in der Vorlesungszeit (Prüfungsvoraussetzung)

verwendet in folgenden Studiengängen:

Diplom Maschinenbau 2017

Master Informatik 2013

Master Informatik 2021

Master Optische Systemtechnik/Optronik 2017

## Modul: IT-Sicherheitsmanagement

Modulabschluss: Prüfungsleistung schriftlich 60 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkenn.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200805 Prüfungsnummer: 2500562

Modulverantwortlich: Prof. Dr. Dirk Stelzer

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 116	SWS: 3.0							
Fakultät für Wirtschaftswissenschaften und Medien			Fachgebiet: 2533							
SWS nach Fach- semester	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P

### Lernergebnisse / Kompetenzen

Nachdem Studierende die Vorlesung besucht haben,

- kennen sie relevante Gefahren und Maßnahmen für die IT-Sicherheit,
- kennen sie wesentliche Bestandteile von Managementsystemen für Informationssicherheit,
- können sie Sicherheitskonzepte entwickeln,
- wissen sie, aus welchen Elementen ein IT-Notfallplan besteht,
- kennen sie die wichtigsten Bestimmungen des Datenschutzes und deren Konsequenzen für die betriebliche Praxis,
- wissen sie, wie IT-Sicherheitsprodukte zertifiziert werden können,
- kennen sie wesentliche Inhalte des IT-Sicherheitsgesetzes und wissen, welche Konsequenzen für Unternehmen damit verbunden sind und
- sie können Grundlagen der IT-Forensik erklären.

Mithilfe von Anwendungsbeispielen, Fallbeispielen und rechnergestützten Gruppenarbeiten während der Übung haben die Studierenden die Inhalte der Vorlesung vertieft und die können die in der Vorlesung vermittelten Instrumente und Methoden anwenden. Dies versetzt sie in die Lage, die Beiträge ihrer Kommilitonen besser zu würdigen und zu kritisieren.

### Vorkenntnisse

Grundlagen des Informationsmanagements

### Inhalt

- Grundlagen des IT-Sicherheitsmanagements
- Managementsysteme für Informationssicherheit
- Entwicklung von Sicherheitskonzepten
- IT-Notfallmanagement
- Datenschutz
- IT-Sicherheitskriterien und Produktzertifizierung
- IT-Sicherheitsgesetz
- IT-Forensik

### Medienformen

Vortrag, Präsentation und Interaktives Tafelbild

Skripte der Vorlesung und Begleitmaterial der Übungen sind auf der Webseite des Fachgebiets Informations- und Wissensmanagement bzw. in moodle abrufbar.

In den Übungen wenden die Studierenden in der Vorlesung vermittelte Instrumente und Methoden an. Geplant ist die Durchführung von Praxisprojekten in Kooperation mit dem Thüringer Ministerium für Infrastruktur und Landwirtschaft (TMIL).

### Literatur

- Matt Bishop: Computer Security. Art and Science. Boston et al. 2003.  
 Claudia Eckert: IT-Sicherheit. Konzepte - Verfahren - Protokolle. 8. Aufl., München 2013.  
 Dieter Gollmann: Computer Security. 2. Aufl., Chichester 2006.



#### Detailangaben zum Abschluss

Im Rahmen der Übungen (Praxisprojekte in Zusammenarbeiten mit Unternehmen und Behörden) können Bonuspunkte erreicht werden.

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Kognitive Robotik

Modulabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten  
 Sprache: Deutsch Pflichtkenn.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200083 Prüfungsnummer: 220453

Modulverantwortlich: Prof. Dr. Horst-Michael Groß

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2233							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

Nach Absolvierung des Moduls "Kognitive Robotik" verfügen die Studenten über die Begrifflichkeiten und das Methodenspektrum der Kognitiven Robotik. Sie haben übergreifende Ansätze zur Konzeption und der Realisierung von Robotik-Komponenten aus der Sicht von Sensorik, Aktorik und kognitiver Informationsverarbeitung verstanden. Sie kennen Techniken der Umgebungswahrnehmung und der lokalen und globalen Navigation von Kognitiven Robotern in komplexer realer Einsatzumgebung. Die Studierenden sind in der Lage, Fragestellungen aus dem o. g. Problembereichen zu analysieren, durch Anwendung des behandelten Methodenspektrums Lösungskonzepte für unterschiedliche Fragestellungen der Service- und Assistenzrobotik zu entwerfen und umzusetzen, sowie bestehende Lösungskonzepte zu bewerten. Vor- und Nachteile der Komponenten und Verfahren im Kontext praktischer Anwendungen sind den Studierenden bekannt. Mit den Python-Implementierungen (Teilleistung 2) verfügen die Studierenden über praktische Verfahren bei der Implementierung von Navigationsalgorithmen für die Robotik. Nach intensiven Diskussionen während der Übungen und zur Auswertung der Python-Implementierung können die Studierenden Leistungen ihrer Mitkommilitonen richtig einschätzen und würdigen. Sie berücksichtigen Kritik, beherzigen Anmerkungen und nehmen Hinweise an.

### Vorkenntnisse

Pflichtmodul "Neuroinformatik und Maschinelles Lernen"

### Inhalt

Das Modul vermittelt das erforderliche Methodenspektrum aus theoretischen Grundkenntnissen und praktischen Fähigkeiten zum Verständnis, zur Implementierung und zur Anwendung von Verfahren der Roboternavigation sowie zur Informations- und Wissensverarbeitung in Kognitiven Robotern. Sie vermittelt sowohl Faktenwissen, begriffliches und algorithmisches Wissen aus folgenden Themenkomplexen:

- Begriffsdefinitionen (Kognitive Robotik, Servicerobotik, Assistenzrobotik), Anwendungsbeispiele und Einsatzgebiete
    - Basiskomponenten Kognitiver Roboter
    - Sensorik und Aktuatorik: aktive und passive / interne und externe Sensoren; Antriebskonzepte und Artikulationstechniken
      - Basisoperation zur Roboternavigation: Lokale Navigation und Hindernisvermeidung incl. Bewegungssteuerung (VFH, VFH+, DWA); Anbindung an die Motorsteuerung; Arten der Umgebungsmodellierung und -kartierung; probabilistische Selbstlokalisierung (Bayes-Filter, Kalman-Filter, Partikel-Filter, MCL); Simultaneous Localization and Mapping (SLAM) Techniken (online SLAM, Full SLAM); Pfadplanung (Dijkstra, A\*, D\*, E\*, Rapidly-Exploring Random Trees (RRTs) )
        - Steuerarchitekturen nach Art der Problemdekomposition und der Ablaufsteuerung
        - Leistungsbewertung und Benchmarking Kognitiver Roboter (Metriken und Gütemaße, Gestaltung von Funktionstests)
      - Aktuelle Entwicklungen der Service- und Assistenzrobotik mit Zuordnung der vermittelten Verfahren
- Im Rahmen der Teilleistung 2 werden die behandelten methodischen und algorithmischen Grundlagen der Roboternavigation (Erzeugung einer Occupancy Grid Maps, Pfadplanung (Dijkstra und A\* Algorithmus), Selbstlokalisierung mittels Partikelfilter) durch die Studierenden selbst softwaretechnisch umgesetzt und im Rahmen eines vorgefertigten Python-Frameworks implementiert.

### Medienformen

Präsenzvorlesung mit Powerpoint, Arbeitsblätter zur Vorlesung, Übungsaufgaben, Videos, Python Apps, studentische Demo-Programme, e-Learning mittels "Jupyter Notebook", Moodle-Kurs

## Literatur

- Hertzberg, J., Lingemann, K., Nüchter: A. Mobile Roboter; Springer Vieweg 2012
- Siciliano, B., Khatib: O. Springer Handbook of Robotics, Springer 2016
- Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics, MIT Press 2005
- Siegwart, R., Nourbakhsh, I. R.: Introduction to Autonomous Mobile Robots, MIT Press 2004

## Detailangaben zum Abschluss

**Das Modul Kognitive Robotik mit der Prüfungsnummer 220453 schließt mit folgenden Leistungen ab:**

- schriftliche Prüfungsleistung über 90 Minuten mit einer Wichtung von 100% (Prüfungsnummer: 2200739)
- alternativ semesterbegleitende Studienleistung mit einer Wichtung von 0% (Prüfungsnummer: 2200740)

Details zum Abschluss Teilleistung 2:

Python-Implementierungen von Navigationsaspekten und Übungsaufgaben

verwendet in folgenden Studiengängen:

Master Informatik 2021

Master Mechatronik 2017

## Modul: Komplexitätstheorie

Modulabschluss: Prüfungsleistung mündlich 30 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch      Pflichtkenn.: Wahlmodul      Turnus: ganzjährig

Modulnummer: 200073      Prüfungsnummer: 2200726

Modulverantwortlich: Prof. Dr. Martin Dietzfelbinger

Leistungspunkte: 5      Workload (h): 150      Anteil Selbststudium (h): 116      SWS: 3.0  
 Fakultät für Informatik und Automatisierung      Fachgebiet: 2242

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

**Fachkompetenz:** Die Studierenden kennen das Konzept von polynomiellen Suchproblemen und polynomiellen Optimierungsproblemen. Sie kennen verschiedene Reduktionskonzepte (Turing, polynomielle Reduktion) sowie den Begriff der NP-Vollständigkeit und den Satz von Cook/Levin. Sie kennen die Implikationen der Eigenschaft "NP-vollständig". Die Studierenden kennen die 20 wichtigsten NP-vollständigen Probleme sowie das Konzept der starken NP-Vollständigkeit. Sie kennen die wesentlichen randomisierten Komplexitätsklassen, die polynomielle Hierarchie und Beziehungen zwischen beiden. Sie kennen die Grundbegriffe der PCP-Theorie.

**Methodenkompetenz:** Den Studierenden stehen die genannten Grundbegriffe als Basis für Argumentationen zur Verfügung. Sie sind in der Lage, den Satz von Cook/Levin zu beweisen, und auch die NP-Vollständigkeit für die in der Vorlesung behandelten Probleme und abgewandelte Versionen hiervon. Sie können wesentliche Berechnungsprobleme komplexitätstheoretisch einordnen.

**Sozialkompetenz:** Selbst in der Vorlesung ist Interaktion stets möglich, ja erwünscht, darin sind die Studierenden geübt. Die Studierenden konnten die Erfahrung machen, dass Fragen unmittelbar geklärt werden, dass Diskussionen auf Augenhöhe stattfinden und Beiträge wertschätzend aufgenommen werden. In der Übung waren die Studierenden zu Präsentationen aufgerufen. Sie konnten wertvolle Erfahrung in der Rolle des Präsentierenden sammeln. Die Teilnahme erforderte und trainierte ein hohes Maß an Selbstorganisation.

### Vorkenntnisse

Berechenbarkeit und Komplexitätstheorie  
 Effiziente Algorithmen

### Inhalt

Theorie der NP-Vollständigkeit, polynomielle Hierarchie, randomisierte Komplexitätsklassen, Grundzüge der PCP-Theorie und Nicht-Approximierbarkeit

### Medienformen

Tafelvortrag, Folien, teilweise schriftliche Ausarbeitung, Übungsblätter

### Literatur

- I. Wegener, Komplexitätstheorie - Grenzen der Effizienz von Algorithmen, Springer, 2003
- G. Ausiello et al., Complexity and Approximation, Springer, 1999
- M. Garey, D. Johnson, Computers and Intractability, W. H. Freeman and Co., 1979
- C. Papadimitriou, Computational Complexity, Addison-Wesley, 1995
- S. Arora, B. Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 2009
- O. Goldreich, Computational Complexity - A Conceptual Perspective, Cambridge University Press, 2008

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Leistungsbewertung technischer Systeme

Modulabschluss: Prüfungsleistung mündlich 25 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200014 Prüfungsnummer: 2200647

Modulverantwortlich: Prof. Dr. Armin Zimmermann

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2236							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

Die Studierenden verstehen detailliert Hintergrund und Funktionsweise von Verfahren der Modellierung und quantitativen Bewertung technischer Systeme. Die Studierenden sind fähig, quantitative Aspekte technischer Systeme beim Entwurf zu untersuchen und zu bewerten. Die Studenten haben Kenntnisse in Anwendungsgebieten der Leistungsbewertung. Methodenkompetenz: Die Studierenden sind in der Lage, Methoden des quantitativen Systementwurfs, der Modellierung und Bewertung auf konkrete Problemstellungen anzuwenden. Die Studierenden sind in der Lage, passende Modelle und Werkzeuge auszuwählen und einzusetzen. Sozialkompetenz: Die Studierenden sind in der Lage, praktische Problemstellungen der Leistungsbewertung in der Gruppe zu lösen und zu präsentieren. In der Übung wurde zu jeder Aufgabe ein vergleichbares Beispiel vorgerechnet bzw. in der gemeinsamen Diskussion erarbeitet. Dabei erkannten die Studierenden unterschiedliche Herangehensweisen. Sie können in kleinen Teams eigenverantwortlich eine Lösung für gegebene Aufgabenstellungen entwickeln. Dabei berücksichtigen sie verschiedene Vorschläge und Einflussfaktoren, diskutieren Lösungsideen und setzen sie gemeinsam um. Bei der Besprechung der Aufgabe können sie Anmerkungen und konstruktive Kritik nehmen und geben.

### Vorkenntnisse

Bachelor-Abschluss im Studiengang Ingenieurinformatik / Informatik bzw. weitgehend äquivalentem Studiengang

### Inhalt

Modellierung und Leistungsbewertung diskreter technischer Systeme  
 Grundlagen (Stochastische Grundlagen, Stochastische Prozesse)  
 Modelle (Markov-Ketten, stochastische Petri-Netze, farbige stochastische Petri-Netze)  
 Bewertungsverfahren (numerische Analyse, Simulation, Beschleunigungsverfahren)  
 Ausgewählte Anwendungsgebiete, Bewertung zuverlässiger Systeme

### Medienformen

Folien und Aufgabenzettel: verfügbar über Webseite der Lehrveranstaltung bzw Moodle-Kurs.  
 Ergänzende Informationen als Tafelanschrieb. Aufzeichnung der Vorlesung 2020 als Video.

### Literatur

Jain: The Art of Computer System Performance Evaluation  
 Law/Kelton: Simulation Modeling and Analysis  
 Cassandras/Lafortune: Introduction to Discrete Event Systems  
 Bolch, Greiner, de Meer, Trivedi: Queueing Networks and Markov Chains  
 Zimmermann: Stochastic Discrete Event Systems  
 Murata: Petri Nets: Properties, Analysis and Applications  
 Ajmone Marsan: Stochastic Petri Nets: An Elementary Introduction

### Detailangaben zum Abschluss

mündliche Prüfung

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Lernen in kognitiven Systemen

Modulabschluss: mehrere Teilleistungen

Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkenn.: Wahlmodul

Turnus: Sommersemester

Modulnummer: 200085

Prüfungsnummer: 220455

Modulverantwortlich: Prof. Dr. Horst-Michael Groß

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2233							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

Im Wahlmodul "Lernen in Kognitiven Systemen" haben die Studierenden aufbauend auf den Modulen "Neuroinformatik und Maschinelles Lernen" und "Deep Learning für Computer Vision" die konzeptionellen, methodischen und algorithmischen Grundlagen des Maschinellen Lernens zum Erwerb komplexer Verhaltensleistungen in kognitiven Systemen (Autonome Systeme, Roboter, Prozesssteuerungen, Spiele) durch Lernen aus Erfahrungen verstanden. Sie verfügen über Kenntnisse zur grundsätzlichen Herangehensweise dieser Form des Wissenserwerbs und zur Generierung von handlungsorientiertem Wissen aus Beobachtungen und Erfahrungen. Die Studierenden haben sich die wesentlichen Konzepte, Lösungsansätze sowie Modellierungs- und Implementierungstechniken beim Einsatz von Verfahren des Reinforcement Learnings und dessen Spielarten angeeignet. Sie sind in der Lage, praxisorientierte Fragestellungen aus dem o. g. Problemkreis zu analysieren, durch Anwendung des behandelten Methodenspektrums auf Fragestellungen aus den behandelten Bereichen neue Lösungskonzepte zu entwerfen und algorithmisch umzusetzen sowie bestehende Lösungen zu bewerten. Vor- und Nachteile der Komponenten und Verfahren im Kontext praktischer Anwendungen sind den Studierenden bekannt. Nach intensiven Diskussionen während der Übungen und zur Auswertung der Python-Implementierung (Teilleistung 2) können die Studierenden Leistungen ihrer Mitkommilitonen richtig einschätzen und würdigen. Sie berücksichtigen Kritik, beherzigen Anmerkungen und nehmen Hinweise an.

