

# STUDIENARBEIT

des Studiengangs Informationstechnik

der Dualen Hochschule Baden-Württemberg Mannheim

---

## THEMA

**Entwicklung eines Systems zur Verfolgung von Objekten mit Hilfe einer Kamera**

---

**Julius Klodt**

19. April 2022

---

Bearbeitungszeitraum	01.10.2021 - 19.04.2022
Matrikelnummer, Kurs	8431855, TINF19IT1
Ausbildungsfirma	Deutsches Zentrum für Luft- und Raumfahrt e.V.
Betreuer	Jürgen Schultheis
Unterschrift des Betreuers	<hr/>

# Erklärung

Ich erkläre hiermit ehrenwörtlich:

1. dass ich meine Studienarbeit mit dem Thema *Entwicklung eines Systems zur Verfolgung von Objekten mit Hilfe einer Kamera* ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Studienarbeit bei keiner anderen Prüfung vorgelegt habe;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Mannheim, 9. März 2022

---

Julius Klodt

**Kurzfassung**

**Abstract**

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Aufgabenstellung</b>	<b>2</b>
<b>3 Methoden und Verfahren</b>	<b>4</b>
3.1 C . . . . .	4
3.2 Python . . . . .	4
3.2.1 OpenCV . . . . .	4
3.3 FreeCAD . . . . .	5
3.4 Ultimaker Cura . . . . .	5
3.5 3D-Drucker . . . . .	5
3.6 Blender . . . . .	6
<b>4 Durchführung</b>	<b>7</b>
4.1 Hardware . . . . .	9
4.1.1 Präzise Anforderungen an die Hardware stellen . . . . .	9
4.1.2 Vorhandene Projekte sichten und Umsetzungsideen sammeln . . . . .	9
4.1.3 Konzepte erstellen und Hardware konzipieren . . . . .	9
4.1.4 Hardware mit Hilfe von FreeCAD konstruieren . . . . .	9
4.1.5 Benötigte Hardware einkaufen . . . . .	9
4.1.6 Hardware zusammenbauen . . . . .	9
4.2 Software . . . . .	9
4.2.1 Anforderungen an die Software stellen . . . . .	9
4.2.2 UML-Diagramm konzipieren . . . . .	9
4.2.3 Hardwareansteuerung implementieren . . . . .	9
4.2.4 Gesichtserkennung implementieren . . . . .	9
4.2.5 Algorithmus zur Verarbeitung der Position des Gesichts zu Motor- bewegungen entwickeln und implementieren . . . . .	9
4.2.6 Hardwareansteuerung, Gesichtserkennung und Algorithmus zur Ver- arbeitung zusammenführen . . . . .	9
4.3 Testen . . . . .	9
4.3.1 Hardware auf Funktionstüchtigkeit und Grenzen testen . . . . .	9
4.3.2 Software auf Funktionstüchtigkeit und Grenzen testen . . . . .	9
4.4 Erweiterungen . . . . .	9

<b>5</b>	<b>Ergebnis</b>	<b>10</b>
<b>6</b>	<b>Reflektion</b>	<b>11</b>
<b>7</b>	<b>Ausblick</b>	<b>12</b>
<b>8</b>	<b>Danksagung</b>	<b>13</b>
	<b>Literatur</b>	<b>I</b>
	<b>Anhang</b>	<b>II</b>

# Abkürzungsverzeichnis

DHBW	Duale Hochschule Baden-Württemberg
IDE	Integrated Development Environment
CAD	Computer Aided Design
PLA	Polylactic acid

# Abbildungsverzeichnis

2.1	Pan-Bewegung einer Kamera <a href="#">[6]</a>	2
2.2	Tilt-Bewegung einer Kamera <a href="#">[7]</a>	2
3.1	“Cubic“ - Struktur	5
3.2	“Grid“ - Struktur	5

# Tabellenverzeichnis



# 1 Einleitung

In Zeiten der Covid-19-Pandemie wurden Videokonferenzen immer populärer. In Schulen, bei der Arbeit, beim Studium und selbst beim Sport werden Videokonferenzen mittlerweile genutzt. Dabei kommt es nicht selten vor, dass sich die Position einer Person verändert und nur noch schlecht im Bild zu sehen ist. Bei Verwendung mehrerer Bildschirme entsteht zusätzlich das Problem, dass je nach dem auf welchen Bildschirm geguckt wird, nur das Profil und nicht die Frontalansicht des Gesichts zu sehen ist.

