



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Masterstudiengang

Medizintechnik

Modulhandbuch

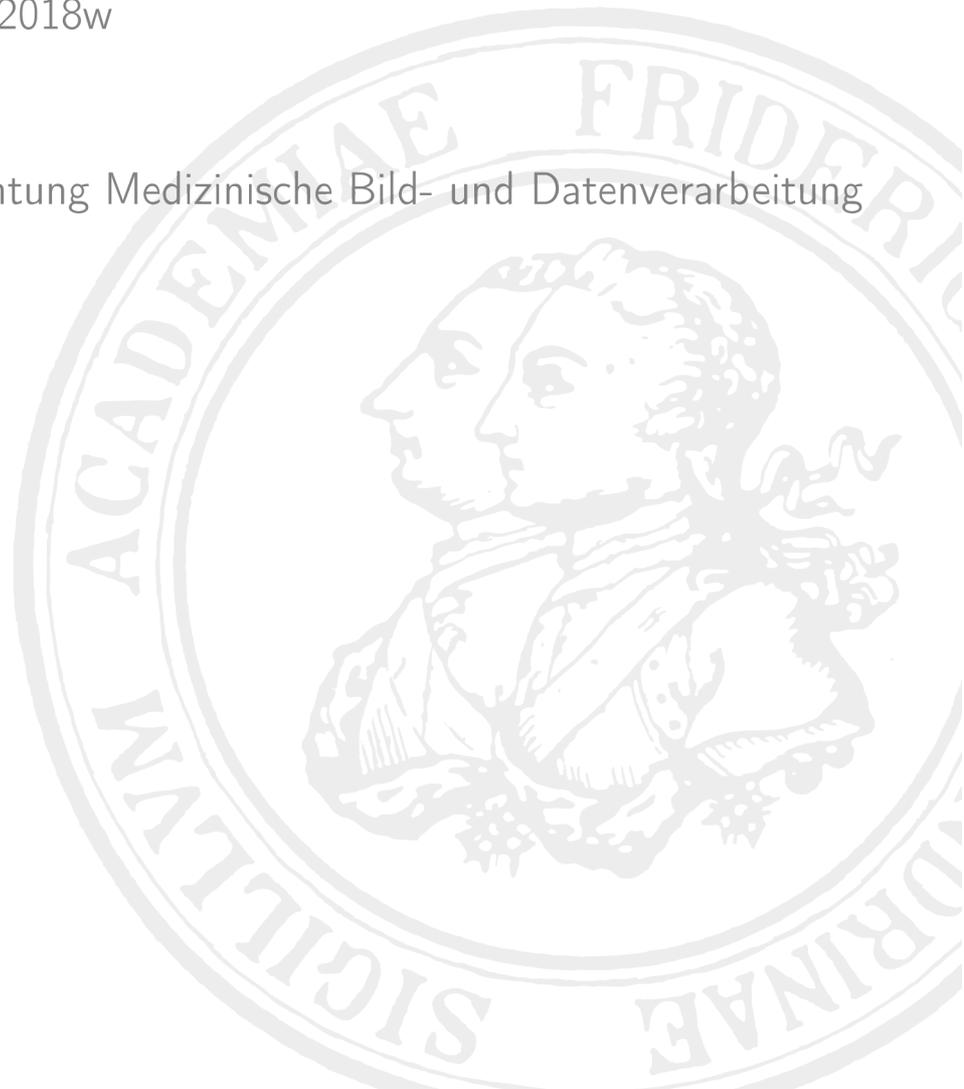
SS 2019

Prüfungsordnungsversion: 2018w

Teilauszug Abschnitt

Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung

Modulhandbuch generiert aus *UnivIS*
Stand: 29.08.2021 20:35



Medizintechnik (Master of Science)

SS 2019; Prüfungsordnungsversion: 2018w

M2 Ingenieurwissenschaftliche Kernmodule (BDV)

- Cyber-Physical Systems, 5 ECTS, Torsten Klie, SS 2019 5
- Transformationen in der Signalverarbeitung, 2.5 ECTS, Jürgen Seiler, SS 2019 7
- Digitale Übertragung, 5 ECTS, Robert Schober, SS 2019 9
- Künstliche Intelligenz II, 7.5 ECTS, Michael Kohlhase, SS 2019 11
- Konzeptionelle Modellierung, 5 ECTS, Richard Lenz, SS 2019 13
- Deep Learning, 5 ECTS, Andreas Maier, Tobias Würfl, Vincent Christlein, Lennart Hus- 15
vogt, SS 2019
- Pattern Analysis, 5 ECTS, Christian Riess, SS 2019 17
- Sprach- und Audiosignalverarbeitung, 5 ECTS, Walter Kellermann, SS 2019 20
- Grundlagen der Systemprogrammierung, 5 ECTS, Wolfgang Schröder-Preikschat, Jürgen 22
Kleinöder, SS 2019
- Applied Visualization, 5 ECTS, Roberto Grosso, SS 2019 23
- Kanalcodierung, 5 ECTS, Clemens Stierstorfer, SS 2019 25
- Systemnahe Programmierung in C, 5 ECTS, Volkmar Sieh, Jürgen Kleinöder, SS 2019 29
- Hardware-Software-Co-Design, 5 ECTS, Jürgen Teich, SS 2019 31
- Hardware-Software-Co-Design mit erweiterter Übung, 7.5 ECTS, Jürgen Teich, SS 2019 33
- Parallele Systeme, 5 ECTS, Jürgen Teich, Frank Hannig, SS 2019 35
- Signale und Systeme II, 5 ECTS, André Kaup, Christian Herglotz, Fabian Brand, SS 2019 38
- Heterogene Rechnerarchitekturen Online, 5 ECTS, Marc Reichenbach, Thomas Heller, 40
Johannes Hofmann, SS 2019
- Optimierung für Ingenieure, 5 ECTS, Johannes Hild, Martin Gugat, SS 2019 42
- Parallele Systeme mit erweiterter Übung, 7.5 ECTS, Jürgen Teich, Frank Hannig, SS 2019 44
- Information Theory and Coding, 5 ECTS, Ralf Müller, SS 2019 47
- Systemprogrammierung, 10 ECTS, Wolfgang Schröder-Preikschat, Jürgen Kleinöder, SS 50
2019, 2 Sem.
- Numerik II für Ingenieure, 5 ECTS, Wilhelm Merz, SS 2019 52

M3 Medizintechnische Kernmodule (BDV)

- Image and Video Compression, 5 ECTS, André Kaup, Daniela Lanz, SS 2019 53
- Wearable and Implantable Computing, 5 ECTS, Oliver Amft, und Mitarbeiter/innen, SS 55
2019

- Magnetic Resonance Imaging 2, 2.5 ECTS, Frederik Laun, Andreas Maier, Armin Nagel, 57

SS 2019

- Computer Architectures for Medical Applications, 5 ECTS, N.N., Gerhard Wellein, SS 59
2019

- Diagnostic Medical Image Processing (VHB-Kurs), 5 ECTS, Andreas Maier, SS 2019 60

- Interventional Medical Image Processing (Online-Kurs), 5 ECTS, Andreas Maier, SS 2019 62

- Magnetic Resonance Imaging 2 + Übung, 5 ECTS, Frederik Laun, Andreas Maier, Armin
Nagel, SS 2019 64

- Geometry Processing, 5 ECTS, Roberto Grosso, SS 2019 65

M5 Medizintechnische Vertiefungsmodule (BDV)

- Forensische Informatik, 5 ECTS, Felix Freiling, Christian Moch, Stefan Vömel, SS 2019 67

- Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (V+Ü), 5 ECTS, Peter Ulbrich, Peter
Wägemann, SS 2019 68

- Knowledge Discovery in Databases, 2.5 ECTS, Klaus Meyer-Wegener, SS 2019 70

- Informationssysteme in der Intensivmedizin, 5 ECTS, Stefan Kraus, Dennis Toddenroth,
Ixchel Castellanos, SS 2019 72

- Human Computer Interaction, 5 ECTS, Björn Eskofier, SS 2019 74

- Interaktive Computergraphik, 5 ECTS, Marc Stamminger, SS 2019 76

UnivIS: 29.08.2021 20:35

3

- Human Factors in Security and Privacy, 5 ECTS, Zinaida Benenson, SS 2019 78

- Security in Embedded Hardware, 5 ECTS, Jürgen Teich, SS 2019 80

- Organic Computing, 5 ECTS, Rolf Wanka, SS 2019 82

- Datenstromsysteme, 5 ECTS, Klaus Meyer-Wegener, SS 2019 84

- Verteilte Systeme - V+Ü, 5 ECTS, Tobias Distler, Jürgen Kleinöder, Michael Eischer, 87 Laura Lawniczak,
SS 2019

Modulbezeichnung:	Cyber-Physical Systems (CPS) (Cyber-Physical Systems)	5 ECTS
Modulverantwortliche/r:	Torsten Klie	
Lehrende:	Torsten Klie	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

- Cyber-Physical Systems (SS 2019, Vorlesung, 2 SWS, Torsten Klie)
- Übung zu Cyber-Physical Systems (SS 2019, Übung, 2 SWS, Torsten Klie)

Inhalt:

Klassische Computersysteme zeichnen sich durch eine strikte Trennung von realer und virtueller Welt aus. Moderne Steuerungssysteme, die z.B. in modernen Fahrzeugen verbaut sind und die aus einer Vielzahl von Sensoren und Aktoren bestehen, entsprechen diesem Bild nur sehr eingeschränkt. Diese Systeme, oft "Cyber-Physical Systems (CPS)" genannt, erkennen ihre physische Umgebung, verarbeiten diese Informationen und können die physische Umwelt auch koordiniert beeinflussen. Hierzu ist eine starke Kopplung von physischem Anwendungsmodell und dem Computer-Steuerungsmodell nötig. Im Unterschied zu Eingebetteten Systemen bestehen CPS meist aus vielen vernetzten Komponenten, die sich selbständig untereinander koordinieren. Diese Vorlesung spannt den Bogen von kontrolltheoretischen Grundlagen über Selbstorganisationsprinzipien bis hin zu visionären Anwendungen aus den Bereichen Verkehr und Medizintechnik. Ferner werden Entwurfsmethoden für Cyber-Physical Systems vorgestellt.

Lernziele und Kompetenzen:

Die Studierenden

- erläutern, was Cyber-Physical Systems sind und auf welchen technologischen Grundlagen sie aufbauen, insbesondere in den Bereichen Regelungstechnik, Ablaufplanung, Kommunikation und Selbstorganisation
- bewerten CPS in verschiedenen Anwendungsgebieten
- stellen den Entwurfsprozess von CPS dar, insbesondere die Modellierung und die grundlegende Programmierung
- entdecken wesentliche Herausforderungen beim Entwurf, Ausbringung und Einsatz von CPS

Literatur:

Otto Föllinger Regelungstechnik. Hüthig 1992.

Rolf Isermann Mechatronische Systeme. Springer 2008.

Hilmar Jaschek und Holger Voos Grundkurs der Regelungstechnik. Oldenbourg 2010.

Jörg Kahlert Crash-Kurs Regelungstechnik. VDE Verlag 2010.

Wolfgang Schneider Praktische Regelungstechnik. Vieweg+Teubner 2008.

Wayne Wolf Computers as Components: Principles of Embedded Computing System Design. Elsevier 2008

Dave Hakkens Projekt „Phonebloks“. 2013

Dustin Earley Motorola's Project Ara is the modular phone of the future. Android and Me 2013

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of

Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Cyber-Physical Systems (Prüfungsnummer: 451696)

(englische Bezeichnung: Cyber-Physical Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil
an der Berechnung der Modulnote: 0%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Torsten Klie

Modulbezeichnung: Transformationen in der Signalverarbeitung (TSV) 2.5 ECTS
 (Transforms in Signal Processing)

Modulverantwortliche/r: Jürgen Seiler

Lehrende: Jürgen Seiler

Startsemester: SS 2019 Dauer: 1 Semester Turnus: jährlich (SS)

Präsenzzeit: 30 Std. Eigenstudium: 45 Std. Sprache: Englisch

Lehrveranstaltungen:

Transformationen in der Signalverarbeitung (SS 2019, Vorlesung, 2 SWS, Jürgen Seiler)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird:

Signale und Systeme I

Signale und Systeme II

Inhalt:

Die Lehrveranstaltung "Transformationen in der Signalverarbeitung" behandelt mehrere verschiedene Transformationen, die im Rahmen der Signalverarbeitung Verwendung finden. Dabei werden zuerst die grundlegenden Konzepte von Transformationen diskutiert und die Vorteile die Transformationen mit sich bringen erläutert. Im Anschluss daran werden die grundlegenden Eigenschaften von Integraltransformationen betrachtet und die Laplace- und die Fourier-Transformation im Detail untersucht. Um auch zeitlich veränderliche Signale gut transformieren zu können werden danach die Kurzzeit-Fourier-Transformation und die Gabor-Transformation eingeführt. Im Anschluss daran erfolgt eine Betrachtung der Auswirkung der Abtastung auf transformierte Signale, bevor die z-Transformation als Transformation für diskrete Signale behandelt wird. Abschließend erfolgt die Betrachtung weiterer Transformationen für diskrete Signale wie der Diskreten Fourier-Transformation oder linearer Block-Transformationen. The lecture "Transforms in Signal Processing" covers several different transforms which are used in the field of signal processing. For this, first the basic concepts of transforms are discussed and the advantages which are offered by the different transforms are presented. Subsequent to this, fundamental properties of integral transforms are considered and the Laplace- and the Fourier-Transform are examined in detail. To be able to transform time-varying signals, the Short-Time Fourier-Transform and the Gabor-Transform are introduced, afterwards. Subsequent to this, the impact of sampling on transformed signals is analyzed before the z-Transform as a transform for discrete signals is covered. Finally, further transforms for discrete signals like the Discrete Fourier-Transform or Linear-Block Transforms are discussed.

Lernziele und Kompetenzen:

Die Studierenden können nach Besuch der Vorlesung

- Anwendungsmöglichkeiten von Transformationen bestimmen
- Integraltransformationen gegenüberstellen und untersuchen
- die Existenz von Transformationen hinterfragen
- die Eindeutigkeit von Transformationen überprüfen
- Sätze und Eigenschaften von Transformationen entwickeln
- zu Transformationen zugehörige inverse Transformationen einschätzen
- die Zusammenhänge zwischen verschiedenen Transformationen einschätzen
- auf Zusammenhänge zwischen Ausgangssignalen und transformierten Signalen folgern
- Symmetriebeziehungen von Transformationen ausarbeiten
- Zusammenhänge zwischen kontinuierlichen und diskreten Signalen ausarbeiten

Educational Objectives and Competences: After attending the lecture, students will be able to

- determine applications of transforms
- contrast and examine integral transforms

- question the existence of transforms
- evaluate the uniqueness of transforms
- develop theorems and properties of transforms
- evaluate to transforms corresponding inverse transforms
- evaluate the relationships between different transforms
- asses the relationship between original signal and transformed signals
- devise the symmetry properties of transforms
- devise the relationship between continuous and discrete signals Literatur:

K. Krüger, Transformationen - Grundlagen und Anwendungen in der Nachrichtentechnik, Vieweg Verlag, Braunschweig

B. Girod, R. Rabenstein, A. Stenger, Einführung in die Systemtheorie, B. G. Teubner Verlag, Stuttgart

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Transformationen in der Signalverarbeitung_ (Prüfungsnummer: 498723)

(englische Bezeichnung: Transforms in Signal Processing_)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Seiler

Organisatorisches:

Signale und Systeme I und II, Digitale Signalverarbeitung

Modulbezeichnung:	Digitale Übertragung (DÜ) (Digital Communications)	5 ECTS
-------------------	---	--------

Modulverantwortliche/r:	Robert Schober, Laura Cottatellucci
-------------------------	-------------------------------------

Lehrende:	Robert Schober
-----------	----------------

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
------------------------	-------------------	-----------------------

Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
----------------------	-----------------------	------------------

Lehrveranstaltungen:

Digitale Übertragung (SS 2019, Vorlesung, 3 SWS, Robert Schober)

Übungen zur Digitalen Übertragung (SS 2019, Übung, 1 SWS, Wayan Wicke)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird:

Nachrichtentechnische Systeme

Inhalt:

Alle modernen Kommunikationssysteme basieren auf digitalen Übertragungsverfahren. Diese Vorlesung befasst sich mit den Grundlagen der Analyse und des Entwurfs digitaler Sender und Empfänger. Dabei wird zunächst von einem einfachen Kanalmodell bei dem das Empfangssignal nur durch additives weißes Gaußsches Rauschen gestört wird ausgegangen. Im Verlauf der Vorlesung werden aber auch Kanäle mit unbekannter Phase sowie verzerrende Kanäle betrachtet. Behandelt werden unter anderem digitale Modulationsverfahren (z.B. Pulsamplitudenmodulation (PAM), digitale Frequenzmodulation (FSK), und Kontinuierliche-Phasenmodulation (CPM)), Orthogonalkonstellationen, das Nyquistkriterium in Zeit- und Frequenzbereich, optimale kohärente und inkohärente Detektions- und Decodierungsverfahren, die Signalraumdarstellung digital modulierter Signale, verschiedene Entzerrungsverfahren, und Mehrträger-Übertragungsverfahren.

Lernziele und Kompetenzen:

Die Studierenden

- analysieren und klassifizieren digitale Modulationsverfahren hinsichtlich ihrer Leistungs- und Bandbreiteneffizienz sowie ihres Spitzenwertfaktors,
 - ermitteln notwendige Kriterien für impulsinterferenzfreie Übertragung,
 - charakterisieren digitale Modulationsverfahren im Signalraum,
 - ermitteln informationsverlustfreie Demodulationsverfahren,
 - entwerfen optimale kohärente und inkohärente Detektions- und Decodierungsverfahren,
 - vergleichen verschiedene Entzerrungsverfahren hinsichtlich deren Leistungsfähigkeit und Komplexität,
 - entwerfen einfache digitale Übertragungssysteme mit vorgeschriebenen Leistungs- und Bandbreiteneffizienzen sowie Spitzenwertfaktoren.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Digitale Übertragung (Prüfungsnummer: 35101)

(englische Bezeichnung: Digital Communications)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Robert Schober

Modulbezeichnung:	Künstliche Intelligenz II (KI II) (Artificial Intelligence II)	7.5 ECTS
Modulverantwortliche/r:	Michael Kohlhasse	
Lehrende:	Michael Kohlhasse	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

- Künstliche Intelligenz II (SS 2019, Vorlesung, 4 SWS, Michael Kohlhasse)
- Übungen zu Künstliche Intelligenz II (SS 2019, Übung, 2 SWS, Dennis Müller)

Inhalt:

Dieser Kurs beschäftigt sich mit den Grundlagen der Künstlichen Intelligenz (KI), insbesondere mit Techniken des Schliessens unter Unsicherheit, des maschinellen Lernens und dem Sprachverstehen. Der Kurs baut auf der Vorlesung Künstliche Intelligenz I vom Wintersemester auf und führt diese weiter.

Lernziele und Kompetenzen:

Fach- Lern- bzw. Methodenkompetenz

- Wissen: Die Studierenden lernen grundlegende Repräsentationsformalismen und Algorithmen der Künstlichen Intelligenz kennen.
- Anwenden: Die Konzepte werden an Beispielen aus der realen Welt angewandt (Übungsaufgaben).
- Analyse: Die Studierenden lernen über die Modellierung in der Maschine menschliche Intelligenzleistungen besser einzuschätzen. Sozialkompetenz
- Die Studierenden arbeiten in Kleingruppen zusammen um kleine Projekte zu bewältigen.

This course covers the foundations of Artificial Intelligence (AI), in particular reasoning under uncertainty, machine learning and (if there is time) natural language understanding. This course builds on the course Artificial Intelligence I from the preceding winter semester and continues it. Learning Goals and Competencies Technical, Learning, and Method Competencies

- Knowledge: The students learn foundational representations and algorithms in AI.
- Application: The concepts learned are applied to examples from the real world (homeworks).
- Analysis: By modeling human cognitive abilities, students learn to assess and understand human intelligence better.
- Social Competencies: Students work in small groups to solve the and machine learning challenge/competition.

Literatur:

Die Vorlesung folgt weitgehend dem Buch Stuart Russell und Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.

Deutsche Ausgabe: Stuart Russell und Peter Norvig: Künstliche Intelligenz: Ein Moderner Ansatz. Pearson-Studium, 2004 (Übersetzung der 2. Auflage).

ISBN: 978-3-8273-7089-1.

Literature The course follows the following textbook: Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Künstliche Intelligenz II (Prüfungsnummer: 532733)

(englische Bezeichnung: Artificial Intelligence II)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Prüfung besteht aus einer Klausur (90 Minuten). Auf Basis der Bewertungen der abgegebenen Übungsaufgaben können bis zu 10 % Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden. Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer) 1.

