

**Modulhandbuch**  
**Master-Studiengang**  
**„Angewandte Informatik“**

**Ruprecht-Karls-Universität Heidelberg**  
**Fakultät für Mathematik und Informatik**

**Fassung vom 18.04.2018 zur Prüfungsordnung vom 22.07.2010**  
**(mit letzter Änderung vom 07.02.2013)**

**Studienform:** Vollzeit

**Art des Studiengangs:** Konsekutiv

**Regelstudienzeit:** 4 Semester

**Anzahl der im Studiengang zu erwerbenden Leistungspunkte:** 120

**Studienstandort:** Heidelberg

**Anzahl der Studienplätze:** Keine Zulassungsbeschränkung

**Gebühren/Beiträge:** Gemäß allgemeiner Regelung der Universität Heidelberg

# Präambel

## Einordnung und Gesamtdarstellung des Master-Studiengangs Angewandte Informatik

Anknüpfend an ihr Leitbild und ihre Grundordnung verfolgt die Universität Heidelberg in ihren Studiengängen fachliche, fachübergreifende und berufsfeldbezogene Ziele in der umfassenden akademischen Bildung und für eine spätere berufliche Tätigkeit ihrer Studierenden. Das daraus folgende Kompetenzprofil wird als für alle Disziplinen gültiges Qualifikationsprofil in den Modulhandbüchern aufgenommen und in den spezifischen Qualifikationszielen sowie dem Curriculum und Modulen des Master-Studiengangs Angewandte Informatik umgesetzt:

- Entwicklung von fachlichen Kompetenzen mit ausgeprägter Forschungsorientierung;
- Entwicklung transdisziplinärer Dialogkompetenz;
- Aufbau von praxisorientierter Problemlösungskompetenz;
- Entwicklung von personalen und Sozialkompetenzen;
- Förderung der Bereitschaft zur Wahrnehmung gesellschaftlicher Verantwortung auf der Grundlage der erworbenen Kompetenzen.

Der Master-Studiengang Angewandte Informatik wird von der Fakultät für Mathematik und Informatik getragen. Der Master-Studiengang ist forschungsorientiert. Er vertieft und verbreitert die Fachkenntnisse, befähigt zum selbstständigen wissenschaftlichen Arbeiten, legt die Voraussetzungen zur Weiterentwicklung des Faches und bereitet auf eine anspruchsvolle Berufstätigkeit oder eine Promotion vor. AbsolventInnen sind qualifiziert für eigenverantwortliche und leitende Tätigkeiten. Der Master-Studiengang erlaubt eine sehr freie Gestaltung des Studiums um sowohl einen frühen Einstieg in forschungsnahen als auch innovative praktische Themengebiete zu ermöglichen. Weiterhin ermöglicht er eine Vertiefung in Themengebieten der Informatik, die insbesondere in den Natur-, Lebens- und Geisteswissenschaften zur Anwendung kommen.

### Qualifikationsziele des Master-Studiengangs Angewandte Informatik

Die Absolventen des Studiengangs sollen nach Abschluss des Studiums folgende grundlegende Kompetenzen überfachlicher Art im Kontext der Informatik besitzen.

- Sie besitzen Problemlösungskompetenz und beherrschen die Wissensanwendung im Bereich der Informatik und zusätzlich in einem breiteren fachlichen Zusammenhang oder verwandten Disziplinen. Darüber hinaus sind sie in der Lage, diese Kompetenzen auch in neuen, unvertrauten Situationen anzuwenden.
- Sie haben die Kompetenz zur Arbeit in einem Team sowie zur Übernahme von herausgehobener Verantwortung in einem Team (Teamleitung).

- Sie können eigene Schlussfolgerungen auf aktuellem Stand von Forschung und Anwendung vermitteln und sich fachbezogen auf wissenschaftlichem Niveau austauschen.
- Sie besitzen die Kompetenz zu selbständiger Informationssammlung, Urteilsfähigkeit und selbständiger Aneignung von Wissen im Bereich der Informatik sowie verwandten Disziplinen. Insbesondere sind sie befähigt zur Rezeption und Interpretation von Forschungsliteratur und zur Bewertung alternativer Lösungsansätze in fachlicher Hinsicht im Bereich der Informatik als auch fachübergreifend.
- Darüber hinaus beherrschen sie den effektiven Umgang mit komplexen Fachproblemen und Situationen, verfügen über Entscheidungsfähigkeit, sowie können selbständig forschungs- oder anwendungsorientierte Projekte durchführen.
- Sie können in fachlichen Angelegenheiten mündlich und schriftlich in deutscher und englischer Sprache effektiv kommunizieren.

In fachlicher Hinsicht beherrschen die AbsolventInnen des Master-Studiengangs Angewandte Informatik insbesondere die Kompetenzen der Bachelor-AbsolventInnen, im Detail:

- Sie verfügen über Kenntnisse der Praktischen, Theoretischen, Technischen und Angewandten Informatik und der Methoden der Mathematik und können diese zur Lösung von konkreten informatischen Problemen anwenden.
- Sie können eine informatische Aufgabe eigenverantwortlich planen, durchführen, dokumentieren und präsentieren.
- Sie können innerhalb einer vorgegebenen Frist ein Problem aus dem Bereich der Informatik mit wissenschaftlichen Methoden bearbeiten und Lösungsvorschläge entwickeln und präsentieren.
- Sie beherrschen wissenschaftlich fundierte Methoden der Programmierung und können diese in Projekten praktisch anwenden. Dazu gehören die wissenschaftlichen Methoden des Entwurfs, der Implementierung und des Debuggens von Software.
- Sie kennen die Konzepte für den Entwurf und die Analyse von effizienten Algorithmen und können diese bei der Erstellung von Software selbständig einsetzen.
- Sie kennen die Grundlagen der Verwendung von Betriebssystemen und Verwaltung von Ressourcen und sind in der Lage, diese Kenntnisse bei dem Entwurf, der Umsetzung und der Optimierung von informatischen Systemen einzusetzen.
- Sie kennen die Probleme und Bedeutung der Verlässlichkeit in modernen Computersystemen und Rechenverbunden und können diese Kenntnisse bei der Planung, Umsetzung als auch der Pflege solcher Systeme praktisch berücksichtigen.

Zusätzlich beherrschen die AbsolventInnen des Master-Studiengangs Angewandte Informatik folgende fachliche Qualifikationen über die Lernergebnisse des Bachelorstudiengangs hinaus.

- Sie sind in der Lage, umfangreiche informatische Systeme unter vorgegebenen technischen und ökonomischen Randbedingungen selbständig zu planen, zu entwerfen und zu evaluieren, sowie dazugehörige Softwareprojekte zu leiten.
- Sie verfügen über vertiefte Kenntnisse in einem oder mehreren Spezialgebieten der Informatik wie Datenanalyse, Requirements Engineering, Verteilte Systeme, Informationssysteme, und können diese Kenntnisse bei dem Entwurf und der Entwicklung von informatischen Systemen praktisch einsetzen.
- Sie können komplexe informatische Systeme in abstrakte Komponenten (Software und Hardware) zerlegen und dafür Realisierungsmöglichkeiten gemäß vorgegeben Randbedingungen ermitteln und bewerten, sowie diese Realisierung planen und umsetzen.
- Sie sind in der Lage, sich selbständig in zukünftige Techniken der Informatik also auch fachübergreifende Gebiete einzuarbeiten, diese in Projekten anzuwenden, sie fachlich zu kommunizieren, und in wissenschaftlicher Hinsicht zu entwickeln.

Aktuelle Forschungsschwerpunkte und Details zum Master-Studium Angewandte Informatik finden sich auf der Webseite [www.informatik.uni-heidelberg.de](http://www.informatik.uni-heidelberg.de).

# Inhaltsverzeichnis

<b>1 Pflichtbereich</b>	<b>7</b>
Wissenschaftliches Arbeiten . . . . .	8
Seminar . . . . .	9
Masterarbeit . . . . .	10
Anwendungsgebiet . . . . .	11
<b>2 Wahlpflichtbereich</b>	<b>13</b>
2.1 Gebietszuordnung der Module . . . . .	13
2.2 Vertiefungen . . . . .	16
Bildverarbeitung . . . . .	16
Computergraphik und Visualisierung . . . . .	17
Information Systems Engineering . . . . .	19
Optimierung . . . . .	21
Theoretische Informatik . . . . .	23
Wissenschaftliches Rechnen . . . . .	24
2.3 Module aus der Informatik . . . . .	25
Advanced Machine Learning . . . . .	26
Advanced Topics in Text Mining . . . . .	27
Artificial Intelligence . . . . .	29
Big Data . . . . .	30
Cloud Computing 1 . . . . .	32
Compilerbau . . . . .	34
Complex Network Analysis . . . . .	36
Computer Graphics . . . . .	38
Computergraphik 1 . . . . .	40
Computergraphik 2 . . . . .	42
Computerspiele . . . . .	43
Computer Vision: Scene Reconstruction and Understanding . . . . .	44
Cybersicherheit und Internet Governance . . . . .	46
Deep Vision . . . . .	50
Effiziente Algorithmen 1 . . . . .	51
Effiziente Algorithmen 2 . . . . .	53
Einführung in das maschinelle Sehen in 3D . . . . .	55
Formale Sprachen und Automatentheorie . . . . .	56
Fundamentals of Machine Learning . . . . .	59
Funddokumentation mittels optischer 3D-Scanner . . . . .	60
Gaussian Processes for Machine Learning . . . . .	61

	Gemischt-ganzzahlige Programmierung und kombinatorische Optimierung	63
	Geometric Modeling and Animation	65
	Inverse Problems	67
	IT-Projektmanagement	68
	IT-Sicherheit	70
	Knowledge Discovery in Databases	72
	Machine Learning	74
	Mining Massive Datasets	76
	Modellierung und Simulation in den Neurowissenschaften	78
	Object-Oriented Programming for Scientific Computing	80
	Object Recognition and Image Understanding	81
	Optimization for Machine Learning	83
	Parallele Lösung großer Gleichungssysteme	85
	Paralleles Höchstleistungsrechnen	86
	Praktische Geometrie	88
	Projektseminar Biomedizinische Bildanalyse	90
	Qualitätsmanagement	92
	Randomisierte Algorithmen	93
	Requirements Engineering	95
	Scientific Visualization	96
	Selected Topics in the Theory of the Computably Enumerable Sets and Degrees	98
	Software Evolution	99
	Software Ökonomie	101
	Software-Praktikum für Fortgeschrittene	103
	Spatial and Spatio-temporal Databases	104
	Verteilte Systeme I	106
	Visualisierung im Bereich Cultural Heritage	108
	Volumenvisualisierung	109
	Wissensmanagement und Entscheidungen im Software Engineering	111
2.4	Module aus dem B.Sc./M.Sc. Mathematik	113
2.5	Module aus dem M.Sc. Scientific Computing	114
	Fundamentals of Computational Environmental Physics	114
2.6	Module aus dem M.Sc. Technische Informatik	116
2.7	Module aus dem B.Sc./M.Sc. Physik	117
	Physics of Imaging	117
	Image Processing	118
	Introduction to Image Processing on the GPU	119
	Fast Parallel Implementations of Image Labeling Problems	121
2.8	Module aus dem B.S./M.Sc. Biologie	123
	Model-based segmentation of biomedical images (Modellbasierte Segmentierung Biomedizinischer Bilder)	123

# 1 Pflichtbereich

Im Master-Studiengang Angewandte Informatik ist das Modul *Wissenschaftliches Arbeiten* ein Pflichtmodul. Weiterhin müssen auch zwei Seminare absolviert werden. Nachfolgend werden beide Module sowie die Module *Masterarbeit* und *Anwendungsgebiet* beschrieben.

## Wissenschaftliches Arbeiten

<b>Code</b> IWA	<b>Name</b> Wissenschaftliches Arbeiten	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Wintersemester
<b>Lehrform</b> 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium 30 h Selbststudium und praktische Übungen (eventuell in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>kennen die wichtigsten Literaturquellen der Informatik;</li> <li>wissen, welche Tools und Techniken zur Verwaltung von Literatur existieren und wie diese verwendet werden;</li> <li>sind in der Lage, wissenschaftliche Texte (z.B. aus Tagungsbänden oder Journals) und Vorträge kritisch zu lesen und zu bewerten und sie kompakt zusammenzufassen;</li> <li>kennen die einschlägigen Techniken zur Präsentation eines wissenschaftlichen Vortrags;</li> <li>kennen die Möglichkeiten des wissenschaftlichen Publizierens und die Organisation von wissenschaftlichen Tagungen;</li> <li>kennen unterschiedliche Forschungsmethoden ;</li> <li>kennen die Möglichkeiten einer Tätigkeit im wissenschaftlichen Umfeld nach dem Studium;</li> <li>kennen aktuelle Forschungsarbeiten in der Informatik;</li> <li>haben einen Überblick über die Wege der Finanzierung von Forschungsarbeiten;</li> <li>kennen die Anforderungen an die Struktur von Anträgen zur Forschungsförderung.</li> </ul>	
<b>Inhalt</b>	<p>Literaturrecherche und -verwaltung Wissenschaftliches Vortragen, Schreiben, Publizieren und Begutachten Forschungsförderung über Drittmittel Forschungsmethoden und aktuelle Forschungsprojekte Wissenschaftliches Arbeiten nach dem Studium</p>	
<b>Voraussetzungen</b>	keine	
<b>Prüfungsmodalitäten</b>	Regelmäßige Teilnahme, Teilnahme an praktischen Übungen (u.a. durch Bearbeitung von Hausaufgaben und Projekten)	
<b>Nützliche Literatur</b>		

## Seminar

<b>Code</b> IS	<b>Name</b> Seminar	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Semester
<b>Lehrform</b> Seminar 2 + 2 SWS (Seminar/Tutorium)	<b>Arbeitsaufwand</b> 120 h; davon 30 h Präsenzstudium 90 h Vorbereitung Vortrag	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Kenntnis von Techniken des wissenschaftlichen Schreibens (insbesondere auch Literaturrecherche) Fähigkeit, komplexe wissenschaftliche Literatur zu erschließen</p> <p>Erweiterte Fähigkeit, komplexe wissenschaftliche Literatur in einem Vortrag zu präsentieren</p> <p>Erweiterte Fähigkeit, zu Vorträgen zu diskutieren und Feedback zu geben</p> <p>Fähigkeit, ein kurze wissenschaftliche Ausarbeitung zu einem komplexen Thema zu erstellen</p>	
<b>Inhalt</b>	<p>Einführung in und Einübung von Techniken des wissenschaftlichen Schreibens</p> <p>Vertiefte Einübung der Erschließung und Präsentation wissenschaftlicher Literatur</p> <p>Fortgeschritteneres Informatikthema</p>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse im Themengebiet des Seminars	
<b>Prüfungsmodalitäten</b>	<p>Regelmäßige Teilnahme an den Seminarterminen und Mitwirkung in den Diskussionen,</p> <p>Ausarbeitung und Halten eines Vortrages von etwa 60 Minuten Dauer (inklusive Diskussion),</p> <p>schriftliche Ausarbeitung von ca. 10 Seiten</p>	
<b>Nützliche Literatur</b>		

## Masterarbeit

<b>Code</b> IMa	<b>Name</b> Masterarbeit	
<b>Leistungspunkte</b> 30 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Betreutes Selbststudium 2 SWS, Kolloquium 1 SWS	<b>Arbeitsaufwand</b> 900 h; davon 810 h Bearbeitung eines individuellen Themas (Forschungs- und Entwicklungsarbeiten) und schriftliche Ausarbeitung 90 h Ausarbeitung Vortrag und Präsentation und Mitwirkung Kolloquium	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Lernziele</b>	Einsatz der erlernten Fachkenntnisse und Methoden zum selbstständigen Lösen einer komplexen Problemstellung aus der Informatik und ihren Anwendungen Fähigkeit, in großem Umfang selbstständig eine anspruchsvolle wissenschaftlichen Arbeit zu erstellen Fähigkeit, eine eigene, anspruchsvolle Arbeiten in einem wissenschaftlichen Vortrag darzustellen	
<b>Inhalt</b>	selbstständiges wissenschaftliches Bearbeiten einer beschränkten Aufgabenstellung aus der Informatik und ihren Anwendungen wissenschaftlicher Vortrag über die Problemstellung und die erarbeitete Lösung	
<b>Voraussetzungen</b>	nach Prüfungsordnung mindestens 45 LP; weiterhin sind empfohlen: Wahlpflichtvorlesungen und Module Seminar (IS) und Fortgeschrittenenpraktikum (IFP)	
<b>Pruefungs- modalitaeten</b>	regelmäßige Treffen mit der/dem BetreuerIn, schriftliche Ausarbeitung und Präsentation	
<b>Nuetzliche Literatur</b>		

## Anwendungsgebiet

<b>Code</b> IAG	<b>Name</b> Anwendungsgebiet	
<b>Leistungspunkte</b> 18 LP	<b>Dauer</b>	<b>Turnus</b>
<b>Lehrform</b> Vorlesung, Übung und / oder Praktikum	<b>Arbeitsaufwand</b> 540 h, Aufteilung in Präsenz- / Übungs- und Praktikumszeit in Absprache mit den Dozentinnen bzw. Dozenten	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Lernziele</b>	Vertiefte Kenntnisse und Fähigkeiten in einem Anwendungsgebiet	
<b>Inhalt</b>	<p>Wahl eines Anwendungsgebietes nach den Regeln der Prüfungsordnung Festlegung von und Teilnahme an Modulen aus dem Anwendungsgebiet (die LP entsprechen dabei den Vorgaben aus dem Anwendungsgebiet). Dabei ist sicherzustellen, dass keine Module aus dem Anwendungsgebiet gewählt werden, die schon im Bachelor-Studium eingebracht wurden.</p> <p>(optional) Definition und Durchführung eines interdisziplinären Projektes, d.h. Festlegung eines Dozenten / einer Dozentin aus dem Anwendungsgebiet und der Informatik Gemeinsame Festlegung des Projektziels durch die Dozentinnen bzw. Dozenten und den/die Studierende. Das Projektziel umfasst eine informatische Leistung im Anwendungsgebiet.</p> <p>Festlegung des Arbeitsaufwandes und damit der LP für das Projekt Durchführung des Projekts entsprechend des Projektziels unter Anleitung der Dozentinnen bzw. Dozenten Dokumentation des Ergebnisses Erstellung eines Projektbericht Präsentation des Ergebnisses</p>	
<b>Voraussetzungen</b>	empfohlen ist das gleiche Anwendungsgebiet wie im Bachelor	
<b>Prüfungsmodalitäten</b>	<p>Von den 18 LP sind mindestens 8 LP durch ein nicht-informatische Modul auf Masterniveau zu erbringen (Ausnahmegenehmigung möglich), die restlichen 10 LP können durch ein Projekt oder Module auf Bachelorniveau erbracht werden.</p> <p>Prüfungsleistungen in dem Anwendungsgebiet und (optional) Prüfungsleistungen für das interdisziplinäre Projekt analog zum Modul IFP, gewichtet nach dem jeweiligen Anteil der LP Die Module im IAG müssen benotet sein, unbenotete Module werden nur in begründeten Ausnahmefällen zugelassen.</p>	

<b>Nuetzliche Literatur</b>	
---------------------------------	--

## 2 Wahlpflichtbereich

Im Folgenden sind die Wahlpflichtmodule des Masterstudiengangs Angewandte Informatik beschrieben. Es können (aber müssen nicht) Vertiefungen gewählt werden, dabei sind die nachfolgenden Anmerkungen zu beachten. Wie in der Prüfungsordnung beschrieben, sind bei der Wahl der Module Gebiete abzudecken. Die Zuordnung der Module zu den Gebieten ist im nachfolgenden Kapitel beschrieben. Anschließend folgen die Beschreibungen der Vertiefungen und dahinter die einzelnen Modulbeschreibungen.

