



FRIEDRICH-ALEXANDER  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
TECHNISCHE FAKULTÄT

Masterstudiengang

# Medizintechnik

Modulhandbuch

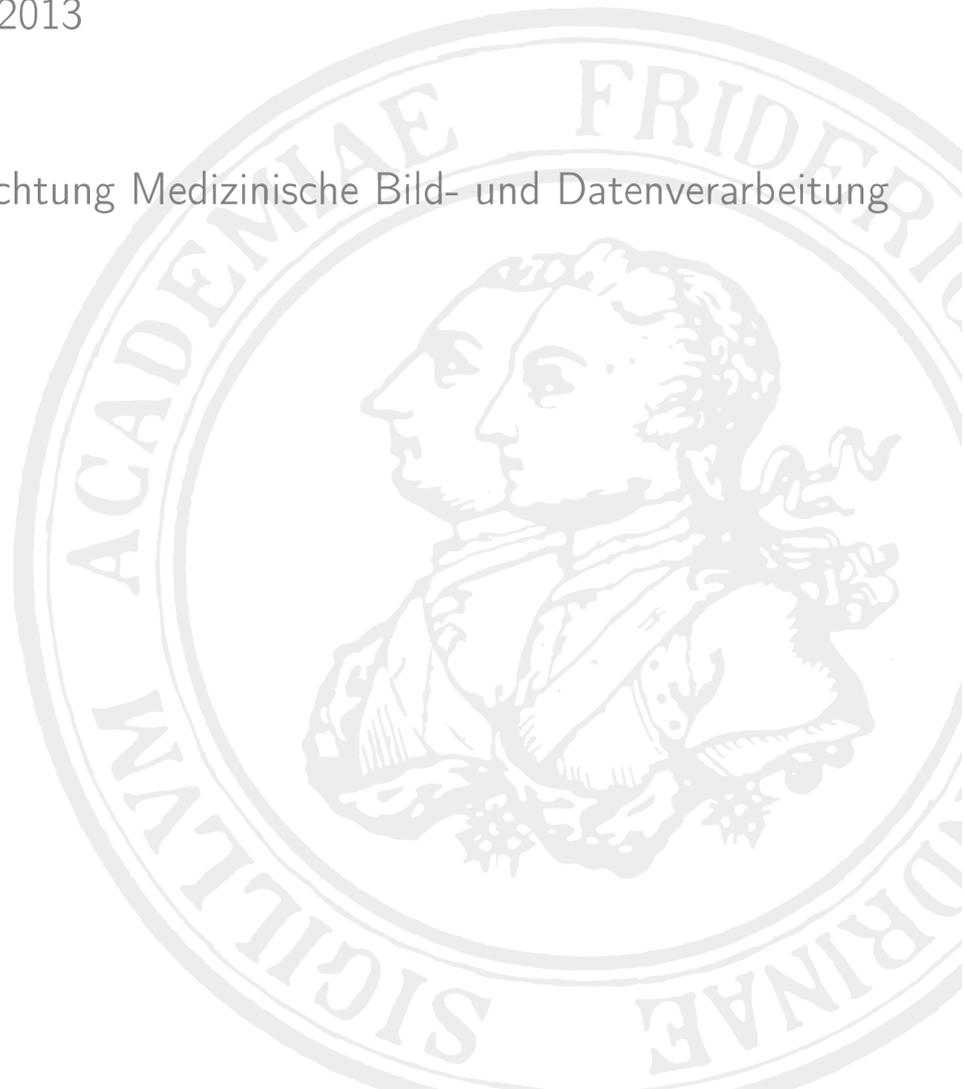
SS 2015

Prüfungsordnungsversion: 2013

Teilauszug Abschnitt

Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung

Modulhandbuch generiert aus *UnivIS*  
Stand: 29.08.2021 21:09





# Medizintechnik (Master of Science)

SS 2015; Prüfungsordnungsversion: 2013

Auflagen Bachelor

M2 Ingenieurwissenschaftliche Kernmodule (BDV)

- Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme, 5 ECTS, Peter Ulbrich, Wolfgang Schröder-Preikschat, SS 2015 4

Schröder-Preikschat, SS 2015

- Informationstheorie, 5 ECTS, Johannes Huber, SS 2015 6
- Systemprogrammierung, 10 ECTS, Wolfgang Schröder-Preikschat, Jürgen Kleinöder, SS 2015, 2 Sem. 8

2015, 2 Sem.

- Konzeptionelle Modellierung, 5 ECTS, Richard Lenz, SS 2015 9
- Digitale Übertragung, 5 ECTS, Robert Schober, SS 2015 11
- Cyber-Physical Systems, 5 ECTS, Torsten Klie, SS 2015 13
- Pattern Analysis (lecture + exercises), 7.5 ECTS, Joachim Hornegger, Marco Bögel, SS 2015 15

2015

- Pattern Analysis (lecture only), 5 ECTS, Joachim Hornegger, SS 2015 18
- Kanalcodierung, 5 ECTS, Clemens Stierstorfer, SS 2015 21
- Statistische Signalverarbeitung, 5 ECTS, Walter Kellermann, SS 2015 24
- Applied Visualization, 5 ECTS, Peter Hastreiter, SS 2015 26
- Grundlagen der Systemprogrammierung, 5 ECTS, Wolfgang Schröder-Preikschat, Jürgen Kleinöder, SS 2015 28

Kleinöder, SS 2015

- Signale und Systeme II, 5 ECTS, André Kaup, Christian Herglotz, Andreas Heindel, SS 2015 30

2015

- Hardware-Software-Co-Design, 5 ECTS, Jürgen Teich, SS 2015 32
- Systemnahe Programmierung in C, 5 ECTS, Jürgen Kleinöder, SS 2015 34

M3 Medizintechnische Kernmodule (BDV)