### Vorkenntnisse

Pflichtmodul "Neuroinformatik und Maschinelles Lernen", Wahlmodul "Deep Learning für Computer Vision"

### Inhalt

Das Modul vermittelt das erforderliche Methodenspektrum aus theoretischen Grundkenntnissen und praktischen Fähigkeiten zum Verständnis, zur Implementierung und zur Anwendung neuronaler und probabilistischer Techniken des Erwerbs von Handlungswissen durch Lernen aus evaluativ bewerteten Erfahrungsbeispielen. Sie vermittelt sowohl Faktenwissen, begriffliches und algorithmisches Wissen aus folgenden Themenkomplexen:

- Begriffliche Grundlagen: Verhalten; Agent; Zweck und Bedeutung von Lernprozessen; Stability-Plasticity Dilemma; Exploration-Exploitation Dilemma
- Reinforcement Learning (RL): Grundidee des RL; General RL-Task; Schwache und starke RL-Verfahren, RL als Markov Decision Process (MDP); Basiskomponenten eines RL-Agenten; Value/Action-Value Iteration und das Bellman'sche Optimalitätsprinzip; Q-Learning, Boltzmann-Aktionsauswahl; SARSA-Learning; On-policy und off-policy Verfahren; Eligibility Traces; RL und teilweise Beobachtbarkeit; Lösungsansätze zur Behandlung von POMDP
  - Neuronale Umsetzung von RL-Agenten: Value Approximation am Beispiel TD-Gammon; NFQ-Verfahren; ADHDP-Verfahren; Grundidee von Policy Search Algorithmen
  - Deep Reinforcement Learning (DRL) als Form des End-to-End Learnings: Atari Deep RL; AlphaGo; DeepControl
- Learning Classifier Systems (LCS)
- Multi-Agenten Systeme (MAS); Motivation und Arten von Multi-Agentensystemen; Konzepte zur Koordinierung von Agenten; Koordination mittels W-Lernen
- Exemplarische Software-Implementierungen von RL-Verfahren für Navigationsaufgaben, Spiele,

## Prozesssteuerungen (Teilleistung 2)

Im Rahmen der Teilleistung 2 sollen in C++ oder Python eigene Plugins zur Anwendung des Reinforcement Learnings am Beispiel der Roboternavigation im Simulator erstellt und experimentell untersucht werden.

### Medienformen

Präsenzvorlesung mit Powerpoint, Arbeitsblätter zur Vorlesung, Übungsaufgaben, Videos, Python Apps, studentische Demo-Programme, e-Learning mittels "Jupyter Notebook"

### Literatur

- Sutton, R., Barto, A. Reinforcement Learning - An Introduction. MIT Press 1998 / 2018 <http://incompleteideas.net/book/RLbook2018.pdf> )
- Alpaydin, Ethem. Maschinelles Lernen, Oldenbourg Verlag, 2008 - Bishop, Chr. Neural Networks for Patter Recognition, Oxford Press 1997

### Detailangaben zum Abschluss

**Das Modul Lernen in kognitiven Systemen mit der Prüfungsnummer 220455 schließt mit folgenden Leistungen ab:**

- schriftliche Prüfungsleistung über 90 Minuten mit einer Wichtung von 100% (Prüfungsnummer: 2200743)
- alternativ semesterbegleitende Studienleistung mit einer Wichtung von 0% (Prüfungsnummer: 2200744)

Details zum Abschluss Teilleistung 2:

eigene C++ oder Python-Implementierungen und Übungsaufgaben

verwendet in folgenden Studiengängen:

Master Informatik 2013

Master Informatik 2021



## Modul: Logik in der Informatik

Modulabschluss: Prüfungsleistung mündlich 20 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200052 Prüfungsnummer: 2200697

Modulverantwortlich: Prof. Dr. Dietrich Kuske

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 116 SWS: 3.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2241

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studierenden kennen Beispiele typischer Sätze & Methoden der algorithmischen Modelltheorie. Sie können diese anwenden, beweisen und fundierte Vermutungen über Erweiterungen aufstellen und begründen. Die Studierenden können kritische Fragen zum behandelten Stoff, Probleme bei der Erarbeitung des Wissens bzw. bei der Lösung der Aufgaben klar formulieren und in Diskussionen mit Kommilitonen und Lehrenden vertreten.

### Vorkenntnisse

Prädikatenlogik (vgl. z. B. Modul "Logik und Logikprogrammierung")  
 Komplexitätstheorie

### Inhalt

- Ehrenfeucht-Fräissé-Spiele, Lokalität der Prädikatenlogik 1. Stufe, Nicht-Ausdrückbarkeits-Beweise
- Deskriptive Modelltheorie (Zusammenhang zwischen logischen Ausdrucksmitteln & Komplexität)
- zufällige Strukturen

### Medienformen

Tafel

### Literatur

- Ebbinghaus, Flum, "Finite Model Theory"
- Libkin, "Elements of Finite Model Theory"

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Parallel Computing

Modulabschluss: mehrere Teilleistungen

Art der Notengebung: Generierte Noten

Sprache: Deutsch/Englisch

Pflichtkenn.: Wahlmodul

Turnus: Sommersemester

Modulnummer: 200003

Prüfungsnummer: 220424

Modulverantwortlich: Prof. Dr. Patrick Mäder

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0																		
Fakultät für Informatik und Automatisierung			Fachgebiet: 2234																		
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS											
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																					

### Lernergebnisse / Kompetenzen

Professional Competence mostly gained in lectures and evaluated in the oral exam:

- Students have knowledge about the fundamental concepts and terminology of parallel systems.
- Students have knowledge about different taxonomies to classify parallel hardware and the advantages and disadvantages per class.
  - Students know different methodologies for decomposing, agglomerating, and mapping a given problem into a set of parallel executable tasks.
  - Students know and can apply different synchronization techniques for parallel programs.
- Students have knowledge about different metrics for evaluating parallelization success and are informed about best practices and problems when profiling parallel software.

Methodological Competence mostly gained in seminars and evaluated in the aPI (assignments):

- Students gained the ability to implement parallel programs on different hardware platforms including the ability to analyze and decompose a given problem for parallel computing.
- Students are able to independently develop individual parallel implementations to a given problem and are able to judge and compare the quality and success in terms of parallelization.
- Students gained the ability to evaluate and troubleshoot parallel programs.
- Students gained the ability to use development tools and computational resources (e.g., cloud computing instances) for programming parallel programs.

Social Competence gained through lectures and seminars:

- Students can discuss advantages and disadvantages of different deep learning approaches among each other and with their lecturers.

### Vorkenntnisse

- basic programming skills in C are beneficial

### Inhalt

The goal of this master-level course is giving a structured introduction into the concepts of parallel programming. Students will learn fundamental concepts of parallelization and will be able to judge the correctness, performance and construction of parallel programs using different parallelization paradigms (e.g. task parallelization, data parallelization) and mechanisms (e.g. threads, task, locks, communication

channels). The course also provides an introduction to the concepts of programming and practical aspects of programming massively parallel systems and cloud computing applications (using Amazon AWS). At the end of the course, students shall be able to design and implement working parallel programs, using shared memory programming on CPU (using pThreads and OpenMP) and GPU (using Cuda) as well as distributed memory programming (using MPI) models. The concepts conveyed in lectures are deepened by practical programming exercises.

The following topics will be covered through lecture and seminar:

- Fundamentals of parallel algorithms
  - Decomposition, Communication, Agglomeration, and Mapping of parallel tasks
  - Styles of parallel programs
- Shared-memory programming
  - Processes, threads, and synchronisation
    - pThreads
    - OpenMP
  - Hardware architecture for parallel computing
    - Shared and distributed memory
    - Flynn's Taxonomy
    - Cache Coherence
    - Interconnection networks und routing
  - Distributed-memory programming
    - Message passing programming
      - MPI
- Analytical program models
  - Amdahl's law, etc.
  - Metrics
  - Profiling
- Parallel algorithms
- Programming massivly parallel systems
  - GPU und CUDA Programmierung
  - OpenCL
  - Warehouse-scale computing

#### Medienformen

- Lecture and seminar slide decks through Moodle
- Tutorials, white-papers and scientific papers
- Development tools
- Extracts of development projects
- Assigments managed through Moodle

- Amazon AWS compute instances to perform assignment and seminar work (require student's personal computer)

## Literatur

- Introduction to Parallel Computing: Zbigniew J. Czech, Cambridge University Press (2017)
- Introduction to Parallel Computing (Second Edition): Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Addison Wesley (2003), ISBN 0-201-64865-2
- Programming Massively Parallel Processors: A Hands-on Approach, D.B. Kirk and W.W. Hwu, Morgan Kaufmann, 2. Ed. (2012)
- Parallelism in Matrix Computations, E. Gallopoulos, B. Philippe, A.H. Sameh, Springer (2015)
- Parallel Programming, T. Rauber and G. Runger, Springer (2013)

## Detailangaben zum Abschluss

**Das Modul Parallel Computing mit der Prufungsnummer 220424 schliet mit folgenden Leistungen ab:**

- mundliche Prufungsleistung uber 20 Minuten mit einer Wichtung von 60% (Prufungsnummer: 2200630)
- alternativ semesterbegleitende Prufungsleistung mit einer Wichtung von 40% (Prufungsnummer: 2200631)

Details zum Abschluss Teilleistung 1:

- oral exam after the lecture period with appointments negotiated during the final lectures

Details zum Abschluss Teilleistung 2:

- one or multiple assignment projects to be solved at home and turned-in via Moodle at a defined due date announced with the task
- assignments are accompanied by a short physical, oral presentation and discussion in front of the peer group OR a short video presentation; students will be informed about the selected form upon announcing assignment topics
- students must register via thoska for this exam, typically within the 3rd and 4th week of the semester

verwendet in folgenden Studiengangen:

Diplom Maschinenbau 2017  
Master Informatik 2021

## Modul: Schutz von Kommunikationsinfrastrukturen

Modulabschluss: Prüfungsleistung mündlich 20 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkenn.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200045 Prüfungsnummer: 2200690

Modulverantwortlich: Prof. Dr. Günter Schäfer

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 116	SWS: 3.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2253							
SWS nach Fachsemester	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P

### Lernergebnisse / Kompetenzen

- . Fachkompetenz: Die Studierenden verfügen über Kenntnisse zu Risiken und Bedrohungen sowie Maßnahmen zum Schutz von Kommunikationsinfrastrukturen. Sie kennen die speziellen Techniken und Gefahren von Sabotageangriffen und können die spezifischen Risiken bei der Einführung neuer Gegenmaßnahmen gegen Sabotageangriffe analysieren und bewerten.
- . Methodenkompetenz: Die Studierenden können bewerten, ob ein Systementwurf bzw. eine -implementierung, sicherheitsgerecht ist, und wie eine Angriffserkennung und Reaktion auf Angriffe durchgeführt werden kann.
- . Systemkompetenz: Die Studierenden verstehen das grundsätzliche Zusammenwirken der Maßnahmen zum Schutz von Kommunikationsinfrastrukturen.
- . Die Studierenden besitzen die grundlegende Fähigkeit, sich in die Perspektive eines Angreifers zu versetzen und aus diesem Blickwinkel heraus Schwachstellen in Protokollen und Systemen zu erkennen. Dabei haben sie gelernt, unterschiedliche Motivationen zu berücksichtigen und begreifen die Notwendigkeit, sich für schützenswerte Werte durch Implementierung entsprechender Gegenmaßnahmen einzusetzen. Auf der Grundlage der behandelten Beispielangriffe können die Studierenden potentiell gesellschaftsbedrohende Angriffe auf essentielle Infrastrukturen antizipieren und im gemeinsamen Diskurs Gegenmaßnahmen und Lösungsvorschläge entwickeln.

### Vorkenntnisse

Bachelorstudium Informatik, Semester 1-4  
 Der vorherige Besuch der Vorlesung "Network Security" im Bachelorstudium ist hilfreich, stellt jedoch keine notwendige Voraussetzung dar.

### Inhalt

Die Lehrveranstaltung behandelt Risiken und Bedrohungen sowie Maßnahmen zum Schutz von Kommunikationsinfrastrukturen. Aufbauend auf einer grundlegenden Klassifikation und einer Abgrenzung zum Inhalt der Grundlagenvorlesung Network Security werden insbesondere die Bereiche Schutz der Verfügbarkeit von Diensten und Systemen, sicherheitsgerechter Systementwurf und -implementierung, Angriffserkennung und Reaktion auf Angriffe, sowie Herausforderungen der Netzsicherheit in Umgebungen mit besonderen Randbedingungen (Adhoc Netze, Sensornetze etc.) thematisiert. 1. Introduction & Motivation 2. Denial of Service Attacks and Countermeasures 3. Protection of IP Packet Transport, Routing and DNS 4. Security Aware System Design and Implementation 5. Intrusion Detection and Response 6. Security in Sensor Networks (Challenges in Constraint Environments)

### Medienformen

Folien, Skripte

### Literatur

- E. Amoroso. Fundamentals of Computer Security Technology. Prentice Hall. 1994.

- E. Amoroso. Intrusion Detection. Intrusion.Net Books, 1999.
- Brent Chapman and Elizabeth Zwicky. Building Internet Firewalls Second Edition. O'Reilly, 2000.
- C. Eckert. IT-Sicherheit: Konzepte, Verfahren, Protokolle. zweite Auflage, Oldenbourg Verlag, 2003.
- Simson Garfinkel and Gene Spafford. Practical Internet & Unix Security, O'Reilly, 1996.
- M.G. Graff, K.R. van Wyck. Secure Coding. O'Reilly, 2003
- S. Northcutt, J. Novak. Network Intrusion Detection - An Analyst's Handbook. second edition, New Riders, 2001.
- G. Schäfer; M.Rossberg. Netzsicherheit. dpunkt.verlag, 676 Seiten, 49,90 Euro, Juli 2014.
- J. Viega, G. McGraw. Building Secure Software. Addison-Wesley, 2003.

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Security in Embedded Systems

Modulabschluss: Prüfungsleistung mündlich 20 min      Art der Notengebung: Gestufte Noten  
 Sprache: Englisch      Pflichtkennz.: Wahlmodul      Turnus: Sommersemester

Modulnummer: 200134      Prüfungsnummer: 2200828

Modulverantwortlich: Prof. Daniel Ziener

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0																								
Fakultät für Informatik und Automatisierung			Fachgebiet: 2231																								
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS																	
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																											

### Lernergebnisse / Kompetenzen

#### Learning Goals

- The students have knowledge about types of attacks.
- The students have knowledge about detection of attacks.
- The students have knowledge about prevention of attacks.
- The students have knowledge about countermeasures against attacks.

#### Expertise

- . The students can show the influence of attacks and the corresponding countermeasures on the dependability of embedded systems
- . The students can describe the different countermeasures of attacks
- . The students can summarize different security facilities and measures for embedded systems
- . The students can show the overhead (area, time) of security facilities
- . The students can classify different types of attack on embedded systems

#### Social Competence

- . The students can develop concepts in groups with subsequent implementations

#### Autonomy

- . The students can acquire new knowledge from specific literature and associate this knowledge with other classes.

### Vorkenntnisse

computer engineering, basic knowledge in embedded systems

### Inhalt

#### Background:

Due to increasing networking of embedded systems, the protection of such systems against attacks on stored or processed data as well as implementation details is an increasingly important but also challenging task. The protection of embedded systems against known as well as new sophisticated attack possibilities is the subject of this lecture. It shows what attacks exist, what countermeasures can be taken and how to design secure embedded systems.

#### Course coverage:

- Attack scenarios
  - Examples of attack scenarios
  - Attacks on cryptographic algorithms and their implementations
- Code injection attacks
  - Different type of code injection attacks
  - Countermeasures
- Invasive physical attacks

- Microprobing
- Prevention and detection of single event effects
- Reverse engineering
- IP Protection
- Watermarking
- Non-invasive logical attacks
  - Phishing
  - Forged authenticity
  - Countermeasures
- Non-invasive physical attacks
  - Eavesdropping
  - Side-channel attacks
- Case study: Security in automotive applications

#### Medienformen

Foliensatz (Projektion und Download), Übungsaufgaben (Download), Praxisbeispiele

#### Literatur

- Catherine H. Gebotys, Security in Embedded Devices. Springer 2010.
- Benoit Badrignans et al., Security Trends for FPGAs. Springer 2011.
- Daniel Ziener, Techniques for Increasing Security and Reliability of IP Cores Embedded in FPGA and ASIC Designs. Dr. Hut 2010.

#### Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2013  
 Master Informatik 2021  
 Master Ingenieurinformatik 2014



## Modul: **Spezielle Kapitel der Komplexitätstheorie und Berechenbarkeit**

Modulabschluss: Prüfungsleistung mündlich 30 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch      Pflichtkennz.: Wahlmodul      Turnus: Sommersemester

Modulnummer: 200121      Prüfungsnummer: 2200806

Modulverantwortlich: PD Dr. Karl-Heinz Niggel

Leistungspunkte: 5      Workload (h): 150      Anteil Selbststudium (h): 105      SWS: 4.0  
 Fakultät für Informatik und Automatisierung      Fachgebiet: 224

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Version 1: Zertifizierung von FP, FLINSPACE und FPSPACE für imperative Programme

**Fachkompetenz:** Die Studierenden kennen die in der Vorlesung vorgestellten Konstruktionen und Beweise im Gebiet der Impliziten Komplexitätstheorie. Sie kennen die notwendigen Grundlagen aus der Komplexitätstheorie.