Prüfer: Michael Kohlhasse

Modulbezeichnung:	Konzeptionelle Modellierung (KonzMod) (Conceptual Modeling)	5 ECTS
Modulverantwortliche/r:	Richard Lenz	
Lehrende:	Richard Lenz	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (WS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

- Konzeptionelle Modellierung (SS 2019, Vorlesung, 2 SWS, Richard Lenz)
- Übungen zu Konzeptionelle Modellierung (SS 2019, Übung, 2 SWS, David Haller)

Inhalt:

- Grundlagen der Modellierung
- Datenmodellierung am Beispiel Entity-Relationship-Modell
- Modellierung objektorientierter Systeme am Beispiel UML
- Relationale Datenmodellierung und Anfragemöglichkeiten
- Grundlagen der Metamodellierung
- XML
- Multidimensionale Datenmodellierung
- Domänenmodellierung und Ontologien

Lernziele und Kompetenzen: Die

Studierenden:

- definieren grundlegende Begriffe aus der Datenbankfachliteratur
- erklären die Vorteile von Datenbanksystemen
- erklären die verschiedenen Phasen des Datenbankentwurfs
- benutzen das Entity-Relationship Modell und das erweiterte Entity-Relationship Modell zur semantischen Datenmodellierung
- unterscheiden verschiedene Notationen für ER-Diagramme
- erläutern die grundlegenden Konzepte des relationalen Datenmodells
- bilden ein gegebenes EER-Diagramm auf ein relationales Datenbankschema ab
- erklären die Normalformen 1NF, 2NF, 3NF, BCNF und 4NF
- definieren die Operationen der Relationenalgebra
- erstellen Datenbanktabellen mit Hilfe von SQL
- lösen Aufgaben zur Datenselektion und Datenmanipulation mit Hilfe von SQL
- erklären die grundlegenden Konzepte der XML
- erstellen DTDs für XML-Dokumente
- benutzen XPATH zur Formulierung von Anfragen an XML-Dokumente
- definieren die grundlegenden Strukturelemente und Operatoren des multidimensionalen Datenmodells
- erklären Star- und Snowflake-Schema
- benutzen einfache UML Use-Case Diagramme
- benutzen einfache UML-Aktivitätsdiagramme
- erstellen UML-Sequenzdiagramme
- erstellen einfache UML-Klassendiagramme
- erklären den Begriff Meta-Modellierung
- definieren den Begriff der Ontologie in der Informatik
- definieren die Begriffe RDF und OWL

Literatur:

- Alfons Kemper, Andre Eickler: Datenbanksysteme : Eine Einführung. 6., aktualis. u. erw. Aufl. Oldenbourg, März 2006. - ISBN-10: 3486576909

- Bernd Oestereich: Analyse und Design mit UML 2.1. 8. Aufl. Oldenbourg, Januar 2006. - ISBN-10: 3486579266
 - Ian Sommerville: Software Engineering. 8., aktualis. Aufl. Pearson Studium, Mai 2007. - ISBN-10: 3827372577
 - Horst A. Neumann: Objektorientierte Softwareentwicklung mit der Unified Modeling Language. (UML). Hanser Fachbuch, März 2002. - ISBN-10: 3446188797
 - Rainer Eckstein, Silke Eckstein: XML und Datenmodellierung. Dpunkt Verlag, November 2003. ISBN-10: 3898642224
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "079#72#H", "079#74#H", "Berufspädagogik Technik (Bachelor of Science)", "Digitale Geistes- und Sozialwissenschaften (Bachelor of Arts (2 Fächer))", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (1. Staatsprüfung für das Lehramt an Realschulen)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Physische Geographie (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Konzeptionelle Modellierung (Klausur) (Prüfungsnummer: 31301)

(englische Bezeichnung: Written examination in conceptual modelling)

Prüfungsleistung, Klausur mit MultipleChoice, Dauer (in Minuten): 90 Anteil
an der Berechnung der Modulnote: 100%

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Richard Lenz

Modulbezeichnung:	Deep Learning (DL) (Deep Learning)	5 ECTS
-------------------	---------------------------------------	--------

Modulverantwortliche/r:	Andreas Maier
-------------------------	---------------

Lehrende:	Tobias Würfl, Lennart Husvogt, Vincent Christlein, Andreas Maier
-----------	--

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
------------------------	-------------------	------------------------------

Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch
----------------------	-----------------------	-------------------

Lehrveranstaltungen:

Deep Learning (SS 2019, Vorlesung, 2 SWS, Andreas Maier et al.)

Deep Learning Exercises (SS 2019, Übung, 2 SWS, Hendrik Schröter et al.)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Introduction to Pattern Recognition

Inhalt:

Deep Learning (DL) has attracted much interest in a wide range of applications such as image recognition, speech recognition and artificial intelligence, both from academia and industry. This lecture introduces the core elements of neural networks and deep learning, it comprises:

- (multilayer) perceptron, backpropagation, fully connected neural networks
- loss functions and optimization strategies
- convolutional neural networks (CNNs)
- activation functions
- regularization strategies
- common practices for training and evaluating neural networks
- visualization of networks and results
- common architectures, such as LeNet, Alexnet, VGG, GoogleNet
- recurrent neural networks (RNN, TBPTT, LSTM, GRU)
- deep reinforcement learning
- unsupervised learning (autoencoder, RBM, DBM, VAE)
- generative adversarial networks (GANs)
- weakly supervised learning
- applications of deep learning (segmentation, object detection, speech recognition, ...)

The accompanying exercises will provide a deeper understanding of the workings and architecture of neural networks.

Lernziele und Kompetenzen:

The students

- explain the different neural network components,
- compare and analyze methods for optimization and regularization of neural networks,
- compare and analyze different CNN architectures,
- explain deep learning techniques for unsupervised / semi-supervised and weakly supervised learning,
- explain deep reinforcement learning,
- explain different deep learning applications,
- implement the presented methods in Python,
- autonomously design deep learning techniques and prototypically implement them,
- effectively investigate raw data, intermediate results and results of Deep Learning techniques on a computer,
- autonomously supplement the mathematical foundations of the presented methods by self-guided study of the literature,
- discuss the social impact of applications of deep learning applications. Literatur:
- Ian Goodfellow, Yoshua Bengio, Aaron Courville: Deep Learning. MIT Press, 2016.
- Christopher Bishop: Pattern Recognition and Machine Learning, Springer Verlag, Heidelberg, 2006
- Yann LeCun, Yoshua Bengio, Geoffrey Hinton: Deep learning. Nature 521, 436 - 444 (28 May 2015)

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Deep Learning (Prüfungsnummer: 901895)

(englische Bezeichnung: Deep Learning)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Maier

Modulbezeichnung:	Pattern Analysis (PA) (Pattern Analysis)	5 ECTS
Modulverantwortliche/r:	Andreas Maier	
Lehrende:	Christian Riess	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

- Pattern Analysis (SS 2019, Vorlesung, 3 SWS, Christian Riess)
 - Pattern Analysis Programming (SS 2019, Übung, 1 SWS, Daniel Stromer et al.)
-

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Pattern Recognition

Inhalt:

Based on the lecture Pattern Recognition, this lecture introduces the design of pattern analysis systems as well as the corresponding fundamental mathematical methods. The lecture comprises:

- an overview over regression and classification, in particular the method of least squares and the Bayes classifier
- clustering methods: soft and hard clustering
- classification and regression trees and forests
- parametric and non-parametric density estimation: maximum-likelihood (ML) estimation, maximum-a-posteriori (MAP) estimation, histograms, Parzen estimation, relationship between folded histograms and Parzen estimation, adaptive binning with regression trees
- mean shift algorithm: local maximization using gradient ascent for non-parametric probability density functions, application of the mean shift algorithm for clustering, color quantization, object tracking
- linear and non-linear manifold learning: curse of dimensionality, various dimensionality reduction methods: principal component analysis (PCA), local linear embedding (LLE), multidimensional scaling (MDS), isomaps, Laplacian eigenmaps
- Gaussian mixture models (GMM) and hidden Markov models (HMM): expectation maximization algorithm, parameter estimation, computation of the optimal sequence of states/Viterbi algorithm, forward-backward algorithm, scaling
- Bayesian networks
- Markov random fields (MRF): definition, probabilities on undirected graphs, Hammersley-Clifford theorem, cliques, clique potentials, examples for MRF-based image pre-processing and processing of image sequences
- Markov random fields and graph cuts: sub-modular functions, global optimization with graph cut algorithms, application examples

Aufbauend auf der Vorlesung Pattern Recognition führt die Vorlesung in das Design von Musteranalyse-Systemen sowie die zugrundeliegenden mathematischen Methoden ein. Die Vorlesung umfasst im Einzelnen:

- Überblick über Regression und Klassifikation, insbesondere die Methode der kleinsten Fehlerquadrate und der Bayes-Klassifikator
- Clustering-Methoden: Soft- und Hard-Clustering
- Klassifikations- und Regressionsbäume/-wälder
- parametrische und nicht-parametrische Dichteschätzung: Verfahren sind ML- und MAP-Schätzung, Histogramme, Parzenschätzung, Zusammenhang gefaltete Histogramme und Parzenschätzung, adaptives Binning mit Regressionsbäumen.

- 'Mean Shift'-Algorithmus: lokale Maximierung durch Gradientenaufstieg bei nicht-parametrischen Dichtefunktionen, Anwendungen des 'Mean Shift'-Algorithmus zum Clustering, Farbquantisierung und Objektverfolgung
- Linear and Non-Linear Manifold Learning: Curse of Dimensionality, Verschiedene Methode zur Dimensionsreduktion: Principal Component Analysis (PCA), Local Linear Embedding (LLE), Multidimensional Scaling (MDS), Isomap, Laplacian Eigenmaps
- Gaußsche Mischverteilungsmodelle (GMM) und Hidden-Markov-Modelle (HMM): 'Expectation Maximization'-Algorithmus, Parameterschätzung, Bestimmung der optimalen Zustandsfolge/Viterbi-Algorithmus, Vorwärts-Rückwärts-Algorithmus, Skalierung
- Bayes-Netze
- Markov-Zufallsfelder: Definition, Wahrscheinlichkeiten auf ungerichteten Graphen, HammersleyClifford-Theorem, Cliques, Cliques-Potenziale, Beispiele zur MRF-basierten Bildvorverarbeitung und Bildfolgenverarbeitung
- Markov Random Fields und Graph Cuts: submodulare Funktionen, globale Optimierung mit 'Graph Cut'-Algorithmen, Anwendungsbeispiele

Lernziele und Kompetenzen: The students

- explain the discussed methods for classification, prediction, and analysis of patterns,
- define regression and classification tasks as optimization problems,
- compare and analyze methods for manifold learning and select a suited method for a given set of features and a given problem,
- compare and analyze methods for probability density estimation and select a suited method for a given set of features and a given problem,
- apply non-parametric probability density estimation to pattern analysis problems,
- apply dimensionality reduction techniques to high-dimensional feature spaces,
- explain statistic modeling of feature sets and sequences of features,
- explain statistic modeling of statistical dependencies,
- implement presented methods in MatLab or Python,
- supplement autonomously the mathematical foundations of the presented methods by self-guided study of the literature,
- discuss the social impact of applications of pattern analysis solutions. Die Studierenden
- erläutern die behandelten Methoden zur Klassifikation, Vorhersage und Analyse von Mustern,
- formulieren Regressions- und Klassifikationsproblemen als Optimierungsaufgaben,
- vergleichen und analysieren Methoden des Manifold Learning und wählen für eine vorgegebene Fragestellung eine geeignete Methode aus,
- vergleichen und analysieren Methoden zur Dichteschätzung und wählen für eine vorgegebene Fragestellung eine geeignete Methode aus,
- wenden nicht-parametrische Dichteschätzung auf Probleme der Musteranalyse an,
- wenden Dimensionsreduktion bei hochdimensionalen Merkmalsräumen an,
- erläutern statistische Modellierung von Merkmalsmengen und Merkmalsfolgen,
- erklären statistische Modellierung abhängiger Größen,
- implementieren vorgestellte Verfahren in MatLab oder Python.
- ergänzen eigenständig mathematische Grundlagen der präsentierten Methoden durch selbstbestimmtes Studium der Literatur
- diskutieren die gesellschaftlichen Auswirkungen von Anwendungen der Musteranalyse Literatur:
- Richard O. Duda, Peter E. Hart und David G. Stork: Pattern Classification, Second Edition, 2004
- Christopher Bishop: Pattern Recognition and Machine Learning, Springer Verlag, Heidelberg, 2006
- Antonio Criminisi and J. Shotton: Decision Forests for Computer Vision and Medical Image Analysis, Springer, 2013
- Kevin P. Murphy: Machine Learning: A Probabilistic Perspective, MIT Press, 2012

- papers referenced in the lecture
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Pattern Analysis (Prüfungsnummer: 41201)

(englische Bezeichnung: Oral Examination on Pattern Analysis)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Christian Riess

Modulbezeichnung: Sprach- und Audiosignalverarbeitung (SAV) 5 ECTS
 (Speech and Audio Signal Processing)

Modulverantwortliche/r: Walter Kellermann

Lehrende: Walter Kellermann

Startsemester: SS 2019 Dauer: 1 Semester Turnus: jährlich (SS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Englisch

Lehrveranstaltungen:

Sprach- und Audiosignalverarbeitung (SS 2019, Vorlesung, 3 SWS, Walter Kellermann)

Übung zur Sprach- und Audiosignalverarbeitung (SS 2019, Übung, 1 SWS, MHD Modar Halimeh)

Empfohlene Voraussetzungen:

Vorlesung Signale und Systeme I & II

Inhalt:

The course concentrates on algorithms for speech and audio signal processing with applications in telecommunications and multimedia, especially

- physiology and models for human speech production and hearing: source-filter model, filterbank model of the cochlea, masking effects,
- representation of speech and audio signals: estimation and representation of short-term and longterm statistics in the time and frequency domain as well as the cepstral domain; typical examples and visualizations
- source coding for speech and audio signals: criteria, scalar and vector quantization, linear prediction, prediction of the pitch frequency; waveform coding, parametric coding, hybrid coding, codec standards (ITU, GSM, ISO-MPEG)
- basic concepts of automatic speech recognition (ASR): feature extraction, dynamic time warping, Hidden Markov Models (HMMs)
- basic concepts of speech synthesis: text-to-speech systems, model-based and data-driven synthesis, PSOLA synthesis system
- signal enhancement for acquisition and reproduction: noise reduction, acoustic echo cancellation, dereverberation using single-channel and multichannel algorithms.

Die Vorlesung behandelt Grundlagen und Algorithmen der Verarbeitung von Sprach- und Audiosignalen mit Anwendungen in Telekommunikation und Multimedia, insbesondere:

- Physiologie und Modelle der Spracherzeugung und des Hörens: Quelle-Filter-Modell, Filterbankmodell der Cochlea; Maskierungseffekte;
- Darstellung von Sprach- und Audiosignalen: Schätzung und Darstellung der Kurzzeit- und Langzeitstatistik in Zeit-, Frequenz- und Cepstralbereich; typische Beispiele, Visualisierungen;
- Quellencodierung für Sprache und Audiosignale: Kriterien; skalare und vektorielle Codierung; lineare Prädiktion; Pitchprädiktion; Wellenform-/Parameter-/Hybrid-Codierung; Standards (ITU, GSM, ISO-MPEG)
- Spracherkennung: Merkmalextraktion, Dynamic Time Warping, Hidden Markov Models
- Grundprinzipien der Sprachsynthese: Text-to-Speech Systeme, modellbasierte und datenbasierte Synthese, PSOLA-Synthese
- Signalverbesserung bei Signalaufnahme und -wiedergabe: Geräuschbefreiung, Echokompensation, Enthaltung mittels ein- und mehrkanaliger Verfahren; Lernziele und Kompetenzen:

The students

- understand basic physiological mechanisms of human speech production and hearing and can apply them for the analysis of speech and audio signals

- apply basic methods for the estimation and representation of the short-term and long-term statistics of speech and audio signals and can analyze such signals by means of these methods
- understand current methods for source coding of speech and audio signals and can analyze current coding standards
- verstehen die Grundbausteine von Spracherkennungssystemen und können deren Funktion mittels Rechnersimulation analysieren
- understand the basic principle of text-to-speech systems and can apply fundamental methods for speech synthesis
- can apply basic algorithms for speech enhancement and understand their functionality for real-world data.

Die Studierenden

- verstehen die grundlegenden physiologischen Mechanismen der Spracherzeugung und des Hörens beim Menschen und können diese zur Analyse von Sprach- und Audiosignalen anwenden
 - wenden die grundlegenden Methoden zur Schätzung und Darstellung der Kurzzeit- und Langzeitstatistik von Sprach- und Audiosignalen an und können diese damit analysieren
 - verstehen die aktuellen Methoden zur Quellencodierung von Sprache- und Audiosignalen und können aktuelle Codierstandards analysieren
 - verstehen die Grundbausteine von Spracherkennungssystemen und können deren Funktion mittels Rechnersimulation analysieren
 - verstehen die Grundprinzipien von Text-to-Speech Systemen und können elementare Algorithmen zur Sprachsynthese anwenden
 - können elementare Algorithmen zur Signalverbesserung anwenden und für reale Daten analysieren
- Literatur:

Gemäß themenbezogenen Angaben in der Lehrveranstaltung

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Speech and Audio Signal Processing (Prüfungsnummer: 64601)

(englische Bezeichnung: Speech and Audio Signal Processing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil
an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Walter Kellermann

Modulbezeichnung:	Grundlagen der Systemprogrammierung (GSP) (Fundamentals of Systems Programming)	5 ECTS
Modulverantwortliche/r:	Wolfgang Schröder-Preikschat	

Lehrende: Wolfgang Schröder-Preikschat, Jürgen Kleinöder

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

Systemprogrammierung 1 (SS 2019, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat et al.)
 Übungen zu Systemprogrammierung 1 (SS 2019, Übung, 2 SWS, Simon Ruderich et al.)
 Rechnerübungen zu Systemprogrammierung 1 und 2 (SS 2019, Übung, 2 SWS, Simon Ruderich et al.)

Inhalt:

- Grundlagen von Betriebssystemen (Adressräume, Speicher, Dateien, Prozesse, Koordinationsmittel; Betriebsarten, Einplanung, Einlastung, Virtualisierung, Nebenläufigkeit, Koordination/Synchronisation)
- Abstraktionen/Funktionen UNIX-ähnlicher Betriebssysteme
- Programmierung von Systemsoftware
- C, Make, UNIX-Shell (Solaris, Linux, MacOS X)

Lernziele und Kompetenzen: Die

Studierenden

- erwerben Kenntnisse über Grundlagen von Betriebssystemen
- verstehen Zusammenhänge, die die Ausführungen von Programmen in vielschichtig organisierten Rechensystemen ermöglichen
- erlernen die Programmiersprache C
- entwickeln Systemprogramme auf Basis der Systemaufrufchnittstelle UNIX-ähnlicher Betriebssysteme

Literatur:

- Lehrbuch: Betriebssysteme Grundlagen, Entwurf, Implementierung, Wolfgang Schröder-Preikschat, 2008
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Grundlagen der Systemprogrammierung (Klausur) (Prüfungsnummer: 31811)

(englische Bezeichnung: Examination (Klausur) on Foundations of System Programming)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Rahmen der Übungen gestellten Übungsaufgaben können abgegeben werden und werden in diesem Fall bewertet. Auf Basis des Ergebnisses dieser Bewertungen können bis zu 10 % Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Kleinöder

Modulbezeichnung:	Applied Visualization (AppVis) (Applied Visualization)	5 ECTS
-------------------	---	--------

Modulverantwortliche/r:	Roberto Grosso
-------------------------	----------------

Lehrende:	Roberto Grosso
-----------	----------------

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
------------------------	-------------------	-----------------------

Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch
----------------------	-----------------------	-------------------

Lehrveranstaltungen:

Applied Visualization (SS 2019, Vorlesung, 2 SWS, Roberto Grosso)
 Tutorials to Applied Visualization (SS 2019, Übung, 2 SWS, Daniel Zint)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Algorithmen und Datenstrukturen

Inhalt:

Die Visualisierung beschäftigt sich mit allen Aspekten, die im Zusammenhang stehen mit der visuellen Aufbereitung der (oft großen) Datenmengen aus technisch-wissenschaftlichen Experimenten oder Simulationen zum Zwecke des tieferen Verständnisses und der einfacheren Präsentation komplexer Phänomene. Die Vorlesung gibt eine Einführung in die grundlegenden Algorithmen und Datenstrukturen, sowie einen Überblick über die verfügbaren Softwarewerkzeuge und verbreiteten Dateiformate. Behandelt werden u.a. folgende Themen:

- Visualisierungsszenarien
- Gitterstrukturen und Interpolation
- Verfahren für 2D Skalar- und Vektorfelder
- Verfahren für 3D Skalar- und Vektorfelder
- Verfahren für multivariate Daten
- Volumenvisualisierung mit Isoflächen
- Direktes Volume-Rendering

In der Übung werden die Vorlesungsinhalte eingeübt und vertieft.

Visualization includes all aspects related to the visual preparation of usually large data sets from technical or scientific experiments and simulation. For a better understanding and a meaningful representation of complex phenomena, methods from interactive computer graphics are applied. This lecture introduces basic algorithms and data structures and gives an overview of available software tools and common data formats.