Das Modul *Künstliche Intelligenz (IKI)* wurde zum WS 2017/18 umbenannt in *Artificial Intelligence (IAI)*.

Das Modul *Parallel data Processing and Analysis (IPDPA)* wurde zum WS 2017/18 umbenannt in *Mining Massive Datasets (IMMD)*.

Das Modul *Objekterkennung und automatisches Bildverstehen (IOAB)* wurde zum SS2018 umbenannt in *Object Recognition and Image Understanding (IORIU)*.

Das Modul *Räumliche Datenbanken (IRDB)* wurde zum SS2018 umbenannt in *Spatial and Spatio-temporal Databases (ISTDB)*.

### 2.1 Gebietszuordnung der Module

Entsprechend der in der Prüfungsordnung genannten Vorgaben ist eine Auswahl aus den folgenden Gebieten abzudecken:

Bildverarbeitung  
Computergraphik und Visualisierung (CGV)  
Datenbanksysteme (DB)  
Optimierung  
Parallele und Verteilte Systeme  
Software Engineering (SWE)  
Technische Informatik  
Theoretische Informatik  
Wissenschaftliches Rechnen

Nachfolgend werden die Module den einzelnen Gebieten zugeordnet.  
Alle nicht genannten Module sind keinem speziellen Gebiet zugeordnet.

#### **Bildverarbeitung**

Artificial Intelligence (IAI)

Deep Vision (IDV)  
Einführung in das maschinelle Sehen in 3D (I3DCV)  
Funddokumentation mittels optischer 3D-Scanner (I3Doc)  
Machine Learning (IML)  
Fundamentals of Machine Learning (IFML)  
Advanced Machine Learning (IAML)  
Projektseminar Biomedizinische Bildanalyse (IPBB)  
Object Recognition and Image Understanding (IORIU)  
Objekterkennung und automatisches Bildverstehen (IOAB) wurde umbenannt in Object Recognition and Image Understanding (IORIU)  
Modellbasierte Segmentierung biomedizinischer Bilder (IMSBI)  
Introduction to Image Processing on the GPU (IGPU)  
Fast Parallel Implementations of Image Labeling Problems (IMLP)

### **Computergraphik und Visualisierung (CGV)**

Artificial Intelligence (IAI)  
Computer Graphics (ICG)  
Computergraphik 1 (ICG1)  
Computergraphik 2 (ICG2)  
Computerspiele (ICS)  
Einführung in das maschinelle Sehen in 3D (I3DCV)  
Geometric Modeling and Animation (IGMA)  
Inverse Probleme (IIP)  
Praktische Geometrie (IPG)  
Scientific Visualization (ISV)  
Visualisierung im Bereich Cultural Heritage (IVCH)  
Volume Visualization (IVV)

### **Datenbanksysteme (DB)**

Advanced Topics in Text Mining (IATM)  
Complex Network Analysis (ICNA)  
Knowledge Discovery in Databases (IKDD)  
Räumliche Datenbanken (IRDB) wurde umbenannt in Spatial and Spatio-temporal Databases (ISTDB)  
Spatial and Spatio-temporal Databases (ISTDB)

### **Optimierung**

Effiziente Algorithmen 1 (IEA1)  
Effiziente Algorithmen 2 (IEA2)  
Mixed-Integer Programming and Combinatorial Optimization (IMIP)  
Lineare Optimierung (MD3)  
Optimization for Machine Learning (IOML)

### **Parallele und Verteilte Systeme**

Big Data (IBD)  
Cloud Computing 1 (ICC1)  
Mining Massive Datasets (IMMD)  
Verteilte Systeme I (IVS1)

### **Software Engineering (SWE)**

IT-Projektmanagement (IPM)  
Software Evolution (ISWEvolv)  
Wissensmanagement und Entscheidungen im Software Engineering (ISWKM)  
Software Ökonomie (ISWÖk)  
Qualitätsmanagement (ISWQM)  
Requirements Engineering (ISWRE)

### **Technische Informatik**

Dieses Gebiet umfasst alle Grundlagen- und Vertiefungsmodule des Masters Technische Informatik.

### **Theoretische Informatik**

Formale Sprachen und Automatentheorie (IFSA)  
Randomisierte Algorithmen (IRA)  
Mathematische Logik (MB9)  
Berechenbarkeit und Komplexität I (MM41)  
Berechenbarkeit und Komplexität II (MM42)

### **Wissenschaftliches Rechnen**

Machine Learning (IML)  
Fundamentals of Machine Learning (IFML)  
Advanced Machine Learning (IAML)  
Modellierung und Simulation in den Neurowissenschaften (IMSN)  
Object-Oriented Programming for Scientific Computing (IOPSC)  
Paralleles Höchstleistungsrechnen (IPHR)  
Parallele Lösung großer Gleichungssysteme (IPLGG)  
Numerik (MD1)

## 2.2 Vertiefungen

### Vertiefung Bildverarbeitung

Die Vertiefung wird gestaltet vom Heidelberger Zentrum für Bildverarbeitung (Heidelberg Collaboratory Image Processing). Sie kann im Master gewählt werden.

Sie umfasst 2 Seminare, Wahlpflichtveranstaltungen und die Masterarbeit. Es sind 2 Seminare und vertiefende Vorlesungen aus dem Gebiet der Bildverarbeitung zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weitere Module in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt werden. Ansprechpartner für diese Vertiefung sind Professor Dr. Jähne, Professor Dr. Hamprecht und Professor Dr. Ommer. Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Die Veranstaltungen aus dem Gebiet der Bildverarbeitung sind:

- Physics of Image Processing (MWInf5) 4 LP
- Bildverarbeitung (MWInf6 oder MH19) 8 LP
- Object Recognition and Image Understanding (IORIU) 8 LP
- Artificial Intelligence (IAI) 6 LP

<b>Veranstaltung</b>	<b>Semester</b>	<b>LP</b>
Wissenschaftliches Arbeiten	1	2
2 Seminare - Bildverarbeitung	1 - 3	8
Bildverarbeitung (MWInf6 oder MH19)	1 - 3	8
Weitere Module aus dem Gebiet der Bildverarbeitung	1 - 3	28
Weitere Module aus mindestens 2 anderen Gebieten	1 - 3	26
Masterarbeit Bildverarbeitung	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		<b>120</b>

## Vertiefung Computergraphik und Visualisierung

Diese Vertiefung befähigt zur Entwicklung von Algorithmen und Anwendungsprogrammen für die visuelle Datenverarbeitung und -analyse. Hierzu gehören Kenntnisse im Bereich Mensch-Maschine-Interaktion, Computergraphik, Datenanalyse und wissenschaftliche Visualisierung. Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst mindestens ein Seminar, ein Praktikum, die Masterarbeit und zusätzliche Wahlpflichtveranstaltungen. Die Module sind aus dem Lehrgebiet Computergraphik und Visualisierung (CGV) zu wählen. Weitere Veranstaltungen sollten aus zwei ergänzenden Lehrgebieten gewählt werden.

Ansprechpartner für diese Vertiefung sind Herr Professor Dr. Sadlo und Frau Dr. Krömker. Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Vertiefende Module aus dem Gebiet CGV sind:

- Computer Graphics (ICG) 8 LP
- Computergraphik 1 (ICG1) 6 LP
- Computergraphik 2 (ICG2) 6 LP
- Computerspiele (ICS) 8 LP
- Einführung in das maschinelle Sehen in 3D (I3DCV) 2 LP
- Geometric Modeling and Animation (IGMA) 8 LP
- Praktische Geometrie (IPG) 4 LP
- Scientific Visualization (ISV) 8 LP
- Visualisierung im Bereich Cultural Heritage (IVCH) 2 LP
- Volume Visualization (IVV) 8 LP

Für das ergänzende Lehrgebiet bieten folgende Bereiche eine gute Erweiterung.

Bildverarbeitung

Datenbanken

Optimierung

Software Engineering

Wissenschaftliches Rechnen

Weitere Bereiche können auf Antrag genehmigt werden.

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar - Lehrgebiet CGV	1 - 3	4
Seminar - CGV oder ergänzendes Lehrgebiet	1 - 3	4
Fortgeschrittenen-Praktikum CGV	1 - 3	8
Vertiefende Module aus dem Bereich CGV	1 - 3	20
Weitere Module aus mindestens 2 anderen Gebieten	1 - 3	34
Masterarbeit CGV	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

Bei Belegung der Vertiefung im Bachelor und Master müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## Vertiefung Information Systems Engineering

Diese Vertiefung befähigt zu Entwicklung, Betrieb und Wartung von komplexen Informationssystemen. Sie kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- bzw. Masterarbeiten und die Wahlpflichtveranstaltungen. Dabei sind die Module zu gleichen Teilen auf die Lehr- und Forschungsgebiete der beiden Arbeitsgruppen Datenbanken (DB) und Software Engineering (SWE) verteilt. Kernstück ist ein die beiden Lehrgebiete übergreifendes Praktikum (genannt ISE-Projekt) im Umfang von 16 LP, in dem im Team für externe Kunden ein Informationssystem mit ingenieurmäßigen Methoden unter Nutzung modernster Technologie entwickelt wird. Dazu gehören u.a. Konzepte und Methoden aus den Bereichen Daten- und Textanalyse, Informationsnetzwerke, Datenmanagement, Softwarequalität und Requirements Engineering. Das ISE-Projekt wird immer im Wintersemester angeboten. Ansprechpartner für diese Vertiefung sind Herr Professor Dr. Gertz (DB) und Frau Professor Dr. Paech (SWE). Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Vertiefende Module aus dem Gebiet SWE:

- Requirements Engineering (ISWRE) 8 LP
- Qualitätsmanagement (ISWQM) 8 LP

Vertiefende Module aus dem Gebiet DB:

- Advanced Topics in Text Mining (IATM) 4 LP
- Knowledge Discovery in Databases (IKDD) 8 LP
- Spatial and Spatio-temporal Databases (ISTDB) 8 LP (vormals Räumliche Datenbanken)

Weitere angebotene Module sind:

DB: Complex Network Analysis (ICNA) 8 LP

SWE: IT-Projektmanagement (IPM) 3 LP

SWE: Software-Ökonomie (ISWÖk) 3 LP

SWE: Software-Evolution (ISWEvolv) 3 LP

SWE: Wissensmanagement in der Softwareentwicklung (ISWKM) (3 LP)

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar - Lehrgebiet SWE	1 - 3	4
Seminar - Lehrgebiet DB	1 - 3	4
ISE-Projekt	1 - 3	16
2 Vertiefende Module Lehrgebiet SWE	1 - 3	16
2 Vertiefende Module Lehrgebiet DB	1 - 3	16
Weitere Module	1 - 3	6
Effiziente Algorithmen 1	1 - 3	8
Masterarbeit SWE oder DB	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

Wurde das ISE-Projekt schon im Bachelor gemacht, so sollten im Master je 1 Praktikum in DB und SWE gemacht werden. Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## Vertiefung Optimierung

Diese Vertiefung hat als Inhalt Entwurf, Analyse und Implementierung von Algorithmen zur Lösung von kontinuierlichen und ganzzahligen Optimierungsproblemen. Sie kann im Bachelor- oder Master-Studium sowie durchgängig vom Bachelor- zum Master-Studium gewählt werden. Sie umfasst neben Vorlesungen, Seminaren und Praktika auch die Bachelorarbeit. Voraussetzungen sind gute bis sehr gute Kenntnisse aus den Pflichtmodulen Mathematik. Ansprechpartner für diese Vertiefung ist Herr Professor Dr. Reinelt. Vor Beginn der Vertiefung wird ein Beratungsgespräch empfohlen.

Den Kern der Veranstaltungen dieser Vertiefung bilden die folgenden Module aus dem Gebiet Optimierung:

- Effiziente Algorithmen 1 (IEA1) 8 LP
- Effiziente Algorithmen 2 (IEA2) 8 LP

Weitere erforderliche Module sind:

mindestens ein Seminar 4 LP,  
mindestens ein Fortgeschrittenen-Praktikum 8 LP.

Empfohlen wird Modul Numerik (MD1).

Möglich sind weiterhin:

- Lineare Optimierung (MD3)
- Mixed-Integer Programming and Combinatorial Optimization (IMIP)

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar Optimierung	1 - 3	4
Seminar (muss nicht Optimierung sein)	1 - 3	4
Fortgeschrittenen-Praktikum Optimierung	1 - 3	8
3 aus den oben genannten Modulen	1 - 3	24
Weitere Module aus dem Gebiet Optimierung	1 - 3	8
Weitere Module aus mindestens 2 anderen Gebieten	1 - 3	22
Masterarbeit Optimierung	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

Für weitere Module werden besonders die Gebiete Wissenschaftliches Rechnen, Bildverarbeitung und Datenbanken empfohlen. Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.



## Vertiefung Theoretische Informatik

Die Vertiefung kann im Master gewählt werden. Sie umfasst Seminare, Wahlpflichtveranstaltungen und die Masterarbeit. Es sind 2 Seminare und 3 vertiefende Vorlesungen aus dem Gebiet der Theoretischen Informatik zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weiter ist die Vorlesung Effiziente Algorithmen 1 verpflichtend. Weitere Lehrveranstaltungen in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt werden. Ansprechpartner für diese Vertiefung ist Herr Professor Dr. Ambos-Spies. Vor Beginn der Vertiefung wird ein Beratungsgespräch empfohlen.

Vertiefende Vorlesungen aus dem Bereich Theoretische Informatik:

- Berechenbarkeit und Komplexität 1 (MH14) 8 LP
- Berechenbarkeit und Komplexität 2 (MH15) 6 LP
- Formale Sprachen und Automatentheorie (IFSA) 8 LP

Auf Antrag kann eine dieser Vorlesungen durch eine vertiefende Vorlesung über effiziente Algorithmen ersetzt werden.

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
2 Seminare - Lehrgebiet Theoretische Informatik	1 - 3	8
3 Vertiefende Vorlesungen - Lehrgebiet Theoretische Informatik	1 - 3	22
Effiziente Algorithmen 1	1 - 3	8
Weitere Module Informatik (Wahlpflichtmodule)	1 - 3	32
Masterarbeit - Lehrgebiet Theoretische Informatik	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

## Vertiefung Wissenschaftliches Rechnen

Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- und Masterarbeiten und die Wahlpflichtveranstaltungen. Ansprechpartner für diese Vertiefung ist Herr Professor Dr. Bastian. Vor Beginn der Vertiefung wird ein Beratungsgespräch empfohlen.

Vertiefende Vorlesungen aus dem Gebiet Wissenschaftliches Rechnen:

- Object-Oriented Programming for Scientific Computing (IOPSC) 6 LP
- Paralleles Höchstleistungsrechnen (IPHR) 8 LP
- Parallele Lösung großer Gleichungssysteme (IPLGG) 8 LP
- Numerik 1 (MD1) 8 LP

Falls die Vertiefung noch nicht im Bachelor gewählt wurde, sollte das Modul MD1 gewählt werden.

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
2 Seminare Wissenschaftliches Rechnen	1 - 3	8
Object-Oriented Programming for Scientific Computing (IOPSC)	1 - 3	6
Paralleles Höchstleistungsrechnen (IPHR)	1 - 3	8
Weitere Module aus dieser Vertiefung	1 - 3	32
Weitere Module aus mindestens 1 anderem Gebiet (empfohlen: SWE)	1 - 3	16
Masterarbeit Wissenschaftliches Rechnen	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## 2.3 Module aus der Informatik

Nachfolgend sind die Module aus der Informatik in alphabetischer Reihenfolge beschrieben.