- Image and Video Compression, 5 ECTS, André Kaup, Wolfgang Schnurrer, SS 2015 36
- Computer Architectures for Medical Applications, 5 ECTS, N.N., Gerhard Wellein, SS 2015 38

2015

- Interventional Medical Image Processing (lecture only), 5 ECTS, Andreas Maier, SS 2015 39
- Interventional Medical Image Processing (lecture + exercises), 7.5 ECTS, Andreas Maier, Matthias Hoffmann, SS 2015 41

Matthias Hoffmann, SS 2015

M5 Medizintechnische Vertiefungsmodule (BDV)

- Human Factors in IT Security, 5 ECTS, Zinaida Benenson, SS 2015 43

- Organic Computing, 5 ECTS, Rolf Wanka, SS 2015 45
- Datenstromsysteme, 5 ECTS, Klaus Meyer-Wegener, SS 2015 47
- Human Computer Interaction, 5 ECTS, Björn Eskofier, SS 2015 50

UnivIS: 29.08.2021 21:09

3

---

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

Modulverantwortliche/r: Wolfgang Schröder-Preikschat

Lehrende: Wolfgang Schröder-Preikschat, Peter Ulbrich

---

Startsemester: SS 2015                      Dauer: 1 Semester                      Turnus: jährlich (SS)

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

---

Lehrveranstaltungen:

Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (SS 2015, Vorlesung, 2 SWS, Peter Ulbrich et al.)  
 Übungen zu Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (SS 2015, Übung, Florian Franzmann 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 eingebettete, 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.

UnivIS: 29.08.2021 21:09

4

- 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. 2013 | Masterprüfung | 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)", "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%

Erstblegung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Wolfgang Schröder-Preikschat

Modulbezeichnung: Informationstheorie (IT) 5 ECTS  
(Information Theory)

Modulverantwortliche/r: Johannes Huber

Lehrende: Johannes Huber

Startsemester: SS 2015 Dauer: 1 Semester Turnus: halbjährlich (WS+SS)

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

**Lehrveranstaltungen:**

Informationstheorie (SS 2015, Vorlesung, 3 SWS, Johannes Huber)

Übungen zur Informationstheorie (SS 2015, Übung, 1 SWS, Christoph Rachinger)

**Inhalt:**

- Grundlegende Definitionen
- Codierung für diskrete Informationsquellen
- Informationstheoretische Beschreibung von Übertragungskanälen
- Informationstheoretische Grundlagen der zuverlässigen Informationsübertragung über gestörte Kanäle durch Blockcodierung
- Kontinuierlich verteilte Zufallsvariablen
- Farbige Rauschen - Signalverzerrungen
- Mehrbenutzerkommunikation

**Lernziele und Kompetenzen:**

Die Studierenden erklären das grundlegende Blockschaltbild für die digitale Nachrichtenübertragung. Bei der informationstheoretischen Modellierung und Analyse der Nachrichtenübertragung verwenden sie diskrete gedächtnislose Informationsquellen, gekoppelte Informationsquellen, das Informationsmaß nach Shannon für diskrete Quellen und das Data Processing Theorem. Sie ermitteln den mittleren Informationsgehalt und bestimmen die relative Entropie. Sie erläutern das Informationsmaß nach Hartley für diskrete Quellen und die Problematik diskreter gedächtnisbehafteter Quellen.

Die Studierenden erklären das Quellencodierungstheorem und analysieren verschiedene Quellencodierverfahren, u.a., mit Codewörtern variabler Länge (variable length code, z.B. Huffman-Code), Blockcodierung für diskrete gedächtnislose Quellen mit Quellenwörtern variabler Länge (Tunstall-Code, LempelZiv) und Blockcodierung für Quellenwörter mit fester Länge.

Die Studierenden bestimmen und analysieren die informationstheoretischen Größen von Übertragungskanälen, insbesondere die Kanalkapazität. Sie untersuchen symmetrische diskrete gedächtnislose Kanäle.

Die Studierenden erläutern die informationstheoretischen Grundlagen der zuverlässigen Informationsübertragung über gestörte Kanäle durch Blockcodierung. Dabei beschreiben und analysieren Sie Blockcodes und deren (optimale) Decodierung (Maximum-a-posteriori- und Maximum-Likelihood- Decodierung) und analysieren die Umkehrung des Kanalcodierungstheorems. Sie schätzen die Wortfehlerwahrscheinlichkeit bei ML-Decodierung nach Bhattacharyya und die Fehlerwahrscheinlichkeit nach Gallager ab. Sie erklären die heuristische Ableitung des Kanalcodierungstheorems basierend auf typischen Sequenzen.

Die Studierenden analysieren die informationstheoretischen Größen in Szenarien mit kontinuierlich verteilten Zufallsvariablen. Dabei untersuchen sie den gedächtnislosen Kanal mit diskreter Eingang und kontinuierlicher reeller Ausgangsvariable sowie den gedächtnislosen Kanal mit kontinuierlichen reellen Ein- und Ausgangsvariablen. Sie bestimmen und analysieren die differentielle Entropie und untersuchen die Kapazität des AWGN-Kanals. Die Studierenden analysieren die Auswirkungen von farbigem Rauschen und Signalverzerrungen auf die Kanalkapazität.

Die Studierenden erläutern die informationstheoretischen Grundlagen der Mehrbenutzerkommunikation. Insbesondere analysieren sie die relevanten Größen von Multiple Access Channel (MAC) und Broadcast Channel (BC).

Als mathematische Hilfsmittel nutzen die Studierenden die Tschebyscheff'sche Ungleichungen, das schwache Gesetz der großen Zahlen, Eigenschaften von konvexen Funktionen (dabei insbesondere die Jensen'sche Ungleichung), die Optimierung mit Berücksichtigung von Nebenbedingungen (Lagrange'sche Optimierung). Sie analysieren die Bedingungen zur Maximierung der Summenkapazität paralleler Kanäle und erklären die Water-Filling-Regel. Literatur:

- Huber, J.: Lecture manuscript;
  - Gallager, R. G.: Information Theory and Reliable Communication, John Wiley & Sons 1968;
  - Cover T., Thomas J.: Elements of Information Theory, John Wiley and Sons, New York, 1991
- 

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "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)", "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 (Written Exam) (Prüfungsnummer: 36001)

(englische Bezeichnung: Information Theory)

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

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Johannes Huber

---

Bemerkungen:

Vorlesung wird auf Englisch gehalten. Eine deutschsprachige Vorlesung folgt im Sommersemester.