The lecture covers the following topics:

- scenarios for visualization
- meshes and data representation
- methods for 2D scalar and vector fields
- methods for 3D scalar and vector fields
- methods for multivariate data
- volume rendering with iso-surfaces
- direct volume rendering

Lernziele und Kompetenzen: Die Studierenden:

- verfügen über tieferes Verständnis der visuellen Aufarbeitung von großen Datenmengen aus technisch-wissenschaftlichen Experimenten oder Simulationen
 - sind mit grundlegenden Algorithmen und Datenstrukturen der Visualisierung wissenschaftlicher Daten vertraut
 - verfügen über einen Überblick über die verfügbaren Softwarewerkzeuge und verbreiteten Dateiformate
 - sind fähig, einfachere Präsentation komplexer Phänomene mit erlernten Methoden selbständig vorzubereiten
 - sind in der Lage, selbstständig einfache Computerprogramme für die Visualisierung anwendungsspezifischer Daten zu entwickeln. Students
 - have a deep understanding of the process of visual processing of large data sets from scientific experiments and simulations
 - can explain and apply fundamentals algorithm and data structures of scientific visualization to common practical problems
 - are familiar with standard software tools in the area of scientific data visualization
 - can carry out simple research projects requiring methods for the visualization of scientific data
 - are able to implement simple algorithms for the visualization of scientific data from common science and engineering applications Literatur:
 - M. Ward, G.G. Grinstein, D. Keim, Interactive Data Visualization: Foundations, Techniques, and Applications, Taylor & Francis, 2010
 - AC. Telea, Data Visualization: Principles and Practice, AK Peters, 2008
 - C.D. Hansen and C.R. Johnson, Visualization Handbook, Academic Press, 2004
 - G.M. Nielson, H. Hagen, H.Müller, Scientific Visualization, IEEE Computer Society Press, Los Alamitos, 1997
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Life Science Engineering (Master of Science)", "Maschinenbau (Bachelor of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Physische Geographie (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Applied Visualization (Prüfungsnummer: 37211)

(englische Bezeichnung: Applied Visualisation)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Klausur in elektronischer Form mit einem Anteil im Antwort-Wahl-Verfahren

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Roberto Grosso

Modulbezeichnung: Kanalcodierung (KaCo)
(Channel Coding)

5 ECTS

Modulverantwortliche/r: Clemens Stierstorfer

Lehrende: Clemens Stierstorfer

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

Tutorial for Channel Coding (SS 2019, Übung, 1 SWS, Arman Ahmadzadeh) Channel Coding (SS 2019, Vorlesung, 3 SWS, Clemens Stierstorfer)

Empfohlene Voraussetzungen:

Es ist hilfreich, wenn die Studierenden die erlernten Algorithmen in eine Programmiersprache (C, Matlab usw.) umsetzen können.

It would be very helpful if the participants can implement the specified algorithms into a programming language (C, Matlab, etc.).

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird:

Digital Communications

Information Theory and Coding

Inhalt:

- 1 Introduction and Motivation 1.1 Definition, Related Fields 1.2 Basic Principles 1.2.1 Schemes 1.2.2 How to Add Redundancy 1.2.3 Applications 1.3 Historical Notes
- 2 Fundamentals of Block Coding 2.1 General Assumptions 2.2 Transmission Channels 2.2.1 Discrete Time AWGN Channel 2.2.2 Binary Symmetric Channel (BSC) 2.2.3 Channels with Memory 2.3 Motivation for Coding 2.4 Fundamentals of Block Coding 2.4.1 Code and Encoding 2.4.2 Decoding
- 3 Introduction to Finite Fields I 3.1 Group 3.1.1 Orders of Elements and Cycles 3.1.2 Subgroups, Cosets 3.2 Field 3.3 Vector Spaces
- 4 Linear Block Codes 4.1 Generator Matrix 4.2 Distance Properties 4.3 Elementary Operations 4.4 Parity-Check Matrix 4.5 Dual Codes 4.6 Syndrome Decoding 4.7 Error Probability and Coding Gain 4.7.1 Error Detection 4.7.2 Error Correction - BMD 4.7.3 Error Correction - ML Decoding 4.7.4 Coding Gain 4.7.5 Asymptotic Results 4.8 Modifications of Codes 4.9 Bounds on the Minimum Distance 4.10 Examples for Linear Block Codes 4.10.1 Binary Hamming Codes ($q=2$) 4.10.2 Simplex Codes 4.10.3 Ternary Golay Code 4.10.4 Reed-Muller Codes
- 5 Linear Cyclic Codes 5.1 Modular Arithmetic 5.2 Generator Polynomial 5.3 Parity-Check Polynomial 5.4 Dual Codes 5.5 Discrete Systems over F_q 5.6 Encoders for Cyclic Codes 5.6.1 Generator Matrix 5.6.2 Non-Systematic Encoding 5.6.3 Systematic Encoding 5.6.4 Systematic Encoding Using $h(x)$ 5.7 Syndrome Decoding 5.7.1 Syndrome 5.7.2 Decoding Strategies 5.8 Examples for Linear Cyclic Block Codes 5.8.1 Repetition Code and Single Parity-Check Code 5.8.2 Binary Hamming Codes 5.8.3 Simplex Codes 5.8.4 Golay Codes 5.8.5 CRC Codes
- 6 Introduction to Finite Fields II 6.1 Extension Fields 6.2 Polynomials over Finite Fields 6.3 Primitive Element 6.4 Existence of Finite Fields 6.5 Finite Fields Arithmetic 6.6 Minimal Polynomials, Conjugate Elements, and Cyclotomic Cosets 6.7 Summary of Important Properties of Finite Fields 6.8 (Discrete) Fourier Transform over Finite Fields
- 7 BCH and RS Codes 7.1 The BCH Bound 7.2 Reed-Solomon Codes 7.3 BCH Codes 7.4 Algebraic Decoding of BCH Codes and RS Codes 7.4.1 Basic Idea 7.4.2 The Berlekamp-Massey Algorithm 7.5 Application: Channel Coding for CD and DVD 7.5.1 Error Correction for the CD 7.5.2 Error Correction for the DVD
- 8 Convolutional Codes 8.1 Discrete Systems over F 8.2 Trellis Coding 8.3 Encoders for Convolutional Codes 8.4 (Optimal) Decoding of Convolutional Codes 8.4.1 Maximum-Likelihood Sequence Estimation (MLSE) 8.4.2 Maximum A-Posteriori Symbol-by-Symbol Estimation

9 Codes with Iterative Decoding 9.1 State of the Art 9.2 Preliminaries 9.2.1 Check Equations 9.2.2 Repetition Code, Parallel Channels 9.2.3 Log-Likelihood Ratios(LLR) 9.3 Turbo Codes 9.4 LDPC Codes
 Lernziele und Kompetenzen:

Das Modul Kanalcodierung umfasst eine umfassende Einführung in die Grundlagen der algebraischen, fehlerkorrigierenden Blockcodes sowie einen Einstieg in die Thematik der Faltungscodes. Iterativ decodierte Codeschemata wie Turbo-Codes und LDPC-Codes werden ebenfalls eingeführt. Im Einzelnen sind die Inhalte oben aufgeführt.

Die Studierenden definieren die Problematik der Kanalcodierung, grenzen sie von anderen Codierverfahren (z.B. der Quellencodierung) ab und kennzeichnen die unterschiedlichen Ansätze zur Fehlerkorrektur und -erkennung. Sie nennen Beispiele für Einsatzgebiete von Kanalcodierung und geben einen Überblick über die historische Entwicklung des Fachgebiets.

Die Studierenden erstellen Übertragungsszenarien für den Einsatz von Kanalcodierung bestehend aus Sender, Übertragungskanal und Empfänger und beachten dabei die Grundannahmen beim Einsatz von Blockcodes bzw. der Modellierung der Kanäle. Sie formulieren mathematische Beschreibungen der Encodierung sowie der optimalen Decodierung bzw. suboptimaler Varianten.

Die Studierenden beherrschen die Grundlagen fehlerkorrigierender linearer Blockcodes, beschreiben diese mathematisch korrekt mittels Vektoren und Matrizen über endlichen Körpern und implementieren und bewerten zugehörige Encoder- und Decoderstrukturen insbesondere Syndromdecoder. Dabei modifizieren sie Generatormatrizen, ermitteln Prüfmatrizen und erstellen Syndromtabellen. Sie schätzen die minimale Hammingdistanz von Codes mittels (asymptotischer) Schranken ab und können den erzielbaren Codegewinn erläutern. Sie kennen und benutzen beispielhaften Codefamilien (z.B. Hamming-Codes, Simplex-Codes, Reed-Muller-Codes).

Die Studierenden erkennen die Vorteile zyklischer linearer Blockcodes und beschreiben diese mit Polynomen über endlichen Körpern. Sie nutzen die Restklassenrechnung bzgl. Polynomen zur Umsetzung systematischer Encoder und zur Realisierung von Syndromdecodern mittels Schieberegisterschaltungen. Sie kennen beispielhafte Codefamilien.

Die Studierenden nutzen Primkörper, Erweiterungskörper, Minimalpolynome und Kreisteilungsklassen sowie die Spektraldarstellung über endlichen Körpern zur Realisierung von BCH- und Reed-SolomonCodes gemäß der BCH-Schranke. Sie verstehen die Grundlagen der Decodierung von BCH- und ReedSolomon-Codes insbesondere des Berlekamp-Massey-Algorithmus. Sie skizzieren und erläutern die Kanalcodierkonzepte von CD und DVD.

Die Studierenden erklären die Unterschiede von Faltungscodes und Blockcodes, skizzieren anhand von tabellierten Generatorpolynomen zugehörige Encoder und erläutern diese. Sie erklären die Funktionsweise des optimalen Decoders (MLSE), demonstrieren diese beispielhaft und vergleichen sie mit symbolweiser Decodierung (MAPSE).

Die Studierenden verstehen die Grundlagen der iterativen Decodierung, insbesondere wenden sie die Grundlagen des Information Combining zur Kombination von verschiedenen Beobachtungen an. Sie verstehen die Bedeutung von Log-Likelihood-Ratios bei iterativen Decodiervorgängen und berechnen diese. Sie skizzieren die Grundlegenden Encoder- und Decoderstrukturen von Turbo-Codes und die Grundzüge der Codierung mit LDPC-Codes u.a. der Decodierung mittels Belief Propagation. Die Vorlesung erfolgt wechselweise auf Deutsch oder Englisch (Winter/Sommer). Die zur Verfügung gestellten Unterlagen sind ausschließlich in Englisch gehalten. Die Studierenden verwenden entweder die englischen Fachtermini sicher oder kennen diese und drücken sich sicher mit den entsprechenden deutschen Fachbegriffen aus.

Die Umsetzung der angegebenen Algorithmen in eine Programmiersprache (C, Matlab usw.) sollten die Studierenden zu diesem Zeitpunkt des Studiums üblicherweise beherrschen. Übungen hierzu bleiben der Eigeninitiative überlassen.

—

Students define the problems of channel coding, how to distinguish it from other coding methods (such as source coding) and how to describe the various different approaches to error correction and detection. They are able to list example application areas of channel coding and give an overview of the historical development of the field. Furthermore, they describe and analyze transmission

scenarios for the application of channel coding which consist of transmitter, transmission channel and receiver, taking into account the general assumptions for applying block codes or modeling the channels. They formulate mathematical descriptions of encoding, optimal decoding and sub-optimal methods. Students illustrate the principles of error-correcting linear block codes and describe them mathematically using vectors and matrices over finite fields. They implement and analyze corresponding encoder and decoder structures, in particular syndrome decoders, and modify generator matrices, construct test matrices and create syndrome tables. They estimate the minimum Hamming distance of codes using (asymptotic) bounds and are able to explain the coding gain that can be achieved in individual cases. They analyze and use example code families (e.g. Hamming codes, simplex codes, Reed-Muller codes).

Students explain the advantages of cyclic linear block codes and how to describe them with polynomials over finite fields. They apply polynomial modular arithmetic to implement systematic encoders and realize syndrome decoders using shift register circuits. They know and use exemplary code families. Students use prime fields, extension fields, minimal polynomials and cyclotomic cosets, and spectral representation over finite fields to implement BCH and Reed-Solomon codes using the BCH bound. They understand the foundations of decoding BCH and Reed-Solomon codes, in particular the BerlekampMassey algorithm, and how to sketch and explain the channel coding concepts of CDs and DVDs. Students are able to describe the differences between convolutional codes and block codes, to sketch the respective encoders based on tabulated generator polynomials and to explain them. They are able to explain how optimal decoders (MLSE) work using examples and compare them with symbol-by-symbol decoding (MAPPSE).

Students sketch the foundations of iterative decoding. In particular, they apply methods of information combining to combine different observations. They use and calculate log-likelihood ratios in iterative decoding processes, sketch the basic encoding and decoding structures of turbo codes and the basics of coding using LDPC codes (including decoding using belief propagation).

Students are able to use the English technical terms correctly or know them and are able to express themselves using the respective technical terms in German. Literatur:

- C. Stierstorfer, R. Fischer, J. Huber: Skriptum zur Vorlesung
 - M. Bossert: Kanalcodierung, Oldenbourg Wissenschaftsverlag, 3. Auflage, 2013
 - M. Bossert: Channel Coding for Telecommunications, John Wiley & Sons, 1999
 - B. Friedrichs: Kanalcodierung, Springer Verlag, 1996
 - S.B. Wicker: Error Control Systems for Digital Communications and Storage, Prentice-Hall, 1995
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Kanalcodierung_ (Prüfungsnummer: 62701)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020, 2. Wdh.: SS 2020

1. Prüfer: Clemens Stierstorfer

Organisatorisches:

Die Unterrichts- und Prüfungssprache (Deutsch oder Englisch) wird in der ersten Lehrveranstaltung mit den Studierenden vereinbart.

Modulbezeichnung: Systemnahe Programmierung in C (SPiC) 5 ECTS
 (System-Level Programming in C)

Modulverantwortliche/r: Volkmar Sieh

Lehrende: Volkmar Sieh, Jürgen Kleinöder

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
------------------------	-------------------	-----------------------

Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
----------------------	-----------------------	------------------

Lehrveranstaltungen:

Systemnahe Programmierung in C (SS 2019, Vorlesung, 2 SWS, Volkmar Sieh et al.)

Übungen zu Systemnahe Programmierung in C (SS 2019, Übung, 2 SWS, Benedict Herzog et al.)

Rechnerübungen zu Systemnahe Programmierung in C (SS 2019, Übung, 2 SWS, Benedict Herzog et al.)

Wiederholungsübungen im WS

Übungen zu Systemnahe Programmierung in C (für Wiederholer) (WS 2019/2020, optional, Übung, 2 SWS, Benedict Herzog et al.)

Rechnerübungen zu Systemnahe Programmierung in C (WS 2019/2020, optional, Übung, 2 SWS, Benedict Herzog et al.)

Empfohlene Voraussetzungen:

Grundlegende Kenntnisse der Programmierung (unabhängig von der Programmiersprache)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird:

Grundlagen der Informatik (als Prüfungsleistung)

Inhalt:

- Grundlegende Konzepte der systemnahen Programmierung
- Einführung in die Programmiersprache C (Unterschiede zu Java, Modulkonzept, Zeiger und Zeigerarithmetik)
- Softwareentwicklung auf „der nackten Hardware“ (ATmega-956;C) (Abbildung Speicher <> Sprachkonstrukte, Unterbrechungen (*interrupts*) und Nebenläufigkeit)
- Softwareentwicklung auf „einem Betriebssystem“ (Linux) (Betriebssystem als Ausführungsumgebung für Programme)
- Abstraktionen und Dienste eines Betriebssystems (Dateisysteme, Programme und Prozesse, Signale, Threads, Koordinierung)

Lernziele und Kompetenzen:

Studierende, die das Modul erfolgreich abgeschlossen haben:

- erläutern die grundlegenden Elemente der Programmiersprache C: Datentypen, Operatoren, Ausdrücke, Kontrollstrukturen, Funktionen, Variablen, Präprozessor.
- bewerten C im Vergleich zu Java im Bezug auf Syntax, Idiomatik und Philosophie.
- nennen wesentliche Unterschiede der Softwareentwicklung für eine Mikrocontrollerplattform versus einer Betriebssystemplattform.
- beschreiben die Funktionsweise von Zeigern.
- beschreiben die Realisierung von Strings und Stringoperationen in C
- verwenden spezifische Sprachmerkmale von C für die hardwarenahe Softwareentwicklung und den nebenläufigen Registerzugriff.
- entwickeln einfache Programme in C für eine Mikrocontroller-Plattform (AVR ATmega) sowohl mit als auch ohne Bibliotheksunterstützung.
- entwickeln einfache Programme für eine Betriebssystemplattform (Linux) unter Verwendung von POSIX Systemaufrufen.

- erläutern Techniken der Abstraktion, funktionalen Dekomposition und Modularisierung in C.
- beschreiben den Weg vom C-Programm zum ausführbaren Binärcode.
- reproduzieren die grundlegende Funktionsweise eines Prozessors mit und ohne Unterbrechungsbearbeitung.
- erläutern Varianten der Ereignisbehandlung auf eingebetteten Systemen.
- verwenden Unterbrechungen und Energiesparzustände bei der Implementierung einfacher Steuergeräte.
- erläutern dabei auftretende Synchronisationsprobleme (lost update, lost wakeup) und setzen geeignete Gegenmaßnahmen um.
- beschreiben Grundzüge der Speicherverwaltung auf einer Mikrocontrollerplattform und einer Betriebssystemplattform (Stackaufbau, Speicherklassen, Segmente, Heap).
- erläutern die Funktionsweise eines Dateisystems.
- verwenden die grundlegende Ein-/Ausgabeoperationen aus der C-Standardbibliothek.
- unterscheiden die Konzepte Programm und Prozess und nennen Prozesszustände.
- verwenden grundlegende Prozessoperationen (fork, exec, signal) aus der C-Standardbibliothek.
- erklären die Unterschiede zwischen Prozessen und Fäden und beschreiben Strategien zur Fadenimplementierung auf einem Betriebssystem.
- erläutern Koordinierungsprobleme auf Prozess-/Fadenebene und grundlegende Synchronisationsabstraktionen (Semaphore, Mutex).
- verwenden die POSIX Fadenabstraktionen zur Implementierung mehrfädiger Programme.

Literatur:

- Manfred Dausmann, Ulrich Bröckl, Dominic Schoop, et al. *C als erste Programmiersprache: Vom Einsteiger zum Fortgeschrittenen*. Vieweg+Teubner, 2010. ISBN: 978-3834812216. URL: <http://www.springerlink.com/content/978-3-8348-1221-6/#section=813748&page=1>
 - Brian W. Kernighan, Dennis M. Ritchie. *The C Programming Language*. Englewood Cliffs, NJ, USA: Prentice Hall PTR, 1988. ISBN: 978-8120305960
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Physik (Bachelor of Science)", "Physik mit integriertem Doktorandenkolleg (Bachelor of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Systemnahe Programmierung in C (Prüfungsnummer: 31701)

(englische Bezeichnung: Machine-oriented programming in C)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Rahmen der Übungen gestellten Übungsaufgaben können abgegeben werden und werden in diesem Fall bewertet. Auf Basis des Ergebnisses dieser Bewertungen können bis zu 10 % Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Volkmar Sieh

Modulbezeichnung:	Hardware-Software-Co-Design (HSCD-VU) (Hardware-Software-Co-Design)	5 ECTS
Modulverantwortliche/r:	Jürgen Teich	
Lehrende:	Jürgen Teich	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
Lehrveranstaltungen:		
Hardware-Software-Co-Design (SS 2019, Vorlesung, 2 SWS, Jürgen Teich)		
Übungen zu Hardware-Software-Co-Design (SS 2019, Übung, 2 SWS, Franz-Josef Streit et al.)		

Inhalt:

1. Überblick und Vergleich von Architekturen und Komponenten in Hardware/Software-Systemen.
2. Aufbau eines Compilers und Codeoptimierungsverfahren für Hardware und Software
3. Hardware/Software-Partitionierung (Partitionierung komplexer Systeme, Schätzungsverfahren, Performanzanalyse, Codegenerierung)
4. Interfacesynthese (Kommunikationsarten, Synchronisation, Synthese)
5. Verifikation und Cosimulation
6. Tafelübungen

Lernziele und Kompetenzen:

Zahlreiche Realisierungen eingebetteter Systeme (z.B. Mobiltelefone, Faxgeräte, Industriesteuerungen) zeichnen sich durch kooperierende Hardware- und Softwarekomponenten aus. Die Popularität solcher Realisierungsformen lässt sich begründen durch 1) die steigende Vielfalt und Komplexität heterogener Systeme, 2) die Notwendigkeit, Entwurfs- und Testkosten zu senken und 3) Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Entwurfsmethoden). Zum Beispiel bieten Halbleiterhersteller kostengünstige ASICs an, die einen Mikrocontroller und benutzerspezifische Peripherie und Datenpfade auf einem Chip integrieren.

Die Synthese solcher Systeme wirft jedoch eine Reihe neuartiger Entwurfsprobleme auf, insbesondere 1) die Frage der Auswahl von Hardware- und Softwarekomponenten, 2) die Partitionierung einer Spezifikation in Hard- und Software, 3) die automatische Synthese von Interface- und Kommunikationsstrukturen und 4) die Verifikation und Cosimulation.

Fachkompetenz Wissen

- Die Studierenden erhalten Einblick in ein aktuelles Forschungsgebiet.

Verstehen

- Die Studierenden verstehen Grundlagen des modernen Systementwurfs.
- Die Studierenden erklären Implementierungsalternativen für digitale Hardware/Software-Systeme.