## Advanced Machine Learning

<b>Code</b> IAML	<b>Name</b> Advanced Machine Learning	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> follows *Fundamentals of Machine Learning*
<b>Lehrform</b> 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials 90h lecture wrap-up and homework 60h graded final report	<b>Verwendbarkeit</b> cannot be combined with *Machine Learning* B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing, M.Sc. Physik (specialization Computational Physics)
<b>Lernziele</b>	Students get to know advanced machine learning methods that define the state-of-the-art and major research directions in the field. Students understand when these methods are called for, what limitations of standard solutions they address, and how they are applied to real-world problems. In addition, students learn how to use Python-based machine learning software such as scikit-learn, theano and OpenGM.	
<b>Inhalt</b>	The lecture, along with its sibling *Fundamentals of Machine Learning*, offers an extended version of the one-semester course *Machine Learning*: Multi-layered architectures (neural networks, deep learning); directed and undirected probabilistic graphical models (Gaussian processes, latent variable models, Markov random fields, structured learning); feature optimization (feature selection and learning, dictionary learning, kernel approximation, randomization); weak supervision (one-class learning, multiple instance learning, active learning, reinforcement learning)	
<b>Voraussetzungen</b>	recommended are: lecture *Fundamentals of Machine Learning* or similar	
<b>Pruefungsmodalitaeten</b>	cannot be combined with *Machine Learning*, successful homework solutions (at least 50% of total achievable points) and written examination (report on a 60h mini-research project)	
<b>Nuetzliche Literatur</b>	David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012 Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006	

## Advanced Topics in Text Mining

<b>Code</b> IATM	<b>Name</b> Advanced Topics in Text Mining	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> one semester	<b>Turnus</b> irregular (every 2nd to 3rd summer semester)
<b>Lehrform</b> Lecture 2 SWS + Exercise course 1 SWS	<b>Arbeitsaufwand</b> 120 h; thereof 30 h lectures 20 h preparation for examination  70 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>Students</p> <ul style="list-style-type: none"> <li>- can apply and evaluate methods of data preparation</li> <li>- know advantages and drawbacks of different data representations</li>   <li>- can apply and evaluate selected methods of text mining</li> <li>- know the theoretical background of machine learning methods deep enough to be able to choose parameters and adapt an algorithm to a given problem</li> <li>- can evaluate and compare text mining models and patterns</li> </ul>	
<b>Inhalt</b>	<p>The lecture introduces the fundamentals as well as selected advanced topics from the domain of text mining.</p> <ul style="list-style-type: none"> <li>- fundamentals of data modeling and preprocessing, in particular for textual data</li> <li>- statistical and algorithmic foundations of the analysis methods</li> <li>- basics of computer linguistics and natural language processing for processing textual data (e.g., morphological analysis, part-of-speech tagging, named entity recognition)</li> <li>- selected and current focus topics such as classification, cluster analysis, sequence pattern mining, association rule mining, topic modeling, and embeddings with an emphasis on the application to textual data</li> </ul>	
<b>Voraussetzungen</b>	recommended are: Algorithmen und Datenstrukturen (IAD), Knowledge Discovery in Databases (IKDD), Einführung in die Wahrscheinlichkeitstheorie und Statistik (MA8)	
<b>Prüfungsmodalitäten</b>	Assignments; at least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final written exam; students can also work on a project (non-graded); final written exam	

<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"><li>- Chru Aggarwal and Zhai ChengXiang: Mining text data. Springer, 2012.</li><li>- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze: Introduction to Information Retrieval, Cambridge University Press. 2008.</li><li>- Jerome H. Friedman, Robert Tibshirani und Trevor Hastie: The Elements of Statistical Learning, 2001.</li><li>- Bing Liu: Web Data Mining (2nd Edition). Springer, 2011.</li></ul>
---------------------------------	---

## Artificial Intelligence

<b>Code</b> IAI	<b>Name</b> Artificial Intelligence	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> one semester	<b>Turnus</b> irregular
<b>Lehrform</b> 2 SWS lecture, 2 SWS tutorial	<b>Arbeitsaufwand</b> 180 h; thereof 60h lectures and tutorials 50h graded final report of mini project 70h lecture wrap-up, programming exercises and homework	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>To be able to develop algorithms for solving problems by informed/uninformed search.</p> <p>To know how to apply logical inference for finding solutions.</p> <p>To have a firm command of applying inference under uncertainty.</p> <p>To have gained a solid understanding of learning agents based on deep learning and neural networks.</p> <p>To know how to apply the developed expertise to different application areas such as Computer Vision or text mining.</p>	
<b>Inhalt</b>	<p>Solving problems by search</p> <p>Game playing</p> <p>Inference using logic</p> <p>Knowledge bases</p> <p>Planning and acting</p> <p>Reasoning under uncertainty</p> <p>Deep Learning and Neural Networks</p> <p>Machine Learning Applications</p>	
<b>Voraussetzungen</b>	recommended are: basic programming skills and knowledge of basic calculus, statistics, and linear algebra	
<b>Pruefungsmodalitaeten</b>	Successful homework solutions and examination (written report on mini research project)	
<b>Nuetzliche Literatur</b>	Russell & Norvig: Artificial Intelligence: A Modern Approach	

## Big Data

<b>Code</b> IBD	<b>Name</b> Big Data	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Hausaufgabenbearbeitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden</p> <p>kennen den prinzipiellen Aufbau einer Big Data Anwendung haben die erforderlichen Grundlagen gelernt (Technologien, Methoden und Konzepte)</p> <p>sind mit den Sicherheitsproblematiken von Big Data Anwendungen vertraut und können eine adäquate Architektur konzipieren</p> <p>können Big Data Konzepte in Anwendungen umsetzen sind in der Lage, öffentliche und private Daten in Anwendungen zu aggregieren</p> <p>können die erworbenen Kenntnisse auf verschiedene Anwendungsbereiche übertragen</p>	
<b>Inhalt</b>	<p>Die Vorlesung bietet eine praktische Einführung in das aktuelle Gebiet Big Data, wobei auch Cloud Ressourcen im Internet für eigene Anwendungen transparent genutzt werden können. Schwerpunkte sind u.a.:</p> <p>Fundamentals: Volume, Variety, Velocity, Veracity, Visualization, Value</p> <p>Architecture: Hadoop as a Service, Data Warehouse</p> <p>Data Management: Movement, Monitoring, Provenance, Preservation, SLAs</p> <p>Data Publication: Preparation, Curation, Discovery, Open Data, Open Access</p> <p>Data Security, Privacy &amp; Trust: Risk Management, Usage Control, Trusted Datastore</p>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie im Modul ISW, IBN und ICC vermittelt werden. Modul IPDPA wird als Ergänzung bzw. Erweiterung empfohlen.	
<b>Prüfungs- modalitäten</b>	Teilnahme an der Vorlesung mit Hausaufgaben und Bestehen einer Klausur oder mündlichen Abschlussprüfung	

<p><b>Nuetzliche Literatur</b></p>	<p>BITKOM Leitfaden Big-Data-Technologien, 2014 (online)          Bücher aus dem O'Reilly Data Science Starter Kit, 2014 (online)          L.Barroso, J.Clidas, U.Hölzle, The Data Center as a Computer:          An Introduction to the Design of Warehouse-Scale Machines,          Morgan&amp;Claypool, 2013 (online)          A.Murthy, V.Vavilapalli, D.Eadline, J.Niemiec, Apache Hadoop          YARN: Moving beyond MapReduce and Batch Processing with          Apache Hadoop 2, Addison Wesley Data&amp;Analytics, 2014          C.Baun, M.Kunze, J.Nimis, S.Tai, Cloud Computing - Web          basierte dynamische IT-Services, Informatik im Fokus, Springer          Verlag 2011, 2. Auflage          C.Bengel, C.Baun, M.Kunze, U.Stucky, Masterkurs Verteilte und          Parallele Systeme, Springer Verlag 2008</p>
--	--

# Cloud Computing 1

<b>Code</b> ICC1	<b>Name</b> Cloud Computing 1	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Hausaufgabenbearbeitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden</p> <p>kennen den prinzipiellen Aufbau einer Computing Cloud</p> <p>haben die erforderlichen Grundlagen gelernt (Virtualisierung, Web Services)</p> <p>sind mit den Sicherheitsproblematiken von Cloud-Systemen vertraut und können eine adäquate Architektur konzipieren</p> <p>können Konzepte wie IaaS, PaaS und SaaS in Anwendungen umsetzen</p> <p>sind in der Lage, öffentliche und private Cloud-Systeme zu nutzen</p> <p>können die erworbenen Kenntnisse auf verschiedene Anwendungsbereiche zu übertragen</p>	
<b>Inhalt</b>	<p>Die Vorlesung bietet eine praktische Einführung in das aktuelle Gebiet Cloud Computing, bei dem Ressourcen im Internet für eigene Anwendungen transparent genutzt werden können.</p> <p>Schwerpunkte sind u.a.:</p> <ul style="list-style-type: none"> <li>Grundlagen</li> <li>Virtualisierung und Web Services</li> <li>Cloud Angebote im Internet (Taxonomie)</li> <li>Cloud Security</li> <li>Programmiermodelle</li> <li>Wiss. Rechnen in der Cloud</li> <li>High Performance Computing as a Service</li> <li>Big Data</li> </ul>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in den Modulen Einführung in Software Engineering (ISW) sowie Betriebssysteme und Netzwerke (IBN) vermittelt werden	
<b>Prüfungsmodalitäten</b>	Teilnahme an der Vorlesung mit Hausaufgaben und Bestehen einer mündlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	C.Baun, M.Kunze, J.Nimis, S.Tai, Cloud Computing - Web basierte dynamische IT-Services, Informatik im Fokus, Springer Verlag 2011, 2. Auflage C.Bengel, C.Baun, M.Kunze, U.Stucky, Masterkurs Verteilte und Parallele Systeme, Springer Verlag 2008
---------------------------------	---

## Compilerbau

<b>Code</b> ICOM	<b>Name</b> Compilerbau	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 20 h Prüfungsvorbereitung 100 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden kennen den prinzipiellen Aufbau eines Compilers haben die erforderlichen Grundlagen zu formalen Sprachen gelernt</p> <p>sind mit den Problematiken von lexikalischer, syntaktischer und semantischer Analyse vertraut kennen die Techniken zur Compilererstellung sind über Optimierungsmöglichkeiten informiert sind in der Lage, vorhandene Werkzeuge zum Compilerbau zu benutzen können die erworbenen Kenntnisse auf andere Anwendungsbereiche, wie z.B. Aufbereitung strukturierter Daten, übertragen</p>	
<b>Inhalt</b>	<p>Überblick über grundlegende Techniken Grundlagen der formale Sprachen Lexikalische Analyse Top-Down-Syntaxanalyse Bottom-Up-Syntaxanalyse Syntaxgesteuerte Übersetzung Semantische Analyse Die C-Maschine 3-Adress-Code Optimierung Datenfluss-Analyse</p>	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50 % der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	z.B. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques and Tools, Pearson - Addison-Wesley, 2006
---------------------------------	--

## Complex Network Analysis

<b>Code</b> ICNA	<b>Name</b> Complex Network Analysis	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> every 2nd wintersemester
<b>Lehrform</b> Lecture 4 SWS, Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h lecture 12 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing B.Sc. Mathematik
<b>Lernziele</b>	<p>Students</p> <ul style="list-style-type: none"> <li>- can describe basic measures and characteristics of complex networks</li> <li>- can implement and apply basic network analysis algorithms using programming environments such as R or Python</li> <li>- can describe different network models and can describe, compute, and analyze characteristic parameters of these models</li> <li>- know how to compute different complex network measures and how to interpret these measures</li> <li>- know different generative models for constructing complex networks, especially scale-free networks</li> <li>- know the fundamental methods for the detection of communities in networks and the analysis of their evolution over time</li> <li>- are familiar with basic concepts of network robustness</li> <li>- understand the principles behind the spread of phenomena in complex networks</li> </ul>	

<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- Graph theory and graph algorithms; basic network measures</li> <li>- Random networks and their characteristics (degree distribution, component sizes, clustering coefficient, network evolution), small world phenomena</li> <li>- Scale-free property of networks, power-laws, hubs, universality</li> <li>- Barabasi-Albert model, growth and preferential attachment, degree dynamics, diameter and clustering coefficient</li> <li>- Evolving networks, Bianconi-Barabasi model, fitness, Bose-Einstein condensation</li> <li>- Degree correlation, assortativity, degree correlations, structural cutoffs</li> <li>- Network robustness, percolation theory, attack tolerance, cascading failures</li> <li>- Communities, modularity, community detection and evolution</li> <li>- Spreading phenomena, epidemic modeling, contact networks, immunization, epidemic prediction</li> </ul>
<b>Voraussetzungen</b>	recommended are: Algorithmen und Datenstrukturen (IAD), Knowledge Discovery in Databases (IKDD), Lineare Algebra I (MA4)
<b>Pruefungs-modalitaeten</b>	Assignments; at least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final written exam; final written exam
<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>- Albert-Laszlo Barabasi: Network Science, Cambridge University Press, 2016.</li> <li>- M.E.J. Newmann: Networks: An Introduction, Oxford University Press, 2010.</li> <li>- Vito Latora, Vincenzo Nicosia, Giovanni Russo: Complex Networks - Principles, Methods and Applications, Cambridge University Press, 2017.</li> <li>- David Easley, Jon Kleinberg: Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.</li> <li>- Stanley Wasserman, Katherine Faust: Social Network Analysis-Methods and Applications, Cambridge University Press, 1994.</li> </ul>

## Computer Graphics

<b>Code</b> ICG	<b>Name</b> Computer Graphics	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> every 3rd semester
<b>Lehrform</b> lecture 4 SWS, exercise 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	<b>Verwendbarkeit</b> cannot be combined with Computergraphik 1 and 2 (ICG1, ICG2) B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	The students understand fundamental and advanced concepts of computer graphics. They understand the mathematical fundamentals, data structures, and implementation aspects. They get to know raster graphics, geometric transforms, color perception and color models, and basics of geometric modeling. The students are able to apply these concepts to real-world problems using existing software packages, and develop small programs using OpenGL 4.	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- Introduction</li> <li>- Perception and Color</li> <li>- Raytracing</li> <li>- Transformations</li> <li>- Rasterization</li> <li>- OpenGL</li> <li>- Textures</li> <li>- Curves</li> <li>- Spatial Data Structures</li> </ul>	
<b>Voraussetzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Pruefungs- modalitaeten</b>	cannot be combined with Computergraphik 1 and 2 (ICG1, ICG2), successful participation in the exercises (more than 50% of the achievable points) and passing of an oral or written exam	

<b>Nuetzliche Literatur</b>	<p>P. Shirley, S. Marschner: Fundamentals of Computer Graphics, 3rd Edition, AK Peters OpenGL Specifications(GL 4.5 + GLSL 4.50) <a href="http://www.opengl.org/registry/">http://www.opengl.org/registry/</a> Optional A. S. Glassner: An Introduction to Ray Tracing, Academic Press T. Akenine-Möller, E. Haines: Real-Time Rendering, AK Peters, 2008</p>
---------------------------------	---

# Computergraphik 1

<b>Code</b> ICG1	<b>Name</b> Computergraphik 1	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> kann nicht kombiniert werden mit Computer Graphics (ICG) B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>Die Studierenden sind in der Lage, einfache Graphikfenster mit Inhalt zu belegen, sind mit 2D-Rastergraphik vertraut und in der Lage, mit Objekten in Weltkoordinaten über Graphikbibliotheken (OpenGL) zu interagieren, können mit homogenen Koordinaten umgehen und Projektions- sowie Transformationsmatrizen bestimmen. Sie kennen die verschiedenen Bufferkonzepte zur schnellen Darstellung realistischer Effekte, sind mit Farbkonzepten und Darstellungsmodellen vertraut, können lokale Lichtmodelle aus optischen Prinzipien herleiten und wissen, wie man Texturen für unterschiedliche Effekte einsetzt. Sie können die physikalischen Grundlagen für globale Lichtmodelle in Fremdsoftware abrufen. Das sichere Beherrschen der Programmierung kleinerer Graphikprogramme und die Generierung von Einzelbildern, 3D Grafiken und Animationen für die unterschiedlichen Zwecke des wissenschaftlichen Rechnens wird auf Basis der Programmbibliothek OpenGL (Open Graphics Library) vermittelt.</p>	
<b>Inhalt</b>	<p>Grundlagen der Graphikprogrammierung          Koordinatensysteme          Projektionen, Transformationen          Zeichenalgorithmen          Bufferkonzepte (z-Buffer, Double-Buffer, Stereo-Buffer)          Shading und Lichtmodelle          Graphikbibliothek OpenGL          Direct Rendering (lokale Verfahren)          Texturen und Abbildungsverfahren          Globale Verfahren          Raytracing (Povray)          Volume Rendering</p>	

<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK)
<b>Pruefungs- modalitaeten</b>	kann nicht kombiniert werden mit Computer Graphics (ICG), Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
<b>Nuetzliche Literatur</b>	J. D. Foley, A. Van Dam, S. K. Feiner, J . F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996 D. Shreiner, M. Woo, J. Neider und T. Davis: OpenGL programming guide: the official guide to learning OpenGL, version 1.4. OpenGL Architecture Review Board, Addison-Wesley, 2004

## Computergraphik 2

<b>Code</b> ICG2	<b>Name</b> Computergraphik 2	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Sommersemester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> kann nicht kombiniert werden mit Computer Graphics (ICG) B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	Die Studierenden sind mit weiterführenden Lichtmodellen vertraut, können vertex- und fragmentbasierte Shader programmieren, kennen die Vor- und Nachteile verschiedener globaler photorealistischer Verfahren, können nichtphotorealistische Verfahren sinnvoll einsetzen und mit Splines als Linien und Flächen umgehen. Das sichere Beherrschen der Programmierung von Grafikkarten (Graphik-Hardware GPU) mit der Programmiersprache *C for graphics* und anderen *Application Programming Interfaces* (API) für globales Rendering wird unter Verwendung von numerischen Algorithmen zur Lösung nichtlinearer Gleichungssysteme vermittelt.	
<b>Inhalt</b>	Shading und Bidirectional Reflectance Distribution Function (BRDF), C for graphics (Cg), Radiosity, Photon Mapping, Nichtphotorealismus (NPR), Splines	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Computergraphik 1 (ICG1)	
<b>Pruefungs- modalitaeten</b>	kann nicht kombiniert werden mit Computer Graphics (ICG), Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung	
<b>Nuetzliche Literatur</b>	J. D. Foley, A. Van Dam, S. K. Feiner, J . F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996 Randima Fernando, Mark J. Kilgard: The Cg Tutorial. Addison-Wesley, 2003	