**Methodenkompetenz:**

Die Studierenden können die Konstruktionsmethoden von Beschränkungspolynomen und Zertifikaten präzise beschreiben und die jeweiligen Kriterien für eine Zertifizierung im Fall einer Schleife benennen. Sie können die wesentlichen Konstruktionen und Beweise nachvollziehen und präzise wiedergeben, insbesondere der Nachweis, dass die von einem zertifizierten Programm berechneten Funktionen in einer der fraglichen Komplexitätsklassen (FP, FLINSPACE oder FPSPACE) liegen. Umgekehrt können sie mittels der Basistheoreme zeigen, dass jede Funktion aus einer der fraglichen Komplexitätsklassen durch ein zertifiziertes Programm berechnet werden kann.

Version 2: Polynomialzeit im Kontext funktionaler Programmierung

**Fachkompetenz:**

Die Studierenden kennen das in der Vorlesung vorgestellte, von Hilbert eingeführte und durch Gödel bekannt gewordene System T, das für das Studium höherstufiger Funktionale und die Entwicklung funktionaler Programmiersprachen grundlegend war. Ferner kennen sie die Funktionenalgebra BC von Bellantoni und Cook, in dem genau die Funktionen aus FP definiert werden können, sowie das System RA ("ramified affinnable Terms") von Bellantoni, N. und Schwichtenberg, ein Teilsystem von System T, jedoch basierend auf Rekursion auf Notation in allen endlichen Typen, in dem genau die Funktionen aus FP durch RA-Programme berechnet werden können.

**Methodenkompetenz:**

Die Studierenden können neben Syntax und denotationaler Semantik auch die operationale Semantik von System T (ein via "Reduktionsregeln" basierter Auswertungsmechanismus von T-Programmen) präzise beschreiben und klassische Ergebnisse wie Korrektheit der operationalen Semantik, Starke Normalisierung und Eindeutigkeit der Normalform präzise formulieren und ihre Beweisstruktur (nach W. Tait) wiedergeben. Ferner können sie die Funktionenalgebra BC präzise beschreiben und die Konstruktion für den Nachweis, dass jede FP-Funktion in BC definierbar ist (die Umkehrung ist einfach), im Kern präzise wiedergeben.

Schließlich können die Studierenden das System RA präzise beschreiben und die Kernideen für seine Konstruktion darlegen. Ferner können sie das induktive Argument, dass jede BC-Funktion in RA definierbar ist, sauber wiedergeben und den vorgestellten Polynomialzeit-Algorithmus zur Auswertung von RA-Programmen in seiner Grundstruktur präzise beschreiben.

Version 3 (alternierend zu Version 1 bzw. Version 2) : Berechenbarkeit

**Fachkompetenz:**

Die Studierenden kennen die in der Vorlesung vertieften partiell rekursiven Funktionen auf Basis der  $\mu$ -rekursiven Funktionen sowie die davon abgeleiteten Begriffe der entscheidbaren und rekursiv aufzählbaren Mengen, Nummerierungen für das Rechnen auf nicht-natürlichen Zahlen und Standardnummerierungen der berechenbaren Zahlenfunktionen und ihre Charakterisierung. Sie sind mit den Konzepten Reduzierbarkeit von Problemen und das Rechnen mit Indizes von berechenbaren Zahlenfunktionen vertraut, insbesondere mit dem Fixpunktsatz, dem Satz von Rice und dem Satz von Rice/Shapiro. Ferner kennen sie die Konzepte einer produktiven, kreativen und simplen Menge zur Klärung der Frage, ob es rekursiv aufzählbare, aber nicht entscheidbare Mengen gibt, die nicht zum Halteproblem H oder dem Selbstanwendungsproblem K äquivalent und somit "leichter" sind. Sie sind mit unlösbaren Problemen wie das Postsche Korrespondenzproblem und der Unentscheidbarkeit der Prädikatenlogik 1. Stufe vertraut, sowie mit einer Variante des ersten Gödelschen Unvollständigkeitssatzes, wonach jedes korrekte Beweissystem für arithmetische Formeln unvollständig ist. Sie kennen die arithmetische Hierarchie und die damit verbundenen Fragestellungen.

#### Methodenkompetenz:

Die Studierenden können mittels  $\mu$ -Kalkül bzw. Kalkül PR der primitiv rekursiven Funktionen nachweisen, dass konkrete einfache Funktionen partiell bzw. primitiv rekursiv sind und grundlegende Abschlusseigenschaften der partiell bzw. primitiv rekursiven Funktionen bzw. der rekursiv aufzählbaren oder entscheidbaren bzw. primitiv rekursiven Mengen gelten. Sie können die Konstruktion einer Aufzählungsfunktion aller  $k$ -stelligen partiell rekursiven Funktionen wiedergeben und mittels des S-m-n-Theorems zeigen, dass man zu jeder aufgezählten Funktion  $f$  mit Nummer  $i$  die Gödelnummer eines GOTO-Programms berechnen kann, das  $f$  berechnet. Ferner können die Studierenden präzise formulieren und beweisen: Kleenes Normalformsatz und Rekursionssatz, sowie Fixpunktsatz, der Satz von Rice und Rice/Shapiro. Sie können damit die Unentscheidbarkeit von grundlegenden konkreten semantischen Fragen an Programmen zeigen. Die Studierenden können die vorgestellte Codierung der prädikatenlogischen Formeln und Herleitungen in einem vollständigen Kalkül des natürlichen Schließens wiedergeben und die Konstruktion angeben, dass jede partiell rekursive Funktion arithmetisch repräsentierbar und jede rekursiv aufzählbare Menge arithmetisch ist. Sie können die arithmetische Hierarchie definieren und die entscheidbaren und rekursiv aufzählbaren Mengen korrekt einordnen sowie von konkreten Problemen ihre Lage in der Hierarchie angeben.

Sozialkompetenz: Die Studierenden könnten in den Übungen, eigene Lösungen präsentieren und sie damit der Diskussion in der Gruppe aussetzen. Wertschätzende Diskussion durch die Gruppe wurde angeleitet, beim Vortrag machten die Studierenden wertvolle Erfahrung in der Rolle der Präsentierenden.

#### Vorkenntnisse

Automaten, Sprachen und Komplexität  
Logik und Logikprogrammierung

#### Inhalt

Version 1:

Die Vorlesung gibt Antworten auf die Grundfrage (der impliziten Komplexitätstheorie), ob man aus der Syntax von imperativen Programmen auf ihre Laufzeit bzw. ihren Platzbedarf schließen kann. Es werden dazu zunächst Stack- und Loop-Programme eingeführt und Ergebnisse (Basistheoreme) aus einer Arbeit von Kristiansen und N. erarbeitet und bewiesen:

- FP = Stackprogramme mit  $\mu$ -Maß 0
- FLINSPACE = Loop-Programme mit  $\mu$ -Maß 0

Das  $\mu$ -Maß ist dabei ein effizientes Verfahren, das jedem solchen Programm eine natürliche Zahl zuordnet. Das Fehlen wesentlicher Konzepte moderner Programmiersprachen -- wie z. B. das Zuweisungskonzept, gemischte Datenstrukturen und benutzerfreundliche Bibliotheksfunktionen -- gibt Anlass zu der Frage, in wieweit die Basistheoreme und die darin tragenden Kernideen auf moderne Programmiersprachen erweitert werden können. Es werden daher imperative Programme betrachtet, die aus beliebigen Basisanweisungen mittels Anweisungsfolgen, bedingten Anweisungen und FOR-Schleifen (loop, foreach und powerloop) aufgebaut sind. Solche Programme arbeiten auf Variablen  $X_1, \dots, X_n$ , wovon jede eine beliebige Datenstruktur (Stack, Register, Array, Baum, Graph,...) repräsentieren kann, solange dafür implizit ein (sinnvoller) Größenbegriff  $|X_i|$  definiert ist. Ziel der Vorlesung ist eine effiziente Methode zur Zertifizierung von

- polynomiellstem Zeitbedarf (FP) und
- linearem bzw. polynomiellstem Platzbedarf (FLINSPACE bzw. FPSPACE)

für solche Programme. Kern der Methode ist ein effizienter Matrizen-Kalkül für die Zertifizierung der

polynomiellen Größenbeschränktheit von solchen imperativen Programmen mit polynomiell größenbeschränkten Basisanweisungen. Das Zertifikat für ein Programm in  $n$  Variablen ist eine  $(n+1) \times (n+1)$ -Matrix über der Vergißmenge  $\{0, 1, \text{unendlich}\}$ . Die Methode ist konstruktiv in dem Sinne, dass neben dem Zertifikat -- wie bei den Basistheoremen -- auch stets Beschränkungspolynome für  $X_1, \dots, X_n$  bestimmt werden. Die folgenden Charakterisierungstheoreme (N. und Wunderlich 2006) werden erarbeitet und bewiesen:

- FP = Zertifizierte Stringprogramme (Stack-Programme mit beliebigen polynomialzeit-berechenbaren Basisanweisungen)
- FLINSPACE = Zertifizierte verallgemeinerte Loop-Programme (Loop-Programme mit beliebigen in linearem Platzbedarf berechenbaren Basisanweisungen)
- FPSPACE = Zertifizierte Power-Stringprogramme (Stringprogramme erweitert um Schleifen, deren Rumpf exponentiell oft in der Größe der Kontrollvariablen iteriert wird, mit beliebigen in polynomiellem Platzbedarf berechenbaren Basisanweisungen)

Das Verfahren steht als Java-Applet zur Verfügung. An Schulbeispielen wie binäres Addieren, binäres Multiplizieren oder Insertion-Sort kann man die Theorie "laufen" lassen, also Zertifikate berechnen und Beschränkungspolynome extrahieren.

Version 2:

Das von Hilbert eingeführte und durch Gödel bekannt gewordene System T war grundlegend für das Studium höherstufiger Funktionale und die Entwicklung von funktionalen Programmiersprachen mit komplexen (Daten) Typen. Zentral darin ist das Prinzip der "höherstufigen Rekursion", eine Verallgemeinerung der primitiv rekursiven Rekursion: Die in einer Rekursion berechneten Objekte sind nicht Grunddaten (z. B. natürliche Zahlen in Binärdarstellungen), sondern "Funktionale beliebigen Typs", d. h. Abbildungen, die andere Abbildungen als Argumente nehmen und wieder Abbildungen oder Grunddaten als Resultate liefern. Objekte in System T sind einfach getypte Lambda-Terme, angereichert um Konstanten für den zugrunde liegenden Datentyp (z. B. Null und Nachfolger zur Darstellung der natürlichen Zahlen) sowie Konstanten  $R_6$  für "Rekursion in allen endlichen Typen 6".

Die Vorlesung bespricht zunächst Syntax, denotationale und operationale Semantik von System T, wobei Letzteres ein Auswertungsmechanismus von Termen in T via "Reduktionsregeln" ist. Es werden u. a. folgende klassische Ergebnisse bewiesen:

- Korrektheit der operationalen Semantik
- Starke Normalisierung und Eindeutigkeit der Normalform (jede Reduktionsfolge für einen Term in T bricht nach endlich vielen Schritten mit einem eindeutig bestimmten Term in Normalform ab).

Geschlossene Terme  $t$  vom Grundtyp (Programme) reduzieren also zu einem eindeutig bestimmten Numeral, das den Wert von  $t$  darstellt. Dies stellt den Auswertungsmechanismus von Programmen in T dar.

Danach wendet sich die Vorlesung der bis vor wenigen Jahren offenen Frage (seit den 1960er Jahren) zu, wie man System T syntaktisch so einschränken kann, dass darin genau die Funktionen aus FP berechnet werden. In T kann man ja schon durch eine einzige höherstufige Rekursion (des Typs  $\text{nat} \rightarrow \text{nat}$ ) -- und darin geschachtelt eine primitive Rekursion -- die Ackermannfunktion definieren, die bekanntlich jedes primitiv rekursive Wachstum überschreitet.

Die Vorlesung bespricht zunächst einen ersten Durchbruch zu dieser Frage für grundstufige Rekursion, nämlich die "Funktionenalgebra"

- $BC = [0, S_0, S_1, c, n; \text{SCOMP}, \text{SRN}]$

von Bellantoni und Cook (1992), ein Baukasten zur Beschreibung von Polynomialzeitalgorithmen zur Berechnung von zahlentheoretischen Funktionen in Binärdarstellung (FPTIME, eine Variante von FP). Darin werden die Argumentpositionen jeder Funktion in "normale" und "sichere" klassifiziert, wobei "normale" Argumentpositionen Rekursionen (auf Binärdarstellungen von natürlichen Zahlen, daher 0 und die binären Nachfolger  $S_0, S_1$ ) kontrollieren können, während "sichere" Argumentpositionen keine Rekursion kontrollieren dürfen. Entsprechend gibt es die Schemata "sichere Komposition" (SCOMP) und "sichere Rekursion auf Notation" (SRN). In der Vorlesung wird dann der folgende wichtige Satz in einer modernen Fassung bewiesen:

- $BC = \text{FPTIME}$

Die darin tragenden Ideen plus "Zähmung" der höherstufigen Rekursion führten 2000 auf eine syntaktische Variante RA von System T, basierend auf Konstanten  $S_0, S_1$  für Binärzahldarstellungen und Konstanten  $RN_6$  für höherstufiger Rekursion auf Notation des Typs

- $6 \rightarrow !( \text{nat} \rightarrow 6 \rightarrow 6 ) \rightarrow ! \text{nat} \rightarrow 6$  für jeden "sicheren" Typ 6.

Im weiteren Fortgang der Vorlesung wird zunächst die Syntax von RA ("ramified affinnable terms") zusammen mit den zentralen Ideen für das Design von RA besprochen und dann der folgende Satz (Bellantoni, N. und

Schwichtenberg) bewiesen:

- $RA = FPTIME$  (in RA sind genau die Funktionen aus FPTIME berechenbar)

Dazu werden die Funktionen aus BC in RA eingebettet und es wird ein Polynomialzeitalgorithmus zur Auswertung von RA-Programmen angegeben.

Version 3:

Die Vorlesung behandelt klassische Resultate aus der Rekursionstheorie zu Fragen der Berechenbarkeit auf Basis der  $\mu$ -rekursiven Funktionen, zu Entscheidbarkeit und Unentscheidbarkeit. Sie beginnt mit den primitiv rekursiven Funktionen PR und Relationen  $Rel(PR)$  und beweist grundlegende Abschusseigenschaften:

- Abgeschlossenheit von PR unter Fallunterscheidung, beschränktem  $\mu$ -Operator, und Wertverlaufsrekursion
- Abgeschlossenheit von  $Rel(PR)$  unter Aussagenlogik und beschränkter Quantifikation

Insbesondere wird eine primitiv rekursive Paarcodierung  $pi$  und Dekodierung  $pi_0$  und  $pi_1$  eingeführt sowie eine darauf aufbauende längen-selbstverwaltende Codierung endlicher Zahlenfolgen (nach W. Buchholz) mit primitiv rekursiver Längen- und Dekodierungsfunktion. Jedem GOTO-Programm kann so in eleganter Weise eine Gödelnummer zugeordnet werden. Die Menge G solcher Gödelnummern ist selbst primitiv rekursiv.

Für festes  $k_1 \geq 1$  wird eine Aufzählungsfunktion  $\phi^{k_1}$  aller  $k$ -stelligen partiell rekursiven Funktionen konstruiert.  $\phi^{k_1}$  ist selbst partiell rekursiv (UTM-Eigenschaft) und mit Hilfe des S-m-n-Theorem wird gezeigt, dass sich jede  $(k+1)$ -stellige Funktion  $f$  in der Form  $f(i, n_1, \dots, n_k) = \phi^{k_1}(s(i), n_1, \dots, n_k)$  für ein  $s$  in PR darstellen lässt (SMN-Eigenschaft). Darauf aufbauend werden gezeigt:

- Die partiell rekursiven Funktionen sind abgeschlossen unter Fallunterscheidung und starker Fallunterscheidung.

- Normalformsatz der partiell rekursiven Funktionen
- Rekursionssatz
- Fixpunktsatz

Nach Einführung der bekannten Konzepte der entscheidbaren, rekursiv aufzählbaren und semi-entscheidbaren Mengen, des Halteproblems H und Selbstanwendungsproblem K sowie der Reduzierbarkeit ( $\leq_m$ ) mit Merkgeln für die Reduktionsmethode werden folg. Themen behandelt:

- Charakterisierung der rekursiv aufzählbaren Mengen
- H und K sind reduktionsäquivalent ( $=_m$ )
- Der Satz von Rice mit Anwendungen bzgl. konkreter unentscheidbarer Mengen wie Tot, Nonempty, Fin, Inf, Cof, Rec, Corf, Equ
- Der Satz von Rice/Shapiro mit Anwendungen -- obige Beispielmengen sind nicht einmal rekursiv aufzählbar.