Im Rahmen einer Studienarbeit der **Duale Hochschule Baden-Württemberg** ([DHBW](#)) Mannheim ist daraus die Idee entstanden, ein System zu entwickeln, welches die genutzte Kamera automatisch so bewegt, dass die gefilmte Person immer mittig und frontal im Bild zu sehen ist.

Dabei soll sowohl Hardware als auch Software erstellt werden. Um die benötigte Präzision der Hardware zu gewährleisten kam die Idee, das System 3D zu drucken. Alle Komponenten sollen dafür modelliert und gedruckt werden. Die Software soll die Motor und Peripherie Ansteuerung beinhalten. Des Weiteren eine Gesichtserkennung und ein Algorithmus zur Weiterverarbeitung der Gesichtsposition zu Motorbewegungen.

## 2 Aufgabenstellung

Dieses Kapitel befasst sich mit den aus der Projektidee resultierenden groben Anforderungen und Aufgabenstellungen.

Damit das System eine Person im Bild einer Kamera verfolgen kann, muss gewährleistet sein, dass das System die Kamera drehen und neigen kann. Diese Drehbewegungen werden als Pan und Tilt bezeichnet.



Abbildung 2.1: Pan-Bewegung einer Kamera[6]



Abbildung 2.2: Tilt-Bewegung einer Kamera[7]

Um zu gewährleisten, dass immer die Frontalansicht einer Person zu sehen ist, muss sich die Kamera mindestens in einer Achse im Raum bewegen können. Diese Arbeit beschränkt sich dabei auf eine gerade Achse.

Im folgenden eine detaillierte Aufgabenstellung:

### **Aufgabe 1:** Hardware

- 1.1. Präzise Anforderungen an die Hardware stellen
- 1.2. Vorhandene Projekte sichten und Umsetzungsideen sammeln
- 1.3. Konzepte erstellen und Hardware konzipieren
- 1.4. Hardware mit Hilfe von FreeCAD konstruieren
- 1.5. Benötigte Hardware einkaufen
- 1.6. Konstruierte Teile mit 3D-Drucker drucken und nachbearbeiten
- 1.7. Hardware zusammenbauen

### **Aufgabe 2:** Software

- 2.1. Anforderungen an die Software stellen
- 2.2. UML-Diagramm konzipieren
- 2.3. Hardwareansteuerung implementieren
- 2.4. Gesichtserkennung implementieren
- 2.5. Algorithmus zur Verarbeitung der Position des Gesichts zu Motorbewegungen
- 2.6. Hardwareansteuerung, Gesichtserkennung und Algorithmus zur Verarbeitung zusammenführen

### **Aufgabe 3:** Testen

- 3.1. Hardware auf Funktionstüchtigkeit und Grenzen testen
- 3.2. Software auf Funktionstüchtigkeit und Grenzen testen

## 3 Methoden und Verfahren

In diesem Kapitel werden die zugrunde liegenden Theoretischen Kenntnisse, die Methodik und Verfahrensgrundlage erörtert. Zu dem wird erklärt, warum diese genutzt wurden.

### 3.1 C

Die Hardwareansteuerung erfolgt mit der Programmiersprache C, welche nativ auf dem verwendeten *Arduino nano* unterstützt wird. C wurde ursprünglich von Dennis Ritchie in den frühen 1970er Jahren an den Bell Laboratories entwickelt und basiert auf der von Ritchie und Ken Thompson entwickelten Programmiersprache B [5].