---

Modulbezeichnung: Systemprogrammierung (SP)

10 ECTS

---

(Systems Programming)

Modulverantwortliche/r: Wolfgang Schröder-Preikschat

Lehrende: Wolfgang Schröder-Preikschat, Jürgen Kleinöder

---

|                        |                   |                       |
|------------------------|-------------------|-----------------------|
| Startsemester: SS 2015 | Dauer: 2 Semester | Turnus: jährlich (SS) |
|------------------------|-------------------|-----------------------|

|                       |                        |                  |
|-----------------------|------------------------|------------------|
| Präsenzzeit: 120 Std. | Eigenstudium: 180 Std. | Sprache: Deutsch |
|-----------------------|------------------------|------------------|

---

Lehrveranstaltungen:

Systemprogrammierung 1 (SS 2015, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat et al.)  
 Übungen zu Systemprogrammierung 1 (SS 2015, Übung, 2 SWS, Jens Schedel et al.)  
 Rechnerübungen zu Systemprogrammierung 1 und 2 (SS 2015, Übung, 2 SWS, Jens Schedel et al.)  
 Systemprogrammierung 2 (WS 2015/2016, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat et al.)  
 Übungen zu Systemprogrammierung 2 (WS 2015/2016, Übung, 2 SWS, Tobias Klaus et al.)  
 Rechnerübungen zu Systemprogrammierung 1 und 2 (WS 2015/2016, Übung, 2 SWS, Tobias Klaus et al.)  
 Übungen zu Systemprogrammierung 1 (für Wiederholer) (WS 2015/2016, optional, Übung, 2 SWS, Tobias Klaus 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
- entwickeln Systemprogramme auf Basis der Systemaufrufschnittstelle 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. 2013 | Masterprüfung | 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 Science)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Wirtschaftsinformatik (Bachelor of Science)" verwendbar.

---

Modulbezeichnung: Konzeptionelle Modellierung (KonzMod)

5 ECTS

---

(Conceptual Modeling)

Modulverantwortliche/r: Richard Lenz

Lehrende: Richard Lenz

---

|                        |                   |                       |
|------------------------|-------------------|-----------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester | Turnus: jährlich (WS) |
|------------------------|-------------------|-----------------------|

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

---

Lehrveranstaltungen:

Konzeptionelle Modellierung (SS 2015, Vorlesung, 2 SWS, Klaus Meyer-Wegener)

Übungen zu Konzeptionelle Modellierung (SS 2015, Übung, 2 SWS, Andreas Maximilian Wahl et al.)

---

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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "079#72#H", "079#74#H", "Berufspädagogik Technik (Bachelor of Science)", "Computational Engineering (Rechnergestütztes Ingenieurwesen) (Master of Science)", "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)", "Informations- und Kommunikationstechnik (Master of Science)", "International Information Systems (IIS) (Master of Science)", "Linguistische Informatik (Bachelor of Arts (2 Fächer))", "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 (Prüfungsnummer: 31301)

(englische Bezeichnung: Conceptual Modeling)

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

Erstablingung: SS 2015, 1. Wdh.: WS 2016/2017

1. Prüfer: Richard Lenz

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

Modulverantwortliche/r: Robert Schober, Johannes Huber

Lehrende: Robert Schober

Startsemester: SS 2015 Dauer: 1 Semester Turnus: jährlich (SS)

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

Lehrveranstaltungen:

Digitale Übertragung (SS 2015, Vorlesung mit Übung, 3 SWS, Robert Schober et al.) Übungen  
zur Digitalen Übertragung (SS 2015, Übung, 1 SWS, Robert Schober)

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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "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)", "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

Erstablegung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Robert Schober

---

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

|                        |                       |                              |
|------------------------|-----------------------|------------------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester     | Turnus: halbjährlich (WS+SS) |
| Präsenzzeit: 60 Std.   | Eigenstudium: 90 Std. | Sprache: Deutsch             |

#### Lehrveranstaltungen:

- Cyber-Physical Systems (SS 2015, Vorlesung, 2 SWS, Torsten Klie)
- Übung zu Cyber-Physical Systems (SS 2015, 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. Lernziele und Kompetenzen:

- Was sind Cyber-physical Systems? (Definitionen, Abgrenzung zu eingebetteten Systemen, Ubiquitous Computing, etc.)
- Kontrolltheorie und Echtzeitanforderungen
- Selbstorganisationsprinzipien ("Self-X", Autonomie, Verhandlungen)
- Anwendungen für Cyber-physical Systems (Beispiele für existierende oder visionäre zukünftige Anwendungen im Bereich Verkehr, Medizintechnik, u.a.)
- Entwurfsmethoden für Cyber-physical Systems (Modellierung, Programmierung, Model-Integrated Development) Literatur:
- Otto Föllinger Regelungstechnik. Hüthig 1992.
- Rolf Isermann Mechatronische Systeme. Springer 2008.
- Hilmar Jaschek und Holger Voos Grundkurs der Regelungstechnik. Oldenbourg 2010.
- Wolfgang Schneider Praktische Regelungstechnik. Vieweg+Teubner 2008.
- Wayne Wolf Computers as Components: Principles of Embedded Computing System Design. Elsevier 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. 2013 | Masterprüfung | 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 Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)" verwendbar.

