

# Lehrveranstaltung

## IBV - Industrielle Bildverarbeitung

---

Version: 2 | Letzte Änderung: 23.09.2019 09:14 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

### ^ Allgemeine Informationen

<b>Langname</b>	Industrielle Bildverarbeitung
<b>Anerkennende LModule</b>	<a href="#">IBV_BaET</a> , <a href="#">IBV_BaTIN</a>
<b>Verantwortlich</b>	Prof. Dr. Lothar Thieling Professor Fakultät IME
<b>Niveau</b>	Bachelor
<b>Semester im Jahr</b>	Sommersemester
<b>Dauer</b>	Semester
<b>Stunden im Selbststudium</b>	78
<b>ECTS</b>	5
<b>Dozenten</b>	Prof. Dr. Lothar Thieling Professor Fakultät IME
<b>Voraussetzungen</b>	Grundlagen der Sinalverarbeitung (Diskrete Signale im Zeit- und Ortsbereich) Grundlagen der Programmierung in Java oder C Grundlagen der Analysis und Linearen Algebra
<b>Unterrichtssprache</b>	deutsch
<b>separate Abschlussprüfung</b>	Ja

## Abschlussprüfung

### Details

Die Studierenden sollen in einer mündlichen Prüfung folgende Kompetenzen nachweisen: 1.) Sicherer Umgang mit grundlegenden Begrifflichkeiten, Mechanismen und Konzepten. 2.) Problemstellungen aus dem Bereich der Industriellen Bildverarbeitung analysieren und mit passenden Methoden lösen können. 3.) Vorliegende Problemlösungen analysieren und die dabei verwendeten algorithmischen und theoretischen Grundlagen erklären können.

## Mindeststandard

Mindestens 50% der möglichen Gesamtpunktzahl.

## Prüfungstyp

Die Studierenden sollen in einer mündlichen Prüfung folgende Kompetenzen nachweisen: 1.) Sicherer Umgang mit grundlegenden Begrifflichkeiten, Mechanismen und Konzepten. 2.) Problemstellungen aus dem Bereich der Industriellen Bildverarbeitung analysieren und mit passenden Methoden lösen können. 3.) Vorliegende Problemlösungen analysieren und die dabei verwendeten algorithmischen und theoretischen Grundlagen erklären können.

# ^ Vorlesung / Übungen

## Lernziele

---

### Kenntnisse

Bildaufbau globale Bildeigenschaften und Zugriff auf Bilddaten

Bildmatrix

Grauwert- und Farbbilder

globale Bildeigenschaften

Mittelwert, Varianz, Entropie

Histogramm, kumulatives Histogramm

Entwicklungsumgebung

Software-Entwicklungsumgebung

Compiler

Linker

Debugger

Entwicklungsumgebung für die Bildverarbeitung und Bildanalyse

programmtechnischer Zugriff auf Bilddaten und Parameter

Überblick über die zur Verfügung stehenden BV-Module

Erstellung eigener BV-Module

Erstellung von "Algorithmenketten" auf Basis von BV-Modulen mittels grafischer Programmierung

---

Grauwerttransformationen

lineare Grauwerttransformation, Histogrammspreizung

nicht lineare Grauwerttransformation, Gammakorrektur

Histogrammasgleich

lokaler Histogrammausgleich

Look-Up-Tabellen

Flaschfarbendarstellung

---

Farbbildanalyse und -verarbeitung

visuelle und technische Farberfassung

additive und subtraktive Farbmischen

RGB-Farbraum

HSI-Farbraum

Transformation zwischen RGB und HSI

---

Rang-Ordnungs-Operatoren (nichtlineare Filterung)

Max, Min, Median

morphologische Operatoren  
Erosion, Dilatation  
Opening, Closing  
Auffinden von Strukturen

---

Analyse und Verarbeitung im Frequenzbereich  
Fourieranalyse und -synthese eindimensionaler digitaler Signale  
reales Spektrum, imaginäres Spektrum  
Amplitudenspektrum, Phasenspektrum  
Filterung im Frequenzbereich  
Fourieranalyse und -synthese von Bildern  
reales Spektrum, imaginäres Spektrum  
Amplitudenspektrum, Phasenspektrum  
Filterung im Frequenzbereich  
richtungunabhängige Filter  
richtungsabhängige Filter  
inverser Filterung