## Computerspiele

<b>Code</b> ICS	<b>Name</b> Computerspiele	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 3 SWS	<b>Arbeitsaufwand</b> 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Die Studierenden lernen die Konzepte von der informatischen Sicht kennen, was Computerspiele ausmacht und welche Herausforderungen damit verbunden sind. Sie lernen, wie man gute und effiziente Architekturkonzepte dafür entwickelt sowie wie man typische Probleme aus Graphik, Kollisionserkennung, Animation/Physik, Pfadplanung/KI umsetzt. Zudem lernen sie in den Übungen, wie man konkrete Spiele entwickelt, so dass sie in der Lage sind, eigene Spieleengines zu realisieren.	
<b>Inhalt</b>	<p>Überblick über die Einteilung von Computerspielen</p> <p>Architektur von Game Engines</p> <p>Vorstellung von OGRE als einer open-source Game Engine</p> <p>Graphik und Computerspiele: ein Überblick</p> <p>Kollisionserkennungstechniken</p> <p>Animationstechniken und Physik bei Computerspielen mit Fokus auf der open source Bibliothek Bullet</p> <p>Pfadplanung und KI</p>	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung	
<b>Nuetzliche Literatur</b>	<p>Gregory et al: Game Engine Architecture</p> <p>Ericson: Real-Time Collision Detection</p> <p>Eberly: Game Physics</p> <p>Millington: Artificial Intelligence for Games</p>	

## Computer Vision: Scene Reconstruction and Understanding

<b>Code</b> ICVSRU	<b>Name</b> Computer Vision: Scene Reconstruction and Understanding	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> one semester	<b>Turnus</b> irregular
<b>Lehrform</b> block course 10 days	<b>Arbeitsaufwand</b> 180 h; thereof 30 h lectures 30 h exercises 20 h revision and home exercise 70 h programming a mini research project 30 h preparation of final report	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
<b>Lernziele</b>	<p>The students</p> <ul style="list-style-type: none"> <li>- Understand the principles behind estimating 3D Point Clouds and Motion from two or more images. They are able to apply this knowledge to new tasks in the field of 3D reconstruction.</li> <li>- Understanding the principles of an image formation process and corresponding Geometry. This can be utilized to design new algorithms, for e.g. 3D motion estimation for autonomous driving.</li> <li>- Understand and implement methods that combine machine learning based methods with classical computer vision based techniques.</li> <li>- Have studied various state-of-the-art computer vision systems and approaches, and are then able to evaluate and classify new systems and approaches.</li> <li>- Understand and implement different approaches for object tracking and object-instance recognition.</li> </ul>	
<b>Inhalt</b>	<p>The lecture will cover areas of computer vision which deal with 3D reconstruction and scene understanding. We will discuss the underlying principles and methods to solve these task. We will also discuss latest state-of-the art techniques, which are often a combination of traditional approaches and modern neural networks. Additionally we will cover the necessary background knowledge, e.g. image formation model, camera models, relevant machine learning. Example topics that will be covered are:</p> <ul style="list-style-type: none"> <li>i) dense motion estimation from two views;</li> <li>ii) full 3D reconstructions from many views;</li> <li>iii) to detect and track objects in the 3D scene.</li> </ul>	
<b>Voraussetzungen</b>	recommended are: Fundamentals of Machine Learning and Advanced Machine Learning or equivalent	

<b>Pruefungs- modalitaeten</b>	graded final report (about 10 pages) or final report (about 5 pages) together with oral exam
<b>Nuetzliche Literatur</b>	

## Cybersicherheit und Internet Governance

<b>Code</b> ICIG	<b>Name</b> Cybersicherheit und Internet Governance	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> 1 Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 120 h; davon 30 h Präsenzstudium 60 h Vor- und Nachbereitung 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik

<p><b>Lernziele</b></p>	<p>Studierende</p> <ul style="list-style-type: none"> <li>* erwerben umfangreiches Wissen über die Funktionsweise und Verwundbarkeiten vernetzter Computersysteme, insbesondere der Internettechnologie und können somit Konzepte zur IT-Sicherheit bewerten und entwerfen</li> <li>* erlangen grundlegende Kenntnisse über die Verwaltung der kritischen Ressourcen des Internets (IP-Adressraum, DNS, Protokolle und Standards) und Entscheidungsfindungsmechanismen internationaler Internet Governance und können die Auswirkung von international festgelegten Vorschriften auf die Umsetzung von IT-Sicherheitskonzepten berücksichtigen</li> <li>* erwerben vertiefte Kenntnisse im Bereich Cybersicherheit; Studium und Nachvollzug typischer Cyberangriffsforme; umfassender Kompetenzerwerb zur Detektion von Cyberangriffen; grundlegender Kompetenzen im Feld der IT-/Cyber-Forensik und sind dadurch in der Lage aktive Maßnahmen im Sinne von Intrusion Detection Systemen für eine Einrichtung zu planen und rechtskonform umzusetzen</li> <li>* erhalten einen detaillierten Überblick über Felder gesellschaftspolitischer Regulierung (Netzneutralität, Datenschutz, Cybersicherheit), einschließlich normativ-rechtlicher Gestaltungsansprüche, konkreter politischer Regulierung (national, EU, international) sowie technischer Voraussetzungen und Umsetzungsmöglichkeiten (Sicherheits-Engineering) und können die Praktikabilität der jeweiligen Konzepte sowohl aus Sicht des gesellschaftlichen Diskurses als auch aus der Perspektive der Technik bewerten</li> </ul> <p>Möglichkeit zur Teilnahme an Aufbaukursen sowohl im technischen/computerwissenschaftlichen (als auch im politik-/sozialwissenschaftlichen Feld)</p> <p>Langfristiges Ausbildungsziel: Einsatz-/Beschäftigungsfähigkeit in der Breite des Arbeitsfeldes IT-Cybersicherheit; durch ganzheitlichen Ansatz der Ausbildung nicht allein in technischen Beschäftigungsfeldern, sondern auch in der Beratung von Unternehmen, Behörden und anderen Einrichtungen</p>
-------------------------	---

<b>Inhalt</b>	<p>Die Digitalisierung und vernetzte Computersysteme prägen gesellschaftliche Austauschprozesse auf nationaler wie internationaler Ebene. Sie haben gravierende Effekte auf das alltägliche Leben der Nutzer, die Wirtschafts- und Arbeitswelt, den internationalen Handel und die Weltpolitik. Gerade die vergangenen Jahre haben gezeigt, wie verwundbar vernetzte Computersysteme sind. Cybersicherheit ist von einer technischen zu einer gesellschaftspolitischen Herausforderung ersten Grades geworden. Um ihr gerecht zu werden, ist es von großer Bedeutung, vernetzte Computersysteme und insbesondere das Internet nicht auf die technischen Aspekte zu reduzieren, sondern das Ineinandergreifen von Technologie auf der einen Seite, normativen und politischen Regulierungsansprüchen sowie vielfältigen Governance-Strukturen und -Prozessen auf der anderen Seite zu begreifen. Hierfür bietet die Vorlesung den teilnehmenden Studierenden einen interdisziplinären Ausgangspunkt, denn sie leistet einen Brückenschlag zwischen der Informatik und der Politischen Wissenschaft.</p> <p>Themen sind u.a.:</p> <ul style="list-style-type: none"> <li>* Angriffsszenarien und Verwundbarkeiten vernetzter Computersysteme;</li> <li>* Internet(un)sicherheit;</li> <li>* Sicherheits-Engineering;</li> <li>* Verwaltung kritischer Internetressourcen;</li> <li>* Netzneutralität;</li> <li>* Content-Regulierung und Zensur;</li> <li>* Cybersicherheit als Gegenstand politischer und rechtlicher Regulierung;</li> </ul>
<b>Voraussetzungen</b>	
<b>Pruefungs-modalitaeten</b>	regelmäßige Teilnahme (überprüft), Abschlussklausur (90 Min.)

<p><b>Nuetzliche Literatur</b></p>	<p>DeNardis, L. (2014). The global war for internet governance. New Haven/London: Yale University Press.</p> <p>Dunn Caveltly, Myriam (2011): Cyber-threats, in: Dunn Caveltly, Myriam/Mauer, Victor (Hg.): The Routledge handbook of security studies. Milton Park [et al.]: Routledge (Routledge handbooks), 180-189.</p> <p>Eckert, C. (2014). IT-Sicherheit: Konzepte, Verfahren, Protokolle (9. Aufl). De Gruyter Studium. Oldenbourg: de Gruyter.</p> <p>Harich, T. (2015). IT-Sicherheit im Unternehmen, MITP Taschenbuch</p> <p>Knake, Robert K. (2010): Internet governance in an age of cyber insecurity. New York, NY: Council on Foreign Relations (Council Special Report).</p> <p>Lange, H.-J., und Bötticher, A. Cyber-Sicherheit. Studien zur Inneren Sicherheit: Band 18.</p> <p>Nissenbaum, Helen (2005): Where Computer Security Meets National Security, in: Ethics and Information Technology 7: 2, 61-73.</p> <p>Weitere Literatur wird in der Vorlesung bekannt gegeben.</p>
--	---

## Deep Vision

<b>Code</b> IDV	<b>Name</b> Deep Vision	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 105 h Selbststudium und Bearbeitung von theoretischen Aufgaben und Programmierübungen 15 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Die Studierenden haben gelernt Algorithmen zu entwickeln, die effizient aus großen Bildmengen robuste Objektmodelle lernen, kennen die Vorzüge/Einschränkungen von Deep Learning Ansätzen im Bereich des maschinellen Sehens, können Inferenz über Objekte in 3D durchführen, können die erworbenen Kenntnisse auf Anwendungsbereiche, wie z.B. die Erkennung von Objekten in 3D Szenen oder biomedizinischen Tomographien übertragen	
<b>Inhalt</b>	High-level Computer Vision 3D Vision and beyond Deep Learning Multi-view Approaches Applications	
<b>Voraussetzungen</b>	keine	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen und Bestehen einer Abschlussprüfung	
<b>Nützliche Literatur</b>		

# Effiziente Algorithmen 1

<b>Code</b> IEA1	<b>Name</b> Effiziente Algorithmen 1	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik M.Sc. Mathematik
<b>Lernziele</b>	Die Studierenden verstehen die grundlegenden Konzepte der Graphentheorie können Fragestellungen als kombinatorische Optimierungsprobleme modellieren können die Komplexität von Optimierungsproblemen analysieren kennen die Methoden zum Beweis der Korrektheit kombinatorischer Algorithmen und der Analyse ihrer Laufzeit kennen die grundlegenden algorithmischen Ansätze sind mit den Fragen der effizienten Implementierung vertraut haben Einblick in die vielfältigen Anwendungsgebiete der kombinatorischen Optimierung	
<b>Inhalt</b>	Grundbegriffe der Graphentheorie Grundlegende Graphenalgorithmien Optimale Bäume und Branchings Kürzeste Wege Das Zuordnungsproblem Maximale Flüsse Minimale Schnitte in ungerichteten Graphen Flüsse mit minimalen Kosten Matchingprobleme	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Lineare Algebra 1	
<b>Prüfungs- modalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	Korte, B., Vygen, J.: Kombinatorische Optimierung, Springer, 2007 Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: Combinatorial Optimization, Wiley, 1997
---------------------------------	--

## Effiziente Algorithmen 2

<b>Code</b> IEA2	<b>Name</b> Effiziente Algorithmen 2	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik M.Sc. Mathematik
<b>Lernziele</b>	<p>Die Studierenden können polynomiale und NP-schwere Optimierungsprobleme klassifizieren</p> <p>kennen die gesamte Bandbreite von Lösungsansätzen für schwierige Probleme</p> <p>können geeignete Modellierungen für Anwendungsprobleme selbst erstellen</p> <p>sind in der Lage, geeignete Lösungsverfahren auszuwählen</p> <p>können selbst Lösungsverfahren konzipieren und implementieren</p>	
<b>Inhalt</b>	<p>NP-schwere Optimierungsprobleme</p> <p>approximative Algorithmen und Heuristiken (Bin-Packing, Scheduling, Knapsack, Traveling Salesman)</p> <p>Relaxierungen (lineare, kombinatorische, semidefinite, Lagrange-Relaxierungen)</p> <p>Verfahren zur Bestimmung optimaler Lösungen (dynamische Optimierung, Branch-and-Bound, Branch-and-Cut)</p> <p>lineare 0/1- Optimierung (Modellierung, Schnittebenen)</p> <p>polyedrische Kombinatorik, Spaltengenerierung und Dekomposition</p> <p>Fallstudien (Traveling-Salesman_Problem, Max-Cut-Problem.)</p> <p>Es wird somit das gesamte Spektrum von Lösungsverfahren für schwierige kombinatorische Probleme vermittelt.</p>	
<b>Voraussetzungen</b>	<p>empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Lineare Algebra 1; Absolvierung des Moduls Effiziente Algorithmen 1 (IEA1) ist nützlich, aber nicht Voraussetzung</p>	
<b>Prüfungsmodalitäten</b>	<p>Erfolgreiche Teilnahme an den Übungen (mehr als 50 Prozent der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung</p>	

<b>Nuetzliche Literatur</b>	Korte, B., Vygen, J.: Kombinatorische Optimierung, Springer, 2007 Nemhauser, G.L., Wolsey, L.A.: Integer Combinatorial Optimization, Wiley, 1988
---------------------------------	---

## Einführung in das maschinelle Sehen in 3D

<b>Code</b> I3DCV	<b>Name</b> Einführung in das maschinelle Sehen in 3D	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Wintersemester
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Die Studierenden sind in der Lage 3D Erfassungs- und Auswertemethoden in praktischen, industriellen und geisteswissenschaftlichen Anwendungen einzusetzen. Sie kennen die Gesetze und Eigenschaften von optischen Linsen, grundlegende Bildfilter, interne und externe Kameraparameter, Stereorektifizierung bzw. Structure from Motion (SfM), Bündelausgleich, Berechnung von Tiefendaten, Datenformate für unstrukturierte Gitter, numerische Krümmungsberechnung auf diskreten Mannigfaltigkeiten. Sie können den Einsatz von sowohl strukturierten Licht als auch Time-of-flight Scannern planen.	
<b>Inhalt</b>	Bildaufnahme Kamerakalibrierung und Merkmalsextraktion Active Range Scanner(s) Shape from Monocular Images Shape from Multiple Images Time-of-flight-scanner (TOF) 3D Objektrepräsentationen 3D Anwendungen	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK)	
<b>Prüfungsmodalitäten</b>	Teilnahme an den Vorlesungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Teilnehmerzahl)	
<b>Nützliche Literatur</b>	J. Bernd, Digital Image Processing., Springer, 6 edition, 2005. E. R. Davies, Machine Vision ? Theory, Algorithms, Practicalities, Elsevier, 2005.	

## Formale Sprachen und Automatentheorie

<b>Code</b> IFSA	<b>Name</b> Formale Sprachen und Automatentheorie	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mindst. jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS mit Hausaufgaben.	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 60 h Prüfungsvorbereitung (und Prüfung) 90 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik

<b>Lernziele</b>	<p>Die Studierenden kennen die Stufen der Chomsky-Hierarchie und deren äquivalente Beschreibung durch Grammatiken und Automatenmodelle, kennen die Charakterisierungen der Klasse der regulären Sprachen durch rechts- bzw. linkslineare Grammatiken, durch deterministische und nichtdeterministische endliche Automaten, durch reguläre Ausdrücke und durch Rechtskongruenzen mit endlichem Index, sind in der Lage, das Pumping-Lemma für reguläre Sprachen zu beweisen und dieses anzuwenden, um zu zeigen, dass bestimmte Sprachen nicht regulär sind, kennen den Nachweis der Eindeutigkeit des Minimalautomaten und können diesen in Form eines reduzierten Automaten oder eines Äquivalenzklassenautomaten konstruieren, können das Pumping-Lemma für kontextfreie Sprachen beweisen und dieses anwenden, um zu zeigen, dass bestimmte Sprachen nicht kontextfrei sind, können die Klasse der kontextfreien Sprachen durch kontextfreie Grammatiken und durch Kellerautomaten beschreiben und die Äquivalenz dieser Darstellungen beweisen, kennen den Satz von Chomsky und Schützenberger und seine anschauliche Bedeutung für die Struktur der Ableitungen in kontextfreien Sprachen, sind mit Normalformen, Abschlusseigenschaften, Entscheidbarkeits- und Komplexitätsaspekten insbesondere der regulären und kontextfreien Sprachen vertraut, können die Klasse der kontextsensitiven Sprachen durch kontextsensitive Grammatiken und Grammatiken vom Erweiterungstyp sowie durch linear beschränkte Automaten beschreiben und die Äquivalenz dieser Darstellungen beweisen.</p>
<b>Inhalt</b>	<p>Die Vorlesung gibt eine Einführung in die Theorie der formalen Sprachen und die dort verwendeten Begriffe und Methoden zur syntaktischen Sprachbeschreibung und -analyse unter besonderer Berücksichtigung algorithmischer Aspekte. Der Schwerpunkt der Vorlesung liegt auf den Sprachklassen der Chomsky-Hierarchie und deren Charakterisierung hinsichtlich der Beschreibbarkeit durch Grammatiken und der Erkennbarkeit durch Automaten.</p>
<b>Voraussetzungen</b>	<p>empfohlen ist: Einführung in die Theoretische Informatik (ITH)</p>
<b>Prüfungsmodalitäten</b>	<p>Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer mündlichen Abschlussprüfung.</p>