Nach Einführung der Konzepte einer produktiven, kreativen und simplen Menge werden gezeigt:

- Merkgeln für produktive Mengen
- Charakterisierung der produktiven und kreativen Mengen
- Es gibt simple Mengen, d. h. rekursiv aufzählbare Mengen, die nicht entscheidbar, aber "leichter" als K und H sind.

Nach Wiederholung des binären Postschen Korrespondenzproblems 0-1-PKP wird behandelt:

- Unentscheidbarkeit der Prädikatenlogik Stufe
- Arithmetische Repräsentierbarkeit der partiell rekursiven Funktionen
- Jede rekursiv aufzählbare Menge ist arithmetisch.
- Codierung der prädikatenlogischen Formeln und Herleitungen in einem vollständigen Kalkül (natürliches Schließen formuliert als Sequenzenkalkül)

- Gödelscher Unvollständigkeitssatz: Jedes korrekte Beweissystem für arithmetische Formeln ist unvollständig.
- Die arithmetische Hierarchie: Eine Schichtung der arithmetischen Mengen in eine echt aufsteigende Hierarchie von Klassen, definiert durch Folgen alternierender Quantorenpräfixe entscheidbarer Mengen. Die unterste Schicht besteht aus den entscheidbaren Mengen, die zweit-unterste aus den rekursiv aufzählbaren Mengen, alle anderen Schichten also aus nicht rekursiv aufzählbaren Mengen.

#### Medienformen

Folien (auch Beamer) mit Definitionen, Lemmata und Theoremen, und Tafelvortrag für Beispiele, Beweisen oder Konstruktionen

#### Literatur

- Kristiansen, K.-H. Niggl. On the computational complexity of imperative programming languages. *Theoretical Computer Science, Special issue on Implicit Computational Complexity*, Editor J.-Y. Marion, 318(1-2):139--161, Elsevier 2004.
- H. Niggl, H. Wunderlich. Certifying polynomial time and linear/polynomial space for imperative programs. *SIAM Journal on Computing*, 35(5):1122--1147, March 3, 2006.
- J. Bellantoni, S. Cook. A new recursion-theoretic characterization of the polytime functions. *Computational Complexity*, 2 (1992), pp. 401-415.
- J. Bellantoni, K.-H. Niggl, H. Schwichtenberg. Higher type recursion, ramification and polynomial time. *Annals of Pure and Applied Logic*, 104:17--30, 2000.
- Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. The MIT Press, Cambridge, Massachusetts, 1987.

#### Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Virtual and Augmented Reality (VR/AR)

Modulabschluss: Prüfungsleistung alternativ

Art der Notengebung: Gestufte Noten

Sprache: Englisch

Pflichtkennz.: Wahlmodul

Turnus: Sommersemester

Modulnummer: 200843

Prüfungsnummer: 2500604

Modulverantwortlich: Prof. Dr. Wolfgang Broll

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0																		
Fakultät für Wirtschaftswissenschaften und Medien			Fachgebiet: 2557																		
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS											
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																					

### Lernergebnisse / Kompetenzen

Through the lecture the students understand the structure and the essential components of VR and AR applications. They know the techniques required for this and can remind them if necessary. Through the seminar they have practical experience in the conception of a VR/AR applications and can implement them as part of the practical course within an application context. Through the group work in the seminar as well as in the practical lab course, the students can correctly assess and appreciate the achievements of their fellow students. They take criticism, heed remarks and accept suggestions.

### Vorkenntnisse

Bachelor level knowledge of computer graphics, programming (e.g. course AuP), VR/AR and games (e.g. course VWDS)

### Inhalt

Virtual Reality (VR) and Augmented Reality (AR) are two technologies which become more and more important today, in particular with the availability of consumer VR displays and the introduction of AR SDKs on all major mobile platforms. This course covers the required knowledge to understand and create VR and AR applications considering their large variety regarding hardware and software.

The course investigates into special VR and AR output devices such as HMD and CAVE, 3D interaction techniques including navigation in VR, selection and manipulation as well as usability aspects. It also deals in detail with real-time aspects. Finally, AR, its different visualization concepts, geometric and photometric registration, as well as the tracking methods underlying AR, such as marker tracking and SLAM. Furthermore, the principle of Mediated Reality is discussed and Diminished Reality is explained in detail.

In the seminar and the practical lab course the students will develop their own small VR or AR application in small groups.

### Medienformen

Powerpoint slides

Lecture content as PDF in Moodle

Videos/Live presentations/Additional material in Moodle

### Literatur

Virtual and Augmented Reality, Dörner et al. (eds), Springer, 2021

### Detailangaben zum Abschluss

Presentation (25%), demonstration (25%), design and implementation (50%) of a VR/AR application (group work)

The binding registration takes place with the registration of the student in a project group.

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Approximationsalgorithmen

Modulabschluss: Prüfungsleistung mündlich 30 min Art der Notengebung: Gestufte Noten

Sprache: Deutsch

Pflichtkenn.: Wahlmodul

Turnus: ganzjährig

Modulnummer: 200077

Prüfungsnummer: 2200731

Modulverantwortlich: Prof. Dr. Martin Dietzfelbinger

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 116	SWS: 3.0																		
Fakultät für Informatik und Automatisierung			Fachgebiet: 2242																		
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS											
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																					

### Lernergebnisse / Kompetenzen

**Fachkompetenz:** Die Studierenden kennen die grundlegenden Konzepte des Bereichs der Approximationsalgorithmen, insbesondere die Definition von Optimierungsaufgaben und die Qualitätsstufen von Approximationsalgorithmen (absolute, relative Approximationsgüte, [voll] polynomielle Approximationsschemata, inputabhängige Approximation). Sie kennen die Wirkungsweise der relevanten Entwurfsprinzipien. Sie kennen die relevanten Analysetechniken. Die Studierenden kennen die zentralen Beispielprobleme, für die Approximationsalgorithmen entwickelt wurden, ihre Performanzparameter und die Analyseverfahren.

**Methodenkompetenz:** Die Studierenden können die Entwurfsprinzipien auf verwandte Problemstellungen anwenden. Sie können Algorithmen und Probleme in die relevanten Klassen APX, PTAS, FPTAS usw. einsortieren. Sie können die zentralen Algorithmen beschreiben und die Analyse durchführen.

**Sozialkompetenz:** Selbst in der Vorlesung ist Interaktion stets möglich, dazu sind die Studierenden befähigt. Die Studierenden haben die Erfahrung gemacht, dass Fragen unmittelbar geklärt werden, dass Diskussionen auf Augenhöhe stattfinden und Beiträge wertschätzend aufgenommen werden. In der Übung konnten sich die Studierenden in Präsentationen üben. Sie haben wertvolle Erfahrung in der Rolle des Präsentierenden.

### Vorkenntnisse

 Berechenbarkeit und Komplexitätstheorie  
 Effiziente Algorithmen

### Inhalt

Grundbegriffe. Einführende Beispiele. Greedy Set Cover. Absolute Approximation: Färbung von planaren Graphen, Kantenfärbungen. Relative Approximation. Greedy-Verfahren und ihre Analyse. MAX-SAT, Metrisches TSP. Asymptotische relative Approximation. Inputabhängige Approximation: Graphfärbungen. Polynomielle Approximationsschemata: Rucksackproblem. Asymptotisches Approximationsschema: Binpacking. Weitere Techniken: Lineare Programmierung und Randomisiertes Runden am Beispiel von MAX-SAT. Derandomisierung. Semidefinite Programmierung am Beispiel von Max-Cut. Approximate Counting und die Monte-Carlo-Methode.

### Medienformen

Folien, Tafel, Übungsblätter

### Literatur

- R. Wanka, Approximationsalgorithmen, Teubner 2006
- K. Jansen, M. Margraf, Approximative Algorithmen und Nichtapproximierbarkeit, de Gruyter 2008
- G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and

Approximation, Springer-Verlag 1999

- D. P. Williamson, D. P. Shmoys, The Design of Approximation Algorithms, Cambridge University Press 2011

- D.-Z. Du, K.-I Ko, X. Hu, Design and Analysis of Approximation Algorithms, Springer 2012

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021



## Modul: Ausgewählte Kapitel der Komplexitätstheorie/Algorithmik

Modulabschluss: Prüfungsleistung mündlich 30 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkenn.: Wahlmodul Turnus: ganzjährig

Modulnummer: 200075 Prüfungsnummer: 2200729

Modulverantwortlich: Prof. Dr. Martin Dietzfelbinger

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 116	SWS: 3.0
Fakultät für Informatik und Automatisierung			Fachgebiet: 2242

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

**Moderne Hashverfahren**

**Fachkompetenz:** Die Studierenden kennen die in der Vorlesung vorgestellten Konstruktionen und Beweise im Gebiet moderner Hashverfahren. Sie kennen die notwendigen Grundlagen aus der linearen Algebra und der Wahrscheinlichkeitsrechnung.

**Methodenkompetenz:** Die Studierenden können die Konstruktionsmethoden beschreiben, ihre Eigenschaften (insbesondere Zuverlässigkeitsaussagen) präzise benennen und die wesentlichen Beweise nachvollziehen und wiedergeben. Sie können Konstruktionen variieren und einschätzen, ob dadurch die Gültigkeit der Beweise eingeschränkt wird. Sie können die Praktikabilität der Verfahren einschätzen.

**Sozialkompetenz:** Selbst in der Vorlesung ist Interaktion stets möglich, ja erwünscht, damit sind die Studierenden vertraut. Die Studierenden haben die Erfahrung gemacht, dass Fragen unmittelbar geklärt werden, dass Diskussionen auf Augenhöhe stattfinden und Beiträge wertschätzend aufgenommen werden. In der Übung waren die Studierenden zu Präsentationen aufgerufen. Sie konnten wertvolle Erfahrung in der Rolle des Präsentierenden sammeln. Die Teilnahme erforderte und trainierte ein hohes Maß an Selbstorganisation.

### Vorkenntnisse

**Berechenbarkeit und Komplexitätstheorie**

**Effiziente Algorithmen**

**Stochastik; günstig: "Randomisierte Algorithmen"**

### Inhalt

**Moderne Hashverfahren**

Hashfunktionen bilden Schlüssel auf eine Indexmenge  $\{1, \dots, m\}$  ab. Aus dieser Grundsituation ergeben sich viele Anwendungen und Fragestellungen. Verschiedene Funktionalitäten, die in der Vorlesung diskutiert werden, sind:

- Dynamische Mengen ("member"-Test)
- Wörterbücher (dynamische Abbildungen mit "member"-Test)
- Retrieval (dynamische Abbildungen ohne "member"-Test)
- Approximative Mengen ("Bloom-Filter"-Funktionalität)
- (Minimale) Perfekte Hashfunktionen
- Analyse von Datenströmen

In der Vorlesung werden klassische und neue Algorithmen und ihre Analysen besprochen. Die hierfür nötigen Techniken aus der Wahrscheinlichkeitsrechnung, der (Hyper-) Graphentheorie und der Linearen Algebra werden bereitgestellt. Die Vorlesung bereitet auch auf weiterführende Arbeiten in dem Gebiet vor.

Konkrete Themen: Universelles Hashing, Konstruktion universeller Klassen, Anwendungen universeller Klassen:  $O(1)$ -Suche und perfekte Hashfunktionen, Momentanalyse bei Datenströmen mit 4-facher Unabhängigkeit, Lineares Sondieren und 5-fache Unabhängigkeit, High-Performance-Hashklassen und ihre Analyse, Verhalten von voll zufälligen Funktionen (Negative Korrelation, Poisson-Approximation, größte Buckets), Simulation von voll zufälligen Funktionen: "Split-and-Share", das Mehrfunktionen-Paradigma (Bloom-Filter, Cuckoo-Hashing, verallgemeinertes Cuckoo-Hashing), zufällige Hypergraphen, bipartite Graphen, Matrizen, "Retrieval": Werte ohne Membership-Test, neuere Konstruktionen perfekter Hashfunktionen

#### Medienformen

hauptsächlich Tafelvortrag, teilweise Folien

#### Literatur

Version 1: Moderne Hashverfahren

- M. Mitzenmacher, E. Upfal, Probability and Computing, Cambridge University Press, 2005
- R. Motwani und P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995
- Originalliteratur, wird in der Vorlesung genannt

#### Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Cellular Communication Systems

Modulabschluss: mehrere Teilleistungen

Art der Notengebung: Generierte Noten

Sprache: Englisch

Pflichtkenn.: Wahlmodul

Turnus: Wintersemester

Modulnummer: 200070

Prüfungsnummer: 220447

Modulverantwortlich: Prof. Dr. Andreas Mitschele-Thiel

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0																														
Fakultät für Informatik und Automatisierung			Fachgebiet: 2235																														
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS																							
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																																	

### Lernergebnisse / Kompetenzen

. Fachkompetenz: Die Studierenden verfügen nach der Vorlesung über Kenntnisse und Wissen zu Aufbau und Funktionsweise moderner zellulärer Mobilkommunikationssysteme, insbesondere von GSM, GPRS/EDGE, UMTS, LTE und 5G und deren Protokolle.

. Methodenkompetenz: Die Studierenden sind in der Lage, komplexe Fragestellungen zellulärer Mobilkommunikationssysteme zu verstehen, dieses Verständnis selbständig zu vertiefen und darauf aufbauend eigene Lösungen zu entwickeln.

. Systemkompetenz: Durch die Kombination aus Vorlesung und individuellen Arbeiten verstehen die Studierenden im Anschluss das Zusammenwirken der Komponenten und Einzelfunktionen des Systems und können den Einfluss von Entwurfsentscheidungen auf das System als Ganzes einschätzen.

. Sozialkompetenz: Die Studierenden sind in der Lage, Problemstellungen zellulärer Mobilkommunikationssysteme selbständig zu lösen und darzustellen. Durch die individuelle Erarbeitung eigener Lösungsvorschläge für ausgewählte Themen und deren Vorstellung und Diskussionen in der Gruppe haben Sie gelernt, Meinungen anderer Studierender zu beachten und diese kritisch zu hinterfragen. Das für die Lösung der Aufgaben benötigte Wissen konnten sie sich selbständig bzw. in Zusammenarbeit mit anderen aus verfügbaren Quellen erarbeiten, wurden sich durch die Präsentation der verschiedenen Möglichkeiten der Herangehensweise bei der Problemlösung bewusst und sind in der Lage die Leistungen Anderer entsprechend zu würdigen.

### Vorkenntnisse

Communication protocols and networks, basics of mobile communication networks

### Inhalt

- Review of mobile communication basics
- Overview on GSM and GPRS
- UMTS architecture (mobility management, connection and session management, wideband CDMA, management of radio resources)
- UMTS radio access network
- High-Speed Packet Access (HSPA)
- Long-Term Evolution (LTE)
- System Architecture Evolution (SAE)
- Self-organization in LTE

### Medienformen

Presentations with beamer, presentation slides

### Literatur

- Kaaranen, Ahtiainen, Laitinen, Naghian, Niemi. UMTS Networks - Architecture, Mobility and Services. Wiley, 2001
- Holma, Toskala. WCDMA for UMTS. revised edition, Wiley, 2002
- Dahlmann, Parkvall, Sköld. 4G: LTE/LTE-Advanced for Mobile Broadband, AP, 2011
- Stefania Sesia, Issam Toufik, Matthew Baker. LTE - The UMTS Long Term Evolution: From Theory to Practice

### Detailangaben zum Abschluss

**Das Modul Cellular Communication Systems mit der Prüfungsnummer 220447 schließt mit folgenden Leistungen ab:**

- alternativ semesterbegleitende Prüfungsleistung mit einer Wichtung von 40% (Prüfungsnummer: 2200720)
- mündliche Prüfungsleistung über 20 Minuten mit einer Wichtung von 60% (Prüfungsnummer: 2200721)

**Details zum Abschluss Teilleistung 1:**

The course consists of two parts: In the first part of the semester, lectures on the material are given. In the second part, individual studies (semester-long research projects that include a term paper and a presentation) help to improve understanding of the material.

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Deep Learning

Modulabschluss: mehrere Teilleistungen      Art der Notengebung: Generierte Noten  
 Sprache: Englisch      Pflichtkenn.: Wahlmodul      Turnus: Wintersemester

Modulnummer: 200131      Prüfungsnummer: 220488

Modulverantwortlich: Prof. Dr. Patrick Mäder

Leistungspunkte: 5      Workload (h): 150      Anteil Selbststudium (h): 105      SWS: 4.0  
 Fakultät für Informatik und Automatisierung      Fachgebiet: 2234

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Professional competence gained through lectures and examined through written exam:

- Students have knowledge about theoretical foundations of deep neural networks.
- Students have knowledge about CNN architectures and their applications.
- Students have knowledge about architectures for sequence modeling and their applications.

Methodological competence gained through seminars and examined through aPI (assignments):

- Students gained the ability to implement and apply a variety of deep learning algorithms.
- Students gained the ability to evaluate and troubleshoot deep learning models.
- Students gained the ability to use computational resources for training and application of deep learning models.

