

# Lehrveranstaltungshandbuch IBA

Industrielle Bildanalyse

Version: 4 | Letzte Änderung: 23.09.2019 09:14 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

## – Allgemeine Informationen

**Langname** Industrielle Bildanalyse

**Anerkennende LModule** [IBA BaET](#), [BV2 BaMT](#),  
[IBA BaTIN](#)

**Verantwortlich** Prof. Dr. Lothar Thieling  
Professor Fakultät IME

**Gültig ab** Sommersemester 2023

**Niveau** Bachelor

**Semester im Jahr** Wintersemester

**Dauer** Semester

**Stunden im Selbststudium** 78

**ECTS** 5

**Dozenten** Prof. Dr. Lothar Thieling  
Professor Fakultät IME

**Voraussetzungen** Grundlagen der  
Signalverarbeitung  
Grundlagen der  
Programmierung in  
Java oder C  
Grundlagen der  
Analysis und Linearen  
Algebra

**Unterrichtssprache** deutsch

**separate  
Abschlussprüfung** Ja

## Literatur

Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Prentice Hall

Scott E Umbaugh, COMPUTER VISION and IMAGE PROCESSING: A Practical Approach Using CVIPtools, Prentice Hall

Wolfgang Abmayer, Einführung in die digitale Bildverarbeitung, Teubner

## 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 Bildanalyse 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**

mündliche Prüfung, strukturierte Befragung

## – Vorlesung / Übungen

### Lernziele

| Zieltyp    | Beschreibung  |
|------------|---|
| Kenntnisse | Bildaufbau und Zugriff auf<br>Bilddaten<br>Bildmatrix<br>Grauwert- und Farbbilder<br>Entwicklungsumgebung<br>Software-Entwicklungsumgebung<br>Compiler<br>Linker<br>Debugger<br>Entwicklungsumgebung für die<br>Bildverarbeitung und Bildanalyse<br>programmtechnischer Zugriff auf<br>Bilddaten und Parameter<br>Überblick über die zur Verfügung<br>stehenden BV-Module<br>Erstellung eigener BV-Module<br>Erstellung von<br>"Algorithmenketten" auf Basis von<br>BV-Modulen mittels grafischer<br>Programmierung |
| Kenntnisse | Segmentierung<br>Histogrammbasierte<br>Segmentierung<br>Histogrammanalyse<br>Shading und dessen Beseitigung<br>flächenbasierte Segmentierung<br>Filling<br>Split and Merge<br>Region Growing<br>kantenbasierte Segmentierung<br>Konturverfolgung<br>Hough-Transformation  |
| Kenntnisse | Merkmalsextraktion<br>geometrische Merkmale<br>grundlegende Merkmale (Fläche,<br>Umfang, Formfaktor)<br>Zentralmomente<br>normierte Zentralmomente<br>Polarabstand<br>Krümmungsverlauf<br>DFT von Polarabstand und/oder<br>Krümmungsverlauf<br>Farbmerkmale (HSI)<br>Texturmerkmale<br>Co-occurrence Matrix<br>Haralick Merkmale  |

### Besondere Voraussetzungen

Grundlagen der industriellen Bildverarbeitung  
(Inhalte der Vorlesung IBV)

### Begleitmaterial

elektronische  
Vortragsfolien zur  
Vorlesung  
, elektronisches  
Entwurfswerkzeug für  
die Entwicklung von  
Bildanalyseanwendungen,  
elektronische Tutorials  
für Selbststudium der  
Entwurfswerkzeuge

### Separate Prüfung

Nein

Kenntnisse      Klassifikation von Merkmalen  
Begriffe und Grundlagen  
Merkmalsvektor, Merkmalsraum,  
Objektklassen ...  
überwachte/unüberwachte  
Klassifikation  
lernende/nicht lernende  
Klassifikation  
"klassische" Verfahren  
Quadermethode  
Minimum-Distance  
Nearest Neighbour  
Maximum-Likelihood  
neuronale Netze  
das künstliche Neuron als  
einfachster Klassifikator  
Arbeitsweise  
Aufgabe der Aktivierungsfunktion  
Aufgabe des Bias  
Training eines Neurons  
(Gradientenabstiegsverfahren)  
Multi-Layer-Perceptron  
Aufbau  
Aufgabe der Layer  
Backpropagation-  
Trainingsalgorithmus  
Entwicklungsumgebung zur  
Erstellung und zum Training  
Neuronaler Netze  
Erstellen und konfiguration  
neuronaler Netze  
Training neuronaler Netze  
Verifikation trainierter Netze  
Erzeugung von C-Funktionen aus  
trainierten Netzen

---

Fertigkeiten      die vorgestellten Verfahren zur  
Segmentierung  
angeben  
beschreiben  
hinsichtlich der Einsatzfelder  
abgrenzen  
hinsichtlich der Vor- und Nachteile  
bewerten  
problemspezifisch parametrieren

---

Fertigkeiten      die vorgestellten Merkmale und  
Verfahren zur Merkmalsextraktion  
angeben  
beschreiben  
hinsichtlich der Einsatzfelder  
abgrenzen  
hinsichtlich der Vor- und Nachteile  
bewerten  
problemspezifisch parametrieren

---

Fertigkeiten die vorgestellten Verfahren zur  
Klassifikation  
angeben  
beschreiben  
hinsichtlich der Einsatzfelder  
abgrenzen  
hinsichtlich der Vor- und Nachteile  
bewerten  
problemspezifisch parametrieren

### Aufwand Präsenzlehre

| <b>Typ</b>               | <b>Präsenzzeit (h/Wo.)</b> |
|--------------------------|----------------------------|
| Vorlesung                | 2                          |
| Übungen (ganzer Kurs)    | 0                          |
| Übungen (geteilter Kurs) | 0                          |
| Tutorium (freiwillig)    | 0                          |

## – Praktikum

### Lernziele

| <b>Zieltyp</b> | <b>Beschreibung</b>  |
|----------------|--|
| Fertigkeiten   | siehe Fertigkeiten, die unter "Vorlesung/Übung->Lernziele->Fertigkeiten" aufgeführt sind |
| Fertigkeiten   | komplexere Aufgaben in einem Kleinteam bewältigen  |

### Besondere Voraussetzungen

Grundlagen der industriellen Bildverarbeitung

### Begleitmaterial

elektronische  
Aufgabenstellung  
(Problembeschreibung)  
,  
Entwicklungsumgebung  
für die industrielle  
Bildanalyse,  
elektronische Tutorials  
für Selbststudium  
Handhabung der  
Entwicklungsumgebung

### Separate Prüfung

Nein

Fertigkeiten

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 Verfahren

Teilsysteme modellieren, implementieren, testen

Teilsysteme soweit möglich auf zur Verfügung 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                   |