Anwenden

- Die Studierenden wenden grundlegende Algorithmen an, zur Analyse und Optimierung von Hardware/Software-Systemen.

Literatur:

Teich, J.; Haubelt, C.: Digitale Hardware/Software-Systeme: Synthese und Optimierung; Springer, Berlin; Auflage: 2. erw. Aufl. (2. März 2007)

Teich, J.: Hardware/Software-Architekturen. Ergänzendes Skriptum zur Vorlesung.

Gajski, D.: Specification and Design of Embedded Systems. Prentice Hall, Englewood Cliffs, NJ, 1994.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Klausur Hardware-Software-Co-Design (Prüfungsnummer: 34901)

(englische Bezeichnung: Examination (Klausur) on Hardware/Software Co-Design)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich

Organisatorisches:

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Hardware-Software-Co-Design mit erweiterter Übung (HSCD-VEU)“ aus. Bemerkungen:
auch für Computational Engineering

Modulbezeichnung: Hardware-Software-Co-Design mit erweiterter 7.5 ECTS Übung (HSCD-VEU)
 (Hardware-Software-Co-Design with Extended Exercises)

Modulverantwortliche/r: Jürgen Teich

Lehrende: Jürgen Teich

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

Hardware-Software-Co-Design (SS 2019, Vorlesung, 2 SWS, Jürgen Teich)
 Übungen zu Hardware-Software-Co-Design (SS 2019, Übung, 2 SWS, Franz-Josef Streit et al.)
 Erweiterte Übungen zu Hardware-Software-Co-Design (SS 2019, Übung, 2 SWS, Franz-Josef Streit)

Inhalt:

1. Überblick und Vergleich von Architekturen und Komponenten in Hardware/Software-Systemen.
2. Aufbau eines Compilers und Codeoptimierungsverfahren für Hardware und Software
3. Hardware/Software-Partitionierung (Partitionierung komplexer Systeme, Schätzungsverfahren, Performanzanalyse, Codegenerierung)
4. Interfacesynthese (Kommunikationsarten, Synchronisation, Synthese)
5. Verifikation und Cosimulation
6. Tafelübungen
7. Demonstrationen mit rechnergestützten Entwurfswerkzeugen und praktische Übungen

Lernziele und Kompetenzen:

Zahlreiche Realisierungen eingebetteter Systeme (z.B. Mobiltelefone, Faxgeräte, Industriesteuerungen) zeichnen sich durch kooperierende Hardware- und Softwarekomponenten aus. Die Popularität solcher Realisierungsformen lässt sich begründen durch 1) die steigende Vielfalt und Komplexität heterogener Systeme, 2) die Notwendigkeit, Entwurfs- und Testkosten zu senken und 3) Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Entwurfsmethoden). Zum Beispiel bieten Halbleiterhersteller kostengünstige ASICs an, die einen Mikrocontroller und benutzerspezifische Peripherie und Datenpfade auf einem Chip integrieren.

Die Synthese solcher Systeme wirft jedoch eine Reihe neuartiger Entwurfsprobleme auf, insbesondere 1) die Frage der Auswahl von Hardware- und Softwarekomponenten, 2) die Partitionierung einer Spezifikation in Hard- und Software, 3) die automatische Synthese von Interface- und Kommunikationsstrukturen und 4) die Verifikation und Cosimulation.

Fachkompetenz Wissen

- Die Studierenden erhalten Einblick in ein aktuelles Forschungsgebiet.

Verstehen

- Die Studierenden verstehen Grundlagen des modernen Systementwurfs.
- Die Studierenden erklären Implementierungsalternativen für digitale Hardware/Software-Systeme.

Anwenden

- Die Studierenden wenden grundlegende Algorithmen an, zur Analyse und Optimierung von Hardware/Software-Systemen.

Sozialkompetenz

- Die Studierenden benutzen aktuelle Entwurfswerkzeuge für die Spezifikation, Optimierung und Prototypisierung von Hardware/Software-Systemen bei der kooperativen Bearbeitung der erweiterten Übung in Gruppen.

Literatur:

Teich, J.; Haubelt, C.: Digitale Hardware/Software-Systeme: Synthese und Optimierung; Springer, Berlin; Auflage: 2. erw. Aufl. (2. März 2007)

Teich, J.: Hardware/Software-Architekturen. Ergänzendes Skriptum zur Vorlesung.

Gajski, D.: Specification and Design of Embedded Systems. Prentice Hall, Englewood Cliffs, NJ, 1994.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Hardware-Software-Co-Design (Prüfungsnummer: 34901)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 66.6666666666667%

weitere Erläuterungen:

Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich

Hardware-Software-Co-Design (Vorlesung mit erweiterter Übung) (Prüfungsnummer: 958291)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Portfolio besteht aus: Klausur (90 min) und erfolgreicher Teilnahme an den erweiterten Übungen (verpflichtend)

Die Modulnote ergibt sich aus der Klausurnote.

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich

Organisatorisches:

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Hardware-Software-Co-Design (HSCDVU)“ aus.

Bemerkungen:

auch für Computational Engineering

Modulbezeichnung:	Parallele Systeme (PSYS-VU) (Parallel Systems)	5 ECTS
Modulverantwortliche/r:	Jürgen Teich	
Lehrende:	Frank Hannig, Jürgen Teich	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

Parallele Systeme (SS 2019, Vorlesung, 2 SWS, Jürgen Teich et al.)
 Übung zu Parallele Systeme (SS 2019, Übung, 2 SWS, Frank Hannig)

Inhalt:

Aktuelle PCs verfügen über Mehrkernprozessoren und Grafikkarten, die wiederum aus hunderten von einfachen Prozessoren bestehen können. Hierdurch wird ein hohes Maß an nebenläufiger Datenverarbeitung möglich, welche bis vor einigen Jahren nur in Großrechnern erreicht werden konnte. Die effiziente Ausnutzung dieser Parallelität bedarf allerdings mehr als nur mehrerer Prozessoren, insbesondere muss das zu lösende Problem Parallelverarbeitung erlauben. In dieser Vorlesung werden Eigenschaften unterschiedlicher paralleler Rechnerarchitekturen und Metriken zu deren Beurteilung behandelt. Weiterhin werden Modelle und Sprachen zum Programmieren paralleler Rechner eingeführt. Neben der Programmierung von allgemeinen Parallelrechnern werden Entwurfsmethoden (CAD) vorgestellt, wie man ausgehend von einer algorithmischen Problemstellung ein massiv paralleles Rechenfeld in VLSI herleiten kann, das genau dieses Problem optimal parallel berechnet. Solche Schaltungen spielen auf der Bit- bzw. Wortebene eine dominante Rolle (Arithmetik) sowie bei Problemen der Signal- und Bildverarbeitung (z.B. Filter).

Im Einzelnen werden behandelt:

1. Theorie der Parallelität (parallele Computermodelle, parallele Spezifikationsformen und -sprachen, Performanzmodelle und -berechnung)
2. Klassifikation paralleler und skalierbarer Rechnerarchitekturen (Multiprozessoren und Multicomputer, Vektorrechner, Datenflussmaschinen, VLSI-Rechenfelder)
3. Programmierbare System-on-Chip (SoC) und Mehrkern-Architekturen (Grafik-Prozessoren, Cell, etc.)
4. Programmierung paralleler Rechner (Sprachen und Modelle, Entwurfsmethoden und Compiler, Optimierung)
5. Massive Parallelität: Vom Algorithmus zur Schaltung

Today's PCs consist of multi-core processors and graphics cards that again comprise hundreds to thousands of simple processors. As a result of this, a very high degree of parallel data processing becomes possible, which was subjected to supercomputers a couple of years ago. The efficient exploitation of parallel processing requires not only multiple processors but also parallelism inherent in the problem to be solved. In this lecture, properties of different parallel computer architectures and corresponding quality metrics are examined. Further, models and parallel programming languages are introduced. In addition to programming general parallel computers, design methods (CAD) are presented that systematically transform an algorithmic problem description into a massive parallel processor array (VLSI), which can optimally execute the given problem in parallel. Such highly parallel circuits play an essential role at the bit level and circuit level (arithmetics) as well as in the case of signal processing and image processing (e.g., filter).

In detail, the following topics are covered:

1. Theory of parallelism (parallel models of computation, parallel specification and parallel languages, performance models)
2. Classification of parallel and scalable computer architectures (multi-processors and multi-computers, vector computers, data-flow machines, VLSI processor arrays)
3. Programmable System-on-Chip (SoC) and multi-core architectures (graphics processors, Cell, etc.)
4. Programming of parallel computers (languages and models, design methods and compiler, optimization)

5. Massive parallelism: From algorithm to circuit Lernziele

und Kompetenzen:

Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen der parallelen Datenverarbeitung. / *The focus of this lecture are foundations of parallel data processing.*

Fachkompetenz Verstehen

- Die Studierenden verstehen grundlegende Konzepte der parallelen Datenverarbeitung, sowohl theoretischer Art anhand von Modellen, als auch an Architekturbeispielen. / *The students*

become familiar with the fundamentals of parallel data processing, theoretic in the form of models as well as by architecture examples.

Anwenden

- Die Studierenden setzen sich mit modernen eingebetteten parallelen Ein-Chip-Architekturen auseinander. / *The students get familiar with modern embedded parallel system-on-chip architectures.*
- Die Studierenden wenden grundlegende Performanzmodelle und Parallelisierungstechniken zur Analyse und Optimierung von parallelen Algorithmen und Architekturen an. / *The students exercise basic performance models and parallelization techniques for the analysis and optimization of parallel algorithms and architectures.*
- Die Studierenden setzen die Modellierung und den Entwurf von massiv-parallelen Prozessorfeldern in konkreten Aufgaben selbstständig um. / *In concrete tasks, the students apply independently the modeling and the design of massively parallel processors arrays.*

Literatur:

Siehe

Webseite:

<https://www.cs12.tf.fau.de/lehre/lehveranstaltungen/vorlesungen/parallelesysteme/>

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Parallele Systeme (Prüfungsnummer: 35101)

(englische Bezeichnung: Parallel Systems)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Mündliche (Dauer: 30 min) oder schriftliche (Dauer: 90 min) Prüfung. Die Modulnote ergibt sich aus der Prüfung.

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich, 2. Prüfer: Frank Hannig

Organisatorisches:

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Parallele Systeme mit erweiterter Übung (PSYS-VEU)“ aus.

Bemerkungen:

auch für Computational Engineering

Modulbezeichnung:	Signale und Systeme II (SISY II) (Signals and Systems II)	5 ECTS
Modulverantwortliche/r:	André Kaup	
Lehrende:	Christian Herglotz, André Kaup, Fabian Brand	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

- Signale und Systeme II (SS 2019, Vorlesung, 2,5 SWS, André Kaup)
- Übung zu Signale und Systeme II (SS 2019, Übung, 1,5 SWS, Christian Herglotz)
- Tutorium zu Signale und Systeme II (SS 2019, optional, Tutorium, 1 SWS, Nils Genser)

Inhalt:

Diskrete Signale
 Elementare Operationen und Eigenschaften, spezielle diskrete Signale, Energie und Leistung, Skalarprodukt und Orthogonalität, Faltung und Korrelation
 Zeitdiskrete Fourier-Transformation (DTFT)
 Definition, Beispiele, Korrespondenzen, inverse zeitdiskrete Fourier-Transformation, Eigenschaften und Sätze
 Diskrete Fourier-Transformation (DFT)
 Definition, Beispiele, Korrespondenzen, Eigenschaften und Sätze, Faltung mittels der diskreten Fourier-Transformation, Matrixschreibweise, schnelle Fourier-Transformation (FFT) z-Transformation
 Definition, Beispiele, Korrespondenzen, inverse z-Transformation, Eigenschaften und Sätze
 Diskrete LTI-Systeme im Zeitbereich
 Beschreibung durch Impulsantwort und Faltung, Beschreibung durch Differenzgleichungen, Beschreibung durch Zustandsraumdarstellung
 Diskrete LTI-Systeme im Frequenzbereich
 Eigenfolgen, Systemfunktion und Übertragungsfunktion, Verkettung von LTI-Systemen, Zustandsraumbeschreibung im Frequenzbereich
 Diskrete LTI-Systeme mit speziellen Übertragungsfunktionen
 Reellwertige Systeme, verzerrungsfreie Systeme, linearphasige Systeme, minimalphasige Systeme und Allpässe, idealer Tiefpass und ideale Bandpässe, idealer Differenzierer
 Kausale diskrete LTI-Systeme und Hilbert-Transformation
 Kausale diskrete LTI-Systeme, Hilbert-Transformation für periodische Spektren, analytisches Signal und diskreter Hilbert-Transformator
 Stabilität diskreter LTI-Systeme
 BIBO-Stabilität, kausale stabile diskrete Systeme, Stabilitätskriterium für Systeme N-ter Ordnung
 Beschreibung von Zufallssignalen
 Erwartungswerte, stationäre und ergodische Zufallsprozesse, Autokorrelations- und Korrelationsfunktion, Leistungsdichtespektrum, komplexwertige Zufallssignale
 Zufallssignale und LTI-Systeme
 Verknüpfung von Zufallssignalen, Reaktion von LTI-Systemen auf Zufallssignale, Wienerfilter Lernziele und Kompetenzen:

Die Studierenden

- analysieren diskrete Signale mit Hilfe der zeitdiskreten Fourier-Transformation und berechnen deren diskrete Fourier-Transformation
- bestimmen die Impulsantwort, Direktformen und Zustandsraumdarstellung für diskrete lineare zeitinvariante Systeme
- berechnen System- und Übertragungsfunktionen für diskrete lineare zeitinvariante Systeme
- analysieren die Eigenschaften von diskreten linearen zeitinvarianten Systemen aufgrund der Zeit- und Frequenzbereichsbeschreibung

- stufen diskrete lineare zeitinvariante Systeme anhand ihrer Eigenschaften Verzerrungsfreiheit, Linearphasigkeit und Minimalphasigkeit ein
 - bewerten Kausalität und Stabilität von diskreten linearen zeitinvarianten Systemen
 - bewerten diskrete Zufallssignale durch Berechnung von Erwartungswerten und Korrelationsfunktionen
 - beurteilen die wesentlichen Effekte einer Filterung von diskreten Zufallssignalen durch diskrete lineare zeitinvariante Systeme
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Technomathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Signale und Systeme II (Prüfungsnummer: 26802)

(englische Bezeichnung: Signals and Systems II)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil
an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: André Kaup

Modulbezeichnung:	Heterogene Rechnerarchitekturen Online (HETRON) (Heterogeneous Computing Architectures Online)	5 ECTS
--------------------------	---	--------

Modulverantwortliche/r:	Dietmar Fey, Marc Reichenbach
--------------------------------	-------------------------------

Lehrende:	Thomas Heller, Johannes Hofmann, Marc Reichenbach
------------------	---

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
------------------------	-------------------	------------------------------

Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch
------------------------	------------------------	-------------------

Lehrveranstaltungen:

Heterogene Rechnerarchitekturen Online (SS 2019, Vorlesung, Marc Reichenbach et al.)

Inhalt:

Whereas heterogeneous architectures and parallel computing has filled an academic niche in the past it has become now a commodity technique with the rising of multi-core processors and programmable graphic cards. Even FPGAs play a role hereby in a certain extent due to their increasing importance as accelerator hardware what is clearly observable in the scientific community. However, on one side parallel hardware like multi-core and GPUs are now available nearly for everybody and not only for a selected selection of people, who have access to a parallel supercomputer. On the other side the knowledge about programming of this commodity hardware, and we mean here in particular hardwareorientated programming in order to squeeze out all offered GFlops and TFlops of such

hardware, is still missing as well as the knowledge about the architecture details. To overcome this lack we offer this course HETRON.

The e-learning course HETRON for the exploitation of parallel and heterogeneous computer architectures) focuses on two main topics which are closely related to each other. This concerns on one side the benefits of using different kinds of multi-core processors and parallel architectures built-up on base of these multicore processors. These architectures differ among each other in the number and in the complexity of its single processing nodes. We distinguish between systems consisting of a large number of simpler, so called fine-grained, processor cores vs. systems consisting of a smaller number of more complex, so called coarse-grained, processor cores. On the other side we lay our focus on that we want to do with these different heterogeneous parallel architectures, namely the execution of parallel programs. Of course this requires the use of parallel programming languages and environments, like CUDA or OpenMP. However, besides these questions of using the right syntax and the right compiler switches to optimize a parallel program it is a pre-requisite to understand how parallel computing really works. This refers (i) to the comprehension which basic mechanisms of parallel computing exist, (ii) where are the limits of getting more performance with parallel computing and (iii) in what context stand these mechanisms to heterogeneous architectures. In other words it handles the question which architecture is the best one for a certain parallelization technique. To teach these three topics, is one main goal we pursuit with the course HETRON, and of course, this more fundamental basics of heterogeneous and parallel computing have to be proven by means of concrete application examples to deepen the acquired knowledge about heterogeneous architectures and parallel computing principles.

Lernziele und Kompetenzen:

Die Studierenden ...

...verstehen die Notwendigkeit sowie grundlegende Anwendungsfälle für heterogene Rechnerarchitekturen.

...können den grundlegenden Aufbau und das Zusammenspiel der Komponenten heterogener Rechnerarchitekturen erklären. ...erläutern grundsätzliche Parallelisierungsprinzipien wie Amdahls Law, HighPerformance- und High-Throughput-Computing sowie Parallelisierungsstrategien. ...können einfache Programme mit Hilfe der vermittelten Parallelisierungsprinzipien (Amdahls Law, High-Performance- und High-Throughput-Computing) analysieren und entsprechende Parallelisierungsstrategien entwickeln. ...erklären den Aufbau sowie Stärken und Schwächen von verschiedenen Architekturen wie CPUs, GPUs, Many-Core Prozessoren und FPGAs.

...implementieren ausgewählte Anwendungsbeispiele (SHA256 Algorithmus, Ising-Modell und FastFourier-Transformation) auf oben genannte Architekturen.

...erforschen und bewerten verschiedener Parallelisierungstechniken in Abhängigkeit der Anwendung und der Architektur.

...erläutern die Grundlagen des Grid- und Cloud-Computings

...sind in der Lage parallele Berechnungen (SHA256) im Grid umzusetzen.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Heterogene Rechnerarchitekturen Online (Prüfungsnummer: 275245)

(englische Bezeichnung: Heterogeneous Computing Architectures Online)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Reichenbach

Modulbezeichnung:	Optimierung für Ingenieure (OptIngV) (Optimization for Engineers)	5 ECTS
Modulverantwortliche/r:	Johannes Hild, Martin Gugat	
Lehrende:	Martin Gugat, Johannes Hild	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 45 Std.	Eigenstudium: 105 Std.	Sprache: Englisch

Lehrveranstaltungen:
 Optimierung für Ingenieure (Optimization for Engineers) (SS 2019, Vorlesung mit Übung, 3 SWS, Johannes Hild)

Empfohlene Voraussetzungen:

Requires contents of the lecture Mathematics for Engineers I, II and III. Especially:

- Linear algebra
- Analysis of real valued functions
- Differential and integral calculus in multi dimensional spaces

Inhalt:

Introduction to continuous optimization problems and methods with and without constraints

- Classification of problem types
- Optimality conditions and termination criterions
- Descent directions and line search methods
- Convergence analysis

Unconstrained optimization

- Steepest descent and conjugate gradient
- Newton-type methods
- Nonlinear Least Squares

Constrained optimization

- Projection methods
- Trust Region
- Barrier and penalty methods Outlook
- Linear programming and simplex method
- Integer programming
- Noisy functions

Lernziele und Kompetenzen:

Fachkompetenz

Wissen

- Students list requirements, strengths and weaknesses of common optimization methods.
- Students recognize crucial components in existence and convergence proofs in the context of minimizing sequences.

Verstehen

- Students explain the different components of optimization methods.
- Students describe the relationship between requirements and conclusions of existence and convergence theorems in the context of minimizing sequences.

Anwenden

- Students check feasibility, well-posedness and constraint qualifications of optimization problems.

- Students formulate and solve optimality conditions analytically.
- Students apply optimization algorithms to optimization problems.

Analysieren

- Students analyse uncommon optimization approaches and extract their requirements, strengths and weaknesses.

Evaluiere (Beurteilen)

- Students evaluate the class and structure of unsolved optimization problems.
- Students choose suitable algorithmic approaches for unsolved optimization problems.

Erschaffen

- Students formulate optimization problems using mathematical methods and structures.
- Students modify and combine common optimization routines to create project-specific algorithms for unsolved optimization problems.

Literatur:

Nocedal, Jorge and Wright, Stephen J.: Numerical Optimization. Springer Serie in Operations Research, 2006.
 Kelley, C. T.: Iterative Methods for Optimization. Frontiers in Applied Mathematics 18, SIAM Philadelphia 1999;
 Polak, E.: Optimization. Algorithms and Consistent Approximations. Applied Mathematical Sciences, Volume 124, Springer-Verlag New York, 1997.
 Jarre, F.: Optimierung, Springer 2003;
 Hamacher, H.W. and K. Klamroth, K.: Linear and Network Optimization: bilingual textbook. Vieweg 2000

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Chemical Engineering - Nachhaltige Chemische Technologien (Master of Science)", "Chemie- und Bioingenieurwesen (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Life Science Engineering (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Optimierung für Ingenieure (Prüfungsnummer: 40501)

(englische Bezeichnung: Optimisation for Engineers)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstabelleung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Johannes Hild

Bemerkungen:

This module aims at students of the Faculty of Engineering of all disciplines and is suitable as an elective subject in the Bachelor's and Master's degree.