---

Lineare Filterung im Ortsbereich  
Flatung, Faltungsmaske und deren Transformierte  
typische Faltungsmasken (Mittelwert, Gauß, Differenz-Operator, Sobel-Operator, Laplace-Operator)  
Gradient und dessen Berechnung mittels Differenz und Sobel-Operator  
Analyse und Bewertung der Operator im Frequenzbereich

---

Tracking  
Normierter Kreuzkorrelation  
Verfolgungen  
ohne Schätzung  
mit Schätzung (Kalmanfilter)

---

Subpixelgenaue Kantenvermessung  
eindimensional  
zweidimensional mittels Gradient

---

## Fertigkeiten

die vorgestellten Verfahren zur Bildverbesserung mittels Grauwerttransformation  
angeben  
beschreiben  
hinsichtlich der Einsatzfelder abgrenzen  
hinsichtlich der Vor- und Nachteile bewerten  
problemspezifisch parametrieren

---

die vorgestellten Farbräume und der Algorithmen zu deren Analyse  
angeben  
beschreiben  
hinsichtlich der Einsatzfelder abgrenzen  
hinsichtlich der Vor- und Nachteile bewerten  
problemspezifisch parametrieren

---

die vorgestellten Verfahren zur nicht linearen Filterung  
angeben

beschreiben

hinsichtlich der Einsatzfelder abgrenzen

hinsichtlich der Vor- und Nachteile bewerten

problemspezifisch parametrieren

---

Spektren von Bildern und/oder Faltungsmasken

analysieren

begründen

konstruieren

diskutieren

---

die vorgestellten Verfahren zur linearen Filterung (Orts- und Frequenzbereich)

angeben

beschreiben

hinsichtlich der Einsatzfelder abgrenzen

hinsichtlich der Vor- und Nachteile bewerten

problemspezifisch parametrieren

## Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	2
Übungen (ganzer Kurs)	0
Übungen (geteilter Kurs)	0
Tutorium (freiwillig)	0

## Separate Prüfung

keine

## ^ Praktikum

### Lernziele

---

#### Fertigkeiten

siehe Fertigkeiten, die unter "Vorlesung/Übung->Lernziele->Fertigkeiten" aufgeführt sind

---

komplexere Aufgaben in einem Kleinteam bewältigen

---

Erarbeitung von komplexeren Problemlösungen die sich mittels Bildverarbeitung und Bildanalyse implementieren lassen

komplexere Problemstellungen verstehen und analysieren

Systemverhalten aus spezifizierenden Texten herleiten

System strukturiert analysieren

sinnvolle Teilsysteme erkennen

Schnittstellen zwischen Teilsystemen erfassen

Gesamtsystem auf Basis von Teilsystemes modellieren

Auswahl geeigneter bekannter Verfahren

Modifikation bekannter Verfahren

Kombination geeigneter Vefahren

Teilsysteme modellieren, implementieren, testen

Teilsysteme soweit möglich auf zur Vefügung stehende Komponenten (BV-Module) abbilden, d.h. Modulauswahl und Parametrierung.

Nicht zur Verfügung stehende aber benötigte BV-Module mittels Software-Entwicklungsumgebung in C implementieren und testen

Compilieren (Finden syntaktischer Fehler und deren Behebung)

Debuggen (Finden semantischer Fehler und deren Behebung)

Gesamtsystem (Problemlösung) implementieren testen und validieren

Erstellung der Problemlösung als "Algorithmenkette" auf Basis von BV-Modulen mittels grafischer Programmierung

Parametrierung der BV-Module

Validierung der Problemlösung

Auf Basis der Validierungsergebnisse in Iterationszyklen die Algorithmenkette und die Parametrierung der BV-Module anpassen. Bei Bedarf auch die BV-Module selbst modifizieren.

## Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Praktikum	2
Tutorium (freiwillig)	0

## Separate Prüfung

keine