#### Studien-/Prüfungsleistungen:

Cyber-Physical Systems (Prüfungsnummer: 44701)

(englische Bezeichnung: Cyber-Physical Systems)

Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30  
Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016  
1. Prüfer: Torsten Klie

---

|                         |  |          |
|-------------------------|--|----------|
| Modulbezeichnung:       | Pattern Analysis (lecture + exercises) (PA-VÜ)<br>(Pattern Analysis (lecture + exercises)) | 7.5 ECTS |
| Modulverantwortliche/r: | Joachim Hornegger  |          |
| Lehrende:               | Joachim Hornegger, Marco Bögel   |          |

|  |                        |                       |
|--|------------------------|-----------------------|
| Startsemester: SS 2015   | Dauer: 1 Semester      | Turnus: jährlich (SS) |
| Präsenzzeit: 80 Std.   | Eigenstudium: 145 Std. | Sprache: Englisch     |
| Lehrveranstaltungen:   |                        |                       |
| Pattern Analysis Exercises (SS 2015, Übung, 1 SWS, Marco Bögel et al.) |                        |                       |
| Pattern Analysis (SS 2015, Vorlesung, 3 SWS, Joachim Hornegger et al.) |                        |                       |

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Pattern Recognition (lecture + exercises)

#### 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 and proof, 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

The accompanying exercises will provide further details on the methods and procedures presented in this lecture with particular emphasis on their application in image processing.

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 Fehlerquadratrate 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 Methoden 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
- Bayessche Netze
- Markov-Zufallsfelder: Definition, Wahrscheinlichkeiten auf ungerichteten Graphen, HammersleyClifford-Theorem und dessen Beweis, 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

In den Übungen werden die in der Vorlesung vorgestellten Verfahren vertieft und auf konkrete Probleme in der Bildverarbeitung angewendet.

Lernziele und Kompetenzen:

The students

- explain the discussed methods for classification, prediction, and analysis of patterns
- define regression and classification tasks as optimization problems
- analyse optimization problems
- adapt (multimodal) objective functions and define new objective functions for regression and classification problems
- understand joint discrete and continuous optimization with continuous and discrete variables and relaxation by transformation of discrete variables into continuous variables, e.g. from  $\{0,1\}$  to  $[0,1]$
- compare 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
- get to know practical applications and apply the presented algorithms to problems in practice Die Studierenden
- erläutern die behandelten Methoden zur Klassifikation, Vorhersage und Analyse von Mustern,
- formulieren Regressions- und Klassifikationsproblemen als Optimierungsaufgaben,
- analysieren Optimierungsprobleme,
- passen (multimodale) Zielfunktionen an und entwickeln für Regressions- und Klassifikationsprobleme neue Zielfunktionen,
- verstehen simultane diskrete und kontinuierliche Optimierung und Relaxation durch Transformation diskreter in kontinuierliche Variable, also z.B. von  $\{0,1\}$  nach  $[0,1]$ ,
- vergleichen Methoden zur Dichteschätzung und wählen für eine vorgegebene Merkmalsmenge und in Abhängigkeit von der 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 bei statistischen Abhängigkeiten,

- implementieren vorgestellte Verfahren in MatLab,
- lernen praktische Anwendungen kennen und wenden die vorgestellten Algorithmen auf konkrete Problemstellungen an.

Literatur:

Christopher Bishop, Pattern Recognition and Machine Learning, Springer Verlag, Heidelberg, 2006

Richard O. Duda, Peter E. Hart und David G. Stork, Pattern Classification, Second Edition, 2004

Trevor Hastie, Robert Tibshirani und Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, Springer Verlag, 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

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

---

Studien-/Prüfungsleistungen:

Pattern Analysis (Vorlesung mit Übung) (Prüfungsnummer: 263497)

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

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Joachim Hornegger

---

|                         |   |                       |                       |
|-------------------------|---|-----------------------|-----------------------|
| Modulbezeichnung:       | Pattern Analysis (lecture only) (PA-V)<br>(Pattern Analysis (lecture only)) |                       | 5 ECTS                |
| Modulverantwortliche/r: | Joachim Hornegger   |                       |                       |
| Lehrende:               | Joachim Hornegger   |                       |                       |
| Startsemester:          | SS 2015   | Dauer: 1 Semester     | Turnus: jährlich (SS) |
| Präsenzzeit:            | 60 Std.   | Eigenstudium: 90 Std. | Sprache: Englisch     |
| Lehrveranstaltungen:    | Pattern Analysis (SS 2015, Vorlesung, 3 SWS, Joachim Hornegger et al.)      |                       |                       |

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Pattern Recognition (lecture only)

#### 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 and proof, 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 und dessen Beweis, 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
- analyse optimization problems
- adapt (multimodal) objective functions and define new objective functions for regression and classification problems
- understand joint discrete and continuous optimization with continuous and discrete variables and relaxation by transformation of discrete variables into continuous variables, e.g. from  $\{0,1\}$  to  $[0,1]$
- compare 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
- erläutern die behandelten Methoden zur Klassifikation, Vorhersage und Analyse von Mustern
- formulieren Regressions- und Klassifikationsproblemen als Optimierungsaufgaben
- analysieren Optimierungsprobleme
- passen (multimodale) Zielfunktionen an und entwickeln für Regressions- und Klassifikationsprobleme neue Zielfunktionen
- verstehen simultane diskrete und kontinuierliche Optimierung mit kontinuierlichen und diskreten Variablen und Relaxation durch Transformation diskreter in kontinuierliche Variablen, also z.B. von  $\{0,1\}$  nach  $[0,1]$
- vergleichen Methoden zur Dichteschätzung und wählen für eine vorgegebene Merkmalsmenge und in Abhängigkeit von der 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 bei statistischen Abhängigkeiten

Literatur: Christopher Bishop, Pattern Recognition and Machine Learning, Springer Verlag, Heidelberg, 2006

Richard O. Duda, Peter E. Hart und David G. Stork, Pattern Classification, Second Edition, 2004