Modulbezeichnung:	Parallele Systeme mit erweiterter Übung (PSYS-VEU) (Parallel Systems with Extended Exercises)	7.5 ECTS
Modulverantwortliche/r:	Jürgen Teich	

Lehrende: Jürgen Teich, Frank Hannig

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 90 Std.	Eigenstudium: 135 Std.	Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

Parallele Systeme (SS 2019, Vorlesung, 2 SWS, Jürgen Teich et al.)
 Erweiterte Übungen zu Parallele Systeme (SS 2019, Übung, 2 SWS, Marcel Brand et al.)
 Übung zu Parallele Systeme (SS 2019, Übung, 2 SWS, Frank Hannig)

Inhalt:

Aktuelle PCs verfügen über Mehrkernprozessoren und Grafikkarten, die wiederum aus hunderten von einfachen Prozessoren bestehen können. Hierdurch wird ein hohes Maß an nebenläufiger Datenverarbeitung möglich, welche bis vor einigen Jahren nur in Großrechnern erreicht werden konnte. Die effiziente Ausnutzung dieser Parallelität bedarf allerdings mehr als nur mehrerer Prozessoren, insbesondere muss das zu lösende Problem Parallelverarbeitung erlauben. In dieser Vorlesung werden Eigenschaften unterschiedlicher paralleler Rechnerarchitekturen und Metriken zu deren Beurteilung behandelt. Weiterhin werden Modelle und Sprachen zum Programmieren paralleler Rechner eingeführt. Neben der Programmierung von allgemeinen Parallelrechnern werden Entwurfsmethoden (CAD) vorgestellt, wie man ausgehend von einer algorithmischen Problemstellung ein massiv paralleles Rechenfeld in VLSI herleiten kann, das genau dieses Problem optimal parallel berechnet. Solche Schaltungen spielen auf der Bit- bzw. Wortebene eine dominante Rolle (Arithmetik) sowie bei Problemen der Signal- und Bildverarbeitung (z.B. Filter).

Im Einzelnen werden behandelt:

1. Theorie der Parallelität (parallele Computermodelle, parallele Spezifikationsformen und -sprachen, Performanzmodelle und -berechnung)
2. Klassifikation paralleler und skalierbarer Rechnerarchitekturen (Multiprozessoren und Multicomputer, Vektorrechner, Datenflussmaschinen, VLSI-Rechenfelder)
3. Programmierbare System-on-Chip (SoC) und Mehrkern-Architekturen (Grafik-Prozessoren, Cell, etc.)
4. Programmierung paralleler Rechner (Sprachen und Modelle, Entwurfsmethoden und Compiler, Optimierung)
5. Massive Parallelität: Vom Algorithmus zur Schaltung
6. Praktische Übungen mit rechnergestützten Werkzeugen

Today's PCs consist of multi-core processors and graphics cards that again comprise hundreds to thousands of simple processors. As a result of this, a very high degree of parallel data processing becomes possible, which was subjected to supercomputers a couple of years ago. The efficient exploitation of parallel processing requires not only multiple processors but also parallelism inherent in the problem to be solved. In this lecture, properties of different parallel computer architectures and corresponding quality metrics are examined. Further, models and parallel programming languages are introduced. In addition to programming general parallel computers, design methods (CAD) are presented that systematically transform an algorithmic problem description into a massive parallel processor array (VLSI), which can optimally execute the given problem in parallel. Such highly parallel circuits play an essential role at the bit level and circuit level (arithmetics) as well as in the case of signal processing and image processing (e.g., filter).

In detail, the following topics are covered:

1. *Theory of parallelism (parallel models of computation, parallel specification and parallel languages, performance models)*
2. *Classification of parallel and scalable computer architectures (multi-processors and multi-computers, vector computers, data-flow machines, VLSI processor arrays)*
3. *Programmable System-on-Chip (SoC) and multi-core architectures (graphics processors, Cell, etc.)*
4. *Programming of parallel computers (languages and models, design methods and compiler, optimization)*
5. *Massive parallelism: From algorithm to circuit*

6. Practical training with computer-aided design tools Lernziele

und Kompetenzen:

Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen der parallelen Datenverarbeitung. / *The focus of this lecture are foundations of parallel data processing.*

Fachkompetenz Verstehen

- Die Studierenden verstehen grundlegende Konzepte der parallelen Datenverarbeitung, sowohl theoretischer Art anhand von Modellen, als auch an Architekturbeispielen. / *The students become familiar with the fundamentals of parallel data processing, theoretic in the form of models as well as by architecture examples.*

Anwenden

- Die Studierenden setzen sich mit modernen eingebetteten parallelen Ein-Chip-Architekturen auseinander. / *The students get familiar with modern embedded parallel system-on-chip architectures.*
- Die Studierenden wenden grundlegende Performanzmodelle und Parallelisierungstechniken zur Analyse und Optimierung von parallelen Algorithmen und Architekturen an. / *The students exercise basic performance models and parallelization techniques for the analysis and optimization of parallel algorithms and architectures.*
- Die Studierenden setzen die Modellierung und den Entwurf von massiv-parallelen Prozessorfeldern in konkreten Aufgaben selbstständig um. / *In concrete tasks, the students apply independently the modeling and the design of massively parallel processors arrays.*
- Die Studierenden wenden das erlernte Wissen in Rechnerübungen an. / *The students apply their learned knowledge in hands-on computer exercises.*

Literatur:

Siehe

Webseite:

<https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/parallelesysteme/>

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Parallele Systeme (Vorlesung mit erweiterten Übungen) (Prüfungsnummer: 687796)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Mündliche (Dauer: 30 min) oder schriftliche (Dauer: 90 min) Prüfung + erfolgreiche Bearbeitung aller Übungsaufgaben in den erweiterten Übungen (verpflichtend). Die Modulnote ergibt sich aus der mündlichen oder schriftlichen Prüfung. Prüfungssprache: Deutsch oder Englisch

Erstabllegung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Jürgen Teich, 2. Prüfer: Frank Hannig

Organisatorisches:

Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Parallele Systeme (PSYS-VU)“ aus.

Bemerkungen:
auch für Computational Engineering

Modulbezeichnung:	Information Theory and Coding (ITC) (Information Theory and Coding)	5 ECTS
Modulverantwortliche/r:	Ralf Müller	
Lehrende:	Ralf Müller	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
Lehrveranstaltungen:		
Informationstheorie und Codierung (SS 2019, Vorlesung, 3 SWS, Ralf Müller)		
Übungen zu Informationstheorie und Codierung (SS 2019, Übung, 1 SWS, Ali Bereyhi et al.)		

Inhalt:

1. Introduction: binomial distribution, (7,4)-Hamming code, parity-check matrix, generator matrix 2. Probability, entropy, and inference: entropy, conditional probability, Bayes' law, likelihood, Jensen's inequality
3. Inference: inverse probability, statistical inference
4. The source coding theorem: information content, typical sequences, Chebychev inequality, law of large numbers
5. Symbol codes: unique decodability, expected codeword length, prefix-free codes, Kraft inequality, Huffman coding
6. Stream codes: arithmetic coding, Lempel-Ziv coding, Burrows-Wheeler transform
7. Dependent random variables: mutual information, data processing lemma
8. Communication over a noisy channel: discrete memory-less channel, channel coding theorem, channel capacity
9. The noisy-channel coding theorem: jointly-typical sequences, proof of the channel coding theorem, proof of converse, symmetric channels
10. Error-correcting codes and real channels: AWGN channel, multivariate Gaussian pdf, capacity of AWGN channel
11. Binary codes: minimum distance, perfect codes, why perfect codes are bad, why distance isn't everything
12. Message passing: distributed counting, path counting, low-cost path, min-sum (=Viterbi) algorithm
13. Exact marginalization in graphs: factor graphs, sum-product algorithm
14. Low-density parity-check codes: density evolution, check node degree, regular vs. irregular codes, girth
15. Lossy source coding: transform coding and JPEG compression

–

1. Einleitung: Binomialverteilung, (7,4)-Hamming-Code, Paritätsmatrix, Generatormatrix
2. Wahrscheinlichkeit, Entropie und Inferenz: Entropie, bedingte Wahrscheinlichkeit, Bayes'sches Gesetz, Likelihood, Jensen'sche Ungleichung
3. Inferenz: Inverse Wahrscheinlichkeit, statistische Inferenz
4. Das Quellencodierungstheorem: Informationsgehalt, typische Folgen, Tschebyschev'sche Ungleichung, Gesetz der großen Zahlen
5. Symbolcodes: eindeutige Dekodierbarkeit, mittlere Codewortlänge, präfixfreie Codes, Kraft'sche Ungleichung, Huffmancodierung
6. Stromcodes: arithmetische Codierung, Lempel-Ziv-Codierung, Burrows-Wheeler-Transformation
7. Abhängige Zufallsvariablen: Transinformation, Datenverarbeitungslemma
8. Kommunikation über gestörte Kanäle: diskreter gedächtnisloser Kanal, Kanalcodierungstheorem, Kanalkapazität
9. Das Kanalcodierungstheorem: verbundtypische Folgen, Beweis des Kanalcodierungstheorems, Beweis des Umkehrsatzes, symmetrische Kanäle

10. Fehlerkorrigierende Codes und reale Kanäle: AWGN-Kanal, mehrdimensionale Gauß'sche WDF, Kapazität des AWGN-Kanals
11. Binäre Codes: Minimaldistanz, perfekte Codes, Warum perfekte Codes schlecht sind, Warum Distanz nicht alles ist
12. Nachrichtenaustausch: verteiltes Zählen, Pfadzählen, günstigster Pfad, Minimumsummenalgorithmus
13. Exakte Marginalisierung in Graphen: Faktorgraph, Summenproduktalgorithmus
14. LDPC-Codes: Dichteevolution, Knotenordnung, reguläre und irreguläre Codes, Graphumfang
15. Verlustbehaftete Quellencodierung: Transformationscodierung und JPEG-Kompression

The students apply Bayesian inference to problems in both communications and everyday's life. The students explain the concept of digital communications by means of source compression and forward-error correction coding.

For the design of communication systems, they use the concepts of entropy and channel capacity.

They calculate these quantities for memoryless sources and channels.

The students prove both the source coding and the channel coding theorem.

The students compare various methods of source coding with respect to compression rate and complexity.

The students apply source compression methods to measure mutual information.

The students factorize multivariate functions, represent them by graphs, and marginalize them with respect to various variables.

The students explain the design of error-correcting codes and the role of minimum distance.

They decode error-correcting codes by means of maximum-likelihood decoding and message passing.

The students apply distributed algorithms to problems in both communications and everyday's life.

The students improve the properties of low-density parity-check codes by widening the girth and/or irregularity in the degree distribution.

The students transform source images into the frequency domain to improve lossy compression.

–

Die Studierenden wenden Bayes'sche Inferenz auf Probleme in der Nachrichtentechnik und im Alltagsleben an.

Die Studierenden erklären die konzeptuelle Trennung von digitaler Übertragung in Quellen- und Kanalcodierung.

Kommunikationssysteme entwerfen sie unter Betrachtung von Entropie und Kanalkapazität.

Sie berechnen diese Größen für gedächtnislose Quellen und Kanäle.

Die Studierenden beweisen sowohl das Quellen- als auch das Kanalcodierungstheorem.

Die Studierenden vergleichen verschiedenartige Quellencodierungsverfahren hinsichtlich Komplexität und Kompressionsrate.

Die Studierenden verwenden Quellencodierverfahren zur Messung von Transinformation.

Die Studierenden faktorisieren Funktionen mehrerer Veränderlicher, stellen diese als Graph dar und marginalisieren sie bezüglich mehrerer Veränderlicher.

Die Studierenden erklären den Entwurf von Kanalcodes und den Einfluss der Minimaldistanz.

Sie decodieren Kanalcodes gemäß maximaler Likelihood und Nachrichtenaustausch.

Die Studierenden wenden verteilte Algorithmen auf Probleme der Nachrichtentechnik und des Alltagslebens an.

Die Studierenden verbessern die Eigenschaften von LDPC-Codes durch Erhöhung des Umfangs und/oder durch irreguläre Knotenordnungsverteilungen.

Die Studierenden transformieren Bildquellen zur Verbesserung verlustbehafteter Kompression in den Frequenzbereich.

Literatur:

MacKay, D.: Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, 2003.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Wirtschaftsingenieurwesen (Bachelor of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Information Theory and Coding / Informationstheorie und Codierung (Prüfungsnummer: 36011)

(englische Bezeichnung: Information Theory and Coding)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Ralf Müller

Bemerkungen:

Schlüsselwörter: ASC

Modulbezeichnung:	Systemprogrammierung (SP) (Systems Programming)	10 ECTS
Modulverantwortliche/r:	Wolfgang Schröder-Preikschat	
Lehrende:	Jürgen Kleinöder, Wolfgang Schröder-Preikschat	

Startsemester: SS 2019	Dauer: 2 Semester	Turnus: jährlich (SS)
Präsenzzeit: 120 Std.	Eigenstudium: 180 Std.	Sprache: Deutsch

Lehrveranstaltungen:

Systemprogrammierung 1 (SS 2019, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat et al.)
 Übungen zu Systemprogrammierung 1 (SS 2019, Übung, 2 SWS, Simon Ruderich et al.)
 Rechnerübungen zu Systemprogrammierung 1 und 2 (SS 2019, Übung, 2 SWS, Simon Ruderich et al.)
 Systemprogrammierung 2 (WS 2019/2020, Vorlesung, 2 SWS, Jürgen Kleinöder et al.)
 Übungen zu Systemprogrammierung 2 (WS 2019/2020, Übung, 2 SWS, Dustin Nguyen et al.)
 Rechnerübungen zu Systemprogrammierung 1 und 2 (WS 2019/2020, Übung, 2 SWS, Dustin Nguyen et al.)
 Übungen zu Systemprogrammierung 1 (für Wiederholer) (WS 2019/2020, optional, Übung, 2 SWS, Dustin Nguyen et al.)

Inhalt:

- Grundlagen von Betriebssystemen (Adressräume, Speicher, Dateien, Prozesse, Koordinationsmittel; Betriebsarten, Einplanung, Einlastung, Virtualisierung, Nebenläufigkeit, Koordination/Synchronisation)
- Abstraktionen/Funktionen UNIX-ähnlicher Betriebssysteme
- Programmierung von Systemsoftware
- C, Make, UNIX-Shell (Solaris, Linux, MacOS X)

Lernziele und Kompetenzen: Die Studierenden

- erwerben fundierte Kenntnisse über Grundlagen von Betriebssystemen
- verstehen Zusammenhänge, die die Ausführungen von Programmen in vielschichtig organisierten Rechensystemen ermöglichen
- erkennen Gemeinsamkeiten und Unterschiede zwischen realen und abstrakten (virtuellen) Maschinen • erlernen die Programmiersprache C
- entwickeln Systemprogramme auf Basis der Systemaufrufchnittstelle UNIX-ähnlicher Betriebssysteme

Literatur:

- Lehrbuch: Betriebssysteme - Grundlagen, Entwurf, Implementierung, Wolfgang Schröder-Preikschat, 2008
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Systemprogrammierung (Klausur) (Prüfungsnummer: 31801)

(englische Bezeichnung: Examination (Klausur) on System programming)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120

Anteil an der Berechnung der Modulnote: 100%

weitere Erläuterungen:

Die Klausur enthält einen kleinen Multiple-Choice-Anteil.

Die Rahmen der Übungen gestellten Übungsaufgaben können abgegeben werden und werden in diesem Fall bewertet. Auf Basis des Ergebnisses dieser Bewertungen können bis zu 10 % Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden. Zu Beginn des Teils Systemprogrammierung 2 wird eine Miniklausur angeboten, deren Ergebnis wie eine Übungsaufgabe behandelt wird.

Erstablingung: WS 2019/2020, 1. Wdh.: SS 2020, 2. Wdh.: WS 2020/2021 1.

Prüfer: Jürgen Kleinöder

Modulbezeichnung:	Numerik II für Ingenieure (NumIng2) (Numerics for Engineers II)	5 ECTS
Modulverantwortliche/r:	Wilhelm Merz	
Lehrende:	Wilhelm Merz	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

- Numerik II für Ingenieure (SS 2019, Vorlesung, 2 SWS, J. Michael Fried)
 - Übungen zur Numerik II für Ingenieure (SS 2019, Übung, 2 SWS, J. Michael Fried)
-

Inhalt:

Numerik partieller Differentialgleichungen
 Finite Differenzenmethode, Stabilität, Konsistenz, Konvergenz, Einführung finite Elementmethode bei elliptischen Problemen, Fehlerschätzer Lernziele und Kompetenzen:

Die Studierenden

- erklären verschiedene Diskretisierungsmethoden
- beurteilen diese Diskretisierungsmethoden
- leiten Finite Elemente Diskretisierungen elliptischer Probleme her
- folgern Aussagen anhand grundlegender Beweistechniken aus oben genannten Bereichen
- konstruieren Algorithmen zu Finite Elemente Diskretisierungen
- erklären Fehlerschätzer Literatur:

Skripte des Dozenten

H. Jung, M. Langer, Methode der Finiten Elemente, Teubner

P. Knabner, L. Angermann, Numerik partieller Differentialgleichungen, Springer

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Maschinenbau (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Vorlesung Numerik II für Ingenieure (Prüfungsnummer: 46311)

(englische Bezeichnung: Lecture: Numerics for Engineers II)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60 Anteil
 an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: J. Michael Fried

Modulbezeichnung:	Image and Video Compression (IVC) (Image and Video Compression)	5 ECTS
Modulverantwortliche/r:	André Kaup	
Lehrende:	Daniela Lanz, André Kaup	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

Image and Video Compression (SS 2019, Vorlesung, 3 SWS, André Kaup)

Übung Image and Video Compression (SS 2019, Übung, 1 SWS, Daniela Lanz)

Empfohlene Voraussetzungen:

Modul „Signale und Systeme II“ und das Modul „Nachrichtentechnische Systeme“

Inhalt:

Multi-Dimensional Sampling

Sampling theorem revisited, 2D sampling, spatiotemporal sampling, motion in 3D sampling

Entropy and Lossless Coding

Entropy and information, variable length codes, Huffman coding, unary coding, Golomb coding, arithmetic coding

Statistical Dependency

Joint entropy and statistical dependency, run-length coding, fax compression standards

Quantization

Rate distortion theory, scalar quantization, Lloyd-Max quantization, entropy coded scalar quantization, embedded quantization, adaptive quantization, vector quantization

Predictive Coding

Lossless predictive coding, optimum 2D linear prediction, JPEG-LS lossless compression standard, differential pulse code modulation (DPCM)

Transform Coding

Principle of transform coding, orthonormal transforms, Karhunen-Loève transform, discrete cosine transform, bit allocation, compression artifacts

Subband Coding

Principle of subband coding, perfect reconstruction property, discrete wavelet transform, bit allocation for subband coding

Visual Perception and Color

Anatomy of the human eye, sensitivity of the human eye, color spaces, color sampling formats

Image Coding Standards

JPEG and JPEG2000

Interframe Coding

Interframe prediction, motion compensated prediction, motion estimation, motion compensated hybrid coding

Video Coding Standards

H.261, H.263, MPEG-1, MPEG-2 / H.262, H.264 / MPEG-4 AVC, H.265 / MPEG-H HEVC Lernziele und

Kompetenzen:

Die Studierenden

- veranschaulichen die mehrdimensionale Abtastung und den Einfluss darauf durch Bewegung im Videosignal
- unterscheiden und bewerten verschiedene Verfahren zur verlustfreien Codierung von Bild- und Videodaten
- verstehen und analysieren Verbundentropie und statistische Abhängigkeiten in Bild- und Videodaten
- berechnen skalare und vektorielle Quantisierer nach unterschiedlichen Optimierungsvorgaben (minimaler mittlerer quadratischer Fehler, entropiecodiert, eingebetteter Quantisierer)
- bestimmen und evaluieren optimale ein- und zwei-dimensionale lineare Prädiktoren
- wenden Prädiktion und Quantisierung sinnvoll in einem gemeinsamen DPCM-System an
- verstehen das Prinzip und die Effekte von Transformations- und Teilbandcodierung für Bilddaten einschließlich optimaler Bitzuteilungen
- beschreiben die Grundzüge der menschlichen visuellen Wahrnehmung für Helligkeit und Farbe
- analysieren Blocksaltbilder und Wirkungsweisen hybrider Coder und Decoder für Videosignale

- kennen die maßgeblichen internationalen Standards aus ITU und MPEG zur Bild- und Videokompression.

The students

- visualize multi-dimensional sampling and the influence of motion within the video signal
- differentiate and evaluate different methods for lossless image and video coding
- understand and analyze mutual entropy and statistical dependencies in image and video data
- determine scalar and vector quantization for different optimization criteria (minimum mean square error, entropy coding, embedded quantization)
- determine and evaluate optimal one-dimensional and two-dimensional linear predictor
- apply prediction and quantization for a common DPCM system
- understand the principle and effects of transform and subband coding for image data including optimal bit allocation
- describe the principles of the human visual system for brightness and color
- analyze block diagrams and the functioning of hybrid coders and decoders for video signals
- know the prevailing international standards of ITU and MPEG for image and video compression.