Social competence gained through lectures and seminars:

- Students gained insights in ethical aspects of machine learning (e.g., bias, autonomous driving) through discussions in lectures and seminars.
- Students can discuss advantages and disadvantages of different deep learning approaches among each other and with their lecturers and gained professionalism in mastering discussions beyond their mother tongue.

### Vorkenntnisse

- basic programming skills in Python
- basic understanding of machine learning preferable

### Inhalt

Deep learning has recently revolutionized a variety of applications like speech recognition, image classification, and language translation mostly driven by large tech companies, but increasingly also small and medium-sized companies aim to apply deep learning techniques for solving an ever increasing variety of problems. This course will give you detailed insight into deep learning, introducing you to the fundamentals as well as to the latest tools and methods in this rapidly emerging field.

Deep learning thereby refers to a subset of machine learning algorithms that analyze data in succeeding stages, each operating on a different representation of the analyzed data. Specific to deep learning is the ability to automatically learn these representations rather than relying on domain experts for defining them manually. The course will teach you the theoretical foundations of deep neural networks, which will provide you with the

understanding necessary for adapting and successfully applying deep learning in your own applications. Additionally, by completing the course, you will be able to implement, parametrize and apply a variety of deep learning algorithms. You will learn how to use deep convolutional neural networks (CNNs) as well as recurrent neural networks (RNNs) for image, text, and time series analysis. You will further become familiar with advanced data science tools and in using computational resources to train and apply deep learning models.

#### Medienformen

- Projector presentation
- Slide decks available through Moodle
- Assignment management through Moodle
- Cloud services (personal computer required)

#### Literatur

- Deep Learning: Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press (2016)
- Pattern Recognition and Machine Learning: Christopher M. Bishop, Springer (2006)
- Hands-On Machine Learning with Scikit-Learn and TensorFlow: Aurélien Géron, O'Reilly Media (2017)

#### Detailangaben zum Abschluss

**Das Modul Deep Learning mit der Prüfungsnummer 220488 schließt mit folgenden Leistungen ab:**

- alternativ semesterbegleitende Prüfungsleistung mit einer Wichtung von 40% (Prüfungsnummer: 2200822)
- Prüfungsleistung mit einer Wichtung von 60% (Prüfungsnummer: 2200823)

Details zum Abschluss Teilleistung 1:

- several assignments evaluating methodological and practical competence in the taught concepts to be solved at home with due date and submission via Moodle
- number of assignments announced in the early lectures of the course
- partial result determined as average across submitted solutions of assignments
- students must register via thoska for this exam, typically within the 3rd and 4th week of the semester

Details zum Abschluss Teilleistung 2:

- written test consisting of multiple choice and free form questions evaluating the professional competence in the topics of the course
- preferably conducted digitally via Moodle
- final results may be scaled or individual questions may be excluded depending on best performing percentile of students

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Eingebettete Computerarchitekturen

Modulabschluss: Prüfungsleistung alternativ

Art der Notengebung: Gestufte Noten

Sprache: Deutsch

Pflichtkenn.: Wahlmodul

Turnus: Sommersemester

Modulnummer: 200140

Prüfungsnummer: 2200835

Modulverantwortlich: Prof. Daniel Ziener

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 128	SWS: 2.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2231							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

#### Fachkompetenz:

Der Student wird nach Abschluss dieses Moduls in der Lage sein, bereits gelernte Prinzipien der Rechnerarchitekturen sowie der parallelen Bearbeitung von Problemen gezielt im Bereich der rekonfigurierbaren Hardware anzuwenden.

Sie verstehen in erster Linie die theoretischen Methoden, und können diese auch praktisch umsetzen. Sie können rekonfigurierbare Architekturelemente in einer Hardwarebeschreibungssprache (VHDL) entwerfen. Sie beherrschen die Anwendung modellbasierter Entwurfsverfahren für den Entwurf rekonfigurierbarer Systeme. Im Ergebnis der praktischen Ausarbeitungen sind sie zu einem konkreten Entwurf praktisch relevanter Beispielsysteme befähigt.

Systemkompetenz: Die Studierenden verstehen das Zusammenwirken der Architekturelemente rekonfigurierbarer Systeme im Zusammenhang mit deren weiteren Elementen und mit dem Verhalten des einbettenden Systems. Sie begreifen die fundamentale Bedeutung durchgängiger Entwurfsverfahren und sind mit den dazugehörigen Vorgehensmodellen vertraut. Sozialkompetenz: Die Studierenden sind in der Lage, grundlegende Problemstellungen bei Planung und Entwurf rekonfigurierbarer Systeme allein und in der Gruppe zu lösen. Die Studierenden können praktische Problemlösungen gemeinsam im Kontext der praktischen Ausarbeitungen diskutieren und können Kritik und Anmerkungen würdigen. Zusätzlich sind durch die Notwendigkeit der frequentierten Lösung kleinerer Teilaufgaben die Selbstkompetenzen zum kontinuierlichen Arbeiten gestärkt.

### Vorkenntnisse

Technische Informatik, Rechnerarchitektur 1+2

### Inhalt

Einführung und Begriffsbestimmung "Eingebettetes System" und "Rekonfigurierbares System"

#### Architekturen:

- Mikrocontroller, DSP, GPU
- Rekonfigurierbare Systeme, VHDL
- System-on-Chip und System-on-Programmable-Chip

#### Entwurf und Implementierung:

- Begriffsklärungen zum Entwurf
- modellbasierter Entwurf
- Design Flow
- HW/SW-Co-Design
- Logiksynthese
- Partitionierung und Platzierung
- Test- und Inbetriebnahmeverfahren

### Medienformen

Folien, Anschriebe, Programmierbeispiele, Aufgabenausarbeitungen & Beispiellösungen

Link zum Moodle-Kurs:

<https://moodle2.tu-ilmenau.de/course/view.php?id=3527>

### Literatur

Peter Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things. Jürgen Teich, Christian Haubelt, Digitale Hardware/Software-Systeme: Synthese und

Optimierung. M. Wolf, Computers as Components: Principles of Embedded Computing System Design. Andre Dehon: Reconfigurable Architectures for General- Purpose Computing K. Compton, S. Hauck: Configurable Computing: A Survey of Systems and Software Dirk Koch: Partial Reconfiguration on FPGAs

#### Detailangaben zum Abschluss

Die semesterbegleitende Prüfung besteht aus zwei separierten Aufgabenteilen:

Teil 1 dem Lösen einer gegebenen praktischen Aufgabenstellung. Hierbei sollen die Studenten ein mathematisches Problem auf einem Mikrocontroller praktisch umsetzen.

Die Aufgabe besteht dabei in der Umsetzung von softwareseitigen Optimierungen für das gegebene Problem, sowie die anschließende fehlerfreie Umsetzung dieser. Die Ergebnisse sollen in einem kurzen Report dokumentiert werden. Dieser bildet die Grundlage für die Bewertung dieser Teilleistung. Teil 2 baut thematisch auf Teil 1 auf. Hier sollen die Studenten das gleiche mathematische Problem auf einer Hardware ihrer Wahl umsetzen (z.B. FPGA, GPU). Dabei sollen vor allem die Möglichkeiten zur Optimierung der gewählten HW-Plattform verwendet werden. Im Anschluss soll die erzielte Lösung mit dem Ergebnis aus Teil 1 verglichen werden. Sowohl die erzielten praktischen Ergebnisse aus Teil 2, als auch der Vergleich sollen im Anschluss in einem weiteren Report dargestellt werden. Im Abschluss wird es eine kurze Präsentation der Ergebnisse geben. Grundlage für die Bewertung der zweiten Teilleistung sind sowohl der zweite Report, als auch der Vortrag.

verwendet in folgenden Studiengängen:

Master Informatik 2021



## Modul: Forschungsprojekt

Modulabschluss: Prüfungsleistung alternativ Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch/Englisch Pflichtkennz.: Wahlmodul Turnus: ganzjährig

Modulnummer: 200739 Prüfungsnummer: 2200846

Modulverantwortlich: Silke Eberhardt-Schmidt

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 150	SWS: 0.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2200							
SWS nach Fach- semester	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P

### Lernergebnisse / Kompetenzen

Im Rahmen eines Projektes mit einer definierten Aufgabe und Zielsetzung sind die Studierenden in der Lage, neue Lösungen in der Informatik oder Ingenieurinformatik zu entwickeln. Sie können in der Projektarbeit ihre Kenntnisse und ihren Sachverstand aus dem Studium zur Lösung der Aufgabe in Anlehnung gezielt einsetzen. Die Studierenden sind in der Lage, komplexe Zusammenhänge zu analysieren, Lösungsalternativen zu entwickeln und zu bewerten, und diese mit Techniken der Ingenieurinformatik umzusetzen und zu überprüfen. Darüber hinaus sind Studierende fähig, Ergebnisse wissenschaftlich vorzustellen und diese zu diskutieren. Sie sind fähig, Kritik anzunehmen und Hinweise zu beachten.

### Vorkenntnisse

### Inhalt

- Bearbeitung eines Projektes aus dem Themenbereich der Informatik oder Ingenieurinformatik unter Betreuung, beispielsweise innerhalb eines laufenden Forschungsprojektes der Fachgebiete
- Dokumentation der Arbeit (Konzeption eines Arbeitsplanes, Literaturrecherche, Stand der Technik)
- Wissenschaftliche Tätigkeiten (z. B. Anforderungsanalyse, Entwurf, Implementierung, Validierung und Testen, Modellierung, Simulation, Auswertung)
- Auswertung und Diskussion der Ergebnisse
- Verfassen einer schriftlichen Projektarbeit
- Vorstellung der Ergebnisse mit anschließender Diskussion

### Medienformen

### Literatur

Themenspezifische Literatur wird zu Beginn der Arbeit vom Betreuer benannt bzw. ist selbstständig zu recherchieren. Empfohlen wird außerdem Literatur zu wissenschaftlichem Arbeiten, Literaturrecherche (beispielsweise Angebote der Bibliothek) und Präsentationstechniken.

### Detailangaben zum Abschluss

- Projektbericht (schriftliche Arbeit)
- Je nach Aufgabenstellung: Softwareentwurf, Implementierung, Simulationsergebnisse etc.
- Präsentation mit Diskussion

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Fortgeschrittene Rechnerarchitekturen

Modulabschluss: Prüfungsleistung mündlich 40 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Wintersemester

Modulnummer: 200122 Prüfungsnummer: 2200807

Modulverantwortlich: Dr. Bernd Däne

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0
Fakultät für Informatik und Automatisierung			Fachgebiet: 2231

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

Lernergebnisse / Kompetenzen

Lernergebnisse und Kompetenzen aus der Vorlesung:

**Fachkompetenz:** Die Studierenden verstehen detailliert gemeinsame Merkmale, Unterscheidungskriterien, Einsatzgebiete, Aufbau und Funktionsweise von Einchipcontrollern und Digitalen Signalprozessoren. Die Studierenden kennen Aufbau und Funktionsweise ausgewählter typischer Vertreter. Die Studierenden verstehen die Funktionen von Softwarewerkzeugen, die in typischen Entwicklungsprozessen für Einchipcontroller und Digitale Signalprozessoren zum Einsatz kommen.

Die Studierenden verstehen detailliert allgemeine Eigenschaften, Vor- und Nachteile, Bedeutung, Aufbau, Funktion und Einsatzmöglichkeiten der behandelten speziellen und innovativen Rechnerarchitekturen. Die Studierenden erkennen die Wirkungsweise ausgewählter Einzelfunktionen anhand beispielhafter Demonstrationen.

**Methodenkompetenz:** Die Studierenden sind in der Lage, die Eigenschaften und Einsatzmöglichkeiten von Einchipcontrollern und Digitalen Signalprozessoren zu analysieren und ihre Eignung für unterschiedliche Aufgaben zu beurteilen. Die Studierenden sind in der Lage, den Einsatz von Einchipcontrollern und Digitalen Signalprozessoren unter Benutzung von Herstellerinformationen zu planen und durchzuführen.

Die Studierenden sind in der Lage, spezielle und innovative Rechnerarchitekturen zu analysieren, ihre Einsatzmöglichkeiten zu beurteilen und ihre Einordnung innerhalb der behandelten Rechnerarchitekturen zu erkennen.

**Systemkompetenz:** Die Studierenden erkennen den Zusammenhang zwischen Architektur und Anwendung auf dem Gebiet von Einchipcontrollern und Digitalen Signalprozessoren. Die Studierenden verstehen die Bedeutung von Einchipcontrollern und Digitalen Signalprozessoren im Zusammenhang mit der Realisierung eingebetteter Systeme.

Die Studierenden erkennen die Vielfalt und Weiterentwicklung der Rechnerarchitekturen als Teil des allgemeinen technischen Fortschritts.

**Sozialkompetenz:** Die Studierenden sind in der Lage, einem Vortrag konzentriert und aufmerksam zu folgen und Störungen zu vermeiden. Sie erkennen die geeigneten Zeitpunkte zum Stellen von Fragen.

Neben der Vorlesung wird als weitere Lernform das quellenbasierte Selbststudium eingesetzt.

Lernergebnisse und Kompetenzen aus dem Selbststudium:

**Fachkompetenz:** Die Studierenden erweitern ihr Wissen über die behandelten Sachverhalte anhand von Quellenmaterial. Sie erarbeiten sich weitere Einzelkenntnisse und vertiefen das Wissen, das sie in der Vorlesung erworben haben.

**Methodenkompetenz:** Die Studierenden sind in der Lage, geeignete und zuverlässige Informationen in der Literatur und im Internet aufzufinden. Sie sind in der Lage, die gefundenen Informationen einzuordnen und mit dem bereits erworbenen Wissen zu verknüpfen. Sie sind darüber hinaus in der Lage, ausgehend von einer konkreten Aufgabe oder Fragestellung die Notwendigkeit eines Quellenstudiums zu erkennen und dieses sachgerecht durchzuführen.

**Systemkompetenz:** Die Studierenden vertiefen ihr Verständnis über die Einordnung und Bedeutung der behandelten einzelnen Architekturen innerhalb der Gesamtbetrachtung der Rechnerarchitekturen.

**Sozialkompetenz:** Die Studierenden besitzen eine Balance zwischen dem selbstständigen, konzentrierten Arbeiten und dem Austausch mit Anderen.

## Vorkenntnisse

Grundkenntnisse zu Rechnerarchitekturen und Technischer Informatik aus den entsprechenden Lehrveranstaltungen.

## Inhalt

- Themenkomplex "Einchipcontroller und Digitale Signalprozessoren"
  - Aufbau, Funktionsweise, Gemeinsamkeiten und Unterscheidungskriterien von Einchipcontrollern (Einchipmikrorechner, EMR; auch: Mikrocontroller,  $\mu$ C) und Digitalen Signalprozessoren (DSP);
  - Detaillierte Betrachtung von EMR an Beispielen: Prozessorkerne, maschinennahe Programmierung, integrierte Peripheriefunktionen; Entwicklungswerkzeuge und Entwicklungsabläufe
  - Detaillierte Betrachtung von DSP an Beispielen: Prozessorkerne, maschinennahe Programmierung, integrierte Peripheriefunktionen; Entwicklungswerkzeuge und Entwicklungsabläufe
  - Themenkomplex "Spezielle und innovative Rechnerarchitekturen"
    - Vektorrechner,
    - Virtuelle Befehlssatzarchitekturen,
    - Datenfluss-Architekturen,
    - Processing in Memory (PIM),
    - Neurocomputer,
    - Tendenzen bei Steuerfluss-Prozessoren,
    - Optische Computer,
  - Quantencomputer

## Medienformen

Anschriebe, Folien, Rechnerdemonstrationen, Downloads

<https://moodle2.tu-ilmenau.de/course/view.php?id=3069>

<https://moodle2.tu-ilmenau.de/course/view.php?id=3099>

## Literatur

Weiterführende Literaturhinweise:

Onlinequellen der Hersteller Infineon und Texas Instruments zu Einchipcontrollern und DSP. Umfangreiche Sammlung von Onlinequellen und Einzelartikeln zu speziellen und innovativen Rechnerarchitekturen.

Diese sind den Webseiten zu entnehmen:

<http://tu-ilmenau.de/?r-dsp>

<http://tu-ilmenau.de/?r-sira>

Die Beschaffung von Literatur ist nicht gefordert.

## Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Mensch-Maschine-Interaktion

Modulabschluss: mehrere Teilleistungen

Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkennz.: Wahlmodul

Turnus: Wintersemester

Modulnummer: 200086

Prüfungsnummer: 220456

Modulverantwortlich: Prof. Dr. Horst-Michael Groß

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2233							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

Im Modul "Mensch-Maschine-Interaktion" haben sich die Studierenden die Begrifflichkeiten und das Methodenspektrum der Mensch-Maschine Interaktion unter Realwelt-Bedingungen angeeignet. Sie beherrschen wichtige Basisoperationen zur (vorrangig visuellen) Wahrnehmung von Menschen und zur Erkennung von deren Intentionen und Zuständen und kennen Techniken zur nutzeradaptiven Dialogführung. Die Studierenden sind in der Lage, Fragestellungen aus dem o. g. Problembereichen zu analysieren, durch Anwendung des behandelten Methodenspektrums Lösungskonzepte für unterschiedliche Fragestellungen der Service- und Assistenzrobotik zu entwerfen und umzusetzen, sowie bestehende Lösungskonzepte zu bewerten. Vor- und Nachteile der Komponenten und Verfahren im Kontext praktischer Anwendungen sind den Studierenden bekannt. Nach intensiven Diskussionen während der Übungen und zur Auswertung der Python-Implementierung können die Studierenden Leistungen ihrer Mitkommilitonen richtig einschätzen und würdigen. Sie berücksichtigen Kritik, beherzigen Anmerkungen und nehmen Hinweise an.