Trevor Hastie, Robert Tibshirani und Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, Springer Verlag, 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

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

---

Studien-/Prüfungsleistungen:

Pattern Analysis (Prüfungsnummer: 41201)

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

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Joachim Hornegger

---

|                         |   |        |
|-------------------------|---|--------|
| Modulbezeichnung:       | Kanalcodierung (KaCo)<br>(Channel Coding) | 5 ECTS |
| Modulverantwortliche/r: | Clemens Stierstorfer                      |        |
| Lehrende:               | Clemens Stierstorfer                      |        |

|                        |                       |                                |
|------------------------|-----------------------|--------------------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester     | Turnus: halbjährlich (WS+SS)   |
| Präsenzzeit: 60 Std.   | Eigenstudium: 90 Std. | Sprache: Deutsch oder Englisch |

#### Lehrveranstaltungen:

Tutorial for Channel Coding (SS 2015, Übung, 1 SWS, Clemens Stierstorfer et al.) Channel Coding (SS 2015, Vorlesung, 3 SWS, Clemens Stierstorfer)

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

Nachrichtentechnische Systeme  
Informationstheorie

#### 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-Solomon-Codes gemäß der BCH-Schranke. Sie verstehen die Grundlagen der Decodierung von BCH- und Reed-Solomon-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. 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "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)", "Informations- und Kommunikationstechnik (Bachelor of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Wirtschaftsingenieurwesen (Master of Science)" verwendbar.

---

Studien-/Prüfungsleistungen:

Kanalcodierung (Prüfungsnummer: 62701)

(englische Bezeichnung: Channel Coding)

Prüfungsleistung, Klausur, Dauer (in Minuten): 90

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

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016, 2. Wdh.: SS 2016 1.

Prüfer: Clemens Stierstorfer

---

---

|                         |   |        |
|-------------------------|---|--------|
| Modulbezeichnung:       | Statistische Signalverarbeitung (STASIP)<br>(Statistical Signal Processing) | 5 ECTS |
| Modulverantwortliche/r: | Walter Kellermann   |        |
| Lehrende:               | Walter Kellermann   |        |

---

|                        |                       |                       |
|------------------------|-----------------------|-----------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester     | Turnus: jährlich (SS) |
| Präsenzzeit: 60 Std.   | Eigenstudium: 90 Std. | Sprache: Englisch     |

---

**Lehrveranstaltungen:**

Statistische Signalverarbeitung (SS 2015, Vorlesung, 3 SWS, Walter Kellermann)

Ergänzungen und Übungen zur statistischen Signalverarbeitung (SS 2015, Übung, 1 SWS, Stefan Meier)

**Empfohlene Voraussetzungen:**

Module ‚Signale und Systeme I‘ und ‚Signale und Systeme II‘, ‚Digitale Signalverarbeitung‘ oder gleichwertige

**Inhalt:**

Zeitdiskrete Zufallsprozesse im Zeit- und Frequenzbereich

Zufallsvariablen (ZVn), Wahrscheinlichkeitsverteilungen und -dichten, Erwartungswerte; Transformation von ZVn; Vektoren normalverteilter ZVn; zeitdiskrete Zufallsprozesse (ZPe): Wahrscheinlichkeitsverteilungen und -dichten, Erwartungswerte, Stationarität, Zyklstationarität, Ergodizität, Korrelationsfunktionen und -matrizen, Spektraldarstellungen; ‚Principal Component Analysis‘, Karhunen-Loeve Transformation;

Schätztheorie

Schätzkriterien; Prädiktion; klassische und Bayes'sche Parameterschätzung (inkl. MMSE, Maximum Likelihood, Maximum A Posteriori); Cramer-Rao-Schranke

Lineare Signalmodelle

Parametrische Modelle (Cepstrale Zerlegung, Paley-Wiener Theorem, Spektrale Glattheit); Nichtparametrische Modelle: ‚Allpole‘-/‚Allzero‘-/‚Pole-zero‘-(AR/MA/ARMA) Modelle; ‚Lattice‘-Strukturen, Yule-Walker Gleichungen, PARCOR-Koeffizienten, Cepstraldarstellungen; Signalschätzung Überwachte Signalschätzung, Problemklassen; Orthogonalitätsprinzip, MMSE-Schätzung, lineare MMSE-Schätzung für Gaußprozesse; Optimale FIR-Filter; Lineare Optimalfilter für stationäre Prozesse; Prädiktion und Glättung; Kalman-Filter; optimale Multikanalfilterung (Wiener-Filter, LCMV, MVDR, GSC);

Adaptive Filterung

Gradientenverfahren; LMS-, NLMS-, APA- und RLS-Algorithmus und Ihr Konvergenzverhalten;

**Lernziele und Kompetenzen:**

Die Studierenden

- analysieren die statistischen Eigenschaften von Zufallsvariablen, -vektoren und stochastischen Prozessen mittels Wahrscheinlichkeitsdichten und Erwartungswerten, bzw. Korrelationsfunktionen, Korrelationsmatrizen und deren Frequenzbereichsdarstellungen
- kennen die spezielle Rolle der Gaußverteilung und ihre Auswirkungen auf die Eigenschaften von Zufallsvariablen, -vektoren und Prozessen
- verstehen die Unterschiede klassischer und Bayes'scher Schätzung, entwerfen und analysieren MMSE- und ML-Schätzer für spezielle Schätzprobleme, insbesondere zur Signalschätzung
- analysieren und evaluieren lineare MMSE-optimale Schätzer (ein- und vielkanalige Wiener-Filter und Kalman-Filter) für direkte und inverse überwachte Schätzprobleme;
- evaluieren adaptive Filter zur Identifikation optimaler linearer Signalschätzer Literatur:

- A. Papoulis, S. Pillai: Probability, Random Variables and Stochastic Processes; McGraw-Hill, 2002 (englisch)
- D. Manolakis, V. Ingle, S. Kogon: Statistical and Adaptive Signal Processing; McGraw-Hill, 2005 (englisch)

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "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)", "Informations- und Kommunikationstechnik (Master of Science)" verwendbar.