Die Möglichkeit auf Hardwarekomponenten direkt zuzugreifen, macht C sehr hardwarenah [2]. Dies bringt auch den Vorteil mit sich, dass C sehr schnell ist. Um mit zu programmieren und den fertigen Code auf den Arduino nano zu übertragen wurde die **I**ntegrated **D**evelopment **E**nvironment (**IDE**) Arduino-Code mit der Version 1.8.13 verwendet.

### 3.2 Python

Für die Gesichtserkennung und die Weiterverarbeitung der Gesichtspositionsdaten wird die Programmiersprache Python verwendet. Python ist eine objektorientierte, höhere Programmiersprache [3] und wird häufig für Anwendungen im Bereich des Maschinellen Lernens verwendet.

Im Rahmen dieser Arbeit wird die Python Version 3.9.7 verwendet.

#### 3.2.1 OpenCV

OpenCV ist eine Open-Source Bibliothek welche für digitale Bildverarbeitung genutzt wird und in den Programmiersprachen Java, C, C++ und Python verwendet werden kann [**OpenCV**]. Die OpenCV Bibliothek bietet unter anderem Algorithmen zur Gesichtserkennung an und wird daher für diese Arbeit mit der Version 4.5.5 genutzt.

### 3.3 FreeCAD

FreeCAD ist ein Open-Source **C**omputer **A**ided **D**esign (**CAD**)-Programm. Es wird für die Konstruktion und Modellierung von Objekten genutzt [1]. In Rahmen dieser Arbeit wird FreeCAD mit der Versionsnummer 0.19 genutzt, um die benötigten Teile für das Verfolgungssystem als 3D-Modelle zu konstruieren.

### 3.4 Ultimaker Cura

“Ultimaker Cura ist eine Open-Source Slicer-Software zur Umwandlung von 3D Modellen in G-Code (Druckerbefehle für 3D-Drucker)“[8].

Für diese Arbeit wurde Ultimaker Cura mit der Version 4.12.1 genutzt. Das Programm bietet vielseitige Möglichkeiten an, wie ein 3D-Modell gedruckt werden soll. Zum Beispiel kann manuell eingestellt werden, wie viel Prozent des Innenraums eines 3D-Modells mit Material gefüllt werden soll. Diese Einstellung wird automatisch mit einer einstellbaren Inneren Struktur umgesetzt. Im folgenden sind zwei unterschiedliche innere Strukturen zu sehen, welche automatisch von Ultimaker Cura produziert worden sind. Beide Abbildungen zeigen die 44. Druckschicht eines Zahnrads mit 20% Füllung des Innenraums.

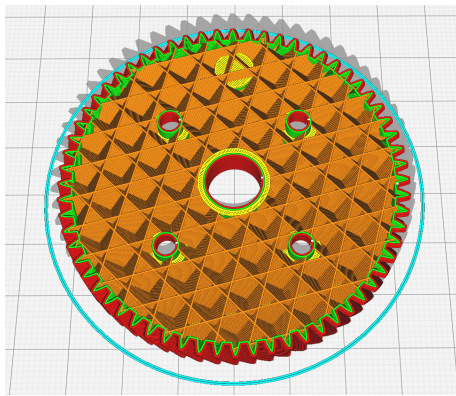


Abbildung 3.1: “Cubic“ - Struktur

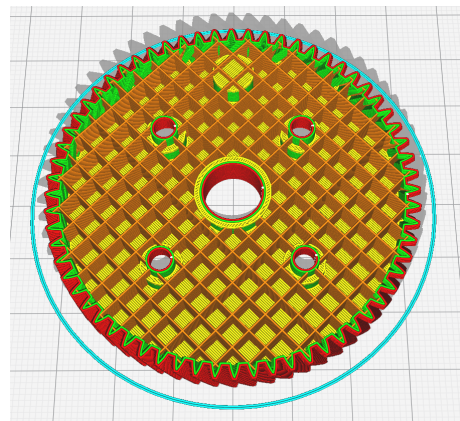


Abbildung 3.2: “Grid“ - Struktur

### 3.5 3D-Drucker

Im Rahmen dieser Arbeit wurden drei verschiedene 3D-Drucker genutzt, um selbst konstruierte Objekte mit dem Kunststoff **P**olylactic **a**cid (**PLA**) zu drucken. 3D-Drucken ist ein additives Verfahren, in welchem ein dreidimensional beweglicher Druckerkopf, Material Schicht für Schicht übereinander lagert. Dieses Verfahren wurde 1945 in einer Science-Fiction Kurzgeschichte beschrieben [4]. Allerdings wurde erst 1987 der erste 3D-Drucker

auf den Markt gebracht. Bis zu den 2000er gab es jedoch wenig Interesse seitens der Wirtschaft. Ab den 2000er wurde 3D-Drucken durch