### Vorkenntnisse

Pflichtmodul "Neuroinformatik und Maschinelles Lernen" und Wahlmodul "Deep Learning für Computer Vision"

### Inhalt

Das Modul vermittelt das erforderliche Methodenspektrum aus theoretischen Grundkenntnissen und praktischen Fähigkeiten zum Verständnis, zur Implementierung und zur Anwendung von Verfahren der Interaktion zwischen Mensch und Maschine (mit Fokus auf vision-basierten Verfahren sowie dem Einsatz auf Robotersystemen) sowie zur erforderlichen Informations- und Wissensverarbeitung. Sie ergänzt das parallel laufende Modul "Robotvision", das sich um Aspekte der Roboternavigation kümmert, um wichtige Erkennungsverfahren der Mensch-Roboter Interaktion (HRI). Das Modul vermittelt das dazu notwendige Faktenwissen sowie begriffliches, methodisches und algorithmisches Wissen aus den folgenden Kernbereichen:

A - Ausgewählte Basisoperationen für viele Erkennungsverfahren

- Basisoperationen der MMI im Rahmen eines Mustererkennungsprozesses
- Leistungsbewertung von Klassifikatoren: Gütemaße; Crossvalidation-Technik; Bewertung von binären Klassifikatoren, Gütemaß ROC/Precision Recall Kurven, usw.
- Bildaufbereitung und Bildanalyse: Beleuchtungs-/ Histogrammausgleich; AuflösungsPyramiden; Lineare Subspace Methoden (HKA / PCA); Gabor-Wavelet-Funktionen (Gaborfilter) zur effizienten Bildbeschreibung;
- Bewegungsanalyse in Videosequenzen
- Techniken zur Repräsentation von Zeit: Dynamic Time Warping, Hidden Markov Modelle (HMMs)
- Bayes Filtering als probabilistische Zustandsschätzer: Grundidee, Markov-Annahme, Grundprinzip des rekursiven Bayes-Filters, Bewegungs- und Sensormodell, Arten der Beliefrepräsentation in Bayes Filtern; Partikel Filter

B - Wichtige Verfahren zur Erkennung von Nutzerzustand & Nutzerintention

- Vision-basierte Nutzerdetektion, Nutzertracking, Nutzeridentifikation
- Zeigeposen- und Gestenerkennung
- Erkennung von Mimik (Emotionen, Stress) und Interaktionsinteresse + aktuelle Entwicklungen
- Multimodale Dialogsysteme: Bestandteile von Dialogsystemen; Besonderheiten multimodaler Dialogsysteme

## C - Anwendungsbeispiele für Assistenzfunktionen in öffentlicher & privater Umgebung

- Soziale Assistenzroboter für die Gesundheitsassistenz
- Robotische Bewegungsassistenz am Beispiel Reha

## D - Gastvorlesung zur sprachbasierten MMI und zu Hidden Markov Modellen sowie deren Einsatz in der Spracherkennung, Unterschriftserkennung und Gestenerkennung

Im Rahmen der Teilleistung 2 werden ausgewählte methodische und algorithmische Grundlagen der MMI durch die Studierenden selbst softwaretechnisch umgesetzt und durch kleine Programmbeispiele vertieft. Als Programmiersprache wird Python verwendet. Für Verfahren des Maschinellen Lernens wird die scikit-Learn Toolbox verwendet.

### Medienformen

Präsenzvorlesung mit Powerpoint, Arbeitsblätter zur Vorlesung, Übungsaufgaben, Videos, Python Apps, e-Learning mittels "Jupyter Notebook", Moodle-Kurs

### Literatur

- Schenk, J, Rigoll, G. Mensch-Maschine-Kommunikation: Grundlagen von sprach- und bildbasierten Benutzerschnittstellen, Springer 2010
- Li, S und Jain, A.: Handbook of Face Recognition, 2004
- Bishop, Ch.: Pattern Recognition and Machine Learning, Springer 2006
- Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction: Foundations and Applications, Studies in fuzziness and soft computing 207, Springer, 2006
- Maltoni, D., et al.: Biometric Fusion, Handbook of Fingerprint Recognition, Kapitel 7, Springer, 2009

### Detailangaben zum Abschluss

**Das Modul Mensch-Maschine-Interaktion mit der Prüfungsnummer 220456 schließt mit folgenden Leistungen ab:**

- schriftliche Prüfungsleistung über 90 Minuten mit einer Wichtung von 100% (Prüfungsnummer: 2200745)
- alternativ semesterbegleitende Studienleistung mit einer Wichtung von 0% (Prüfungsnummer: 2200746)

Details zum Abschluss Teilleistung 2:

eigene Python-Implementierungen von vorgegebenen Algorithmen und Übungsaufgaben

### verwendet in folgenden Studiengängen:

Master Informatik 2013

Master Informatik 2021

## Modul: Modellgetriebene Softwareentwicklung

Modulabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkennz.: Wahlmodul

Turnus: Wintersemester

Modulnummer: 200017 Prüfungsnummer: 220432

Modulverantwortlich: Prof. Dr. Armin Zimmermann

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 94 SWS: 5.0  
Fakultät für Informatik und Automatisierung Fachgebiet: 2236

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studierenden haben Wissen und Fähigkeiten in der Erstellung domänenspezifischer Sprachen (DSL) erworben und können Editoren dafür entwickeln. Sie verstehen Modelltransformationen (M2M & M2T) und können sie im Entwicklungsprozess modellgetriebener Softwareentwicklung erfolgreich einsetzen. Sie haben Kenntnisse von Metameta-Modellen (ECORE, EMOF) für die UML und konnten die grundlegenden OMG Standardspezifikationen erlernen. Zur praktischen Übung konnten die theoretischen Inhalte in einem begleitenden Praktikum eingesetzt werden. Danach waren die Studierenden in der Lage, in kleinen Teams eigene Lösungen für Problemstellungen aus dem Bereich der Vorlesung zu entwickeln. Sie können Herangehensweisen und Lösungswege diskutieren, konstruktive Kritik geben und ihre Lösungen vorstellen.

### Vorkenntnisse

Besuch der Lehrveranstaltung OOM  
Alternativ: Kenntnisse der UML und des Meta-Modells der UML sowie Grundlagen der objektorientierten Programmierung

### Inhalt

Die Model-Driven Architecture (MDA) ist der Object Management Group (OMG) -Ansatz des Model-Driven (Software) Developments (MDD) zur modellgetriebenen und generativen Soft- und Hardwareentwicklung. Ziel der MDA ist es die Lücke zwischen Modell und Quelltext zu schließen und den Automatisierungsgrad der Entwicklung zu erhöhen. Dies erfolgt durch eine automatische Generierung von Quellcode aus Domänenspezifischen Modellen, die auf definierten Domänenspezifischen Sprachen (DSL) beruhen. Im Ergebnis sollen die Fehlerquellen während der Entwicklung reduziert werden und die Software schneller, effizienter, kostengünstiger und qualitativ hochwertiger erstellt werden. Für die Anwendung dieses Ansatzes sind verschiedene Kenntnisse und Fähigkeiten notwendig:

- Kenntnisse in einer Programmiersprache, in der Zielsprache und in der Modellierungssprache
- Es müssen unterschiedliche Modellierungstechniken beherrscht werden
- Eine Kerntechnologie der MDA sind die Transformationstechnologien
- Es existieren viele verschiedene Werkzeuge und recht komplexe Toolchains, die beherrscht werden müssen

Im Rahmen dieser Lehrveranstaltung sollen diese notwendigen Kenntnisse und Fähigkeiten vermittelt werden. Im Seminar sollen mit Hilfe des Eclipse Modeling Projects (EMP) und des Eclipse Sirius Projects praktische Aufgabenstellungen gelöst werden. Hierbei soll ein eigener Editor für eine eigene Domänenspezifische Sprache erstellt werden.

### Medienformen

Präsentationsfolien, alle Unterlagen im Web verfügbar.  
Tafel, Beamer und PC Raum für aPL.  
Moodle: (Link: <https://moodle2.tu-ilmenau.de/course/view.php?id=151>)

### Literatur

- [1] V. Gruhn, D. Pieper, and C. Röttgers, MDA®: Effektives Software-Engineering mit UML2® und Eclipse (TM) (Xpert.press) (German Edition). Dordrecht: Springer, 2007.

- [2] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, EMF: Eclipse modeling framework, 2nd ed. Upper Saddle River, NJ: Addison-Wesley, 2011.

- [3] R. C. Gronback, Eclipse modeling project: A domain-specific language toolkit. Upper Saddle River, N.J: Addison-Wesley, 2009.

- [4] Object Management Group, MDA - The Architecture Of Choice For A Changing World. [Online] Available: <http://www.omg.org/mda/>.

- [5] Object Management Group, OMG Specifications. [Online] Available: <http://www.omg.org/spec/>.

#### Detailangaben zum Abschluss

**Das Modul Modellgetriebene Softwareentwicklung mit der Prüfungsnummer 220432 schließt mit folgenden Leistungen ab:**

- alternativ semesterbegleitende Prüfungsleistung mit einer Wichtung von 40% (Prüfungsnummer: 2200652)
- schriftliche Prüfungsleistung über 90 Minuten mit einer Wichtung von 60% (Prüfungsnummer: 2200653)

Details zum Abschluss Teilleistung 1:

praktische Ausarbeitung mit Dokument; ist organisatorisch vor der sPL abzuschließen

Details zum Abschluss Teilleistung 2:

schriftliche Prüfung, keine Hilfsmittel; Planung als Ausnahme im 2. PZR, damit Projekt vorher abgeschlossen werden kann

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Robotvision

Modulabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkennz.: Wahlmodul

Turnus: Wintersemester

Modulnummer: 200087

Prüfungsnummer: 220457

Modulverantwortlich: Prof. Dr. Horst-Michael Groß

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 105	SWS: 4.0							
Fakultät für Informatik und Automatisierung			Fachgebiet: 2233							
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

Im Modul "Robotvision" haben die Studierenden die Begrifflichkeiten und das Methodenspektrum des Maschinellen Sehens mit Fokus in der mobilen Robotik kennen gelernt. Sie haben das Paradigma der handlungsorientierten Wahrnehmung - insbesondere zur visuellen Roboternavigation in natürlicher Umwelt verstanden. Sie beherrschen wichtige Basisoperationen für die visuelle Wahrnehmung der Umgebung (Tiefe, Bewegung, Hindernisse, Freiraum, Räumlichkeiten, eigene Position in der Welt) und können Handlungskonsequenzen aus der visuellen Wahrnehmung der Umgebung ableiten. Sie haben Techniken der vision-basierten Umgebungswahrnehmung und der lokalen und globalen Navigation von Kognitiven Robotern in komplexer realer Einsatzumgebung kennen gelernt.

Die Studierenden sind in der Lage, Fragestellungen aus dem o. g. Problemkreisen zu analysieren, durch Anwendung des behandelten Methodenspektrums Lösungskonzepte für unterschiedliche Fragestellungen der Service- und Assistenzrobotik zu entwerfen und umzusetzen, sowie bestehende Lösungskonzepte zu bewerten. Vor- und Nachteile der Komponenten und Verfahren im Kontext praktischer Anwendungen sind den Studierenden bekannt. Nach intensiven Diskussionen während der Übungen und zur Auswertung des Praktikums (Teilleistung 2) können die Studierenden Leistungen ihrer Mitkommilitonen richtig einschätzen und würdigen. Sie berücksichtigen Kritik, beherzigen Anmerkungen und nehmen Hinweise an.

### Vorkenntnisse

Kognitive Robotik

### Inhalt

Das Modul vermittelt das erforderliche Methodenspektrum aus theoretischen Grundkenntnissen und praktischen Fähigkeiten zum Verständnis, zur Implementierung und zur Anwendung von Verfahren der vision-basierten Roboternavigation sowie zur erforderlichen Informations- und Wissensverarbeitung. Es vermittelt sowohl Faktenwissen, begriffliches und algorithmisches Wissen aus folgenden Themenkomplexen:

- Basisoperationen d. Roboternavigation
- Neuronale Basisoperationen der visuo-motorischen Verarbeitung - der neuronale Instruktionssatz: funktionelle und topografische Abbildungen (u.a. log-polare Abbildung), AuflösungsPyramiden, neuronale Felddynamik, ortsvariante Informationsverarbeitung
- Basisoperationen & Technologien für die visuelle Umgebungswahrnehmung:
  - Detektoren & Deskriptoren für Interest-Points in 2D-Bildern
  - Bewegungssehen und optischer Fluss
  - Tiefenwahrnehmung, Tiefenkameras (RGB-D Kameras)
  - Detektoren & Deskriptoren für Tiefenbilder (3D-Bilder)
  - Visuelle Odometrie
- Vision-basierte Roboternavigation
  - Hindernisvermeidung (u.a. flussbasiert, Untergrund-Segmentierung)
  - Mapping und Selbstlokalisierung
  - Visuelles SLAM (Simultaneous Localization and Map Building inkl. ORB-SLAM)
- Innovative Entwicklungen (z.B. Semantisches Labeln)
- Exemplarische Software-Implementierungen von Basisoperationen



Im Rahmen des Praktikums werden die behandelten methodischen und algorithmischen Grundlagen der vision-basierten Roboternavigation durch die Studierenden selbst softwaretechnisch umgesetzt und im Rahmen eines vorgefertigten Robotersimulations-Frameworks implementiert (Teilleistung 2).

#### Medienformen

Präsenzvorlesung mit Powerpoint, Arbeitsblätter zur Vorlesung, Übungsaufgaben, Videos, Python Apps, e-Learning mittels "Jupyter Notebook", Moodle-Kurs

#### Literatur

- Hertzberg, J., Lingemann, K., Nüchter, A.: Mobile Roboter, Springer 2012
- Siegwart, R., Nourbakhsh, I. R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots. MIT Press 2004
- Jähne, B. Digitale Bildverarbeitung. Springer Verlag 2005
- Bradsky, G., Kaehler, A. Learning OpenCV: Computer Vision with OpenCV Library
  
- Siciliano, B., Khatib: O. Springer Handbook of Robotics, Springer 2016
  
- Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics, MIT Press 2005

#### Detailangaben zum Abschluss

**Das Modul Robotvision mit der Prüfungsnummer 220457 schließt mit folgenden Leistungen ab:**

- schriftliche Prüfungsleistung über 90 Minuten mit einer Wichtung von 100% (Prüfungsnummer: 2200747)
- Studienleistung mit einer Wichtung von 0% (Prüfungsnummer: 2200748)

Details zum Abschluss Teilleistung 2:

Erfolgreiche Implementierung von zwei vorgegebenen Navigationsalgorithmen im vorhandenen Navigationsframe Simulator

verwendet in folgenden Studiengängen:

Master Informatik 2013  
Master Informatik 2021

**Modul: Spezielle Aspekte Integrierter Hard- und Softwaresysteme**

Modulabschluss: Prüfungsleistung mündlich 20 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch/Englisch Pflichtkenn.: Wahlmodul Turnus: Wintersemester

Modulnummer: 200076 Prüfungsnummer: 2200730

Modulverantwortlich: Prof. Dr. Andreas Mitschele-Thiel

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 105 SWS: 4.0  
 Fakultät für Informatik und Automatisierung Fachgebiet: 2235

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

**Lernergebnisse / Kompetenzen**

**Fachkompetenz:** Die Studierenden verfügen nach Abschluß der Lehrveranstaltung über ein grundlegendes Systemwissen und Vorgehensstrategien, das es ihnen erlaubt, zentrale Entscheidungen bezüglich Verifikation, Validierung und Test von gemischten Hard- und Softwaresystemen zu treffen.

**Methodenkompetenz:** Die Studierenden sind nach Abschluss der Lehrveranstaltung in der Lage, aus einer Auswahl an Validierungstechniken die für ihr Problem geeignetste auszuwählen, geeignete Methoden zur funktionalen und temporalen sowie strukturellen Validierung auszuwählen und anzuwenden und damit gezielt die nötigen Entwurfs- und Testentscheidungen im Entwicklungs- und Implementierungsprozess komplexer HW/SW-Systeme treffen zu können.

**Systemkompetenz:** Die Kombination von Vorlesung und Übung befähigt die Studierenden zu einer Systemsicht, d.h. der Abstraktion von unzähligen Details des Validierungsprozesses, um so das System als Ganzes verifizierbar und testbar zu gestalten und zu entwickeln.