Literatur:

J.-R. Ohm, „Multimedia Communications Technology“, Berlin: Springer-Verlag, 2004

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern " Communications Engineering (Master of Science)", "123#67#H", "Berufspädagogik Technik (Bachelor of Science)", "Berufspädagogik Technik (Master of Education)", "Communications and Multimedia Engineering (Master of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Bachelor of Science)", "Elektrotechnik, Elektronik und Informationstechnik (Master of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Image and Video Compression (Prüfungsnummer: 63101)

(englische Bezeichnung: Image and Video Compression)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil
an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: André Kaup

Modulbezeichnung:	Wearable and Implantable Computing (WIC) (Wearable and Implantable Computing)	5 ECTS
Modulverantwortliche/r:	Oliver Amft	
Lehrende:	und Mitarbeiter/innen, Oliver Amft	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:

WPF MT-MA-BDV ab 1 WPF MT-MA-MEL ab 1 WPF MT-MA-GPP ab 1 WPF MT-BA ab 5

Empfohlene Voraussetzungen:

- Ability to apply sensors, analyse signals, basic signal processing methods.
 - Ability to write scripts in Matlab, Python, or similar.
-

Inhalt:

The course provides an overview on the system design of wearable computing systems and implantable systems. Electronic design topics will be addressed, including bioelectronics, flexible electronics, electronics textile integration, multiprocess additive manufacturing. On the system functional level, frequent sensor and actuators and their designs for on-body and implantable systems are discussed. Powering and energy management concepts will be detailed, including processing and task scheduling, sparse sampling and sparse sample signal processing. Energy harvesting methods for wearable and implantable systems are analysed. Principles of biocompatibility and system validation for remote health monitoring are covered. Concrete design problems related to context awareness, energy-efficient context recognition, and mechanical design in medical applications are demonstrated, prototypes realised and discussed in mini-projects.

Submitting reports for all exercises is compulsory to be accepted for the oral exam.

Lernziele und Kompetenzen:

Fachkompetenz Verstehen

- Gain overview on context awareness, sensors and actuators for context management in digital health.
- Understand design concepts and apply/analyse wearable and implantable system design methods for accessories, smart textiles, skin-attachables using soft substrates, and encapsulation. *Analysieren*
- Analyse the electrical and physical principles, select and optimise on-body energy harvesting and power management techniques.

Evaluieren (Beurteilen)

- Apply system evaluation methods, assess and design for biocompatibility.

Erschaffen

- Create continuous context recognition and energy-efficient processing using sparse sampling, related signal and pattern processing methods.
- Create digital models of wearable systems.

Literatur:

Literature references will be provided during the lecture.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Medizintechnik (Bachelor of Science)" verwendbar.

Modulbezeichnung:	Magnetic Resonance Imaging 2 (MRI2) (Magnetic Resonance Imaging 2)	2.5 ECTS
Modulverantwortliche/r:	Frederik Laun	
Lehrende:	Andreas Maier, Armin Nagel, Frederik Laun	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Deutsch oder Englisch
Lehrveranstaltungen:		
Magnetic Resonance Imaging 2 (SS 2019, Vorlesung, 2 SWS, Frederik Laun et al.)		
Magnetic Resonance Imaging 2 - Übung (SS 2019, Übung, Frederik Laun et al.)		

Inhalt:

In der Vorlesung werden fortgeschrittene Techniken der Magnetresonanztomographie (MRT) erklärt. Vorausgesetzt werden Kenntnisse über Grundlagen des Gebietes, wie sie z.B. in der Vorlesung „Magnetic resonance imaging 1“ behandelt werden (Blochgleichungen, T1- und T2-Wichtung, Schichtselektion, k-Raum-Kodierung). U.a. folgende Themen werden behandelt: Echoplanare Bildgebung; Bildgebung des Flusses, der Perfusion, der Diffusion, der magnetischen Suszeptibilität; funktionelle MRT; X-KernBildgebung; Ultrahochfeld-MRT; CEST-Bildgebung; MRT-Technik; Beschleunigungsverfahren, z.B. parallele Bildgebung, Fingerprinting-Techniken; Angiographie; quantenmechanische Grundlagen; Bewegungskompensation.

The lecture covers advanced topics in magnetic resonance imaging (MRI). Knowledge about the basic principles of MRI are required as they are covered in the lecture „Magnetic Resonance Imaging 1“ (Bloch equations, T1 and T2 weighting, slice selection, k-space encoding). I.a. the following topics will be treated: echo planar imaging; imaging of flow, perfusion, diffusion, magnetic susceptibility; functional MRI; x-nuclei MRI; ultrahigh field MRI; chemical exchange saturation transfer imaging; MRI technique; acceleration methods, e.g. parallel imaging, fingerprinting; angiography; quantum mechanical foundations; motion compensation.

Lernziele und Kompetenzen:

The participants

- understand the principles, properties and limits of advanced MRI techniques
- develop the ability to adapt basic principles of MRI to advanced MRI techniques
- are able to explain MRI techniques, algorithms and concepts of the lecture to other engineers.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Physics (Master of Science)", "Physik (Bachelor of Science)", "Physik mit integriertem Doktorandenkolleg (Bachelor of Science)", "Physik mit integriertem Doktorandenkolleg (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Magnetic Resonance Imaging 2 (Prüfungsnummer: 870721)

(englische Bezeichnung: Magnetic Resonance Imaging 2)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60 Anteil
an der Berechnung der Modulnote: 0%

Erstabelleung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Maier, 2. Prüfer: Frederik Laun

Bemerkungen:

The language of instruction will be decided upon at the first meeting according to the needs of the attending students.

Modulbezeichnung:	Computer Architectures for Medical Applications (CAMA) (Computer Architectures for Medical Applications)	5 ECTS
Modulverantwortliche/r:	Dietmar Fey	
Lehrende:	N.N., Gerhard Wellein	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Englisch

Lehrveranstaltungen:
 Computer Architectures for Medical Applications (SS 2019, Vorlesung, 2 SWS, Dietmar Fey et al.)
 Übung zu Computer Architectures for Medical Applications (SS 2019, Übung, Jakob Peschel et al.)

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Computer Architectures for Medical Applications (Prüfungsnummer: 41451)

(englische Bezeichnung: Computer Architectures for Medical Applications)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Dietmar Fey

Modulbezeichnung:	Diagnostic Medical Image Processing (VHB-Kurs) (DMIP-VHB) (Diagnostic Medical Image Processing (VHB course))	5 ECTS
Modulverantwortliche/r:	Andreas Maier	
Lehrende:	Andreas Maier	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

Lehrveranstaltungen:
 Medical Image Processing for Diagnostic Applications (VHB-Kurs) (SS 2019, Vorlesung, 4 SWS, Julian Hoßbach et al.)

Empfohlene Voraussetzungen: Ingenieurmathematik

Inhalt:

English version:

The contents of the lecture comprise basics about medical imaging modalities and acquisition hardware. Furthermore, details on acquisition-dependent preprocessing are covered for image

intensifiers, flatpanel detectors, and MR. The fundamentals of 3D reconstruction from parallel-beam to cone-beam reconstruction are also covered. In the last chapter, rigid registration for image fusion is explained. In the exercises, algorithms that were presented in the lecture are implemented in Java.

Deutsche Version:

Die Inhalte der Vorlesung umfassen Grundlagen der medizinischen Bildverarbeitung und Aufnahmeprinzipien. Darüber hinaus werden Details der Vorverarbeitung für Bildverstärker, Flachpanel-detektoren und MR erklärt. Die Grundlagen der Rekonstruktion von Parallelstrahl bis hin zur Kegelstrahl-Tomographie werden ebenfalls behandelt. Im letzten Kapitel wird starre Registrierung für Bildfusion erläutert. In den Übungen werden Algorithmen aus der Vorlesung in Java implementiert.

Lernziele und Kompetenzen:

English Version: The participants

- understand the challenges in interdisciplinary work between engineers and medical practitioners.
- develop understanding of algorithms and math for diagnostic medical image processing.
- learn that creative adaptation of known algorithms to new problems is key for their future career.
- develop the ability to adapt algorithms to different problems.
- are able to explain algorithms and concepts of the lecture to other engineers. Deutsche Version:

Die Teilnehmer

- verstehen die Herausforderungen in der interdisziplinären Arbeit zwischen Ingenieuren und Ärzten.
 - entwickeln Verständnis für Algorithmen und Mathematik der diagnostischen medizinischen Bildverarbeitung.
 - erfahren, dass kreative Adaption von bekannten Algorithmen auf neue Probleme der Schlüssel für ihre berufliche Zukunft ist.
 - entwickeln die Fähigkeit Algorithmen auf verschiedene Probleme anzupassen.
 - sind in der Lage, Algorithmen und Konzepte der Vorlesung anderen Studenten der Technischen Fakultät zu erklären.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Diagnostic Medical Image Processing (Prüfungsnummer: 41501)

(englische Bezeichnung: Diagnostic Medical Image Processing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Maier

Modulbezeichnung: Interventional Medical Image Processing 5 ECTS
 (Online-Kurs) (IMIP)
 (Interventional Medical Image Processing (online course))

Modulverantwortliche/r: Andreas Maier

Lehrende: Andreas Maier

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: k.A. Std.	Eigenstudium: 150 Std.	Sprache: Englisch

Lehrveranstaltungen:

Medical Image Processing for Interventional Applications (VHB-Kurs) (SS 2019, Vorlesung, 4 SWS, Julian Hoßbach et al.)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Diagnostic Medical Image Processing (VHB-Kurs)

Inhalt:

English Version:

This lecture focuses on recent developments in image processing driven by medical applications. All algorithms are motivated by practical problems. The mathematical tools required to solve the considered image processing tasks will be introduced.

The lecture starts with an overview on preprocessing algorithms such as scatter correction for x-ray images, edge detection, super-resolution and edge-preserving noise reduction. The second chapter describes automatic image analysis using feature descriptors, key point detection, and segmentation using bottom-up algorithms such as the random walker or top-down approaches such as active shape models. Furthermore, the lecture covers geometric calibration algorithms for single view calibration, epipolar geometry, and factorization. The last part of the lecture covers non-rigid registration based on variational methods and motion-compensated image reconstruction.

Deutsche Version:

Die Vorlesung ist auf die jüngsten Entwicklungen in der Verarbeitung von medizinischen Bildern ausgerichtet. Alle Algorithmen werden durch praktische Probleme motiviert. Die mathematischen Werkzeuge, die für die Bildverarbeitungsaufgaben benötigt werden, werden eingeführt.

Die Vorlesung beginnt mit einem Überblick über Vorverarbeitungsalgorithmen, wie zum Beispiel Streustrahlkorrektur für Röntgenbilder, Kantenerkennung, Superresolution und kantenerhaltende Rauschunterdrückung. Das zweite Kapitel beschreibt die automatische Bildanalyse mit Merkmalsdeskriptoren, Punkterkennung und Segmentierung mit Bottom-up-Algorithmen wie dem Random-Walker oder TopDown-Ansätzen wie aktiven Formmodellen. Darüber hinaus deckt die Vorlesung auch geometrische Kalibrierungsalgorithmen zur Einzelansicht-Kalibrierung, Epipolargeometrie und Faktorisierung ab. Der letzte Teil der Vorlesung deckt nicht-starre Registrierung auf der Grundlage von Variationsmethoden und bewegungskompensierter Bildrekonstruktion ab.

Lernziele und Kompetenzen: English

Version:

The participants

- summarize the contents of the lecture.
- apply pre-processing algorithms such as scatter correction and edge-preserving filtering.
- extract information from images automatically by image analysis methods such as key point detectors and segmentation algorithms.
- calibrate projection geometries for single images and image sequences using the described methods.

- develop non-rigid registration methods using variational calculus and different regularizers.
- adopt algorithms to new domains by appropriate modifications.

Deutsche Version:

Die Teilnehmer

- fassen die Inhalte der Vorlesung zusammen.
 - wenden Vorverarbeitungsalgorithmen wie Streustrahlkorrektur und kantenerhaltende Filterung an.
 - extrahieren automatisch Informationen aus Bildern, indem sie Bildanalyseverfahren wie Punktdetektoren und Segmentierungsalgorithmen verwenden.
 - kalibrieren Projektionsgeometrien für Einzelbilder und Bildsequenzen mit den beschriebenen Methoden.
 - entwickeln nicht-starre Registrierungsmethoden mit Hilfe von Variationsrechnung und unterschiedlichen Regularisierungen.
 - wenden Algorithmen auf neue Modalitäten durch entsprechende Änderungen im Algorithmus an.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Interventional Medical Image Processing (VHB-Kurs) (Prüfungsnummer: 41401)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Maier

Modulbezeichnung:	Magnetic Resonance Imaging 2 + Übung (MRI2+Ü) (Magnetic Resonance Imaging 2 + Exercises)	5 ECTS
Modulverantwortliche/r:	Frederik Laun	
Lehrende:	Frederik Laun, Andreas Maier, Armin Nagel	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch
Lehrveranstaltungen:		
Magnetic Resonance Imaging 2 (SS 2019, Vorlesung, 2 SWS, Frederik Laun et al.)		
Magnetic Resonance Imaging 2 - Übung (SS 2019, Übung, Frederik Laun et al.)		

Inhalt:

In der Vorlesung werden fortgeschrittene Techniken der Magnetresonanztomographie (MRT) erklärt. Vorausgesetzt werden Kenntnisse über Grundlagen des Gebietes, wie sie z.B. in der Vorlesung „Magnetic resonance imaging 1“ behandelt werden (Blochgleichungen, T1- und T2-Wichtung, Schichtselektion, k-Raum-Kodierung). U.a. folgende Themen werden behandelt: Echoplanare Bildgebung; Bildgebung des Flusses, der Perfusion, der Diffusion, der magnetischen Suszeptibilität; funktionelle MRT; X-KernBildgebung; Ultrahochfeld-MRT; CEST-Bildgebung; MRT-Technik; Beschleunigungsverfahren, z.B. parallele Bildgebung, Fingerprinting-Techniken; Angiographie; quantenmechanische Grundlagen; Bewegungskompensation.

The lecture covers advanced topics in magnetic resonance imaging (MRI). Knowledge about the basic principles of MRI are required as they are covered in the lecture „Magnetic Resonance Imaging 1“ (Bloch equations, T1 and T2 weighting, slice selection, k-space encoding). I.a. the following topics will be treated: echo planar imaging; imaging of flow, perfusion, diffusion, magnetic susceptibility; functional MRI; x-nuclei MRI; ultrahigh field MRI; chemical exchange saturation transfer imaging; MRI technique; acceleration methods, e.g. parallel imaging, fingerprinting; angiography; quantum mechanical foundations; motion compensation.

Lernziele und Kompetenzen:

The participants

- understand the principles, properties and limits of advanced MRI techniques
- develop the ability to adapt basic principles of MRI to advanced MRI techniques
- are able to explain MRI techniques, algorithms and concepts of the lecture to other engineers.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Studien-/Prüfungsleistungen:

Magnetic Resonance Imaging 2 + Übung (Prüfungsnummer: 568977)

(englische Bezeichnung: Magnetic Resonance Imaging 2 + Exercises)

Prüfungsleistung, Klausur, Dauer (in Minuten): 120 Anteil
an der Berechnung der Modulnote: 100%

Erstablegung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Andreas Maier, 2. Prüfer: Frederik Laun

Bemerkungen:

The language of instruction will be decided upon at the first meeting according to the needs of the attending students.

Modulbezeichnung:	Geometry Processing (GP) (Geometry Processing)		5 ECTS
Modulverantwortliche/r:	Roberto Grosso		
Lehrende:	Roberto Grosso		
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)	
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch	
Lehrveranstaltungen:			
Geometry Processing (SS 2019, Vorlesung, 2 SWS, Roberto Grosso)			
Tutorials to Geometry Processing (SS 2019, Übung, 2 SWS, Matthias Innmann et al.)			

Empfohlene Voraussetzungen:

Voraussetzung für die Teilnahme sind gute Kenntnisse in der Computer Grafik und der geometrischen Modellierung.

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Geometrische Modellierung - VU

Inhalt:

In dieser Vorlesung werden einige Aspekte der diskreten Geometrieverarbeitung vorgestellt. Schwerpunkte sind dabei Erzeugung, Manipulation und Analyse von Dreiecksnetzen sowie die Flächenrekonstruktion aus 3D Scanner Daten.

Folgende Themen werden in der Vorlesung behandelt:

- Delaunay-Triangulierung und Voronoi-Diagramme • 3D Scanning, Registrierung von 3D Scans
 - Vereinfachung und Reduktion von Dreiecksnetzen
 - Flächenrekonstruktion aus Punktwolken
 - Segmentierung von Dreiecksnetzen
 - Flächenparametrisierung
 - Animation / Deformation von Dreiecksnetzen Lecture's content
 - Delaunay triangulation and Voronoi diagrams
 - 3D scanning and registration of 3D scans
 - Mesh reduction and simplification
 - Surface mesh reconstruction
 - Mesh segmentation
 - Mesh parametrization
 - Animation and deformation of triangle meshes
- Lernziele und Kompetenzen: Die Studierenden:
- verfügen über tieferes Verständnis der grundlegenden Algorithmen der diskreten Geometrieverarbeitung
 - sind mit Datenstrukturen und Softwaremethoden für die Rekonstruktion, Analyse und Verarbeitung diskreter 3D-Modelle vertraut
 - sind fähig, neue Algorithmen für die Rekonstruktion und Manipulation von Dreiecksnetzen zu entwickeln und in einem Computerprogramm unter Zuhilfenahme von modernen Softwarebibliotheken zu implementieren
 - sind in der Lage, selbstständig neue anwendungsnahe Probleme aus dem industriellen Umfeld zu analysieren, Zusammenhänge zu erkennen und neue Lösungswege vorzuschlagen
 - Students
 - have a deep understanding of fundamental algorithms for the discrete geometry processing
 - are familiar with data structures and algorithms for the reconstruction, analysis and processing of 3D discrete models

- can develop and implement new algorithms for the reconstruction and manipulation of triangle meshes, and have the understanding how to use standard libraries
- can carry out research and development on typical problems from industrial applications on their own

Literatur:

- Polygon Mesh Processing. Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez and Bruno Levy, AK Peters 2010
 - Vorlesungsunterlagen
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Geometry Processing (Prüfungsnummer: 666129)

(englische Bezeichnung: Geometry Processing)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Roberto Grosso

Bemerkungen:

Yearly alteration in the language of instruction between German and English.

Modulbezeichnung:	Forensische Informatik (ForensInf) (Forensic Computing)	5 ECTS
Modulverantwortliche/r:	Felix Freiling	
Lehrende:	Christian Moch, Felix Freiling, Stefan Vömel	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch

Lehrveranstaltungen:

Forensische Informatik (SS 2019, Vorlesung mit Übung, 2 SWS, Felix Freiling)

Forensische Informatik - Übung (SS 2019, Übung, 2 SWS, Ralph Palutke)

Inhalt:

Forensische Informatik befasst sich mit der Sammlung, Aufbereitung und Analyse digitaler Beweismittel zur Verwendung vor Gericht. Ausgangspunkt ist jeweils der Verdacht auf einen Computereintrich oder eine Straftat, die mit Hilfe von digitalen Geräten vorgenommen worden ist. Diese Vorlesung gibt einen Überblick über die Methoden der forensischen Informatik aus einer wissenschaftlichen Perspektive. Der Schwerpunkt liegt auf der Analyse von Dateisystemen. Ziel der Vorlesung ist nicht die Ausbildung von Forensik-Praktikern, sondern die Vermittlung von Kenntnissen, die es einem erlauben, Forschung im Bereich Computerforensik zu betreiben. Voraussichtliche Themen:

- Definition forensische Informatik
- Der forensische Prozess und seine wissenschaftliche Fundierung
- Rechtliche Rahmenbedingungen

- Sichern von Festplatten
- Analyse verschiedener Dateisysteme (FAT32, NTFS, Ext2/Ext3)
- Tools

Lernziele und Kompetenzen:

Die Studierenden können Termini und Methoden der digitalen Forensik in die Entwicklung der forensischen Wissenschaften einordnen.

Die Studierenden können die wesentlichen Datenstrukturen verschiedener Dateisysteme erklären. Sie können die für forensische Zwecke wesentlichen Datenstrukturen lokalisieren und geeignete Werkzeuge zu ihrer Analyse auswählen und anwenden.