Studien-/Prüfungsleistungen:

Statistische Signalverarbeitung\_ (Prüfungsnummer: 64301)

(englische Bezeichnung: Statistical Signal Processing)

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

Erläuterungen:

Durch Abgabe der Übungsblätter können Bonuspunkte für die Klausur erarbeitet werden. Wird die Klausur ohne Bonus nicht bestanden, darf der Bonus nicht angerechnet werden. Der Bonus verfällt dann auch für die Wiederholungsklausur. Es gilt folgende Abbildung (bei 100 erreichbaren Punkten in der Klausur): weniger als 4 Übungspunkte = 0 Bonuspunkte in der Klausur, 4 bis 4,5 Übungspunkte = 4 Bonuspunkte in der Klausur, 5 bis 5,5 Übungspunkte = 5 Bonuspunkte in der Klausur, 6 bis 6,5 Übungspunkte = 6 Bonuspunkte in der Klausur, 7 Übungspunkte = 7 Bonuspunkte in der Klausur.

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Walter Kellermann

|                         |   |                       |
|-------------------------|---|-----------------------|
| Modulbezeichnung:       | Applied Visualization (AppVis)<br>(Applied Visualization) | 5 ECTS                |
| Modulverantwortliche/r: | Roberto Grosso  |                       |
| Lehrende:               | Peter Hastreiter  |                       |
| Startsemester: SS 2015  | Dauer: 1 Semester   | Turnus: jährlich (SS) |
| Präsenzzeit: 60 Std.    | Eigenstudium: 90 Std.                                     | Sprache: Deutsch      |

Lehrveranstaltungen:

Applied Visualization (SS 2015, Vorlesung, 2 SWS, Roberto Grosso)

Tutorials to Applied Visualization (SS 2015, Übung, 2 SWS, Christian Siegl et al.)

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.

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. 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. 2013 | Masterprüfung | 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 Science)", "Informatik (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)", "Mechatronik (Bachelor of Science)", "Mechatronik (Master of Science)", "Physische Geographie (Bachelor of Science)" verwendbar.

---

Studien-/Prüfungsleistungen:

Applied Visualization (Lecture and Tutorials) (Prüfungsnummer: 37211)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60 Anteil

an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

Übung: 50% der schriftlichen Aufgaben, Modulnote durch Klausur über 60 Minuten

Erstabllegung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Roberto Grosso

1. Prüfer: Günther Greiner

Applied Visualization (Lecture and Tutorials) (Prüfungsnummer: 70701)

Prüfungsleistung, Klausur, Dauer (in Minuten): 60 Anteil

an der Berechnung der Modulnote: 100% weitere

Erläuterungen:

Übung: 50% der schriftlichen Aufgaben, Modulnote durch Klausur über 60 Minuten

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Roberto Grosso

1. Prüfer: Günther Greiner

---

---

|                         |  |        |
|-------------------------|--|--------|
| Modulbezeichnung:       | Grundlagen der Systemprogrammierung (GSP)<br>(Fundamentals of Systems Programming) | 5 ECTS |
| Modulverantwortliche/r: | Wolfgang Schröder-Preikschat   |        |
| Lehrende:               | Jürgen Kleinöder, Wolfgang Schröder-Preikschat                                     |        |

---

|                        |                       |                       |
|------------------------|-----------------------|-----------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester     | Turnus: jährlich (SS) |
| Präsenzzeit: 60 Std.   | Eigenstudium: 90 Std. | Sprache: Deutsch      |

---

Lehrveranstaltungen:

- Systemprogrammierung 1 (SS 2015, Vorlesung, 2 SWS, Wolfgang Schröder-Preikschat et al.)
  - Übungen zu Systemprogrammierung 1 (SS 2015, Übung, 2 SWS, Jens Schedel et al.)
  - Rechnerübungen zu Systemprogrammierung 1 und 2 (SS 2015, Übung, 2 SWS, Jens Schedel 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 Systemaufrufschnittstelle 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "079#72#H", "079#74#H", "Informatik (1. Staatsprüfung für das Lehramt an Gymnasien)", "Informatik (1. Staatsprüfung für das Lehramt an Realschulen)", "Informatik (Bachelor of Arts (2 Fächer))", "Mathematik (Bachelor of Science)", "Technomathematik (Bachelor of Science)" verwendbar.

---

Studien-/Prüfungsleistungen:

Grundlagen der Systemprogrammierung (Vorlesung mit Übungen) (Prüfungsnummer: 31811)

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 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Jürgen Kleinöder

---

|                         |  |        |
|-------------------------|--|--------|
| Modulbezeichnung:       | Signale und Systeme II (SISY II)<br>(Signals and Systems II) | 5 ECTS |
| Modulverantwortliche/r: | André Kaup   |        |
| Lehrende:               | Christian Herglotz, André Kaup, Andreas Heindel              |        |

|                        |                       |                       |
|------------------------|-----------------------|-----------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester     | Turnus: jährlich (SS) |
| Präsenzzeit: 60 Std.   | Eigenstudium: 90 Std. | Sprache: Deutsch      |

#### Lehrveranstaltungen:

Signale und Systeme II (SS 2015, Vorlesung, 2,5 SWS, André Kaup)  
 Übung zu Signale und Systeme II (SS 2015, Übung, 1,5 SWS, Christian Herglotz)  
 Tutorium zu Signale und Systeme II (SS 2015, optional, Tutorium, 1 SWS, Andreas Heindel)

#### 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "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)", "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%