## **3.6 Blender**

# 4 Durchführung

Dieses Kapitel erörtert die Durchführung der Aufgabenstellung. Die Strukturierung gleicht dem Aufbau der Aufgabenstellung.





## **4.1 Hardware**

**4.1.1 Präzise Anforderungen an die Hardware stellen**

**4.1.2 Vorhandene Projekte sichten und Umsetzungsideen sammeln**

**4.1.3 Konzepte erstellen und Hardware konzipieren**

**4.1.4 Hardware mit Hilfe von FreeCAD konstruieren**

**4.1.5 Benötigte Hardware einkaufen**

**4.1.6 Hardware zusammenbauen**

## **4.2 Software**

**4.2.1 Anforderungen an die Software stellen**

**4.2.2 UML-Diagramm konzipieren**

**4.2.3 Hardwareansteuerung implementieren**

**4.2.4 Gesichtserkennung implementieren**

**4.2.5 Algorithmus zur Verarbeitung der Position des Gesichts zu Motorbewegungen entwickeln und implementieren**

**4.2.6 Hardwareansteuerung, Gesichtserkennung und Algorithmus zur Verarbeitung zusammenführen**

## **4.3 Testen**

**4.3.1 Hardware auf Funktionstüchtigkeit und Grenzen testen**

**4.3.2 Software auf Funktionstüchtigkeit und Grenzen testen**

## **4.4 Erweiterungen**

## 5 Ergebnis

Referenzierung zu Aufgabenstellung.

→ Was habe ich geschafft, was habe ich nicht geschafft?

Als (Not-)Lösung: Tabelle der Aufgaben mit Spalte “konnte erledigt“ werden

## 6 Reflektion

Was habe ICH gelernt (ich erlaubt)

Hier auch Gedanken zu Recycling des 3D-Mülls

# 7 Ausblick

Ausblick mit Erweiterungsvorschlägen

## 8 Danksagung

Hier kommt die Danksagung hin

# Literatur

- [1] FreeCAD. *FreeCAD - Offizielle Website*. URL: <https://www.freecadweb.org/> (besucht am 07.03.2022).
- [2] Tobias Häberlein. *Technische Informatik - Ein Tutorium der Maschinenprogrammierung und Rechnertechnik*. Vieweg+Teubner, 2011. ISBN: 978-3-8348-1372-5.
- [3] Python. *Python - Offizielle Website*. URL: <https://www.python.org/about/> (besucht am 06.03.2022).
- [4] Drew Turney. *Geschichte des 3D-Drucks: Eine innovative Technologie mit Tradition*. Nov. 2021. URL: <https://redshift.autodesk.de/geschichte-des-3d-drucks/> (besucht am 08.03.2022).
- [5] Wikipedia. *C-Programmiersprache*. URL: [https://de.wikipedia.org/wiki/C\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/C_(Programmiersprache)) (besucht am 06.03.2022).
- [6] Wikipedia. *Pan*. URL: [https://en.wikipedia.org/wiki/Panning\\_\(camera\)](https://en.wikipedia.org/wiki/Panning_(camera)) (besucht am 03.03.2022).
- [7] Wikipedia. *Tilt*. URL: [https://en.wikipedia.org/wiki/Tilt\\_\(camera\)](https://en.wikipedia.org/wiki/Tilt_(camera)) (besucht am 02.03.2022).
- [8] Wikipedia. *Ultimaker Cura - Wikipedia Artikel*. URL: <https://www.freecadweb.org/> (besucht am 07.03.2022).

# Anhang