Die Studierenden können digitale Spuren konkreter Fallkonstellationen durch Anwendung von Werkzeugen rekonstruieren, analysieren, interpretieren und dokumentieren. Sie lernen ihre Untersuchungsergebnisse zu präsentieren und gegenüber kritischen Nachfragen zu verteidigen.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Forensische Informatik (Prüfungsnummer: 866129)

(englische Bezeichnung: Forensic Computing)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Felix Freiling

Modulbezeichnung:	Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (V+Ü) (EZS2) (Real-Time Systems 2 - Dependable Real-Time Systems (V+Ü))	5 ECTS
--------------------------	---	--------

Modulverantwortliche/r:	Peter Ulbrich
--------------------------------	---------------

Lehrende:	Peter Wägemann, Peter Ulbrich
------------------	-------------------------------

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
------------------------	-------------------	-----------------------

Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
----------------------	-----------------------	------------------

Lehrveranstaltungen:

Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (SS 2019, Vorlesung, 2 SWS, Peter Ulbrich) Übungen zu Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (SS 2019, Übung, 2 SWS, Simon Schuster et al.)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Echtzeitsysteme-V+Ü

Inhalt:

Viele Echtzeitsysteme sind in Bereiche des täglichen Lebens eingebettet, die hohe Anforderungen an die funktionale Sicherheit dieser Systeme stellen. Beispiele hierfür sind Fahrerassistenzsysteme in modernen Automobilen, medizinische Geräte, Prozessanlagen in Kernkraftwerken oder Chemiefabriken oder Flugzeuge. Fehlfunktionen in diesen Anwendungen ziehen mitunter katastrophale Konsequenzen nach sich - Menschen können ernsthaft verletzt oder sogar getötet werden, Landstriche können unbewohnbar gemacht oder zumindest großer finanzieller Schaden verursacht werden.

Dieses Modul betrachtet Methoden und Werkzeuge, die uns helfen können, einerseits zuverlässig Software zu entwickeln (also Fehler im Programm zu entdecken und zu vermeiden), und andererseits zuverlässige Software zu entwickeln (also Abstraktionen, die auch im Fehlerfall ihre Gültigkeit behalten). Hierbei steht weniger die Vermittlung theoretischer Grundkenntnisse auf diesen Gebieten im Vordergrund, also vielmehr

- die praktische Anwendung existierende Werkzeuge und Methoden
- sowie die Erfahrung und das Verständnis ihrer Grenzen.

Auf diese Weise soll ein Fundament für die konstruktive Umsetzung verlässlicher Echtzeitsysteme gelegt werden. Dieses Modul soll daher fundierte Anknüpfungspunkte für die Entwicklung verlässlicher Echtzeitsysteme vermitteln, die Ad-hoc-Techniken möglichst ersetzen sollen.

Lernziele und Kompetenzen:

Studierende, die das Modul erfolgreich abgeschlossen haben:

- nennen die Konzepte und die Taxonomie verlässlicher Systeme, unterscheiden Software- und Hardwarefehler und klassifizieren Fehler (Defekt, Fehler, Fehlverhalten).
- stellen Fehlerbäume auf.
- organisieren Softwareentwicklungsprojekte mittels der Versionsverwaltung git.
- vergleichen die verschiedenen Arten der Redundanz als Grundvoraussetzung für Fehlererkennung und -toleranz.
- entwickeln fehlertolerante Systeme mittels Replikation.
- diskutieren die Fehlerhypothese und die Sicherstellung von Replikdeterminismus.
- erläutern die Vor- und Nachteile softwarebasierter Replikation und den Einsatz von Diversität.
- wenden Informationsredundanz zur Härtung von Daten- und Kontrollflüssen an.
- bewerten die Effektivität der arithmetischer Codierung von Programmen und verallgemeinern diesen Ansatz auf die verschiedenen Implementierungsebenen (Maschinenprogramm zu Prozessinkarnation).
- interpretieren den Einfluss der Ausführungsplattform (Hardware, Betriebssystem) auf die Leistungsfähigkeit der Fehlererkennung.
- konzipieren eine fehlertolerante Ausführungsumgebung für ein softwarebasiertes TMR-System basierend auf ANBD-Codierung.
- nennen die Grundlagen der systematischen Fehlerinjektion.
- überprüfen die Wirksamkeit von Fehlertoleranzmechanismen mittels Fehlerinjektion auf der Befehlssatzebene.
- entwickeln Testfälle für die Fehlerinjektion mittels des fail* Werkzeugs.
- setzen Messergebnisse in Relation zu dem tatsächlichen Fehlerraum.
- beschreiben die Grundlagen der Fehlererholung (Vorwärts- bzw. Rückwärtskorrektur) und Reintegration fehlgeschlagener Knoten.
- vergleichen den Zustandstransfer am Beispiel der Running bzw. Recursive State Restoration.
- benennen Konzepte der Rückwärtskorrektur durch Entwurfsalternativen (Recovery Blocks).
- fassen die Grundlagen des dynamischen Testens zusammen.
- unterscheiden Black-Box und White-Box Testverfahren.
- konzipieren und implementieren Testfälle.
- überprüfen die Testüberdeckung anhand grundlegender Überdeckungskriterien (Anweisungs- bis Bedingungsüberdeckung).

- geben die Grundlagen der statischen Programmanalyse wieder.
- nennen die Funktionsweise von Hoare- WP-Kalkül.
- verifizieren eine Ampelsteuerung mittels des FramaC Werkzeugs zur statischen Analyse von C Programmen.
- beschreiben den Korrektheitsnachweis mittels abstrakter Interpretation und unterscheiden die konkrete von der abstrakten Programmsemantik.
- erläutern die Funktionsweise von Sammel- und Präfixsemantiken.
- erstellen einen Korrektheitsbeweis für einen a-b-Filter mittels des Astrée Werkzeugs zur abstrakten Interpretation von C Programmen.
- bewerten die Verlässlichkeit kommerzieller, sicherheitskritischer Systeme anhand von Fallstudien (Sizewell B, Airbus A320).
- erschließen sich typische Probleme und Fehlerquellen bei der Programmierung von eingebetteten Systemen im Allgemeinen.
- klassifizieren Fallstricke und Mehrdeutigkeiten in der Programmiersprache C99 im Besonderen.
- können in Gruppen kooperativ und effektiv arbeiten.
- können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
- können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Maschinenbau (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Verlässliche Echtzeitsysteme (Prüfungsnummer: 39451)

(englische Bezeichnung: Dependable Real-Time Systems)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Peter Ulbrich

Modulbezeichnung:	Knowledge Discovery in Databases (KDD) (Knowledge Discovery in Databases)	2.5 ECTS
-------------------	--	----------

Modulverantwortliche/r:	Klaus Meyer-Wegener
-------------------------	---------------------

Lehrende:	Klaus Meyer-Wegener
-----------	---------------------

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
------------------------	-------------------	-----------------------

Präsenzzeit: 30 Std.	Eigenstudium: 45 Std.	Sprache: Englisch
----------------------	-----------------------	-------------------

Lehrveranstaltungen:

Knowledge Discovery in Databases (SS 2019, Vorlesung, 2 SWS, Klaus Meyer-Wegener)

Inhalt:

- Why data mining?
- What is data mining?
- A multi-dimensional view of data mining
- What kinds of data can be mined?
- What kinds of patterns can be mined?
- What technologies are used?
- What kinds of applications are targeted?
- Major issues in data mining
- A brief history of data mining

Lernziele und Kompetenzen:

Die Studierenden

- kennen den typischen KDD-Prozess;
- kennen Verfahren zur Vorbereitung von Daten für das Data Mining;
- definieren Distanz- oder Ähnlichkeits-Funktionen auf einem speziellen Datenbestand;
- überprüfen Attribute eines Datensatzes auf ihre Bedeutung für die Analyse hin und transformieren ggf. Attributwerte geeignet;
- wissen, wie ein typisches Data Warehouse aufgebaut ist;
- kennen die Definition von Distanz- bzw. Ähnlichkeitsfunktionen für die verschiedenen Typen von Attributen;
- sind vertraut mit dem Prinzip des Apriori-Algorithmus zur Bestimmung von Mengen häufiger Elemente (frequent itemsets);
- kennen den FP-Growth-Algorithmus zum schnellen Auffinden von Mengen häufiger Elemente;
- geben die Definitionen von Support und Confidence für Assoziationsregeln wieder;
- beschreiben die Ermittlung von Assoziationsregeln auf der Basis von Mengen häufiger Elemente;
- sind in der Lage, die Vorgehensweise bei Klassifikationsaufgaben darzustellen;
- legen dar, wie ein Entscheidungsbaum auf einem Trainingsdatensatz erzeugt wird;
- stellen das Prinzip der Bayes'schen Klassifikation dar;
- zählen verschiedene Clustering-Verfahren auf; • beschreiben den Ablauf von k-Means-Clustering;
- kennen die verschiedenen Arten von Ausreißern.

The students:

- know the typical KDD process;
- know procedures for the preparation of data for data mining;
- know the definition of distance or similarity functions for the different kinds of attributes;
- define distance and similarity functions for a particular dataset;
- check attributes of a dataset for their meaning with reference to an analysis and transform attribute values accordingly, if required.
- know how a typical data warehouse is structured;
- are familiar with the principle of the Apriori algorithm for the identification of frequent itemsets;
- know the FP-growth algorithm for a faster identification of frequent itemsets;
- present the definitions of support and confidence for association rules;
- describe the construction of association rules based on frequent itemsets;
- are capable of describing the course of action in classification tasks;
- present the construction of a decision tree based on a training dataset;
- present the principle of Bayes' classification;
- enumerate different clustering procedures; • describe the steps of k-means clustering;
- know the different kinds of outliers.

Literatur:

Jiawei Han, Micheline Kamber, and Jian Pei: Data Mining - Concepts and Technologies, 3rd ed. Waltham, MA : Morgan Kaufmann, 2012 (The Morgan Kaufmann Series in Data Management Systems). - ISBN 978-0-12-381479-1

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Studien-/Prüfungsleistungen:

Knowledge Discovery in Databases (Prüfungsnummer: 392229)

(englische Bezeichnung: Knowledge Discovery in Databases)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Klaus Meyer-Wegener

Modulbezeichnung:	Informationssysteme in der Intensivmedizin (MedInfIntens) (Information Systems in Intensive Care)	5 ECTS
Modulverantwortliche/r:	Stefan Kraus, Stefan Kraus	
Lehrende:	Stefan Kraus, Dennis Toddenroth, Ixchel Castellanos	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: halbjährlich (WS+SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
Lehrveranstaltungen:		
Informationssysteme in der Intensivmedizin (SS 2019, Vorlesung mit Übung, 4 SWS, Stefan Kraus et al.)		

Inhalt:

Der gegenüber einer Normalstation deutlich erhöhte ärztliche und pflegerische Aufwand auf einer Intensivstation zieht entsprechend erhöhte Anforderungen an die klinische Dokumentation nach sich. So muss beispielsweise eine Vielzahl von Geräten an die elektronische Patientenakte angebunden werden, die auf Normalstationen gar nicht eingesetzt werden (z.B. Patientenmonitore, Beatmungsmaschinen und Dialysegeräte). Daher gibt es für Intensivstationen einen speziellen Typ von Dokumentationssystemen, das sogenannte PDMS (Patientendatenmanagementsystem). In Rahmen dieser Vorlesung erhalten die Studierenden eine detaillierte Einführung in die Nutzung eines PDMS und in eine Vielzahl von technischen und nichttechnischen Aspekten, die damit zusammenhängen.

- Der Prozess der medizinischen Dokumentation wird erläutert und anhand des Routinesystems am UK Erlangen demonstriert.
- Die Teilnehmer erhalten im Rahmen einer Vor-Ort-Besichtigung eine Einführung in intensivmedizinische Krankheitsbilder und die Arbeitsabläufe auf der größten Intensivstation am UK Erlangen.
- Der Projektleiter PDMS am UK Erlangen erläutert die Installation und den Betrieb des kommerziellen PDMS, das seit vielen Jahren in der klinischen Routine genutzt wird. Dies beinhaltet die Netzwerkarchitektur, die Gerätedatenübernahme und ausgewählte Aspekte der Konfiguration.
- Darüber hinaus wird in verschiedene Themengebiete eingeführt, die mit der klinischen Dokumentation in elektronischen Patientenakten zusammenhängen. Dazu gehören Aspekte des Datenschutzes, die Optimierung der Gebrauchstauglichkeit und der Einsatz wissensbasierter Funktion in der Form standardisierter Wissensmodule.

Lernziele und Kompetenzen:

Die Studierenden

- benennen grundlegende Arbeitsabläufe auf einer Intensivstation und die Besonderheiten der intensivmedizinischen Dokumentation.
- erklären die Architektur eines PDMS und die Eigenheiten seiner Schnittstellen.
- benutzen ein PDMS für grundlegende Dokumentationsschritte.
- erklären den Datenstrom von Medizingeräten in elektronische Patientenakten sowie die zugehörigen Abbildungsprozesse anhand ausgewählter Beispiele.
- benutzen eine Datenzugriffsschnittstelle für die Weiterverarbeitung dokumentierter Daten.
- nutzen die HL7-Inboundschnittstelle eines PDMS.
- kennen ausgewählte Aspekte des Datenschutzes.
- kennen die Grundlagen der medizinischen Wissensverarbeitung.
- Diskutieren die Vor- und Nachteile des wichtigsten Standards zur medizinischen Wissensrepräsentation.
- erstellen einfache Module zur automatisierten Verarbeitung von medizinischem Wissen.

- erklären wichtige Aspekte der Gebrauchstauglichkeit klinischer Systeme.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Mündliche Prüfung zu Informationssysteme in der Intensivmedizin (Prüfungsnummer: 29401)

(englische Bezeichnung: Oral Examination on Intensive Care Information Systems)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 20

Anteil an der Berechnung der Modulnote: 100% Prüfungssprache: Deutsch

Erstablingung: SS 2019, 1. Wdh.: keine Angabe

1. Prüfer: Hans-Ulrich Prokosch

Organisatorisches:

Die Veranstaltung findet als einwöchiger Blockkurs in der zweiten Hälfte des Semesters statt. Die genaue Zeitplanung wird in der Vorbesprechung zu Beginn des Semesters bekannt gegeben.

Die Teilnehmerzahl ist beschränkt. Einschreibungen werden nach Reihenfolge der Anmeldung per Mail an martin.ross@fau.de akzeptiert (bitte Matrikelnummer, Studienfach, Abschluss [Bachelor/Master] nennen).

Bemerkungen: Master

Modulbezeichnung:	Human Computer Interaction (HCI) (Human Computer Interaction)	5 ECTS
Modulverantwortliche/r:	Björn Eskofier	
Lehrende:	Björn Eskofier	

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

- Human Computer Interaction (SS 2019, Vorlesung, 3 SWS, Björn Eskofier)
- Human Computer Interaction Exercises (SS 2019, Übung, 1 SWS, Markus Wirth)

Inhalt:

Die Vorlesung vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet. Die folgenden Themen werden in der Vorlesung behandelt:

- Einführung in die Grundlagen der Mensch-Computer-Interaktion, historische Entwicklung
- Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme
- Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers
- Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides
- Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme
- Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen
- Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge
- Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten
- Akzeptanz, Evaluationsmethoden und Qualitätssicherung

Contents: Aim of the lecture is to teach basic knowledge of concepts, principles, models, methods and techniques for developing highly user-friendly Human Computer Interfaces. Beyond traditional computer system the topic of modern user interfaces is also discussed in the context of automobile and intelligent environments, mobile devices and embedded systems. This lecture addresses the following topics:

- Introduction to the basics of Human Computer Interaction
 - Design principles and models for modern user interfaces and interactive systems
 - Information processing of humans, perception, motor skills, properties and skills of the users
 - Interaction concepts, metaphors, standards, norms and style guides
 - In- and output devices, design space for interactive systems
 - Analysis-, design- and development methodologies and tools for easy to use user interfaces
 - Prototypic implementation of interactive systems
 - Architectures for interactive systems, User Interface Toolkits and components
 - Acceptance, evaluation methods and quality assurance
- Lernziele und Kompetenzen:
- Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion.
 - Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile.
 - Die Teilnahme an der Veranstaltung versetzt Studierende in die Lage, einen Entwicklungsprozess in der Mensch-Computer-Interaktion zu verstehen und umzusetzen.

- Sie werden weiterhin in die Lage versetzt, dies vor dem Hintergrund der Informationsverarbeitungsfähigkeit, Wahrnehmung und Motorik des Benutzers zu gestalten.
- Schlussendlich werden Methoden der Evaluation sowie Akzeptanz- und Qualitätssicherung erlangt.

Learning Objectives and Competences:

- Students develop an understanding for models, methods and concepts in the field of HumanComputer Interaction.
- They learn different approaches for designing, developing and evaluating User Interfaces and their advantages and disadvantages.
- Joining the course enables students to understand and execute a development process in the area of Human-Computer Interaction.
- Student will be able to do an UI evaluation by learning basics about Informationprocessing, perception and motoric skills of the user.
- Additionally, appropriate evaluation method as well as acceptance and quality assurance aspects will be learned.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Human Computer Interaction (Prüfungsnummer: 645618)

(englische Bezeichnung: Human Computer Interaction)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

Anteil an der Berechnung der Modulnote: 0%

weitere Erläuterungen:

90-minütige schriftliche Klausur über den Stoff der Vorlesung und der Übungen. Bei weniger als 25 Teilnehmern erfolgt die Prüfung nach vorheriger Ankündigung mündlich (Dauer: 30 Minuten).

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Björn Eskofier

Organisatorisches:

Folien zur Vorlesung und Organisation über Studon. Organisation and slides via StudOn.

Modulbezeichnung: Interaktive Computergraphik (InCG) 5 ECTS
(Interactive Computer Graphics)

Modulverantwortliche/r: Marc Stamminger

Lehrende: Marc Stamminger

Startsemester: SS 2019 Dauer: 1 Semester Turnus: jährlich (SS)

Präsenzzeit: 60 Std. Eigenstudium: 90 Std. Sprache: Deutsch

Lehrveranstaltungen:

Interaktive Computergraphik (SS 2019, Vorlesung, 2 SWS, Frank Bauer)

Übungen zur Interaktiven Computergraphik (SS 2019, Übung, 2 SWS, Alexander Lier et al.)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Computergraphik-VU

Inhalt:

In der Vorlesung werden Algorithmen der Interaktiven Computergraphik behandelt, die sich für eine hochparallele Ausführung auf spezieller Grafikhardware (GPUs) eignen. Insbesondere sind dies:

- Bump- und Normalmaps zur Erhöhung des visuellen Detailgrads
- Tessellierung auf Grafikhardware
- Berechnung von Schatten, insbesondere auch Filterung von Schatten und Schatten von Flächenlichtquellen
- Vereinfachte Globale Beleuchtungsverfahren
- Tiefen- und Bewegungsunschärfe

Lernziele und Kompetenzen: Die Studierenden

- klassifizieren Schattierungsverfahren
 - bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren - Algorithmen für Ray Tracing und Radiosity
 - ermitteln graphische Algorithmen zur Berechnung von Schatten für unterschiedlichen Lichtquellen
 - veranschaulichen Methoden zur Tessellierung von virtuellen 3D Modellen auf der Graphikhardware
 - klassifizieren Algorithmen zur Simulation von feinen Oberflächenstrukturen zur Erhöhung der Rendering-Detailgrad
 - skizzieren interaktive Algorithmen zur Berechnung globaler Beleuchtungsverfahren
 - erläutern Techniken zur Bestimmung von Tiefen- und Bewegungsschärfe Literatur:
 - Möller, Haines: "Real-Time Rendering"
 - Foley, van Dam, Feiner, Hughes: "Computer Graphics: Principles and Practice"
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Interaktive Computergraphik (Prüfungsnummer: 990298)

(englische Bezeichnung: Interactive Computer Graphics)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0%

Erstablesung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Marc Stamminger

Modulbezeichnung: Human Factors in Security and Privacy (HumSecPri) 5 ECTS
 (Human Factors in Security and Privacy)

Modulverantwortliche/r: Zinaida Benenson

Lehrende: Zinaida Benenson

Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
------------------------	-------------------	-----------------------

Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
----------------------	-----------------------	------------------

Lehrveranstaltungen:

Human Factors in Security and Privacy (SS 2019, Vorlesung, 2 SWS, Zinaida Benenson)

Human Factors in Security and Privacy - Übung (SS 2019, Übung, 2 SWS, Zinaida Benenson)

Empfohlene Voraussetzungen:

Basic knowledge in the area of IT security and privacy, such as security goals (CIA), basic protection mechanisms (symmetric and asymmetric cryptography principles, PKI) is required. This knowledge can be acquired through the attendance of the module "Applied IT Security" or similar modules.

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Angewandte IT-Sicherheit

Inhalt:

People are often said to be "the weakest link" in the chain of IT security measures. This course provides insight into the ways in which IT security is affected by people and why it happens. Special attention will be paid to complex environments such as companies, governmental organizations or hospitals. A number of guest talks from practitioners and researchers highlight some of the issues in greater depth. The course covers the following topics:

- Terminology of security and privacy, technical and non-technical protection measures
- Development and testing of usable security mechanisms (encryption and authentication tools, security policies, security warnings)
- Risk perception and decision making in security and privacy context (usage of security software, reaction to security warnings, divulging information in social media)
- Economics approach to security and privacy decision making (traditional and behavioral economics)
- Trade-offs between the national security and surveillance (psychology behind the EU data retention directive and NSA programs)
- Psychological principles of cyber fraud (scams, phishing, social engineering)
- Security awareness and user education
- Interplay of safety and security in complex systems
- Research methods in human factors (qualitative vs. quantitative research, usability testing, experimental design, survey design, interviews)

The exercises aim at deepening the understanding of the topics and are highly relevant for oral examinations. We plan to conduct approximately eight exercises per semester; the rest of the exercises is reserved for the guest talks. A typical exercise consist of two parts:

- (1) For each topic, the students receive a homework assignment consisting of practical exercises.
- (2) For each topic, the students receive 1-3 papers to read for the next exercise. The papers will be discussed in the class with the teaching assistant.