Erstablesung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: André Kaup

---

|                         |  |                       |
|-------------------------|--|-----------------------|
| Modulbezeichnung:       | Hardware-Software-Co-Design (HSCD-VU)<br>(Hardware-Software-Co-Design) | 5 ECTS                |
| Modulverantwortliche/r: | Jürgen Teich   |                       |
| Lehrende:               | Jürgen Teich   |                       |
| Startsemester: SS 2015  | Dauer: 1 Semester  | Turnus: jährlich (SS) |
| Präsenzzeit: 60 Std.    | Eigenstudium: 90 Std.  | Sprache: Deutsch      |

---

Lehrveranstaltungen:

Hardware-Software-Co-Design (SS 2015, Vorlesung, 2 SWS, Jürgen Teich)

Übungen zu Hardware-Software-Co-Design (SS 2015, Übung, 2 SWS, Sascha Roloff)

---

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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Bachelor 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: 100%

Erstabelleung: SS 2015, 1. Wdh.: WS 2015/2016

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:       | Systemnahe Programmierung in C (SPiC)<br>(System-Level Programming in C) | 5 ECTS |
| Modulverantwortliche/r: | Jürgen Kleinöder   |        |
| Lehrende:               | Jürgen Kleinöder   |        |

|                        |                       |                       |
|------------------------|-----------------------|-----------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester     | Turnus: jährlich (SS) |
| Präsenzzeit: 60 Std.   | Eigenstudium: 90 Std. | Sprache: Deutsch      |

**Lehrveranstaltungen:**

Systemnahe Programmierung in C (SS 2015, Vorlesung, 2 SWS, Jürgen Kleinöder et al.)  
 Übungen zu Systemnahe Programmierung in C (SS 2015, Übung, 2 SWS, Peter Wägemann et al.)  
 Rechnerübungen zu Systemnahe Programmierung in C (SS 2015, Übung, Peter Wägemann et al.)  
 Wiederholungsübungen im WS  
 Übungen zu Systemnahe Programmierung in C (für Wiederholer) (WS 2015/2016, optional, Übung, 2 SWS, Sebastian Maier et al.)  
 Rechnerübungen zu Systemnahe Programmierung in C (WS 2015/2016, optional, Übung, 2 SWS, Sebastian Maier 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

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

Modulbezeichnung: Image and Video Compression (IVC) 5 ECTS  
(Image and Video Compression)

Modulverantwortliche/r: André Kaup

Lehrende: Wolfgang Schnurrer, André Kaup

Startsemester: SS 2015 Dauer: 1 Semester Turnus: jährlich (SS)

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

Lehrveranstaltungen:

Image and Video Compression (SS 2015, Vorlesung, 3 SWS, André Kaup)

Übung Image and Video Compression (SS 2015, Übung, 1 SWS, Wolfgang Schnurrer)

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 Fehler, entropiecodiert, eingebettet)
- 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 Blockschnittbilder und Wirkungsweisen hybrider Coder und Decoder für Videosignale
- kennen die maßgeblichen internationalen Standards aus ITU und MPEG zur Bild- und Videokompression 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "123#67#H", "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)", "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 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: André Kaup

---

|                          |   |        |
|--------------------------|---|--------|
| <b>Modulbezeichnung:</b> | Computer Architectures for Medical Applications (CAMA)<br>(Computer Architectures for Medical Applications) | 5 ECTS |
|--------------------------|---|--------|

|                                |             |
|--------------------------------|-------------|
| <b>Modulverantwortliche/r:</b> | Dietmar Fey |
|--------------------------------|-------------|

|                  |                       |
|------------------|-----------------------|
| <b>Lehrende:</b> | Gerhard Wellein, N.N. |
|------------------|-----------------------|

---

|                               |                          |                              |
|-------------------------------|--------------------------|------------------------------|
| <b>Startsemester:</b> SS 2015 | <b>Dauer:</b> 1 Semester | <b>Turnus:</b> jährlich (SS) |
|-------------------------------|--------------------------|------------------------------|

|                             |                              |                          |
|-----------------------------|------------------------------|--------------------------|
| <b>Präsenzzeit:</b> 60 Std. | <b>Eigenstudium:</b> 90 Std. | <b>Sprache:</b> Englisch |
|-----------------------------|------------------------------|--------------------------|

---

**Lehrveranstaltungen:**

Computer Architectures for Medical Applications (SS 2015, Vorlesung, 2 SWS, Dietmar Fey et al.)  
Übung zu Computer Architectures for Medical Applications (SS 2015, Übung, Johannes Hofmann 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

---

**Studien-/Prüfungsleistungen:**

Computer Architectures for Medical Applications (Vorlesung mit Übung) (Prüfungsnummer: 41451)

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

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Dietmar Fey

---

---

Modulbezeichnung: Interventional Medical Image Processing (lecture only) 5 ECTS  
 (IMIP-V)  
 (Interventional Medical Image Processing (lecture only))

Modulverantwortliche/r: Andreas Maier

Lehrende: Andreas Maier

---

Startsemester: SS 2015 Dauer: 1 Semester Turnus: jährlich (SS)

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

---

Lehrveranstaltungen:

Interventional Medical Image Processing (SS 2015, Vorlesung, 3 SWS, Andreas Maier)

---

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Diagnostic  
 Medical Image Processing (lecture + exercises)

---

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, Epipolar geometrie 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. 2013 | Masterprüfung | 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 Science)", "Informatik (Master of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

---

Studien-/Prüfungsleistungen:

Interventional Medical Image Processing (Lecture) (Prüfungsnummer: 41401)

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

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

Erstablagerung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Andreas Maier

---

Modulbezeichnung: Interventional Medical Image Processing (lecture + 7.5 ECTS excercises)  
(IMIP-VÜ)  
(Interventional Medical Image Processing (lecture + excercises))

Modulverantwortliche/r: Andreas Maier

Lehrende: Matthias Hoffmann, Andreas Maier

Startsemester: SS 2015                      Dauer: 1 Semester                      Turnus: jährlich (SS)

Präsenzzeit: 80 Std.                      Eigenstudium: 145 Std.                      Sprache: Deutsch

Lehrveranstaltungen:

Interventional Medical Image Processing (SS 2015, Vorlesung, 3 SWS, Andreas Maier)

Interventional Medical Image Processing Exercises (SS 2015, Übung, 1 SWS, Matthias Hoffmann)

Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: Diagnostic  
Medical Image Processing (lecture + excercises)

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 pre-processing algorithms such as scatter correction for xray 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.
- are able to develop non-rigid registration methods using variational calculus and different regularizers.
- adopt algorithms to new domains by appropriate modifications.
- implement the algorithms that were covered in the exercises.