<b>Nuetzliche Literatur</b>	
---------------------------------	--

## Fundamentals of Machine Learning

<b>Code</b> IFML	<b>Name</b> Fundamentals of Machine Learning	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> in (irregular) alternation with *Machine Learning*
<b>Lehrform</b> 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials 90h lecture wrap-up and homework 60h graded final report	<b>Verwendbarkeit</b> cannot be combined with *Machine Learning* B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing, M.Sc. Physik (specialization Computational Physics)
<b>Lernziele</b>	Students understand fundamental concepts of machine learning (features vs. response, unsupervised vs. supervised training, regression vs. classification etc.), get to know established learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results. In addition, students learn how to use Python-based machine learning software such as scikit-learn.	
<b>Inhalt</b>	The lecture, along with its sibling *Advanced Machine Learning*, offers an extended version of the one-semester course *Machine Learning*, with more room for regression methods, unsupervised learning and algorithmic details: Classification (nearest neighbor rules, linear and quadratic discriminant analysis, logistic regression, classical and randomized decision trees, support vector machines, ensemble methods); regression (linear and non-linear least squares, regularized and sparse regression, robust regression); unsupervised learning (hierarchical clustering, k-means algorithm, Gaussian mixture models and expectation maximization, principal component analysis, non-linear dimension reduction); evaluation (risk minimization, model selection, cross-validation)	
<b>Voraussetzungen</b>	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
<b>Pruefungsmodalitaeten</b>	cannot be combined with *Machine Learning*, successful homework solutions (at least 50% of total achievable points) and written examination (report on a 60h mini-research project)	
<b>Nuetzliche Literatur</b>	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009	

## Funddokumentation mittels optischer 3D-Scanner

<b>Code</b> I3DOK	<b>Name</b> Funddokumentation mittels optischer 3D-Scanner	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Übung 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Die Studierenden sind mit unterschiedlichen 3D-Nahbereichsscantechniken vertraut, können 3D-Modelle aufbereiten und kennen die Herangehensweise mit 3D Bildverarbeitung zur Erkennung von Merkmalen (Schrift).	
<b>Inhalt</b>	3D-Meßtechnik basierend auf dem Prinzip des Strukturierten Licht, Aufnahme und Verarbeitung von hochauflösenden 3D-Modellen *3D-image-processing pipeline*	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI) oder Computergraphik 1 (ICG1) oder Visualisierung im Bereich Cultural Heritage (IVCH) oder Vermessungskunde (UFG)	
<b>Pruefungs-modalitaeten</b>	Teilnahme an den einleitenden Vorlesungen und den praktischen Übungen. Bestehen einer praktischen Abschlussprüfung.	
<b>Nuetzliche Literatur</b>	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007 Christian Hörr. Algorithmen zur automatisierten Dokumentation und Klassifikation archäologischer Gefäße. Dissertation, TU Chemnitz, 2011, Hubert Mara, Multi-Scale Integral Invariants for Robust Character Extraction from Irregular Polygon Mesh Data, Dissertation, Uni Heidelberg 2012	

## Gaussian Processes for Machine Learning

<b>Code</b> IGPML	<b>Name</b> Gaussian Processes for Machine Learning	
<b>Leistungspunkte</b> 5 LP	<b>Dauer</b> one semester	<b>Turnus</b> irregular
<b>Lehrform</b> Lecture 2 SWS	<b>Arbeitsaufwand</b> 150 h; thereof 30 h lecture 100 h project 20 h report	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>To build a solid background on both the theory of Gaussian processes (GPs) and how they are used in practice to build effective machine learning models.</p> <ul style="list-style-type: none"> <li>* Firm theoretical knowledge on how to use GPs for machine learning.</li> <li>* Knowledge on how big data can be modeled with GPs.</li> <li>* Practice on how to design, develop, and evaluate a powerful machine learning model.</li> </ul>	
<b>Inhalt</b>	<p>This module covers the following topics:</p> <ul style="list-style-type: none"> <li>* Introduction to and motivation for GPs.</li> <li>* Predicting real-valued and categorical output with GPs.</li> <li>* Approximate inference of GPs.</li> <li>* Modeling big data with GPs.</li> <li>* Exploratory data analysis and knowledge discovery with GPs.</li> <li>* Time series modeling with GPs.</li> <li>* Deep learning with GPs.</li> <li>* GPs for alternative learning setups.</li> </ul>	
<b>Voraussetzungen</b>	recommended are: basic background on probability and statistics, basic knowledge on machine learning and linear algebra	
<b>Pruefungsmodalitaeten</b>	<p>Successfully handing in the source code of a working GP-based machine learning model implementation and the related report (approximately four pages, single column). The students are free to choose the learning setup to study with GPs and the data set to evaluate the learning model on.</p>	

<b>Nuetzliche Literatur</b>	Carl E. Rasmussen, Christopher I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006 (online) Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2007
---------------------------------	--

## Gemischt-ganzzahlige Programmierung und kombinatorische Optimierung

<b>Code</b> IMIP	<b>Name</b> Gemischt-ganzzahlige Programmierung und kombinatorische Optimierung	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, praktischer Teil (Übungen) 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 20 h Prüfungsvorbereitung 100 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>kennen die Modellierungsmöglichkeiten der gemischt-ganzzahligen Optimierung</li> <li>sind mit einer kommerziellen MIP-Software vertraut</li> <li>haben einen guten Überblick über die Anwendungsmöglichkeiten</li> <li>sind mit den polyedertheoretischen Grundlagen der Optimierung vertraut</li> <li>kennen die wichtigsten Optimierungstechniken</li> <li>haben die erforderlichen Kenntnisse zur Beurteilung von Möglichkeiten und Grenzen der praktischen Verwendbarkeit der erlernten Techniken</li> <li>haben Expertise in der Modellierung diskreter und ganzzahliger Optimierungsprobleme und dem Einsatz kommerzieller Optimierungssoftware</li> </ul>	

<b>Inhalt</b>	Lineare Programmierung (Polyedertheorie, Dualität, Simplex-Algorithmus, postoptimale Analyse) gemischt-ganzzahlige Modellierungen Kombinatorische Optimierungsprobleme Bestimmung von Optimallösungen Polyhedrische Kombinatorik Kombinatorische Polytope Relaxierungen (kombinatorische, LP-, semidefinite , Lagrange-Relaxierungen) Branch-and-Cut Algorithmen Zulässige Ungleichungen, Schnittebenen Presolve Techniken für große Probleme (Spaltengenerierung, Benders- und Dantzig-Wolfe-Dekomposition) Anwendungen
<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse Mathematik, Programmierkurs (IPK)
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
<b>Nützliche Literatur</b>	z.B. Kallrath, J. and Wilson, J.M.: Business Optimisation using Mathematical Programming, Macmillan Press, 1997 Williams, H.P.: Model Building, Wiley, 1999

## Geometric Modeling and Animation

<b>Code</b> IGMA	<b>Name</b> Geometric Modeling and Animation	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> every 3rd semester
<b>Lehrform</b> Lecture 4 SWS, Exercise 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>The students</p> <p>know the mathematical foundations of geometric modeling</p> <p>know the mathematical and physical foundations of computer animation</p> <p>know the algorithms and implementation aspects</p> <p>are familiar with the basics of animated movies</p> <p>are able to apply existing tools for geometric modeling and animation</p>	
<b>Inhalt</b>	<p>Introduction to curves</p> <p>Interpolating curves</p> <p>Bézier curves</p> <p>B-Splines</p> <p>Rational curves</p> <p>Introduction to surfaces</p> <p>Tensor product surfaces</p> <p>Transfinite surfaces and extrusion</p> <p>Subdivision</p> <p>Subdivision surfaces</p> <p>Animation and simulation</p> <p>Rigid body kinematics</p> <p>Particle systems</p> <p>Mass-spring models</p> <p>Cloth modeling</p> <p>Numerical methods for differential equations</p> <p>Collision detection and handling</p> <p>Fluid simulation and natural phenomena</p>	
<b>Voraussetzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Prüfungsmodalitäten</b>	Successful participation in the exercises (more than 50 % have to be scored) und passing a written or oral exam	

<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>- Curves and Surfaces for CAGD ? A Practical Guide, G. Farin, Morgan Kaufmann, 2002</li> <li>- Computer Animation ? Algorithms and Techniques, R. Parent, Morgan Kaufmann, 2002</li> <li>- 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, D. Eberly, Morgan Kaufmann, 2000</li> <li>- Graphische Datenverarbeitung I, J. Encarnacao, W. Straßer, R. Klein, 4. Auflage, Oldenbourg 1996</li> <li>- Advanced Animation and Rendering Techniques, A. Watt, M. Watt, Addison-Wesley, 1992</li> <li>- Grundlagen der geometrischen Datenverarbeitung, J. Hoschek, D. Lasser, Teubner 1992</li> <li>- Numerical Recipes ? The Art of Scientific Computing, W.H. Press, P. Flannery, S.A. Teukolsky, W.T. Vetterling, Cambridge University Press, 1986</li> </ul>
---------------------------------	--

## Inverse Problems

<b>Code</b> IIP	<b>Name</b> Inverse Problems	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS und Hausarbeiten	<b>Arbeitsaufwand</b> 240 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 165 h Selbststudium und Aufgabenbearbeitung, Hausarbeiten	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	Die Studierenden verstehen, was inverse Probleme sind und warum sie schwer zu lösen sind. Sie lernen die Prinzipien kennen, wie man sowohl deterministische Probleme, als auch stochastische Probleme löst und dabei auch die Regularisierungsparameter geeignet wählt. Schließlich erfahren sie die neuesten Entwicklungen im Bereich compressed sensing. Alle Prinzipien werden an zwei ausgewählten Gebieten, der Tomographie und des Deblurrings dargestellt. Sie erhalten damit die Kompetenz komplexe Probleme zu lösen, die mit klassischen Techniken nicht stabil lösbar sind und sind damit in der Lage auch komplexe experimentelle Messungen adäquat auswerten zu können.	
<b>Inhalt</b>	Deterministische inverse Probleme Stochastische inverse Probleme Wahl der Regularisierungsparameter Compressed sensing Tomographie Deblurring	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Numerische Mathematik	
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 60% der Punkte müssen erreicht werden) und Bestehen einer mündlichen Abschlussprüfung	
<b>Nuetzliche Literatur</b>	M. Bertero, P. Boccacci: Introduction to Inverse Problems in Imaging, IoP, 2002 web-Page and book: <a href="http://www.slaney.org/pct/pct-toc.html">http://www.slaney.org/pct/pct-toc.html</a>	

## IT-Projektmanagement

<b>Code</b> IPM	<b>Name</b> IT-Projektmanagement	
<b>Leistungspunkte</b> 3 LP als Spezialvorlesung Angewandte Informatik 6 LP als EPG	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung+Übung 2 SWS für eine Anrechnung als EPG mit zusätzlichem Forschungsprojekt	<b>Arbeitsaufwand</b> Vorlesung + Übung 90 h insgesamt, davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)  für eine Anrechnung als EPG zusätzlich: 90 h Forschungsprojekt, davon 15 h Einarbeitung und Literaturrecherche 15 h Vorbereitung einer Studie, z.B. Interview-Training 20 h Durchführung einer Studie 20 h Auswertung 20 h Abschlusspräsentation + Berichterstellung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Die Teilnehmer/innen können ein Projekt planen und überwachen, verstehen, wie Projekte in Organisationen eingebettet sind und haben Grundkenntnisse in vertraglichen Themen. Die Teilnehmer/innen des EPG können ein Forschungsprojekt selbständig durchführen und kennen Forschungsergebnisse aus der Gender-Forschung.	

<b>Inhalt</b>	Projektplanung, Projektorganisation Kostenschätzung Angebot/ Vertrag, Verhandeln Vorgehensmodelle Risikomanagement Controlling IT-Vertragsrecht Änderungsmanagement Zeitmanagement Projektabschluss Verteilte Softwareentwicklung
<b>Voraussetzungen</b>	keine
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Hausaufgaben (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer Abschlussprüfung. Für EPG: Zusätzlich erfolgreiche Teilnahme am Forschungsprojekt. Die Note für das Forschungsprojekt geht zur Hälfte in die Endnote ein.
<b>Nuetzliche Literatur</b>	PMI (Project Management Institute): A Guide to the Project Management Body of Knowledge (PM BOK ® Guide), 4. Ausgabe 2008

## IT-Sicherheit

<b>Code</b> IITS	<b>Name</b> IT-Sicherheit	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 120 h, davon 30 h Präsenzzeit 60 h Vor- und Nachbereitung 30 h Klausurvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Studierende</p> <ul style="list-style-type: none"> <li>* erwerben umfangreiches Wissen über die Funktionsweise und Verwundbarkeiten vernetzter Computersysteme, und können somit Konzepte zur IT-Netzsicherheit bewerten und entwerfen</li> <li>* erlangen grundlegende Kenntnisse über die Sicherung großer Netzwerke und der Kommunikationsinfrastruktur (Routing, Namensauflösung, Internet-Firewalls, Intrusion Detection Systeme)</li> <li>* erwerben Kenntnisse im Bereich Kryptographie: Theorie der Kryptographie und praktische Umsetzung typischer kryptographischer Verfahren im Zusammenhang mit kryptographischen Prüfverfahren, symmetrischen und asymmetrischen Chiffrierverfahren</li> <li>* erlangen grundlegende Expertise im Bereich der Kryptoanalyse</li> <li>* erwerben umfassende Kompetenzen zur Detektion von Cyberangriffen; grundlegende Kompetenzen im Feld der IT-/Cyber-Forensik</li> <li>* erwerben praktische Erfahrungen bei der Verwendung von dedizierter Software zur Detektion von Angriffsszenarien im Datennetz</li> </ul> <p>Langfristiges Ausbildungsziel: Einsatz-/Beschäftigungsfähigkeit in der Breite des Arbeitsfeldes IT-Sicherheit;</p>	

<b>Inhalt</b>	<p>Der IT-Sicherheit kommt bei der allgegenwärtigen Digitalisierung eine Schlüsselrolle zu. Diese Vorlesung vermittelt methodische Ansätze zur Modellierung und Bewertung von Angriffsszenarien, auf Basis welcher wirksame technische Gegenmaßnahmen umgesetzt werden können. Insbesondere werden folgende Schwerpunkte adressiert:</p> <ul style="list-style-type: none"> <li>- Sicherheitsmodelle und Bewertungskriterien</li> <li>- Kryptographische Prüfwerte: Modifikationserkennungs- und Nachrichtenauthentisierungswerte</li> <li>- Symmetrische und asymmetrische kryptographische Verfahren</li> <li>- Kryptographische Protokolle</li> <li>- Zugriffskontrolle</li> <li>- Schutz von Kommunikationsinfrastruktur</li> <li>- Digitale Identität</li> </ul> <p>Mit Hilfe von virtuellen Maschinen in einem geschützten Bereich werden klassische Angriffs- und Schutzszenarien praktisch untersucht. Hier werden sog. Experimentierblätter samt Daten zur Verfügung gestellt.</p>
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Einführung in die Technische Informatik (ITE), Algorithmen und Datenstrukturen (IAD)
<b>Pruefungs-modalitaeten</b>	Regelmäßige Teilnahme, Abschlussklausur (90 Min.). Art und Zeitpunkt einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
<b>Nuetzliche Literatur</b>	<p>C. Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle De Gruyter Studium. Oldenbourg: de Gruyter, 2014.</p> <p>T.W. Harich, IT-Sicherheitsmanagement: Arbeitsplatz IT Security Manager. MITP, 2012.</p> <p>K. Schmeh, Kryptographie: Verfahren, Protokolle, Infrastruktur. 5. Auflage. Heidelberg, Dpunkt Verlag, 2013.</p>

## Knowledge Discovery in Databases

<b>Code</b> IKDD	<b>Name</b> Knowledge Discovery in Databases	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> every 2nd winter semester
<b>Lehrform</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h lecture 20 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>Students</p> <ul style="list-style-type: none"> <li>- understand the KDD process and when to apply different KDD tasks</li> <li>- are able to apply suitable data mining techniques to specific data analysis problem</li> <li>- know the foundations of statistics and probability theory underlying diverse data mining techniques</li> <li>- can apply and adopt different data clustering algorithms and models</li> <li>- can apply and adopt different data classification algorithms and models</li> <li>- understand different methods and metrics to evaluate the quality of data mining results</li> <li>- can describe different pattern detection methods to obtain frequent patterns from diverse types of data sets</li> <li>- are familiar with the foundations of models and techniques to extract patterns from graph data</li> <li>- can apply and realize the different algorithms and data mining procedures in software environments such as R or Python</li> </ul>	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- KDD process and tasks</li> <li>- Data, statistics, and probability theory</li> <li>- Clustering models, techniques, and algorithms</li> <li>- Classification models, techniques, and algorithms</li> <li>- Frequent pattern mining approaches</li> <li>- Outlier detection concepts</li> <li>- Graph mining models and methods</li> </ul>	
<b>Voraussetzungen</b>	recommended are: Algorithmen und Datenstrukturen (IAD), Effiziente Algorithmen 1 (IEA1), Einführung in die Wahrscheinlichkeitstheorie und Statistik (MA8)	

<b>Pruefungs- modalitaeten</b>	Assignments; at least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final written exam; students can also work on a project (non-graded); final written exam
<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>- Jiawei Han, Micheline Kamber, and Jian Pei: Data Mining. Concepts and Techniques, Morgan Kaufmann Series in Data Management Systems, 2011 (3rd Edition).</li> <li>- Charu Aggarwal: The textbook. Springer, 2015.</li> <li>- Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: Introduction to Data Mining. Addison Wesley, 2005.</li> </ul>

## Machine Learning

<b>Code</b> IML	<b>Name</b> Machine Learning	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> in (irregular) alternation with *Fundamentals of Machine Learning* + *Advanced Machine Learning*
<b>Lehrform</b> 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials 120h lecture wrap-up and homework 30h preparation for examination	<b>Verwendbarkeit</b> cannot be combined with *Fundamentals of Machine Learning* or *Advanced Machine Learning* B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing, M.Sc. Physik (specialization Computational Physics)
<b>Lernziele</b>	Students understand a broad range of machine learning concepts, get to know established and advanced learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results. In addition, students learn how to use Python-based machine learning software such as scikit-learn.	
<b>Inhalt</b>	This lecture is a compact version of the two-semester course *Fundamentals of Machine Learning* + *Advanced Machine Learning*: Classification (linear and quadratic discriminant analysis, neural networks, linear and kernelized support vector machines, decision trees and random forests), least squares and regularized regression, Gaussian processes, unsupervised learning (density estimation, cluster analysis, Gaussian mixture models and expectation maximization, principal component analysis, bilinear decompositions), directed probabilistic graphical models, optimization for machine learning, structured learning	
<b>Voraussetzungen</b>	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
<b>Pruefungsmodalitaeten</b>	cannot be combined with *Fundamentals of Machine Learning* or *Advanced Machine Learning*, successful homework solutions (at least 50% of total achievable points) and oral examination	