Lernziele und Kompetenzen:

The main goal of this course is for the students to develop a mindset that naturally takes into account typical psychological and physical characteristics of the users. In particular, when developing or evaluating security- and privacy-enhancing technologies or policies, the students are able to:

- critically appraise technological solutions or policies for likely "human factors" weaknesses in design and usage
- choose appropriate techniques for testing and evaluation of the design and usage
- develop and test improvements

More precisely, after the successful completion of the course the students are able to:

- discuss the meanings of the terms "security" and "privacy"
 - identify main research questions in the area of human factors in security and privacy
 - demonstrate specific difficulties in developing and testing of usable security mechanisms
 - compare different approaches to the development of usable security features
 - apply elements of the mental models approach and of user-centered design to development and evaluation of security- and privacy-enhancing techniques
 - contrast the approaches of traditional and behavioral economics to the explanation of security- and privacy-related behavior
 - illustrate the influence of the psychological risk perception principles (especially under- and overestimation of risk) on security and privacy decision making
 - argue advantages and disadvantages of mass surveillance and other kinds of mass data collection for security and privacy of citizens
 - explain main psychological principles behind the cyber fraud
 - illustrate specific difficulties in awareness campaigns and user training in the realms of security and privacy
 - critically appraise design and results of published user studies
 - plan and conduct small user studies
 - scan research papers and other materials for important points that clarify and deepen course contents
 - prepare and conduct a discussion in the class on a given topic, using research papers and other materials
 - develop well-founded personal opinions on the course topics and defend them in the class discussions
- Literatur:
- L. F. Cranor, S. Garfinkel. Security and usability: designing secure systems that people can use. O'Reilly Media, Inc., 2005.
 - Schneier, Bruce. "Beyond fear." Copernicus Book, 2003.
 - Anderson, Ross. Security engineering. 2nd edition, John Wiley & Sons, 2008. <http://www.cl.cam.ac.uk/rja14/book.html>
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mechatronik (Master of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Human Factors in Security and Privacy (Prüfungsnummer: 658644)

(englische Bezeichnung: Human Factors in Security and Privacy)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0% weitere

Erläuterungen:

Voraussetzung für die Zulassung zur Prüfung: erfolgreicher Vortrag in der Übung

Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Zinaida Benenson

Organisatorisches:

Die Modulsprache ist Deutsch, Folien sind auf Englisch.

This module will be held in German, slides are in English.

Modulbezeichnung: Security in Embedded Hardware (SEH) 5 ECTS
(Security in Embedded Hardware)

Modulverantwortliche/r: Jürgen Teich

Lehrende: Jürgen Teich

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

Security in Embedded Hardware (SS 2019, Vorlesung, 2 SWS, Jürgen Teich)

Übung zu Security in Embedded Hardware (SS 2019, Übung, 2 SWS, Jürgen Teich)

Inhalt:

Der Schutz eingebetteter Systeme gegenüber Angriffe Dritter auf gespeicherte Daten und Implementierungen, stellt eine immer wichtigere, jedoch auch durch zunehmende Vernetzung herausfordernde Aufgabe dar. Der Schutz der eingebetteten Systeme gegenüber bekannten als auch neueren ausgeklügelten Angriffsmöglichkeiten ist Gegenstand dieser Vorlesung. Es wird gezeigt, welche Angriffe existieren, welche Gegenmaßnahmen man ergreifen kann und wie man sichere eingebettete Systeme entwirft.

Einleitung und Motivation

- Was ist Security?
- Die Bedeutung von Security für zuverlässige Systeme
- Klassifikation von Angriffen
- Entwurf eingebetteter Systeme Angriffsszenarien
- Beispiele von Angriffsszenarien
- Kryptographischer Algorithmen als Ziel von Angriffen Angriffe durch Einschleusen von Code (Code Injection Attacks)
- Welche Arten von Code Injection-Angriffe gibt es? • Gegenmaßnahmen

Invasive physikalische Angriffe (Invasive Physical Attacks)

- Microprobing
- Reverse Engineering
- Differential Fault Analysis
- Gegenmaßnahmen

Nichtinvasive softwarebasierte Angriffe (Non-Invasive Logical Attacks)

- Erlangen von nicht autorisiertem Zugriff
- Gegenmaßnahmen

Nichtinvasive physikalische Angriffe (Non-Invasive Physical Attacks)

- Abhören
- Seitenkanalangriffe
- Gegenmaßnahmen

Lernziele und Kompetenzen:

Fachkompetenz Wissen

- Die Studierenden legen die entsprechenden Gegenmaßnahmen dar

- Die Studierenden nennen verschiedene Sicherheitseinrichtungen und -maßnahmen in eingebetteten Systemen

Verstehen

- Die Studierenden zeigen den Einfluss von Angriffen und deren Gegenmaßnahmen auf die Verlässlichkeit eines eingebetteten Systems auf
- Die Studierenden zeigen den zusätzlichen Aufwand (Fläche, Rechenzeit) von Sicherheitseinrichtungen auf

Analysieren

- Die Studierenden klassifizieren verschiedene Angriffstypen auf eingebettete Systeme

Sozialkompetenz

- Die Studierenden erarbeiten kooperativ in Gruppen Lösungskonzepte und implementieren diese gemeinsam 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.

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Security in Embedded Hardware (Prüfungsnummer: 172338)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020 (nur für Wiederholer) 1.

Prüfer: Jürgen Teich

Modulbezeichnung: Organic Computing (OC) 5 ECTS
(Organic Computing)

Modulverantwortliche/r: Rolf Wanka

Lehrende: Rolf Wanka

Startsemester: SS 2019

Dauer: 1 Semester

Turnus: jährlich (SS)

Präsenzzeit: 60 Std.

Eigenstudium: 90 Std.

Sprache: Deutsch oder Englisch

Lehrveranstaltungen:

Organic Computing (SS 2019, Vorlesung, 2 SWS, Rolf Wanka)

Übungen zu Organic Computing (SS 2019, Übung, 2 SWS, Alexander Raß)

Inhalt:

Unter Organic Computing (OC) versteht man den Entwurf und den Einsatz von selbst-organisierenden Systemen, die sich den jeweiligen Umgebungsbedürfnissen dynamisch anpassen. Diese Systeme

zeichnen sich dadurch aus, dass sie die sog. Self*-Eigenschaft besitzen, d.h. sie sind selbst-konfigurierend, selbst-optimierend, selbst-heilend, selbst-schützend, selbst-erklärend, ...

Als Vorbild für solche technischen Systeme werden Strukturen und Methoden biologischer und anderer natürlicher Systeme gewählt.

Lernziele und Kompetenzen:

Fachkompetenz

Wissen

Lernende können Wissen abrufen und wiedergeben. Sie kennen konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, Regeln, Theorien, Merkmale, Kriterien, Abläufe. Sie lernen den Begriff des Organic Computings von anderen Paradigmen zu unterscheiden.

Verstehen

Lernende können Beispiele anführen und Aufgabenstellungen interpretieren.

Anwenden

Lernende können ein neues Problem wie z.B. Ranking-Erstellung durch Transfer des Wissens lösen. *Analysieren*

Lernende können ein Problem in einzelne Teile zerlegen und so die Struktur des Problems verstehen.

Sozialkompetenz

Fähigkeit und Bereitschaft, zielorientiert mit anderen zusammenzuarbeiten. Literatur:

- Ch. Müller-Schloer, Ch. von der Malsburg, R. P. Würt. Organic Computing. Informatik-Spektrum, Band 27, Nummer 4, S. 332-336. (LINK)
 - I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85 (2003) 317-325. (LINK)
 - J. M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM 46 (1999) 604-632. (LINK)
 - M. Dorigo. V. Maniezzo. A Colorni. Ant system: an autocatalytic optimizing process. Technical Report 91-016, Politecnico di Milano, 1991. (LINK)
 - A. Badr. A. Fahmy. A proof of convergence for Ant algorithms. Information Sciences 160 (2004) 267-279.
 - M. Clerc. J. Kennedy. The particle swarm - Explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 8 (2002) 58-73.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Information and Communication Technology (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Organic Computing (Prüfungsnummer: 39701)

(englische Bezeichnung: Organic Computing)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90 Anteil
an der Berechnung der Modulnote: 100%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Rolf Wanka

Bemerkungen:
Auch für CE

Modulbezeichnung:	Datenstromsysteme (DSSmUe) (Datastream Systems)	5 ECTS
Modulverantwortliche/r:	Klaus Meyer-Wegener	
Lehrende:	Klaus Meyer-Wegener	
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch
Lehrveranstaltungen:		
Datenstromsysteme (SS 2019, Vorlesung, 2 SWS, Klaus Meyer-Wegener)		
Übungen zu Datenstromsysteme (SS 2019, Übung, 2 SWS, Klaus Meyer-Wegener)		

Inhalt:

Datenströme gibt es schon lange. Sie fließen als Nachrichten durch Netze, werden in Protokolldateien abgelegt (und dort oft vergessen oder nur bei Problemen im nachhinein konsultiert), werden zur Anzeige gebracht, gefiltert, ausgewertet, transformiert usw. Beispiele sind die unterschiedlichsten Arten von Messwerten (Temperatur, Helligkeit, Feuchtigkeit, Druck, chemische Zusammensetzung), Börsen- und Wirtschaftsnachrichten (Preise), Gebote in Online-Auktionen, Aktionen in Rechnersystemen (Anmeldungen, Zugriffe, Programmausführungen), um nur ein paar zu nennen.

Bislang wurden die Systeme zur Verarbeitung dieser Datenströme als Individuallösungen erstellt, oft sogar auf sehr speziellen Plattformen (SPS). Um das Jahr 2000 herum begann sich auch die Datenbankforschung mit diesem Problem zu befassen, weil man erkannt hatte, dass sich eine Reihe von Techniken dieses Gebiets auch auf Datenströme anwenden lassen, vor allem die Anfrageverarbeitung und -optimierung, aber auch Verteilung und Datenunabhängigkeit. Damit lassen sich auch für die Verarbeitung von Datenströmen generische Lösungen erstellen - so wie Datenbanksysteme eine generische Lösung für die Auswertung gespeicherter Daten sind.

Die nun entstehenden Datenstromsysteme werden in dieser Vorlesung vorgestellt. Diskutiert werden dabei u.a. • Anfragesprachen

- Schemastrukturen
- Sensornetze als eine der wichtigsten Datenquellen
- Anfrageauswertung und -optimierung
- Architekturen
- Anwendungsszenarien

Diese Lehrveranstaltung wurde von Prof. Meyer-Wegener und Dr. Daum erstellt und im Sommersemester 2007 erstmalig angeboten.

Lernziele und Kompetenzen:

DSS:

Die Studierenden:

- beschreiben Datenstromsysteme als neues Systemkonzept;
- grenzen Datenstromsysteme von Datenbanksystemen ab;
- erklären ein Datenstrommanagementsystem;
- grenzen Datenstromverarbeitung von Complex-Event Processing ab;
- charakterisieren Datenstromquellen und ihre Anbindung an das System;
- unterscheiden Zeitmodelle für Datenströme;
- zählen die wichtigsten Datenstrom-Operatoren auf;
- trennen blockierende von nicht blockierenden Operatoren;
- unterscheiden die verschiedenen Arten von Fensterdefinitionen;
- benennen den Ansatz und die Vorgehensweise des DSAM-Projekts;
- charakterisieren kurz die Datenstromsysteme Aurora, STREAM, Cayuga, Esper und PIPES;
- erläutern den Umgang mit Abweichungen von der Ordnung in Aurora;

- erklären die Operatoren NEXT und FOLD in Cayuga;
- verstehen die drei Datenstrommodelle von PIPES;
- erklären den Begriff "Schnappschuss-Reduzierbarkeit";
- beschreiben die unterschiedlichen Semantiken von zeitbasierten Fenstern anhand des SECRETModells;
- können Techniken der lokalen und verteilten Optimierung von Datenstromanfragen nennen;
- erklären Eddies als Mittel der dynamischen Optimierung;
- zählen verschiedene Anwendungsszenarien von Datenstromsystemen auf;
- formulieren Anfragen in einer Anfragesprache für Datenstromsysteme;
- überprüfen Aufgabenstellungen bei der Auswertung von Daten daraufhin, ob sie mit Hilfe eines Datenstromsystems gelöst werden können. bei DSSmUeb zusätzlich: Die Studierenden
- kennen die Vor- und Nachteile der Techniken zur Tupelinvalidierung;
- kennen das Problem, das bei der Kombination von Anwendungszeit und tupelbasierten Fenstern entsteht;
- erstellen eine SQL-Anfrage, die aus einem Einzelmessungsstrom einen Turnstile-Strom erzeugt;
- erklären die Bedeutung der beiden Funktionen "windows" und "wids", die Dave Maier eingeführt hat;
- diskutieren den Resample-Operator von Aurora.

Literatur:

Die Literatur zu Datenstromsystemen ist derzeit noch sehr forschungsorientiert; es sind einzelne Aspekte in Artikeln dargestellt, aber es gibt nur wenige Übersichtsarbeiten und schon gar kein Standardwerk. Insofern ist die folgende Liste zwangsläufig unvollständig und laufenden Änderungen unterworfen.

- GOLAB, Lukasz ; ÖZSU, M. Tamer: Data Stream Management. Milton Keynes UK : Morgan & Claypool Publishers, 2010 (Synthesis Lectures on Data Management 5). - ISBN 9781608452729. 14GI mat 17.6.5-126
- Aggarwal, Charu C. (Hrsg.): *Data Streams : Models and Algorithms*. Boston/Dordrecht/London : Kluwer Academic Publishers, 2006
- Motwani, Rajeev ; Widom, Jennifer, u.a.: Query Processing, Resource Management, and Approximation in a Data Stream Management System. In: *Proc. 1st Biennial Conf. on Innovative Data Systems Research (CIDR, Asilomar, CA, USA, January 5-8, 2003)*. - Online Proceedings
- Golab, Lukasz ; Özsü, M. Tamer: Data Stream Management Issues - A Survey. Technical Report CS-2003-08, Waterloo 2003
- Golab, Lukasz ; Özsü, M. Tamer: Issues in Data Stream Management. *ACM SIGMOD Record* 32 (June 2003) issue 2, pp. 5-14
- Arasu, Arvind ; Babu, Shivnath ; Widom, Jennifer: The CQL Continuous Query Language : Semantic Foundations and Query Execution. *VLDB Journal* 15 (June 2006) no. 2, pp. 121-142
- Bry, Francois ; Furche, Tim ; Olteanu, Dan: Datenströme. Technical Report PMS-FB-2004-2, München 2004
- Koudas, Nick ; Srivastava, Divesh: Data Stream Query Processing. In: *Proc. VLDB 2003*

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Datenstromsysteme mit Übungen (Prüfungsnummer: 139414)

(englische Bezeichnung: Datastream Systems with Exercises)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0% Prüfungssprache: Deutsch oder Englisch

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Klaus Meyer-Wegener

Modulbezeichnung:	Verteilte Systeme - V+Ü (VS) (Distributed Systems - L+E)		5 ECTS
Modulverantwortliche/r:	Tobias Distler		
Lehrende:	Michael Eischer, Laura Lawniczak, Jürgen Kleinöder, Tobias Distler		
Startsemester: SS 2019	Dauer: 1 Semester	Turnus: jährlich (SS)	
Präsenzzeit: 60 Std.	Eigenstudium: 90 Std.	Sprache: Deutsch	

Lehrveranstaltungen:

Verteilte Systeme (SS 2019, Vorlesung, 2 SWS, Tobias Distler et al.)
 Übungen zu Verteilte Systeme (SS 2019, Übung, 2 SWS, Michael Eischer et al.)

Empfohlene Voraussetzungen: Gute Programmierkenntnisse in Java

Inhalt:

Verteilte Systeme bestehen aus mehreren Rechnern, die über ein Netzwerk miteinander verbunden sind und einen gemeinsamen Dienst erbringen. Obwohl die beteiligten Rechner hierfür in weiten Teilen unabhängig voneinander agieren, erscheinen sie ihren Nutzern gegenüber in der Gesamtheit dabei trotzdem als ein einheitliches System. Die Einsatzmöglichkeiten für verteilte Systeme erstrecken sich über ein weites Spektrum an Szenarien: Von der Zusammenschaltung kleinster Rechenknoten zur Sammlung von Daten im Rahmen von Sensornetzwerken über Steuerungssysteme für Kraftfahrzeuge und Industrieanlagen bis hin zu weltumspannenden, Internet-gestützten Infrastrukturen mit Komponenten in Datenzentren auf verschiedenen Kontinenten.

Ziel dieses Moduls ist es, die sich durch die speziellen Eigenschaften verteilter Systeme ergebenden Problemstellungen zu verdeutlichen und Ansätze zu vermitteln, mit deren Hilfe sie gelöst werden können; Beispiele hierfür sind etwa die Interaktion zwischen heterogenen Systemkomponenten, der Umgang mit erhöhten Netzwerklatenzen sowie die Wahrung konsistenter Zustände über Rechengrenzen hinweg. Gleichzeitig zeigt das Modul auf, dass die Verteiltheit eines Systems nicht nur Herausforderungen mit sich bringt, sondern auf der anderen Seite auch Chancen eröffnet. Dies gilt insbesondere in Bezug auf die im Vergleich zu nicht verteilten Systemen erzielbare höhere Widerstandsfähigkeit eines Gesamtsystems gegenüber Fehlern wie den Ausfällen ganzer Rechner oder sogar kompletter Datenzentren. Ausgehend von den einfachsten, aus nur einem Client und einem Server bestehenden verteilten Systemen, beschäftigt sich die Vorlesung danach mit der deutlich komplexeren Replikation der Server-Seite und behandelt anschließend die Verteilung eines Systems über mehrere, mitunter weit voneinander entfernte geografische Standorte. In allen Abschnitten umfasst die Betrachtung des jeweiligen Themas eine Auswahl aus Grundlagen, im Praxiseinsatz befindlicher Ansätze und Techniken sowie für den aktuellen Stand der Forschung repräsentativer Konzepte.

Im Rahmen der Übungen wird zunächst ein plattformunabhängiges Fernaufrufsystem schrittweise entwickelt und parallel dazu getestet. Als Vorlage und Orientierungshilfe dient dabei das in der Praxis weit verbreitete Java RMI. In den weiteren Übungsaufgaben stehen anschließend klassische Problemstellungen von verteilten Systemen wie fehlertolerante Replikation und verteilte Synchronisation im Mittelpunkt.

Lernziele und Kompetenzen:

Studierende, die das Modul erfolgreich abgeschlossen haben:

- beschreiben charakteristische Merkmale und Eigenschaften verteilter Systeme sowie grundlegende Probleme im Zusammenhang mit ihrer Realisierung.
- untersuchen die Unterschiede zwischen lokalen Methodenaufrufen und Fernmethodenaufrufen.
- vergleichen Ansätze zur Konvertierung von Nachrichten zwischen verschiedenen Datenrepräsentationen.
- konzipieren eine eigene auf Java RMI basierende Anwendung.

- entwickeln ein eigenes Fernaufrufsystem nach dem Vorbild von Java RMI.
 - gestalten ein Modul zur Unterstützung verschiedener Fernaufrufsemantiken (Maybe, Last-of-Many) für das eigene Fernaufrufsystem.
 - beurteilen auf Basis eigener Experimente mit Fehlerinjektionen die Auswirkungen von Störeinflüssen auf verschiedene Fernaufrufsemantiken.
 - klassifizieren Mechanismen zur Bereitstellung von Fehlertoleranz, insbesondere verschiedene Arten der Replikation (aktiv vs. passiv).
 - entwickeln ein eigene aktiv replizierte Anwendung zur Erforschung der mit dieser Replikationsart verbundenen Problemstellungen (z.B. Determinismus).
 - vergleichen verschiedene Konsistenzgarantien georeplizierter Systeme.
 - illustrieren das Problem einer fehlenden gemeinsamen Zeitbasis in verteilten Systemen.
 - erforschen logische Uhren als Mittel zur Reihenfolgebestimmung und Methoden zur Synchronisation physikalischer Uhren.
 - unterscheiden grundlegende Zustellungs- und Ordnungsgarantien beim Multicast von Nachrichten.
 - gestalten ein Protokoll für den zuverlässigen und totalgeordneten Versand von Nachrichten in einer Gruppe von Knoten.
 - entwickeln einen Dienst zur Verwaltung verteilter Sperrobjekte auf Basis von Lamport-Locks.
 - bewerten die Qualität einer aktuellen Publikation aus der Fachliteratur.
 - erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz) und Fehlerquellen bei der Programmierung verteilter Anwendungen.
 - können in Kleingruppen kooperativ arbeiten.
 - können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.
 - reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
 - können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.
-

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

Das Modul ist im Kontext der folgenden Studienfächer/Vertiefungsrichtungen verwendbar:

[1] Medizintechnik (Master of Science)

(Po-Vers. 2018w | Gesamtkonto | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Arts (2 Fächer))", "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Verteilte Systeme - V+Ü (Prüfungsnummer: 890298)

(englische Bezeichnung: Distributed Systems - L+E)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30

Anteil an der Berechnung der Modulnote: 0%

Erstablingung: SS 2019, 1. Wdh.: WS 2019/2020

1. Prüfer: Tobias Distler