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.
  - implementieren Algorithmen aus der Übung.
- 

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2013 | Masterprüfung | 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 Science)", "Informatik (Master of Science)", "Mathematik (Bachelor of Science)", "Medizintechnik (Bachelor of Science)" verwendbar.

---

Studien-/Prüfungsleistungen:

Interventional Medical Image Processing (Lecture and Exercises) (Prüfungsnummer: 37601)

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

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Andreas Maier

---

---

Modulbezeichnung: Human Factors in IT Security (HumITSec) 5 ECTS  
 (Human Factors in IT Security)

Modulverantwortliche/r: Zinaida Benenson

Lehrende: Zinaida Benenson

---

Startsemester: SS 2015 Dauer: 1 Semester Turnus: jährlich (SS)

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

---

Lehrveranstaltungen:

Human Factors in IT Security (SS 2015, Vorlesung, 2 SWS, Zinaida Benenson)

Human Factors in IT Security - Übung (SS 2015, Übung, 2 SWS, Lena Reinfelder)

---

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) will be very helpful. 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 exercise will be divided in two parts:
  - (1) After each lecture, the students will receive a homework assignment consisting of practical exercises.
  - (2) The students will be divided into groups, and each group will prepare a 30-minutes long presentation with the following discussion for the class on a given topic. Materials such as papers and key discussion questions will be provided. Participation in an exercise group is a prerequisite for participation in the exam.

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, of the human-in-the-loop framework and of usercentered 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. 2013 | Masterprüfung | 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)" verwendbar.

---

Studien-/Prüfungsleistungen:

Human Factors in IT Security (Prüfungsnummer: 900203)

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

Anteil an der Berechnung der Modulnote: 100%

Erstabelleung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Zinaida Benenson

---

|                         |   |        |
|-------------------------|---|--------|
| Modulbezeichnung:       | Organic Computing (OC)<br>(Organic Computing) | 5 ECTS |
| Modulverantwortliche/r: | Rolf Wanka                                    |        |
| Lehrende:               | Rolf Wanka                                    |        |

---

|                        |                       |                                |
|------------------------|-----------------------|--------------------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester     | Turnus: jährlich (SS)          |
| Präsenzzeit: 60 Std.   | Eigenstudium: 90 Std. | Sprache: Deutsch oder Englisch |

---

Lehrveranstaltungen:

- Organic Computing (SS 2015, Vorlesung, 2 SWS, Rolf Wanka)
- Übungen zu Organic Computing (SS 2015, 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. 2013 | Masterprüfung | 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 Science)", "Informatik (Master

of Science)", "Informations- und Kommunikationstechnik (Master of Science)", "Mathematik (Bachelor of Science)" verwendbar.

---

Studien-/Prüfungsleistungen:

Organic Computing (Vorlesung mit Übung) (Prüfungsnummer: 39701)

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

Anteil an der Berechnung der Modulnote: 100%

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Rolf Wanka

---

Bemerkungen:

Auch für CE

---

Modulbezeichnung: Datenstromsysteme (DSSmUe) 5 ECTS  
 (Datastream Systems)

Modulverantwortliche/r: Klaus Meyer-Wegener

Lehrende: Klaus Meyer-Wegener

---

|                        |                   |                       |
|------------------------|-------------------|-----------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester | Turnus: jährlich (SS) |
|------------------------|-------------------|-----------------------|

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

---

Lehrveranstaltungen:

Datenstromsysteme (SS 2015, Vorlesung, 2 SWS, Klaus Meyer-Wegener)

Übungen zu Datenstromsysteme (SS 2015, Übung, 2 SWS, Sebastian Herbst)

---

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 ; ÖZSU, M. Tamer: Data Stream Management Issues - A Survey. Technical Report CS-2003-08, Waterloo 2003
  - Golab, Lukasz ; ÖZSU, 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. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

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

---

Studien-/Prüfungsleistungen:

Datenstromsysteme mit Übungen (Prüfungsnummer: 139414)

(englische Bezeichnung: Datastream Systems with Exercises)

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

Erstablingung: SS 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Klaus Meyer-Wegener

---

---

Modulbezeichnung: Human Computer Interaction (HCI) 5 ECTS  
 (Human Computer Interaction)

Modulverantwortliche/r: Björn Eskofier

Lehrende: Björn Eskofier

---

|                        |                       |                                |
|------------------------|-----------------------|--------------------------------|
| Startsemester: SS 2015 | Dauer: 1 Semester     | Turnus: jährlich (SS)          |
| Präsenzzeit: 60 Std.   | Eigenstudium: 90 Std. | Sprache: Deutsch oder Englisch |

---

Lehrveranstaltungen:

Human Computer Interaction (SS 2015, Vorlesung, 3 SWS, Björn Eskofier)

Human Computer Interaction Exercises (SS 2015, Übung, 1 SWS, Stefan Gradl)

---

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 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.
- 

Verwendbarkeit des Moduls / Einpassung in den Musterstudienplan:

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

[1] Medizintechnik (Master of Science)

(Po-Vers. 2013 | Masterprüfung | Studienrichtung Medizinische Bild- und Datenverarbeitung)

Dieses Modul ist daneben auch in den Studienfächern "Informatik (Bachelor of Science)", "Informatik (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: 100% 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 2015, 1. Wdh.: WS 2015/2016

1. Prüfer: Björn Eskofier

---

Organisatorisches:

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