<b>Nuetzliche Literatur</b>	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009; David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012
---------------------------------	---

## Mining Massive Datasets

<b>Code</b> IMMD	<b>Name</b> Mining Massive Datasets	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 105 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>* kennen ausgewählte Ansätze und Programmierparadigmen der parallelen Datenverarbeitung</li> <li>* können Tools zur parallelen Datenverarbeitung (u.a. Apache Hadoop und Spark) anwenden</li> <li>* wissen die Anwendungsbereiche der Analyse großer Datenmengen</li> <li>* kennen Methoden der parallelen Vorverarbeitung von Daten</li> <li>* kennen Analyseverfahren wie Klassifikation, Regression, Clustering sowie von deren parallelen Implementierungen</li> <li>* wissen die theoretischen und praktische Aspekte der Skalierbarkeit der parallelen Datenverarbeitung</li> </ul>	
<b>Inhalt</b>	<p>Zu den Inhalten zählen:</p> <ul style="list-style-type: none"> <li>* Programmierparadigmen für parallel-verteilte Datenverarbeitung, insbes. Map-Reduce und Spark-Programmiermodell</li> <li>* Praktische Kenntnisse von Apache Hadoop, Pig, und Hive, sowie Spark und ggf. anderer Frameworks für parallel-verteilte Datenverarbeitung</li> <li>* Anwendungsbereiche der parallelen Datenanalyse u.a. Clustering, Recommendation, Suche nach ähnlichen Objekten, Mining von Datenströmen</li> <li>* Verfahren zur parallelen Vorverarbeitung der Daten</li> <li>* Grundlagen der Analysetechniken wie Klassifikation, Regression, Clustering und Evaluation der Ergebnisse</li> <li>* Parallele Algorithmen für die Datenanalyse und ihre Umsetzung</li> </ul> <p>* Theorie und Praxis der Skalierbarkeit, Tuning der Algorithmen und Frameworks</p>	

<b>Voraussetzungen</b>	empfohlen sind: Programmierkenntnisse (z.B. in Java) und elementare Wahrscheinlichkeitsrechnung; Modul Knowledge Discovery in Databases (IKDD) ist empfohlen aber nicht notwendig. Modul Big Data (IBD) wird als Ergänzung bzw. Erweiterung empfohlen
<b>Pruefungsmodalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>* Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman: Mining of Massive Datasets, Cambridge University Press, Version 2.1 von 2014 (<a href="http://www.mmids.org/">http://www.mmids.org/</a>)</li> <li>* Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009 (<a href="http://statweb.stanford.edu/~tibs/ElemStatLearn/">http://statweb.stanford.edu/~tibs/ElemStatLearn/</a>)</li> <li>* Ron Bekkerman, Misha Bilenko, John Langford: Scaling Up Machine Learning, Cambridge University Press, 2012</li> <li>* Jiawei Han, Micheline Kamber, Jian Pei: Data Mining: Concepts and Techniques, Morgan Kaufmann, (third edition), 2012</li> <li>* Bücher aus dem O'Reilly Data Science Starter Kit, 2014 (<a href="http://shop.oreilly.com/category/get/data-science-kit.do">http://shop.oreilly.com/category/get/data-science-kit.do</a>)</li> </ul>

## Modellierung und Simulation in den Neurowissenschaften

<b>Code</b> IMSN	<b>Name</b> Modellierung und Simulation in den Neurowissenschaften	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 25 h Prüfungsvorbereitung 95 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden kennen den makro-/mikroskopischen Aufbau eines Gehirns und verstehen die Prinzipien neuronaler Signalverarbeitungsprozesse. sind in der Lage die Eigenschaften individueller Neuronenmodelle mit Methoden aus dem Bereich der dynamischen Systeme zu analysieren.</p> <p>können mathematische Modelle von Einzelneuronen selbst entwerfen und numerische Verfahren zur Lösung bestimmen.</p> <p>haben die Fähigkeit die erforderlichen numerischen Methoden in einer Programmiersprache (C/C++ ) eigenständig umzusetzen.</p> <p>haben die Kenntnis die Simulationsergebnisse mathematisch-informatisch zu analysieren und zu interpretieren.</p>	

<b>Inhalt</b>	<p>Grundkenntnisse in Neuroanatomie und -physiologie          Bioelektrizität: Biophysikalische Grundlagen von Ionenströmen, Elektrophysiologie, Nernstpotential          Biomembranmodelle          Modellierung von Signalverarbeitungsprozessen im extrazellulären Raum          Integrate-and-Fire Modelle / Neuronenmodelle mit einem Kompartiment          Numerische Lösungsmethoden für Einkompartimentmodelle          Dynamische Systeme: Begriffsbildungen, Phasenraumanalyse, Stabilität und Hysterese, Bifurkationen, Spiking, Bursting          Numerische Lösungsverfahren für steife Systeme          Modellierung und Simulation der passiven Kabelgleichung          Diskretisierung der Kabelgleichung          Das Hodgkin-Huxley Modell (HH) der aktiven Signalverarbeitung            Methoden zur nichtlineare Kopplung von HH und Kabelgleichung            Simulation stochastischer Prozesse</p>
<b>Voraussetzungen</b>	empfohlen sind: Grundlagen in Numerik wie z:B. aus MA7
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
<b>Nuetzliche Literatur</b>	C. Koch: Biophysics of Computation: Information Processing in Single Neurons. Oxford Univ. Press, 1999, ISBN 0-19-518199-9

## Object-Oriented Programming for Scientific Computing

<b>Code</b> IOPSC	<b>Name</b> Object-Oriented Programming for Scientific Computing	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> one semester	<b>Turnus</b> every summer semester
<b>Lehrform</b> Lecture 2 SWS, Exercise on computer 2 SWS	<b>Arbeitsaufwand</b> 180 h; thereof 60 h lecture 105 h self-study and working on assignments 15 h preparation for exam	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	The students are proficient in the programming language C++, can assess the performance of different programming techniques, know template programming techniques, and can use the Standard Template Library (STL). They can apply their new skills to solve selected problems of Scientific Computing.	
<b>Inhalt</b>	This module deepens the skills in object-oriented programming obtained in the basic lecture Einführung in die Praktische Informatik (IPI) with special emphasis on Scientific Computing: Class concept Dynamic memory allocation Exception handling Resource allocation and initialization Constness Static versus dynamic polymorphism Traits and Policies Standard Template Library Template Metaprogramming Parallel programming techniques	
<b>Voraussetzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), alternatively basic knowledge of an object-oriented programming language	
<b>Pruefungs- modalitaeten</b>	Successful participation in the exercises (getting at least 50% of the points), written exam at the end of the semester	
<b>Nuetzliche Literatur</b>		

## Object Recognition and Image Understanding

<b>Code</b> IORIU	<b>Name</b> Object Recognition and Image Understanding	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> at least every 4th semester
<b>Lehrform</b> Lecture 4 h + Exercise course 2 h	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials 100h lecture wrap-up and homework 50h preparation for project work, examination	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik M.Sc. Scientific Computing
<b>Lernziele</b>	<p>To have reached understanding of the state-of-the-art in mid- and high-level Computer Vision and to have the ability to relate and contrast different concepts.</p> <p>To be able to apply essential algorithms from pattern recognition and deep learning to current problems in machine vision.</p> <p>To be capable of understanding and analyzing the latest publications in Computer Vision and to evaluate their strengths and weaknesses.</p> <p>To have a firm command of the algorithmic basics and to be able to analyze and solve object recognition problems in novel application areas.</p> <p>To know the most relevant methods for robust object representation and to judge them based on their applicability and restrictions.</p>	
<b>Inhalt</b>	<p>Methods in mid- and high-level Computer Vision, esp.:</p> <ul style="list-style-type: none"> <li>- object detection and classification</li> <li>- deep learning in Computer Vision and convolutional neural networks</li> <li>- machine learning approaches for object representation</li> <li>- video analysis</li> <li>- recognition of human actions</li> <li>- local and global feature extraction</li> <li>- model based approaches</li> <li>- view based approaches</li> <li>- generative/discriminative methods</li> <li>- supervised/unsupervised methods</li> <li>- registration</li> <li>- shape analysis</li> <li>- voting and hashing methods</li> <li>- hierarchical object representations</li> </ul>	

<b>Voraussetzungen</b>	recommended are: solid programming skills and knowledge of basic calculus, statistics, and linear algebra
<b>Pruefungsmodalitaeten</b>	Successful homework solutions, project work with final report or oral/written examination
<b>Nuetzliche Literatur</b>	Will be presented in class

## Optimization for Machine Learning

<b>Code</b> IOML	<b>Name</b> Optimization for Machine Learning	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> every winter semester
<b>Lehrform</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 60 h lectures 30 h exercises 24 h preparation for exam 126 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik M.Sc. Angewandte Informatik M.Sc. Mathematik M.Sc. Scientific Computing
<b>Lernziele</b>	<p>The students</p> <ul style="list-style-type: none"> <li>- can analyze optimization methods for machine learning problems and estimate the area of their potential application</li> <li>- can competently apply existing algorithms and program packages for inference and learning with graphical models and neural networks</li> <li>- know typical optimization techniques for inference and learning with graphical models and neural networks</li> <li>- understand the basics of convex analysis, convex optimization, convex duality theory, (integer) linear programs and their geometry</li> </ul>	

<b>Inhalt</b>	<p>The course presents various existing optimization techniques for such important machine learning tasks, as inference and learning for graphical models and neural networks. In particular, it addresses such topics as combinatorial algorithms, integer linear programs, scalable convex and non-convex optimization and convex duality theory. Graphical models and neural networks play a role of working examples along the course. The content of the course includes:</p> <ul style="list-style-type: none"> <li>- Convex analysis and optimization: convex sets and functions, polyhedra, (integer) linear programs, basic first-order convex optimization methods and their stochastic variants, LP and Lagrange relaxations</li> <li>- Graphical Models: dynamic programming, sub-gradient and block-coordinate ascent inference methods, min-cut/max-flow based inference, structured risk minimization for graphical models</li> <li>- neural networks: architectures, backpropagation algorithm, stochastic gradient descent and its variants for training neural networks.</li> </ul>
<b>Voraussetzungen</b>	recommended are: linear algebra, analysis and any universal programming language (e.g. C/C++/Pascal/python)
<b>Pruefungs-modalitaeten</b>	Oral or written exam (is defined depending the number of students at the beginning of the course). At least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final exam.
<b>Nuetzliche Literatur</b>	Will be given by lecturer at the beginning of the course

## Parallele Lösung großer Gleichungssysteme

<b>Code</b> IPLGG	<b>Name</b> Parallele Lösung großer Gleichungssysteme	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	Die Studierenden beherrschen die Diskretisierung elliptischer Differentialgleichungen mittels Finiten Elemente verstehen das abstrakte Konzept der Teilraumkorrekturverfahren können dies im Rahmen von überlappenden und nichtüberlappenden Gebietszerlegungsverfahren sowie Mehrgitterverfahren umsetzen beherrschen die Konvergenztheorie zu diesen Verfahren können ausgewählte Verfahren praktisch auf Parallelrechnern umsetzen und der Leistungsfähigkeit bewerten	
<b>Inhalt</b>	Grundlagen der Diskretisierung elliptischer partieller Differentialgleichungen, Teilraumkorrekturverfahren überlappende und nichtüberlappende Gebietszerlegungsverfahren mit Konvergenztheorie geometrische Mehrgitterverfahren mit Konvergenztheorie algebraische Mehrgitterverfahren	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse einer objektorientierten Programmiersprache, MA7, Kenntnisse wie sie das Modul IPHR oder ICC1 vermittelt, Vorkenntnisse in Numerik, insbesondere Differentialgleichungen (MH7).	
<b>Prüfungs- modalitäten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte) und Bestehen einer Abschlussprüfung.	
<b>Nützliche Literatur</b>		

## Paralleles Höchstleistungsrechnen

<b>Code</b> IPHR	<b>Name</b> Paralleles Höchstleistungsrechnen	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>Die Studierenden</p> <p>kennen die grundlegenden Architekturen paralleler Höchstleistungsrechner</p> <p>kennen die grundlegenden Synchronisations-mechanismen in parallelen Systemen inklusive Performanzaspekten</p> <p>beherrschen die wichtigsten Programmierparadigmen für parallele Systeme</p> <p>sind in der Lage grundlegende Synchronisationsaufgaben zu lösen</p> <p>beherrschen Algorithmen der parallelen linearen Algebra</p> <p>können die Qualität paralleler Algorithmen und deren praktische Implementierung bewerten</p>	
<b>Inhalt</b>	<p>Rechner mit globalem Adressraum</p> <p>Cache-Kohärenz</p> <p>Rechner mit lokalem Adressraum und Nachrichtenaustausch</p> <p>kritischer Abschnitt, Bedingungsynchronisation Semaphore</p> <p>Posix Threads</p> <p>Programmierung von Grafikkarten</p> <p>Nachrichtenaustausch</p> <p>MPI</p> <p>Client-Server Architekturen, entfernter Prozeduraufruf</p> <p>Bewertung paralleler Algorithmen</p> <p>Lastverteilung</p> <p>Algorithmen für vollbesetzte Matrizen</p> <p>Lösung dünnbesetzter Gleichungssysteme</p> <p>Partikelmethode</p> <p>Paralleles Sortieren</p>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse einer Programmiersprache (Programmieren und Softwaretechnik) und grundlegender Algorithmen (Algorithmen und Datenstrukturen (IAD)).	

<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50%) und Bestehen einer Abschlussprüfung.
<b>Nuetzliche Literatur</b>	

## Praktische Geometrie

<b>Code</b> IPG	<b>Name</b> Praktische Geometrie	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 1 SWS	<b>Arbeitsaufwand</b> 120 h; davon 45 h Präsenzstudium 60 h Aufgabenbearbeitung 15 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Verständnis grundlegender geometrischer Konzepte zur Datenanalyse sowie effektive Punktsuche und Weiterverarbeitung von Messdaten Souveräner Umgang mit Projektionen und Beschreibungen jenseits der dreidimensionalen Erfahrungswelt Berechnung geometrischer Invarianten, Distanzen, Krümmungen aus Messdaten, rekonstruierten und generierten Flächen	
<b>Inhalt</b>	Grundlegende Gebiete der Geometrie mit Relevanz in Computergraphik, Bildverarbeitung, Mustererkennung, Computer Vision und Geometrischem Modellieren (i) Analytische Geometrie: Operationen auf Vektorräumen mit geeigneten Koordinaten und Abbildungen (Affinitäten, Kollinearitäten), geometrische Ausgleichsprobleme aus fehlerbehafteten Messdaten (ii) Projektive Geometrie: Zentralprojektion und inverse Rekonstruktion von 3D-Objekten aus ebenen Bildern (Computer Vision, Geodäsie), Unterschiede zwischen B-Spline-Kurven und -Flächen und der Klasse der NURBS, Freiformgeometrien in CAD-Systemen (iii) Differentialgeometrie: Parameterdarstellungen in der geometrischen Datenverarbeitung, implizite Darstellungen (level sets), Abschätzung von Invarianten aus diskreten Daten (Triangulierungen, Punktwolken)	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker (IMI1 und 2) oder Lineare Algebra (MA4)	
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50 % der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	Geometrie für Informatiker, Skript TU Wien 2004, Helmut Pottmann Aktuelle Fachveröffentlichungen
---------------------------------	--

## Projektseminar Biomedizinische Bildanalyse

<b>Code</b> IPBB	<b>Name</b> Projektseminar Biomedizinische Bildanalyse	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> 2 Teile Seminar und Projekt, 4 SWS	<b>Arbeitsaufwand</b> 180 h (je zur Hälfte Seminar und Projekt) 60 h Präsenzstudium 120 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden erlangen vertiefte Kenntnisse und Fähigkeiten im Gebiet Biomedizinische Bildanalyse</p> <p>lernen fortgeschrittene Methoden und Algorithmen zur automatischen Analyse biomedizinischer Bilder</p> <p>lernen wie man Algorithmen und Software für automatische Bildanalyse entwickelt</p> <p>erweitern ihre Fähigkeiten Projektergebnisse mündlich zu präsentieren und schriftlich zu dokumentieren</p> <p>erweitern ihre Fähigkeiten zur Teamarbeit und zur Strukturierung von Projekten</p>	
<b>Inhalt</b>	<p>Die Studierenden arbeiten in Teams an ausgewählten fortgeschrittenen Themen der Biomedizinischen Bildanalyse. Der Schwerpunkt liegt auf der automatischen Analyse von Zellmikroskopiebildern und medizinischen tomographischen Bildern. Beispiele für Themen sind die Segmentierung und Verfolgung (Tracking) von Zellen in Mikroskopiebildern, die Segmentierung von Blutgefäßen in tomographischen Bildern sowie die Registrierung von Magnetresonanz (MR) Bildern des menschlichen Gehirns. Die Veranstaltung besteht aus einem Seminarteil (Einarbeitung in die relevante Literatur, Erarbeitung der theoretischen Grundlagen, Vortragspräsentation) und einem Projektteil (Spezifikation eines Softwaresystems, Entwurf von Algorithmen und Implementierung von Bildanalyseverfahren, Test und Evaluierung der Verfahren, Präsentation der Ergebnisse).</p>	
<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse in Bildverarbeitung (Computer Vision, Image Analysis), Programmierkenntnisse, Kenntnisse in Software Engineering	