**Sozialkompetenz:** Die Studierenden sind in der Lage, Problemstellungen der Entwicklung integrierter HW/SW-Systeme bezüglich Validierung und Testbarkeit selbständig zu lösen und darzustellen. Durch praktische Übungen in Kleingruppen haben Sie gelernt, Meinungen anderer Studierender zu beachten und diese kritisch zu hinterfragen. Das für die Lösung der Aufgaben benötigte Wissen erarbeiteten sie sich selbständig bzw. in Zusammenarbeit mit anderen aus verfügbaren Quellen. Sie wurden sich durch die Präsentation der verschiedenen Möglichkeiten der Herangehensweise bei der Problemlösung bewusst und sind in der Lage die Leistungen Anderer entsprechend zu würdigen.

**Vorkenntnisse**

Inhalte der Veranstaltung Schaltsysteme

**Inhalt**

Auswahl von Themen zum fortgeschrittenen Stand des Gebietes Integrierte Hard- und Softwaresysteme, wie z. B.:

- Validierung, Verifikation und Test
- Testverfahren
- Fehlerabdeckung
- Boundary Scan
- Built in self Test
- Fehlerdiagnose

**Medienformen**

kurzfristig unter Lehrmaterial auf den WEB-Seiten der beteiligten Fachgebiete abrufbare pdf-Dateien

**Literatur**

Literaturangaben individuell zu den behandelten Themen in der Vorlesung bzw. im bereitgestellten Lehrmaterial

**Detailangaben zum Abschluss**

**verwendet in folgenden Studiengängen:**

Master Informatik 2021

## Modul: Systemtechnik und Systemtheorie der Bildverarbeitung

Modulabschluss: mehrere Teilleistungen Art der Notengebung: Generierte Noten

Sprache: Deutsch

Pflichtkennz.: Wahlmodul

Turnus: Wintersemester

Modulnummer: 200241

Prüfungsnummer: 230482

Modulverantwortlich: Prof. Dr. Gunther Notni

Leistungspunkte: 5	Workload (h): 150	Anteil Selbststudium (h): 94	SWS: 5.0							
Fakultät für Maschinenbau		Fachgebiet: 2362								
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS
Fach-	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P	V S P
semester										

### Lernergebnisse / Kompetenzen

Der Hörer hat einen umfassenden Überblick zu systemtechnischen und -theoretischen Aspekten der Bildverarbeitung. Wichtiges Hilfsmittel der Wissensvermittlung sind zahlreiche Praxisbeispiele in Vorlesung und Übungen. Zusammen mit dem Dozenten kann der Hörer diese analysieren und diskutieren.

#### Fachkompetenz:

Die Studierenden beherrschen die Grundbegriffe der Systemtechnik der Bildverarbeitung, können Kamera- und Beleuchtungssysteme bewerten und sind fähig, Aufgaben der Bildverarbeitung in unterschiedlichen Anwendungsszenarien zu analysieren. Sie sind in der Lage, Bildverarbeitungssysteme zu konzipieren, auszulegen, Lösungen zum praktischen Einsatz zu entwerfen und die Eigenschaften der Systeme und von Einzelkomponenten zu bewerten.

Nach dem zugehörigen Seminar und den praktischen Anwendungen sind die in der Vorlesung erworbenen Kenntnisse in vier Versuchen mit den Inhalten: Charakterisierung von Kamerasystemen (EMVA-Standard 1288) und Methoden der 3D-Datenerfassung, gefestigt.

#### Methodenkompetenz:

Im Ergebnis ist der Hörer in der Lage, Probleme der Bildverarbeitung zu analysieren und zu klassifizieren sowie wichtige Schritte der Problemlösung abzuleiten. Mit den vermittelten Kompetenzen ist der Hörer befähigt, in konkreten Anwendungen der Bildverarbeitung entwickelnd tätig zu werden.

#### Sozialkompetenz:

Sie haben gelernt, Aufgaben der industriellen Bildverarbeitung im Team im Rahmen von Praktikumsgruppen (3-4 Studenten) zu lösen, die Leistungen ihrer Mitkommilitonen anzuerkennen und Meinungen anderer zu berücksichtigen.

### Vorkenntnisse

Naturwissenschaftliche und ingenieurwissenschaftliche Fächer des Grundstudiums

### Inhalt

Systemtechnik und Systemtheorie der Bildverarbeitung

Die Vorlesung Systemtechnik und Systemtheorie der Bildverarbeitung vermittelt vertiefendes Wissen im systemtechnischen und systemtheoretischen Bereich der Bildverarbeitung.

#### Inhaltliche Schwerpunkte bilden:

Gewinnung digitaler Bildsignale,  
Bildsensoren Detektoren vom Röntgen bis FIR-Spektralbereich, elektronische und optische Systemkomponenten der Bildverarbeitung

Konzepte von Abbildungs- und Beleuchtungssystemen  
Methoden der Bildsignalverarbeitung sowie der Systemtheorie  
Applikationen (Robotik, Qualitätssicherung, Prüftechnik, Mensch-Maschine Kommunikation)  
Aufbau und Auslegung von Bildverarbeitungssystemen in industriellen Anwendungen;

Die Vorlesung wird durch Praktikumsversuche unterstützt und gibt den Studierenden die Möglichkeit einer praktischen Erprobung der vermittelten Inhalte.

#### Medienformen

Tafel, Beamer, Vorlesungsscript ppt-Datei "Systemtechnik und Systemtheorie der Bildverarbeitung",  
Versuchsanleitungen im Internet

#### Literatur

Pedrotti u.a.: Optik für Ingenieure, Springer Verlag, 2008  
R.Dohlus "Lichtquellen", De Gruyter 2015

R.D. Fiete "Modelling the Imaging Chain of Digital Cameras", SPIE Press (2010)  
Th. Luhmann "Nahbereichsphotogrammetrie" 4.Auflage 2019

M. Sackewitz (Hrsg.), Handbuch zur Industriellen Bildverarbeitung (2017) Fraunhofer IRB Verlag  
B. Jähne "Digitale Bildverarbeitung", Springer Verlag 2012  
J. Beyerer, F.P. Leon, Ch. Frese.: Automatische Sichtprüfung, Springer Vieweg 2012

#### Detailangaben zum Abschluss

**Das Modul Systemtechnik und Systemtheorie der Bildverarbeitung mit der Prüfungsnummer 230482 schließt mit folgenden Leistungen ab:**

- mündliche Prüfungsleistung über 30 Minuten mit einer Wichtung von 100% (Prüfungsnummer: 2300674)
- Studienleistung mit einer Wichtung von 0% (Prüfungsnummer: 2300675)

Details zum Abschluss Teilleistung 2:  
Praktika gemäß Testatkarte in der Vorlesungszeit

#### verwendet in folgenden Studiengängen:

Master Informatik 2021  
Master Optische Systemtechnik/Optronik 2017

## Modul: Graphen & Algorithmen

Modulabschluss: Prüfungsleistung mündlich 30 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Wintersemester

Modulnummer: 200408 Prüfungsnummer: 2400760

Modulverantwortlich: Prof. Dr. Matthias Kriesell

Leistungspunkte: 5 Workload (h): 150 Anteil Selbststudium (h): 116 SWS: 3.0  
 Fakultät für Mathematik und Naturwissenschaften Fachgebiet: 2411

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studierenden kennen nach der Vorlesung typische Berechnungsprobleme und Algorithmen zu deren Lösung, wissen diese zu beschreiben. Sie haben dadurch auch grundlegende Kenntnisse der Theorie endlicher Graphen. Nach den Übungen sind sie fähig, die o. g. Kenntnisse zur Lösung einfacher anwendungsnaher Probleme einzusetzen, andererseits können sie die in der Vorlesung verwendeten Beweistechniken anwenden.

### Vorkenntnisse

Lineare Algebra I und Lineare Algebra II

### Inhalt

I. Bäume (Breiten- und Tiefensuchbäume, Matroidmethoden, Approximation optimaler Rundreisen, Baumweite, der Satz von Courcelle) II. Matchings (bipartiter Fall, allgemeine Faktorsätze) III. Flüsse (die Sätze von Ford-Fulkerson, Menger, Gutnikov) IV. Färbungen (Greedy-Färbung, die Sätze von Brooks und Vizing, Komplexität von Färbungsproblemen, der 4-Farben-Satz)

Die Inhalte können abhängig vom Dozenten variieren: Das Grundthema läßt hier ungeheuer viel Spielraum.

### Medienformen

Tafel

### Literatur

Die einschlägigen Lehrbücher von Diestel und Bondy-Murty.

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Graphentheorie

Modulabschluss: Prüfungsleistung mündlich 30 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Wintersemester

Modulnummer: 200431 Prüfungsnummer: 2400783

Modulverantwortlich: Prof. Dr. Matthias Kriesell

Leistungspunkte: 10 Workload (h): 300 Anteil Selbststudium (h): 232 SWS: 6.0  
 Fakultät für Mathematik und Naturwissenschaften Fachgebiet: 2411

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studierenden haben vertiefte Kenntnisse der Graphentheorie. Sie sind nach der Vorlesung befähigt, aktuelle Forschungspaper auf dem Gebiet der Graphentheorie zu verstehen. Nach den Übungen können sie vornehmlich von der Fähigkeit zur Anwendung typischer graphentheoretischer Beweistechniken Gebrauch machen.

### Vorkenntnisse

Graphen & Algorithmen

### Inhalt

Ausgewählte Kapitel der Graphentheorie, in denen zugleich zahlreiche aktuelle Forschungsfragen aufgehen, die genügend Potential für Themen qualifizierender Arbeiten geben:

- I. Packungs- und Überdeckungssätze II. Minoren: Wohlquasiordnung von Wurzelbäumen III. Kritischer und minimaler mehrfacher Zusammenhang IV. Minoren: Der Satz von Kuratowski V. Der Satz von Turán VI. Gradfolgen VII. Minoren: Hadwigers Vermutung VIII. Flußvermutungen.

Auch hier variiert der Inhalt je nach Dozent infolge der großen Breite des Grundthemas. Graphentheorie ist ein seit vielen Jahrzehnten sehr stark in Ilmenau vertretenes Forschungsgebiet.

### Medienformen

Tafel

### Literatur

Die einschlägigen Lehrbücher von Diestel und Bondy-Murty.

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Kombinatorik

Modulabschluss: Prüfungsleistung mündlich 30 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch      Pflichtkennz.: Wahlmodul      Turnus: Wintersemester

Modulnummer: 200432      Prüfungsnummer: 2400784

Modulverantwortlich: Prof. Dr. Matthias Kriesell

Leistungspunkte: 10      Workload (h): 300      Anteil Selbststudium (h): 232      SWS: 6.0  
 Fakultät für Mathematik und Naturwissenschaften      Fachgebiet: 2411

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Nach der Vorlesung kennen die Studierenden die grundlegenden Begriffe und Methoden der Kombinatorik. Nach den Übungen können sie einerseits die o. g. Begriffe und Methoden anhand typischer Beispiele anwenden, andererseits können die Studierenden die aus der Vorlesung bekannten Beweistechniken in verwandten Fällen anwenden.

### Vorkenntnisse

Lineare Algebra I/II sowie Graphen & Algorithmen

### Inhalt

Ausgewählte Kapitel der Kombinatorik: Kombinatorik von Mengensystemen (darunter Designs und projektive Geometrien), lateinische Quadrate, Schemata, Polyá-Theorie.

### Medienformen

Tafel

### Literatur

Die einschlägigen Lehrbücher von Aigner, van Lint-Wilson, et al.

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Kombinatorische Optimierung

Modulabschluss: Prüfungsleistung mündlich 30 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch      Pflichtkennz.: Wahlmodul      Turnus: Wintersemester

Modulnummer: 200435      Prüfungsnummer: 2400787

Modulverantwortlich: Prof. Dr. Matthias Kriesell

Leistungspunkte: 10      Workload (h): 300      Anteil Selbststudium (h): 232      SWS: 6.0  
 Fakultät für Mathematik und Naturwissenschaften      Fachgebiet: 2411

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Nach der Vorlesung können die Studierenden typische Probleme der Kombinatorischen Optimierung und Verfahren zu deren Lösung benennen, wissen dieses zusammenzufassen. Sie sind anhand der in der Vorlesung vorgestellten Beispiele für Fragen der Berechnungskomplexität sensibilisiert und sind fähig, typische Probleme hinsichtlich ihrer Berechnungskomplexität einzuordnen. Nach den Übungen können die Studierenden einerseits die o. g. Kenntnisse und Methoden zur Lösung von Beispielaufgaben anwenden, andererseits können sie kombinatorische Sachverhalte beweisen.

### Vorkenntnisse

Lineare Algebra I/II sowie Graphen & Algorithmen

### Inhalt

Grundlagen der Kombinatorischen Optimierung. Ausgewählte Probleme.

### Medienformen

Tafel

### Literatur

Das Lehrbuch von Korte-Vygen

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021



## Modul: Large Networks & Random Graphs

Modulabschluss: Prüfungsleistung mündlich 30 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch/Englisch      Pflichtkennz.: Wahlmodul      Turnus: Wintersemester

Modulnummer: 200439      Prüfungsnummer: 2400791

Modulverantwortlich: Prof. Dr. Yury Person

Leistungspunkte: 5      Workload (h): 150      Anteil Selbststudium (h): 116      SWS: 3.0  
 Fakultät für Mathematik und Naturwissenschaften      Fachgebiet: 241D

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studierenden kennen nach der Vorlesung verschiedene Modelle zufälliger Graphen, deren Anwendungsmöglichkeiten sowie Vor- und Nachteile. Sie sind u.a. durch die Übungen in der Lage, für ein Anwendungsproblem ein passendes Modell auszuwählen, dieses methodisch zu untersuchen und Algorithmen darauf anzuwenden und zu entwickeln. Sie sind ebenso befähigt, aktuelle Forschungsarbeiten zu lesen und die Ergebnisse zu präsentieren, zu besprechen und zu reflektieren.

### Vorkenntnisse

Analysis I&II, Diskrete Stochastik

### Inhalt

Modelle zufälliger Graphen und deren wichtigste Eigenschaften wie Schwellenwertfunktionen, Algorithmen,...

### Medienformen

Tafel, Aufgaben

### Literatur

- B. Bollobás: Random Graphs, 2nd edition; Cambridge University Press, 2001
- A. Frieze, M. Karonski: Introduction to Random Graphs; Cambridge University Press, 2015
- S. Janson, T. Luczak, A. Rucinski: Random Graphs; Wiley, 2000.

Aktuelle Forschungspublikationen.

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Angewandte Optimierung

Modulabschluss: Prüfungsleistung mündlich 30 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch      Pflichtkennz.: Wahlmodul      Turnus: Sommersemester

Modulnummer: 200400      Prüfungsnummer: 2400751

Modulverantwortlich: Prof. Dr. Gabriele Eichfelder

Leistungspunkte: 5      Workload (h): 150      Anteil Selbststudium (h): 105      SWS: 4.0  
 Fakultät für Mathematik und Naturwissenschaften      Fachgebiet: 2415

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studierenden erkennen durch Vorlesung und Übung, dass elementare Ergebnisse aus der Linearen Algebra und der Analysis im Bereich der Optimierung Anwendung finden. Darüber hinaus sind sie in die Lage versetzt, die grundlegenden Ideen und Herangehensweisen in der linearen und nichtlinearen Optimierung darzustellen, zu klassifizieren, zu vergleichen und zu erklären. Anhand dieser Erkenntnisse gelingt es den Studierenden in den Übungen weitere theoretische Resultate abzuleiten und vorgegebene konkrete Optimierungsprobleme sowie Probleme aus Anwendungsfragestellungen mathematisch zu modellieren, zu bearbeiten und unter Zuhilfenahme von Software zu lösen. Dabei können sie verschiedene Lösungsstrategien erkennen, analysieren, vergleichen und die erhaltenen Ergebnisse bewerten. Insgesamt sind die Studierenden aufgrund der erworbenen Kompetenzen in die Lage versetzt, in ihrer weiteren akademischen oder beruflichen Laufbahn Optimierungsprobleme zu erkennen, zu modellieren und dies fachfremden Kollegen zu erklären bzw. zu motivieren, sowie schließlich Lösungsstrategien für diese Optimierungsprobleme, gegebenenfalls auch im Team zusammen mit anderen Spezialisten, zu konzipieren und umzusetzen. Sie können dabei Anmerkungen beachten und Kritik würdigen.