<b>Pruefungs- modalitaeten</b>	Vortragspraesentationen von Zwischen- und Endergebnissen (jeder Studierende 4 Vortraege je ca. 10 Min. und anschließender Diskussion) Schriftliche Ausarbeitung der theoretischen Grundlagen, der verwendeten Methoden und der Ergebnisse (jeder Studierende ca. 10 Seiten)
<b>Nuetzliche Literatur</b>	Bekanntgabe in der Lehrveranstaltung

## Qualitätsmanagement

<b>Code</b> ISWQM	<b>Name</b> Qualitätsmanagement	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Sommersemester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 3SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung im Team	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Dieses Modul vertieft die Grundkenntnisse und Fähigkeiten des Software Engineering insbesondere im Bereich des Qualitätsmanagement. Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge Fähigkeit, Qualitätssicherung, Qualitätsmanagement und Prozessverbesserung und -management einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen. Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit *echten* Kunden)	
<b>Inhalt</b>	Methoden, Prozesse und Werkzeuge für Qualitätssicherung Qualitätsmanagement Verbesserung von Softwareentwicklungsprozessen Management von Softwareentwicklungsprozessen	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (durch Erwerb einer entsprechenden Punktzahl) und erfolgreiche Teilnahme an der mündlichen oder schriftlichen Abschlussprüfung.	
<b>Nützliche Literatur</b>	Wird jährlich aktualisiert	

## Randomisierte Algorithmen

<b>Code</b> IRA	<b>Name</b> Randomisierte Algorithmen	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mindst. jedes 4. Semester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 1 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 40 h Prüfungsvorbereitung 80 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Auf der Grundlage der behandelten Anwendungsbeispiele aus verschiedenen Teilgebieten der Informatik können die Studierenden die probabilistische Betrachtungs- und Vorgehensweise anwenden bei der Konstruktion und Analyse von probabilistischen und deterministischen Algorithmen, auf kombinatorische Fragestellungen, um spieltheoretische Situationen zu analysieren, auf kryptographische Fragestellungen.	
<b>Inhalt</b>	Elementare Wahrscheinlichkeitsrechnung Das Tenure-Spiel Derandomisierungstechniken Die probabilistische Methode Byzantinische Übereinkunft Stabile Heiraten und der Gale-Shapley-Algorithmus Das Minimax-Prinzip von Yao Komplexitätsanalyse des randomisierten Sortierens Randomisierte Fehlersuche und -korrektur Das Local-Lemma von Lovasz PAC-Lernen und VC-Dimension Wahrscheinlichkeitsverstärkung und Fehlerschranken Lokale Suche für k-SAT Kryptographische Protokolle	
<b>Voraussetzungen</b>	empfohlen sind: elementare Grundkenntnisse in Algorithmen wie sie z.B. im Modul Algorithmen und Datenstrukturen (IAD) vermittelt werden.	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	R. Motwani und P. Raghavan, Randomized Algorithms, Cambridge University Press 1995. M. Mitzenmacher und E. Upfal, Probability and Computing, Cambridge University Press, 1995. N. Alon und J. H. Spencer, The Probabilistic Method, John Wiley and Sons, 2008.
---------------------------------	---

## Requirements Engineering

<b>Code</b> ISWRE	<b>Name</b> Requirements Engineering	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Sommersemester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 3SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung im Team	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Dieses Modul vertieft die Grundkenntnisse und Fähigkeiten des Software Engineering insbesondere im Bereich des Requirements Engineering. Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge Fähigkeit, Unternehmensmodellierung, Anforderungserhebung und -verhandlung und Requirements Management in einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen. Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit *echten* Kunden)	
<b>Inhalt</b>	Methoden, Prozesse und Werkzeuge für Unternehmensmodellierung Prozessverbesserung in Unternehmen Anforderungserhebung und -verhandlung Requirements Management (Verbreitung, Prüfung und Aktualisierung von Anforderungen)	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (durch Erwerb einer entsprechenden Punktzahl) und erfolgreiche Teilnahme an der mündlichen oder schriftlichen Abschlussprüfung.	
<b>Nuetzliche Literatur</b>	Wird jährlich aktualisiert	

## Scientific Visualization

<b>Code</b> ISV	<b>Name</b> Scientific Visualization	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> every 3rd semester
<b>Lehrform</b> Lecture 4 SWS, Exercise 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	The students understand fundamental and advanced concepts of scientific visualization. They understand the mathematical fundamentals, data structures, and implementation aspects. They get to know schemes for interpolation and integration, mapping for scalar, vector, and tensor fields, and derived approaches. The students understand approaches for direct and indirect volume rendering, feature extraction, and topology-based analysis. The students are able to apply these concepts to real-world problems using existing software packages, and develop small programs using visualization libraries.	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- Introduction</li> <li>- Visualization Process</li> <li>- Data Sources and Representation</li> <li>- Interpolation and Filtering</li> <li>- Approaches for Visual Mapping</li> <li>- Scalar Field Visualization: Advanced Techniques for Contour Extraction, Classification, Texture-Based Volume Rendering, Volumetric Illumination, Advanced Techniques for Volume Visualization, Pre-Integration, Cell Projection, Feature Extraction</li> <li>- Vector Field Visualization: Vector Calculus, Particle Tracing on Grids, Vector Field Topology, Vortex Visualization, Feature Extraction, Feature Tracking</li> <li>- Tensor Field Visualization: Glyphs, Hue-Balls and Lit-Tensors, Line-Based Visualization, Tensor Field Topology, Feature Extraction</li> </ul>	
<b>Voraussetzungen</b>	strongly recommended is: Computer Graphics (ICG) recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Prüfungsmodalitäten</b>	successful participation in the exercises (more than 50% of the achievable points) and passing of an oral or written exam	

<b>Nuetzliche Literatur</b>	C.D. Hansen, C.R. Johnson, The Visualization Handbook, 2005.
---------------------------------	--

## Selected Topics in the Theory of the Computably Enumerable Sets and Degrees

<b>Code</b> ITCESD	<b>Name</b> Selected Topics in the Theory of the Computably Enumerable Sets and Degrees	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS mit Übung 1 SWS	<b>Arbeitsaufwand</b> 120 h; davon 45 h Präsenzstudium 60 h Aufgabenbearbeitung 15 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Lernziele</b>	Kennenlernen von zentralen Konzepten und Ergebnissen in der Theorie der aufzählbaren Mengen, sowie das Erlernen der in diesem Gebiet verwendeten Methoden (insbesondere Prioritätsargumente)	
<b>Inhalt</b>	Computably enumerable (c.e.) sets play an important role in the foundations of mathematics, and the investigation of these sets and their degrees of unsolvability is one of the central areas in computability theory (recursive function theory). Many of the fundamental techniques of computability theory - like the priority method - have been introduced and developed in this area. In this advanced course we discuss some selected topics in this area including some very recent results. These topics include the low-high hierarchy, simplicity properties of c.e. sets (as simple, hyper-simple, hyper-hyper-simple and maximal and variations hereof), and the investigation of various types of multiple permitting arguments. (The lectures are in English.)	
<b>Voraussetzungen</b>	Empfohlen werden gute Grundkenntnisse der Berechenbarkeitstheorie	
<b>Pruefungs-modalitaeten</b>	Lösen von Übungsaufgaben und benotete mündliche Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nuetzliche Literatur</b>	Robert I. Soare. Turing Computability. Springer, 2016. Robert I. Soare. Recursively Enumerable Sets and Degrees. Springer, 1987. Rodney G. Downey; Denis R. Hirschfeldt. Algorithmic Randomness and Complexity. Springer, 2010.	

## Software Evolution

<b>Code</b> ISWEvol	<b>Name</b> Software Evolution	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90h; davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden können nach der Vorlesung: ein Software-Reengineering-Projekt fachlich planen und beurteilen, bei der Ersterstellung von Software die Evolutionsfähigkeit konzeptuell sicherstellen, ein Wartungskonzept für eine erstellte Software aufbauen.</p> <p>Sie kennen die Unterschiede und Herausforderungen der Software-Weiterentwicklung versus der Softwareneuentwicklung ? und worauf die/der InformatikerIn hierbei achten muss, sowohl aus Sicht eines Softwareherstellers als auch aus der Sicht der NutzerInnen von Software, die klassischen Techniken der Softwaresanierung, die Typologie der Softwarewartung und das Management der Fehlerbehebung, die Relevanz der Thematik in der Praxis der industriellen Softwareerstellung.</p>	

<b>Inhalt</b>	<p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen für ein erfolgreiches Lebenszyklusmanagement von Software nach ihrer Ersterstellung. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Dozenten in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <p>Einleitung          Begriffsklärung, Grundlagen          Softwareevolution          Softwarewartung, Softwareerhaltung          Software-Reengineering          Evolution und Weiterentwicklung          Management der Softwareevolution          Zusammenfassung</p>
<b>Voraussetzungen</b>	<p>empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden</p>
<b>Pruefungs-modalitaeten</b>	<p>Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Anzahl der Teilnehmer)</p>
<b>Nuetzliche Literatur</b>	<p>Arnold, R. (Hrsg.): Software Reengineering. IEEE Computer Society Press, Los Alamitos 1993</p> <p>Fowler, M.: Refactoring ? Improving the Design of Existing Code. Addison-Wesley, Reading, Massachusetts, 1999</p> <p>von Hahn, E.: Werterhaltung von Software. DUV, Wiesbaden 2005</p> <p>Müller, B.: Reengineering. Eine Einführung. Teubner, Stuttgart 1997</p> <p>Sneed, H.M.; Hasitschka, M.; Teichmann, M.-T.: Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungs-systeme. dpunkt, Heidelberg 2005</p> <p>Smith, D.D.: Designing Maintainable Software. Springer, Heidelberg 1999</p>

## Software Ökonomie

<b>Code</b> ISWÖk	<b>Name</b> Software Ökonomie	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90h; davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Die Studierenden können nach der Vorlesung grob den Preis und die Lizenzierung einer erstellten Software ermitteln, die Vermarktung von Software planen und anstoßen, grob die Bilanz sowie die Gewinn- und Verlustrechnung eines Softwareherstellers verstehen, den Wert einer Software mit seinen verschiedenen Komponenten beurteilen, aus Sicht des Herstellers sowie aus Sicht der Nutzer, Preisverhandlungen zu Softwareprojekten planen.</p> <p>Sie kennen</p> <ul style="list-style-type: none"> <li>die Grundzüge der Kosten- und Leistungsrechnung (soweit sie für die Softwareerstellung relevant ist),</li> <li>die unterschiedlichen Vertragsarten, die im Umfeld der Softwareerstellung zum Einsatz kommen,</li> <li>die wichtigsten Verhandlungsstrategien bei der Verhandlung von Softwareverträgen,</li> <li>rechtliche Aspekte im Bereich der IT-Kriminalität,</li> <li>die Relevanz der Vorlesungsthemen in der Praxis der industriellen Softwareerstellung.</li> </ul>	

<b>Inhalt</b>	<p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen der Wirtschaftsinformatik, wie sie für die Softwareerstellung relevant sind. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Dozenten in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <p>Einleitung  Begriffsklärung, Grundlagen  Softwareökonomie  Management von Softwareprojekten  Wertermittlung von Software  Bepreisung von Software  Software-Marketing  Verhandlungen und Verträge  Bilanzierung und Rechnungslegung  IT-Kriminalität  Schadensabwendung  Zusammenfassung</p>
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden
<b>Pruefungs-modalitaeten</b>	Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Anzahl der TeilnehmerInnen)
<b>Nuetzliche Literatur</b>	<p>Buxmann, P.; Diefenbach, H.; Hess, T.: Die Softwareindustrie. Ökonomische Prinzipien, Strategien, Perspektiven. Heidelberg, 2008</p> <p>Herzwurm, G.; Pietsch, W.: Management von IT-Produkten. Heidelberg, 2009</p> <p>Mertens, P. (Hrsg.): Lexikon der Wirtschaftsinformatik, 3. Auflage, Heidelberg 2001</p> <p>Versteegen, G.: Marketing in der IT-Branche. Heidelberg 2003</p> <p>Wöhe, G.: Einführung in die Allgemeine Betriebswirtschaftslehre. München 2010</p>

## Software-Praktikum für Fortgeschrittene

<b>Code</b> IFM	<b>Name</b> Software-Praktikum für Fortgeschrittene	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b>	<b>Turnus</b> jedes Semester
<b>Lehrform</b> Praktikum 6 SWS	<b>Arbeitsaufwand</b> 240 h; davon mind. 25 h Präsenzzeit 10 h Vorbereitung Vortrag	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik Fachübergreifende Kompetenzen: B.Sc. Mathematik, M.Sc. Mathematik
<b>Lernziele</b>	<p>Die Studierenden erlangen vertiefende Problemlösungskompetenz für komplexe Entwurfs- und Implementierungsaufgaben können Problemanalyse- und Beschreibungstechniken klar darstellen, differenzieren und anwenden vertiefen Programmierkenntnisse in der jeweiligen für das Projekt erforderlichen Programmiersprache sind in der Lage, das Projekt mit Hilfe einer Softwareentwicklungsumgebung durchzuführen</p> <p>Zusätzlich werden die projektypischen Kompetenzen vertieft, insbesondere das Arbeiten im Team (von bis zu drei Studierenden):</p> <p>Durchführung und Evaluation von Projekten und ihrer Phasenstruktur</p> <p>Planung und Durchführung von Projekt- und Teamarbeit.</p> <p>Zu den zu trainierenden Softskills zählen somit insbesondere Teamfähigkeit, Verfeinerung von Präsentationstechniken, etwaige Erschließung wissenschaftlicher Literatur sowie eigenverantwortliches Arbeiten.</p>	
<b>Inhalt</b>	<p>Domänenkenntnisse abhängig von den DozentInnen; allgemeine Lerninhalte sind:</p> <p>Vertiefung in die Projektarbeit</p> <p>Eigenständige Entwicklung von komplexer Software und deren Dokumentation</p>	
<b>Voraussetzungen</b>	keine	
<b>Pruefungs-modalitaeten</b>	Bewertung der dokumentierten Software, des Projektberichts und des Vortrags	
<b>Nuetzliche Literatur</b>		

## Spatial and Spatio-temporal Databases

<b>Code</b> ISTDB	<b>Name</b> Spatial and Spatio-temporal Databases	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> every 2nd to 3rd winter semester
<b>Lehrform</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h lecture 20 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>Students</p> <ul style="list-style-type: none"> <li>- have a solid understanding of concepts, models, and techniques underlying the modeling, management, processing, and querying of diverse types of spatial and spatio-temporal data</li> <li>- know important spatial and spatio-temporal indexing structures</li> <li>- can employ core algorithmic frameworks for processing spatial and spatio-temporal queries</li> <li>- can apply key algorithms adopted from computational geometry</li> <li>- can apply suitable data management techniques using systems such as PostgreSQL or PostGIS</li> </ul>	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- Principles and requirements of managing spatial data</li> <li>- Application domains of managing and analyzing spatial data</li> <li>Support of managing spatial data using commercial and open-source DBMS</li> <li>- Concepts and methods for representing spatial data in 2d and 3d; tessellation and vector models, groups of spatial objects, abstract data types for spatial data</li> <li>- Fundamentals of computational geometry (e.g., convex hull, sweep-line techniques, polygon partitioning, cut of polygons)</li> <li>- Access structures for spatial data, in particular grid-files, kd-tree, quad-tress, and R-trees</li> <li>- Algorithms and cost models for using spatial index structures</li> <li>- Principles of spatial query processing and optimization, in particular spatial join techniques</li> <li>- Spatio-temporal databases and index structures</li> <li>- Moving objects: applications, querying, and index structures</li> <li>- Introduction to mining spatio-temporal data</li> </ul>	
<b>Voraussetzungen</b>	recommended are: Algorithmen und Datenstrukturen (IAD), Datenbanken (IDB)	

<b>Pruefungs- modalitaeten</b>	Assignments; at least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final written exam; students can also work on a project (non-graded); final written exam
<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>- Philippe Rigaux, Michel Scholl, Agnes Voisard: Spatial Databases ? With Applications to GIS. Morgan Kaufmann, 2001.</li> <li>- Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars: Computational Geometry: Algorithms and Applications Springer, Berlin, 2008.</li> <li>- Xiaofeng Meng, Zhiming Ding, Jiajie Xu: Moving Objects Management: Models, Techniques and Applications. Springer, 2016.</li> </ul>

## Verteilte Systeme I

<b>Code</b> IVS1	<b>Name</b> Verteilte Systeme I	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 105 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>Verständnis der unterschiedlichen parallelen Architekturen und der Besonderheiten von verteilten Systemen</p> <p>Kenntnis der grundlegenden theoretischen Probleme und Algorithmen in verteilten Systemen (z.B. Skalierbarkeit)</p> <p>Fähigkeit zur Erstellung von parallelen und verteilten Programmen, insbes. im Spark-Programmiermodell</p> <p>Kenntnis der praktischen Anwendung diverser Programmierparadigmen und Frameworks (Pthreads, MPI, Hadoop, Spark) für parallele oder verteilte Programmierung</p> <p>Vertrautheit mit skalierbarer Verarbeitung von Daten</p> <p>Kenntnisse über Probleme und Lösungen in Bereichen Fehlertoleranz und Verlässlichkeit der verteilten Systeme</p>	
<b>Inhalt</b>	<p>Das Modul behandelt die grundlegenden Prinzipien der parallelen und verteilten Systeme im Kontext ihrer Programmierung, insbesondere zum Zwecke der skalierbaren Verarbeitung von Daten. Es werden Konzepte aus den Bereichen Architekturen, Protokolle, Algorithmen, Implementierung und Softwareframeworks vorgestellt. Ein wesentlicher Teil der Vorlesung widmet sich der praktischen parallelen und verteilten Programmierung. Dazu gehörten u.a. Ansätze wie MPI, Map-Reduce, Spark-Programmiermodell und Actors. Ergänzende Themen umfassen Fehlertoleranz, effiziente Protokolle und Skalierbarkeit. Die Umsetzung in die Praxis erfolgt an Beispielen der Verarbeitung großer Datenmengen. Das Modul soll die Studierenden befähigen, Spezifika und Probleme der verteilten Systeme zu verstehen, und effiziente verteilte Anwendungen mit Softwareframeworks wie Apache Spark zu erstellen.</p>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse in Java (z.B. durch Einführung in Software Engineering (ISW)) oder Python	

<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung.
<b>Nuetzliche Literatur</b>	<p>George Coulouris, Jean Dollimore, Tim Kindberg: Distributed Systems: Concepts and Design (4th ed.), Addison-Wesley, 2005.</p> <p>Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia: Learning Spark: Lightning-Fast Big Data Analysis, O'Reilly Media, 2015.</p> <p>Jure Leskovec, Anand Rajaraman, Jeff Ullman: Mining of Massive Datasets, 2nd edition (v2.1), Online: <a href="http://www.mmds.org/">http://www.mmds.org/</a></p> <p>Grams, A., Gupta, A., Karypis, G., Kumar V.: Introduction to Parallel Computing, Addison-Wesley, 2. Auflage, 2003.</p> <p>Andrew S. Tanenbaum, Maarten Van Steen: Distributed Systems: Principles and Paradigms, Prentice Hall, 2006.</p>

## Visualisierung im Bereich Cultural Heritage

<b>Code</b> IVCH	<b>Name</b> Visualisierung im Bereich Cultural Heritage	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Die Studierenden sind mit unterschiedlichen Scantechniken vertraut und können Georadardaten interpretieren. Sie beherrschen den Umgang mit 3D Scan-, Georadar- und Magnetfelddaten, geophysikalischer Prospektion und weiteren Untersuchungen von Messdaten und kennen die Herangehensweise mit 2D und 3D Bildverarbeitung zur Erkennung von Merkmalen (Schrift). Sie wissen um die ethischen Grundsätze bei der Rekonstruktion, Befund und Hypothese (London Charter).	
<b>Inhalt</b>	Weißlicht- und Time-of-flight-Scanner, Rekonstruktionen von Gefäßen und Gebäuden, 3D-Puzzle, Skelettierung, ethische Grundsätze	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Computergraphik 1 (ICG1)	
<b>Prüfungsmodalitäten</b>	Teilnahme an den Vorlesungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Teilnehmerzahl)	
<b>Nützliche Literatur</b>	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007 Christian Hörr. Algorithmen zur automatisierten Dokumentation und Klassifikation archäologischer Gefäße. Dissertation, TU Chemnitz, 2011	

## Volumenvisualisierung

<b>Code</b> IVV	<b>Name</b> Volumenvisualisierung	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 3 SWS	<b>Arbeitsaufwand</b> 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziele</b>	<p>Die Studierenden verstehen, wie experimentelle Daten zustande kommen und welche prinzipiellen Informationselemente diese enthalten, die für eine wissenschaftliche Visualisierung notwendig sind. Sie lernen die Prinzipien kennen, wie man zwischen kontinuierlichen (wirkliche Verteilung) und diskreten Signalen (Darstellung im Computer) wechseln kann. Zudem werden sie eingeführt in Methoden der Konversion von Oberflächendaten in Volumendaten und umgekehrt. Schließlich werden ausgehend von theoretischen physikalischen Prinzipien sie in der Lage sein, verschiedene Visualisierungstechniken abzuleiten und die Näherungen, die hierfür gemacht werden, zu verstehen.</p> <p>Sie sind damit in der Lage, komplexe volumenorientierte Daten adäquat mathematisch zu repräsentieren, zu transformieren und die wesentlichen Strukturen mit adäquat darauf angepassten Verfahren der Visualisierung darzustellen.</p>	
<b>Inhalt</b>	<p>Einführung in die Visualisierung wissenschaftlicher Daten aus den Natur- und Lebenswissenschaften.</p> <p>Diskrete und kontinuierliche Repräsentation von Daten sowie numerische und computergraphische Methoden der Interpolation.</p> <p>Methoden der Konversion von Oberflächenrepräsentationen in Volumenrepräsentationen und umgekehrt, Optimierungstechniken für effiziente Algorithmen</p> <p>Ableitung und Varianten von Volumenvisualisierungstechniken</p> <p>Beschleunigungsverfahren und Parallelisierungsmethoden für Volumenvisualisierung</p> <p>Programmiertechnik: GPU-Programmierung</p>	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	

<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
<b>Nuetzliche Literatur</b>	Engel et al: Real-Time Volume Graphics <a href="http://www.real-time-volume-graphics.org">www.real-time-volume-graphics.org</a> , Schroeder et al: VTK Textbook <a href="http://www.kitware.com/products/books/vtkbook.html">http://www.kitware.com/products/books/vtkbook.html</a>

## Wissensmanagement und Entscheidungen im Software Engineering

<b>Code</b> ISWKM	<b>Name</b> Wissensmanagement und Entscheidungen im Software Engineering	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung+Übung 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	Die Teilnehmer/innen kennen vertiefende Software Engineering Methoden, die Entscheidungen unterstützen bei Anforderungspriorisierung, Entwurf, Managemententscheidungen und Risikomanagement. Sie wissen, wie man im Arbeitsalltag Wissen verwaltet und haben eine Einführung in die Entscheidungstheorie erhalten.	
<b>Inhalt</b>	Wissensmanagement Ontologien Rationale Re-engineering learning organization Entscheidungen Management-Entscheidungen, Business Case Risikomanagement Anforderungspriorisierung Entscheidungen im Entwurf: ATAM, SAAM, CBAM Entscheidungstheorie Entscheiden unter Ungewissheit Mathematical Economics Entscheidung mit mehreren Parteien: Harvard-Konzept, Verhandlungen Spieltheorie Fehlentscheidungen/ Decision Traps/ Biases	
<b>Voraussetzungen</b>	empfohlen sind: Vorlesung und Übung Einführung in Software Engineering (ISW) oder vergleichbare Vorkenntnisse	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Hausaufgaben (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer Abschlussprüfung.	

<b>Nuetzliche Literatur</b>	Raiffa, Howard; Richardson, John; Metcalfe, David: Negotiation analysis - the science and art of collaborative decision making, Belknap, Cambridge, 2002
---------------------------------	--

## 2.4 Module aus dem B.Sc./M.Sc. Mathematik

**Bitte beachten: Welche Module bzw. Veranstaltungen aus dem Bachelor und Master Mathematik anrechenbar sind, wird gerade überarbeitet. In der nachfolgenden Aufzählung sind Module bzw. Veranstaltungen gelistet, die auf jeden Fall anrechenbar sind. Diese Liste wird im Sommersemester noch um weitere Module bzw. Veranstaltungen insbesondere aus dem Master Mathematik ergänzt werden.**

Aus dem Bachelor Mathematik mit 100% Fachanteil sind folgende Module anrechenbar:

- Wahrscheinlichkeitstheorie (MC4)
- Numerik (MD1)
- Statistik (MD2)
- Lineare Optimierung (MD3)
- Computational Statistics (MD6)
- Mathematische Logik (ME3)

Aus dem Master Mathematik sind aus den folgenden Modulen die folgenden Veranstaltungen anrechenbar:

- Grundmodul Numerik und Optimierung:  
Nichtlineare Optimierung  
Numerische Optimierung bei Differentialgleichungen I
- Grundmodul Statistik und Wahrscheinlichkeitsrechnung:  
Wahrscheinlichkeitstheorie II  
Statistik II

Aus den Ergänzungsmodulen:

- Berechenbarkeit und Komplexität I
- Berechenbarkeit und Komplexität II

## 2.5 Module aus dem M.Sc. Scientific Computing

### Fundamentals of Computational Environmental Physics

<b>Code</b> FCEP	<b>Name</b> Fundamentals of Computational Environmental Physics	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b>
<b>Lernziel</b>	Die Studenten – Lernen die Modellierung fundamentaler Prozesse der Umweltphysik mit Methoden der Kontinuumsmechanik – Lernen die Simulation dieser Modelle mit modernen Methoden der numerischen Mathematik kennen – Wenden diese Methoden auf konkrete umweltphysikalische Fragestellungen an	
<b>Inhalt</b>	Elementare, lineare Modelle 1) Einphasen-Strömung in porösen Medien / Elliptische partielle Differentialgleichungen (PDGLn) 2) Skalarer Transport / Hyperbolische Gleichung erster Ordnung 3) Wärmeleitung / parabolische PDGLn 4) Wellenausbreitung / Hyperbolische PDGLn zweiter Ordnung  Nichtlineare Modelle: 5) Gekoppelte elementare Modelle 6) Strömungsmechanik / Stokes- und Navier-Stokes-Gleichung	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Einführung in die Numerik, Numerik	

<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
<b>Nützliche Literatur</b>	

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 11.07.2013, ins Deutsche übersetzt

## 2.6 Module aus dem M.Sc. Technische Informatik

Alle fachlichen Module (nicht Fachübergreifende Kompetenzen) aus dem Master Technische Informatik können – entsprechend der inhaltlichen Voraussetzungen – auch im Master Angewandte Informatik belegt werden. Das Angebot findet sich im jeweils gültigen Modulhandbuch des Masters Technische Informatik.

## 2.7 Module aus dem B.Sc./M.Sc. Physik

### Physics of Imaging

<b>Code</b> MWInf5	<b>Name</b> Physics of Imaging	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b>	<b>Turnus</b>
<b>Lehrform</b> Lecture 4 SWS	<b>Arbeitsaufwand</b> 120 h	<b>Verwendbarkeit</b>
<b>Lernziel</b>	Basics of the Physics of Imaging; common principles and techniques of imaging for atomic to astronomical scales.	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>– Projective geometry, optics, wave optics, Fourier optics and lens aberrations</li> <li>– Radiometry of imaging</li> <li>– Methods of imaging: scanning electron microscopy, X-ray, EDX, FLIM, FRET, fluorescence imaging, near-field imaging</li> <li>– CCD and CMOS technology</li> <li>– Holography, ultrasound imaging, CT-computer tomography, magnetic resonance imaging</li> <li>– Satellite imaging, synthetic aperture radar, radio astronomy</li> </ul>	
<b>Voraussetzungen</b>	recommended are: UKInf2, PEP1 - PEP4	
<b>Prüfungsmodalitäten</b>	Defined by lecturer before beginning of course	
<b>Nützliche Literatur</b>		

Entnommen aus dem Modulhandbuch MScPhysik, Version 2012.2/v3.

## Image Processing

<b>Code</b> MWInf6	<b>Name</b> Image Processing	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b>	<b>Turnus</b>
<b>Lehrform</b> Lecture 4 SWS, Exercises 2 SWS	<b>Arbeitsaufwand</b> 210 h	<b>Verwendbarkeit</b>
<b>Lernziel</b>	Learn how to analyze signals from time series, images, and any kind of multidimensional signals and to apply it to problems in natural sciences, life sciences and technology.	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- Continuous and discrete signals, sampling theorem, signal representation</li> <li>- Fourier transform</li> <li>- Random variables and fields, probability density functions, error propagation</li> <li>- Homogeneous and inhomogeneous point operations</li> <li>- Neighbourhood operations, linear and nonlinear filters, linear system theory</li> <li>- Geometric transformations and interpolation</li> <li>- Multi-grid signal presentation and processing</li> <li>- Averaging, edge and line detection, local structure analysis, local phase and wave numbers</li> <li>- Motion analysis in image sequences</li> <li>- Segmentation</li> <li>- Regression, globally optimal signal analysis, variation approaches, steerable and nonlinear filtering, inverse filtering</li> <li>- Morphology and shape analysis, moments, Fourier descriptors</li> <li>- Bayesian image restoration</li> <li>- Object detection and recognition</li> </ul>	
<b>Teilnahme-Voraussetzungen</b>		
<b>Voraussetzungen</b>	recommended are: UkInfl	
<b>Prüfungs-modalitäten</b>	Defined by lecturer before beginning of course	
<b>Nützliche Literatur</b>	B. Jähne, Digital Image Processing, 6th edition, Springer	

Entnommen aus dem Modulhandbuch MScPhysik, Version 2012.2/v3

## Introduction to Image Processing on the GPU

<b>Code</b> IGPU	<b>Name</b> Introduction to Image Processing on the GPU	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Four day block course, equivalent of 2 SWS	<b>Arbeitsaufwand</b> 60 h, thereof 30 h Lectures and lab exercises 30 h Preparation and home exercises	<b>Verwendbarkeit</b>
<b>Lernziel</b>	<p>For certain computation tasks which are colloquially called „embarrassingly parallel“, and which occur quite frequently for example in image analysis, an implementation on a GPU (i.e. graphics card) can be orders of magnitude faster than a similar CPU implementation.</p> <p>In this short course, the students learn:</p> <ul style="list-style-type: none"> <li>- what kind of code can and can not be parallelized on a graphics processing unit (GPU)</li> <li>- the structure of a GPU, the different types of memory and how and when to use them</li> <li>- how to program the GPU using nVidia CUDA</li> <li>- how to perform various image analysis tasks using the GPU</li> <li>- ways to optimize the speed of GPU code</li> </ul>	
<b>Inhalt</b>	<p>The four days will be broken up into theoretical and practical segments. In the (short) theoretical lectures, I will introduce the concepts necessary for GPU programming and basic image analysis, which you will immediately try out afterwards in lab sessions. This way, the course will be highly practical and interactive. Students will work in small groups, each of which will have the task to write a small program solving an image analysis problem of their choice. Note that while image analysis tasks are a focus of the course because they give immediately visible results, the acquired techniques can be of course be employed in other fields as well, e.g. for solving large linear algebra problems or PDEs.</p>	
<b>Teilnahme-Voraussetzungen</b>	none	
<b>Voraussetzungen</b>	<p>recommended are: Solid knowledge of C programming, in particular no fear of pointers and direct memory access Some basics of image processing are helpful, but not strictly necessary</p>	

<b>Prüfungs- modalitäten</b>	Active participation in the sessions and completion of the assigned program.
<b>Nützliche Literatur</b>	e.g. nVidia CUDA Programming Guide, available in the „CUDA zone“ on <a href="http://www.nvidia.com">www.nvidia.com</a>

## Fast Parallel Implementations of Image Labeling Problems

<b>Code</b> IMLP	<b>Name</b> Fast Parallel Implementations of Image Labeling Problems	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> 2 SWS Lecture, 1 SWS Exercise	<b>Arbeitsaufwand</b> 120 h; thereof 45 h Lectures and lab exercises 60 h Revision and home exercises 15 h Exam preparation	<b>Verwendbarkeit</b>
<b>Lernziel</b>	<p>Image labeling problems are a fundamental class of problems appearing in image analysis, which dominate tasks in low-level computer vision like depth and motion estimation. Recently, many algorithms have been developed to solve this kind of problems in a variational framework, which allows for fast parallel implementations on the GPU.</p> <p>In this short course, the students learn:</p> <ul style="list-style-type: none"> <li>- Theoretical background for solving labeling problems in a variational framework</li> <li>- Efficient algorithms to solve the related class of optimization problems with parallel algorithms, which can be implemented on the GPU</li> <li>- How to implement these algorithms using nVidia CUDA</li> <li>- Techniques and tricks to make the implementations efficient</li> </ul>	
<b>Inhalt</b>	<p>The lecture is aimed at students who either participated in my CUDA course at the beginning of the semester and want to learn more about image analysis and the theoretical background, or participated in a theoretical course on variational image analysis and want to learn more about state-of-the art labeling algorithms and the practical side of their implementation.</p> <p>If you are new to both topics, you might still take part in the course, but must be prepared to take (potentially a lot of) additional time learning the prerequisites.</p>	
<b>Teilnahme-Voraussetzungen</b>	none	
<b>Voraussetzungen</b>	recommended are: either „Introduction to Image Processing on the GPU“ offered at the beginning of the semester, or an introduction to Variational Image Analysis from e.g. last semester.	

<b>Prüfungsmodalitäten</b>	Active participation in lecture and exercises, oral exam.
<b>Nützliche Literatur</b>	e.g. Chambolle et al. 2010 „An Introduction to Total Variation for Image Analysis“ for theoretical background, and nVidia CUDA Programming Guide, available in the „CUDA zone“ on <a href="http://www.nvidia.com">www.nvidia.com</a> , for the practical one.

## 2.8 Module aus dem B.S./M.Sc. Biologie

### Model-based segmentation of biomedical images (Modellbasierte Segmentierung Biomedizinischer Bilder)

<b>Code</b> IMSBI	<b>Name</b> Model-based segmentation of biomedical images (Modellbasierte Segmentierung Biomedizinischer Bilder)	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 120h insgesamt, davon 30h Präsenzstudium 30h Prüfungsvorbereitung 60h Selbststudium und Aufgabenbearbeitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Ziel der Vorlesung ist es, einen Überblick über 2D und 3D deformierbare Modelle und deren mathematische Grundlagen für die modellbasierte Segmentierung biomedizinischer Bilder zu geben.	
<b>Inhalt</b>	<p>In der Vorlesung werden drei wesentliche Klassen von deformierbaren Modellen (aktive Konturen, statistische Formmodelle, analytische parametrische Modelle) sowie starre Template Modelle behandelt. Die verschiedenen Segmentierungsverfahren werden an aktuellen Beispielen aus der 2D und 3D biomedizinischen Bildverarbeitung veranschaulicht. Geplante Themen sind unter anderem:</p> <ul style="list-style-type: none"> <li>- Introduction and overview of image analysis, image segmentation, and biomedical images</li> <li>- Template models, linear transformations, Hough transform</li> <li>- Active contour models: snakes, level sets, and geodesic active contours</li> <li>- Statistical shape models: active shape models, active appearance models, and extensions</li> <li>- Shape correspondence estimation: landmark-based, intensity-based, and hybrid</li> <li>- Analytic parametric models: contour-based and intensity-based</li> <li>- Performance evaluation and applications</li> </ul>	

<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse Signal- oder Bildverarbeitung (z.B. Vorlesung Bildverarbeitung oder Bioinformatik)
<b>Prüfungsmodalitäten</b>	Bestehen einer Abschlussprüfung
<b>Nützliche Literatur</b>	Bekanntgabe in der Vorlesung