### Vorkenntnisse

Lineare Algebra und Grundlagen der Analysis

### Inhalt

Anwendungsprobleme und Modellierung, konvexe Mengen, konvexe Funktionen, Lösungsverhalten linearer Ungleichungssysteme, Dualität, Optimalitätskriterien der linearen Optimierung, Lösungsverfahren, Optimalitätsbedingungen der nichtlinearen Optimierung, Überblick zu Verfahren der restriktionsfreien nichtlinearen Optimierung und Ansätze zu Verfahren der restringierten nichtlinearen Optimierung

### Medienformen

Tafel, Folien, Beamer

### Literatur

- A. Ben-Tal und A. Nemirovski, Lectures on modern convex optimization (MPS-SIAM Series on Optimization, 2001).
- M. Gerdt und F. Lempio, Mathematische Optimierungsverfahren des Operations Research (De Gruyter, Berlin, 2011).
- C. Geiger und C. Kanzow, Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben (Springer, Berlin, 1999).
- C. Geiger und C. Kanzow, Theorie und Numerik restringierter Optimierungsaufgaben (Springer, Berlin, 2002).
- F. Jarre und J. Stoer, Optimierung (Springer, Berlin, 2004).
- R. Reemtsen, Lineare Optimierung (Shaker Verlag, Aachen, 2001).

### Detailangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Numerik 1

Modulabschluss: Prüfungsleistung mündlich 30 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch Pflichtkennz.: Wahlmodul Turnus: Sommersemester

Modulnummer: 200404 Prüfungsnummer: 2400755

Modulverantwortlich: Prof. Dr. Karl Worthmann

Leistungspunkte: 10	Workload (h): 300	Anteil Selbststudium (h): 210	SWS: 8.0																		
Fakultät für Mathematik und Naturwissenschaften			Fachgebiet: 2413																		
SWS nach	1.FS	2.FS	3.FS	4.FS	5.FS	6.FS	7.FS	8.FS	9.FS	10.FS											
Fach-	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P
semester																					

### Lernergebnisse / Kompetenzen

Nach der Vorlesung beherrschen die Studierenden die grundlegenden Begriffe / Konzepte, Resultate und Beweismethoden der numerischen Mathematik; Sie sind fähig, die allgemeinen Resultate auf Spezialfälle anzuwenden; sie können die numerischen Verfahren nach der Übung auf konkrete (Anwendungs-) Beispiele, zum (Groß-) Teil mit Rechnerunterstützung, anwenden.

### Vorkenntnisse

### Inhalt

Numerische Grundkonzepte wie Kondition, Lineare Gleichungssysteme: Kondition, direkte und indirekte Verfahren, Gradientenverfahren, Eigenwertprobleme, Interpolation: Lagrange-Polynome, (kubische) Splines, Numerische Integration, Extrapolation, nichtlineare Gleichungssysteme, lineare und nichtlineare Ausgleichsrechnung, gewöhnliche Differentialgleichungen: Einschrittverfahren, lineare Mehrschrittverfahren, Stabilität.

### Medienformen

Tafel, Arbeitsblätter

### Literatur

S. Kurz, M. Stoll und K. Worthmann: Angewandte Mathematik - Ein Lehrbuch für Lehramtsstudierende, Springer: Lehrbuch, 2018.  
 Andreas Meister und Thomas Sonar: Numerik - Eine lebendige und gut verständliche Einführung mit vielen Beispielen, Springer: Lehrbuch, 2019.  
 Claus-Dieter Munz und Thomas Westermann: Numerische Behandlung gewöhnlicher und partieller Differentialgleichungen - Ein anwendungsorientiertes Lehrbuch für Ingenieure, Springer, 4. Auflage, 2019.  
 Karl Strehmel, Helmut Podhaisky und Rüdiger Weiner: Numerik gewöhnlicher Differentialgleichungen - Nichtsteife, steife und differential-algebraische Gleichungen, Springer Spektrum: Studium, 2. Auflage, 2012.  
 J.A. Trangenstein: Scientific Computing - Vol. I - Linear and Nonlinear Equations, Springer: Texts in computational science and engineering 18, 2017.  
 J.A. Trangenstein: Scientific Computing - Vol. II - Eigenvalues and Optimization, Springer: Texts in computational science and engineering 19, 2017.  
 J.A. Trangenstein: Scientific Computing - Vol. III - Approximation and Integration, Springer: Texts in computational science and engineering 20, 2017.

### Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Modul: Ordnungs- und Verbandstheorie

Modulabschluss: Prüfungsleistung mündlich 30 min      Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch      Pflichtkennz.: Wahlmodul      Turnus: Sommersemester

Modulnummer: 200441      Prüfungsnummer: 2400793

Modulverantwortlich: Prof. Dr. Matthias Kriesell

Leistungspunkte: 5      Workload (h): 150      Anteil Selbststudium (h): 116      SWS: 3.0  
 Fakultät für Mathematik und Naturwissenschaften      Fachgebiet: 2411

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studierenden haben nach der Vorlesung Grundkenntnisse der Ordnungs- und Verbandstheorie. Sie sind mit typischen Beweismethoden der Ordnungstheorie vertraut. Nach den Übungen sind die Studierenden in der Lage, die in der Vorlesung erlernten Begriffe und Methoden auf konkrete Beispiele anzuwenden.

### Vorkenntnisse

Lineare Algebra I/II, Algebra

### Inhalt

Ordnungen, Quasiordnungen, totale Ordnungen. Ordinal- und Kardinalzahlen, Supremums- und Infimumshalbverbände, Boolesche Verbände, Hüllenoperatoren, Topologien vs. Ordnungen.

### Medienformen

Tafel

### Literatur

Einschlägige Lehrbücher z. Bsp. von Erné.

### Detaillangaben zum Abschluss

### verwendet in folgenden Studiengängen:

Master Informatik 2021

## Lehrveranstaltung aus dem aktuellen Katalog

Fachabschluss: Prüfungsleistung Art der Notengebung: Gestufte Noten  
 Sprache: Pflichtkennz.:Wahlmodul Turnus:ganzjährig

Fachnummer: 0000 Prüfungsnummer:90401

Fachverantwortlich:

Leistungspunkte: 5 Workload (h):150 Anteil Selbststudium (h):150 SWS:0.0  
 Fakultät für Informatik und Automatisierung Fachgebiet:22

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

Lernergebnisse / Kompetenzen

Vorkenntnisse

Inhalt

Medienformen

Literatur

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

- Bachelor Medienwirtschaft 2015
- Master Wirtschaftsingenieurwesen 2014 Vertiefung BT
- Master Elektrotechnik und Informationstechnik 2014 Vertiefung MNE
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2008 Vertiefung
- Bachelor Informatik 2010
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2008
- Master Medien- und Kommunikationswissenschaft/Media and Communication Science 2011
- Master Allgemeine Betriebswirtschaftslehre 2013
- Bachelor Wirtschaftsingenieurwesen 2013 Vertiefung MB
- Master Wirtschaftsingenieurwesen 2015 Vertiefung MB
- Bachelor Angewandte Medien- und Kommunikationswissenschaft 2012
- Master Informatik 2021
- Bachelor Mathematik 2013
- Bachelor Wirtschaftsingenieurwesen 2013 Vertiefung ET
- Bachelor Angewandte Medien- und Kommunikationswissenschaft 2013
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013
- Master Wirtschaftsingenieurwesen 2011
- Bachelor Fahrzeugtechnik 2021
- Bachelor Informatik 2021
- Bachelor Technische Kybernetik und Systemtheorie 2013
- Master Ingenieurinformatik 2014
- Bachelor Mathematik 2009
- Master Medientechnologie 2013
- Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2008 Vertiefung

Master Biotechnische Chemie 2020  
Master Medienwirtschaft 2018  
Master Elektrotechnik und Informationstechnik 2014 Vertiefung AST  
Master Wirtschaftsingenieurwesen 2014 Vertiefung MB  
Master Regenerative Energietechnik 2013  
Master Technische Kybernetik und Systemtheorie 2014  
Master Technische Physik 2013  
Master Technische Physik 2011  
Master Elektrochemie und Galvanotechnik 2013  
Master Biomedizinische Technik 2014  
Bachelor Ingenieurinformatik 2013  
Bachelor Maschinenbau 2021  
Master Mathematik und Wirtschaftsmathematik 2008  
Master Research in Computer & Systems Engineering 2016  
Master Micro- and Nanotechnologies 2013  
Bachelor Biotechnische Chemie 2013  
Master Werkstoffwissenschaft 2013  
Bachelor Medienwirtschaft 2013  
Master Wirtschaftsinformatik 2018  
Master Wirtschaftsinformatik 2014  
Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung AM  
Bachelor Technische Physik 2013  
Master Wirtschaftsingenieurwesen 2009  
Bachelor Technische Kybernetik und Systemtheorie 2010  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2008 Vertiefung  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung  
Master Wirtschaftsingenieurwesen 2018 Vertiefung MB  
Master Optische Systemtechnik/Optronik 2017  
Master Wirtschaftsingenieurwesen 2014  
Master Medienwirtschaft 2013  
Master Maschinenbau 2009  
Master Micro- and Nanotechnologies 2016  
Diplom Elektrotechnik und Informationstechnik 2017  
Master Research in Computer & Systems Engineering 2012  
Bachelor Wirtschaftsingenieurwesen 2015 Vertiefung MB  
Bachelor Mechatronik 2021  
Master Medien- und Kommunikationswissenschaft/Media and Communication Science 2013  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013 Vertiefung  
Master Elektrotechnik und Informationstechnik 2014 Vertiefung IKT  
Bachelor Elektrotechnik und Informationstechnik 2008  
Master Elektrotechnik und Informationstechnik 2014 Vertiefung ATE  
Bachelor Wirtschaftsingenieurwesen 2015 Vertiefung ET  
Master Maschinenbau 2017  
Master Technische Physik 2008  
Bachelor Angewandte Medienwissenschaft 2011  
Bachelor Elektrotechnik und Informationstechnik 2013  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2013 Vertiefung  
Master Ingenieurinformatik 2009  
Master Wirtschaftsingenieurwesen 2018  
Master Medientechnologie 2017  
Master Medien- und Kommunikationswissenschaft/Media and Communication Science 2009  
Bachelor Angewandte Medienwissenschaft 2008  
Master Wirtschaftsingenieurwesen 2013  
Master Elektrotechnik und Informationstechnik 2014 Vertiefung EWT  
Bachelor Angewandte Medien- und Kommunikationswissenschaft 2014  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2008 Vertiefung  
Master Communications and Signal Processing 2013  
Bachelor Medientechnologie 2013  
Master Medienwirtschaft 2014  
Master Electrical Power and Control Engineering 2008  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Elektrotechnik 2008  
Master Elektrotechnik und Informationstechnik 2014 Vertiefung EET

Master Maschinenbau 2011  
Master Wirtschaftsingenieurwesen 2015 Vertiefung BT  
Master Fahrzeugtechnik 2009  
Master Wirtschaftsinformatik 2015  
Bachelor Optische Systemtechnik/Optronik 2013  
Bachelor Angewandte Medienwissenschaft 2009  
Bachelor Technische Physik 2011  
Master Wirtschaftsingenieurwesen 2018 Vertiefung BT  
Master Wirtschaftsingenieurwesen 2010  
Master Mathematik und Wirtschaftsmathematik 2013 Vertiefung WM  
Master Wirtschaftsingenieurwesen 2013 Vertiefung BT  
Master Wirtschaftsingenieurwesen 2015  
Master Medienwirtschaft 2015  
Master Electrical Power and Control Engineering 2013  
Master Informatik 2013  
Master Regenerative Energietechnik 2016  
Bachelor Polyvalenter Bachelor mit Lehramtsoption für berufsbildende Schulen - Metalltechnik 2013  
Master Wirtschaftsinformatik 2013  
Diplom Maschinenbau 2017

## Fachpraktikum

Fachabschluss: Studienleistung alternativ 20 Wochen      Art der Notengebung: Testat unbenotet  
Sprache:      Pflichtkennz.: Pflichtmodul      Turnus: unbekannt

Fachnummer: 200931      Prüfungsnummer: 90020

Fachverantwortlich:

Leistungspunkte: 30      Workload (h): 900      Anteil Selbststudium (h): 900      SWS: 0.0  
Fakultät für Informatik und Automatisierung      Fachgebiet: 22

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS		
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P

Lernergebnisse / Kompetenzen

Vorkenntnisse

Inhalt

Medienformen

Literatur

Detailangaben zum Abschluss

verwendet in folgenden Studiengängen:

Master Informatik 2021



## Modul: Masterarbeit

Modulnummer: 8222

Modulverantwortlich: Prof. Dr. Winfried Kühnhauser

Modulabschluss: Fachprüfung/Modulprüfung generiert

### Lernergebnisse

Vertiefung der bisher erworbenen Kompetenzen in einem individuellen Thema. Die Studierenden erwerben die Fähigkeit, eine komplexe Problemstellung zu beurteilen und unter Anwendung der bisher erworbenen Theorie- und Methodenkompetenzen weitgehend selbstständig zu bearbeiten, gemäß wissenschaftlicher Standards schriftlich zu dokumentieren und mündlich zu präsentieren. Die Studierenden erwerben Problemlösungskompetenz und lernen es, die eigene Arbeit zu bewerten und einzuordnen.

### Voraussetzungen für die Teilnahme

Erfüllung der Zulassungsvoraussetzungen

### Detailangaben zum Abschluss

Schriftliche Arbeit und Kolloquium

## Abschlusskolloquium zur Master-Arbeit

Fachabschluss: Prüfungsleistung mündlich 30 min Art der Notengebung: Gestufte Noten  
 Sprache: Deutsch und Englisch Pflichtkennz.: Pflichtmodul Turnus: ganzjährig

Fachnummer: 200929 Prüfungsnummer: 99002

Fachverantwortlich: Prof. Dr. Winfried Kühnhauser

Leistungspunkte: 6	Workload (h): 180	Anteil Selbststudium (h): 180	SWS: 0.0
Fakultät für Informatik und Automatisierung		Fachgebiet: 22	

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS					
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

### Lernergebnisse / Kompetenzen

Die Studierenden vertiefen in einem speziellen fachlichen Thema ihre bisher erworbenen Kompetenzen. Sie werden befähigt, eine komplexe und konkrete Problemstellung zu beurteilen und unter Anwendung der bisher erworbenen Theorie- und Methodenkompetenzen selbstständig zu bearbeiten. Das Thema ist gemäß wissenschaftlicher Standards zu dokumentieren und die Studierenden werden befähigt, entsprechende wissenschaftlich fundierte Texte zu verfassen. Die Studierenden erwerben Problemlösungskompetenz und lernen es, die eigene Arbeit zu bewerten und einzuordnen.

### Vorkenntnisse

Zulassung zur Masterarbeit durch den Prüfungsausschuss

### Inhalt

siehe Modulbeschreibung

### Medienformen

wissenschaftlicher Vortrag

### Literatur

Literatur wird mit Ausgabe des Themas bekannt gegeben oder ist selbstständig zu recherchieren.

### Detaillangaben zum Abschluss

Prüfungsform: mündlich

Dauer: 30 min

Abschluss: Prüfungsleistung

verwendet in folgenden Studiengängen:

Master Informatik 2021

## Masterarbeit

Fachabschluss: Masterarbeit schriftlich 6 Monate      Art der Notengebung: Generierte Note mit  
 Sprache:deutsch oder englisch      Pflichtkennz.:Pflichtmodul      Turnus:ganzjährig

Fachnummer: 200930      Prüfungsnummer:99001

Fachverantwortlich: Prof. Dr. Winfried Kühnhauser

Leistungspunkte: 24      Workload (h):720      Anteil Selbststudium (h):720      SWS:0.0  
 Fakultät für Informatik und Automatisierung      Fachgebiet:22

SWS nach Fach- semester	1.FS			2.FS			3.FS			4.FS			5.FS			6.FS			7.FS			8.FS			9.FS			10.FS								
	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P	V	S	P			

**Lernergebnisse / Kompetenzen**

Die Studierenden vertiefen in einem speziellen fachlichen Thema ihre bisher erworbenen Kompetenzen. Sie werden befähigt, eine komplexe und konkrete Problemstellung zu beurteilen und unter Anwendung der bisher erworbenen Theorie- und Methodenkompetenzen selbstständig zu bearbeiten. Das Thema ist gemäß wissenschaftlicher Standards zu dokumentieren und die Studierenden werden befähigt, entsprechende wissenschaftlich fundierte Texte zu verfassen. Die Studierenden erwerben Problemlösungskompetenz und lernen es, die eigene Arbeit zu bewerten und einzuordnen.

**Vorkenntnisse**

Zulassung zur Masterarbeit durch den Prüfungsausschuss

**Inhalt**

siehe Modulbeschreibung

**Medienformen**

wissenschaftlicher Vortrag

**Literatur**

Literatur wird mit Ausgabe des Themas bekannt gegeben oder ist selbstständig zu recherchieren.

**Detailangaben zum Abschluss**

Prüfungsform: schriftlich  
 Abschluss: Prüfungsleistung

**verwendet in folgenden Studiengängen:**

Master Informatik 2021



## **Glossar und Abkürzungsverzeichnis:**

LP	Leistungspunkte
SWS	Semesterwochenstunden
FS	Fachsemester
V S P	Angabe verteilt auf Vorlesungen, Seminare, Praktika
N.N.	Nomen nominandum, Platzhalter für eine noch unbekannte Person (wikipedia)
Objektypen lt. Inhaltsverzeichnis	K=Kompetenzfeld; M=Modul; P,L,U= Fach (Prüfung, Lehrveranstaltung, Unit)