

Eingereicht von

Lukas Hornung
Lukas Luschin
Moritz Schmidt
Timmo Waller-Ehrat



Teamprojekt

Mehrbildkamarasystem zur
räumlichen Detektion von
Modellhubschraubern

Extended Abstract

Thema:	Mehrbildkamarasystem zur räumlichen Detektion von Modellhubschraubern
Teammitglieder:	Lukas Hornung, Lukas Luschin, Moritz Schmidt, Timmo Waller-Ehrat
Betreuer:	Hochschule für Technik, Wirtschaft und Gestaltung HTWG Konstanz, Institut für Optische Systeme Prof. Dr. Georg Umlauf, Prof. Dr Matthias O. Franz

Unser Projekt behandelt das räumliche Detektieren eines Modellhubschraubers. Die Detektion soll unter Laborbedingungen, das heißt, der Helikopter befindet sich vor einer weißen Wand, stattfinden. Mittels der Detektion soll auf einem Bild angezeigt werden, wo sich der Mittelpunkt des Helikopters befindet. Auch die Tiefe (Entfernung zur Kamera) des Helikopters soll ermittelt werden. Für diese Detektion sollen zwei oder mehr Kameras verwendet werden. Bei diesen handelt es sich um HIERKAMERA EINFÜGEN, die mit dem Computer über ein FireWire-Kabel verbunden sind.

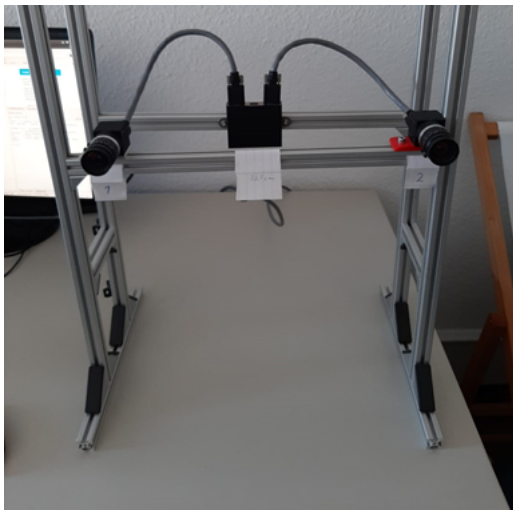


Abbildung 1: Kamera-System



(a) Kamera-Kalibrierung



(b) Helikopter-Punktewolke

Abbildung 2: Kamera-Kalibrierung und Punktewolke

Das Projekt wurde erfolgreich umgesetzt. Mittels zwei Kameras, die auf einer geraden Linie angebracht sind, kann der Mittelpunkt des Helikopters und dessen Abstand zur Kamera ermittelt werden.

Unser Programm kalibriert als erstes die Kameras einzeln und anschließend zu einander. Das Kalibrieren erfolgt über ein Schachbrett-Muster. Sind die Kameras zueinander kalibriert, kann mittels eines Feature-Detektors eine Punktewolke des Helikopters generiert und der Mittelpunkt berechnet werden.

Eine mögliche Erweiterung des Projekts wäre das Kalibrieren von zwei Stereo-Systemen zu einander, um eine noch höhere Genauigkeit zu erlangen. Dies wurde versucht umzusetzen, ist allerdings gescheitert.

Abstract

Unser Projekt behandelt das räumliche Detektieren eines Modellhubschraubers. Die Detektion soll unter Laborbedingungen, das heißt, der Helikopter befindet sich vor einer weißen Wand, stattfinden. Mittels der Detektion soll auf einem Bild angezeigt werden, wo sich der Mittelpunkt des Helikopters befindet. Auch die Tiefe (Entfernung zur Kamera) des Helikopters soll ermittelt werden. Für diese Detektion sollen zwei oder mehr Kameras verwendet werden.

Inhaltsverzeichnis

1	Einleitung	
1.1	Aufgabenstellung und Zielsetzung
1.2	Motivation
1.3	Aufbau
2	Technologien	
2.1	Software
2.1.1	OpenCV
2.1.2	scikit-learn
2.1.3	Open3D
2.1.4	PyCapture
2.2	Hardware
2.2.1	Kamera
3	Bildverarbeitung und Umsetzung	
3.1	Kalibrierung
3.1.1	Fehler Überprüfung
4	Probleme	

1

Einleitung

1.1. Aufgabenstellung und Zielsetzung

Im Rahmen dieses Teamprojekts stand die Entwicklung eines Mehrbildkamarasystems zur räumlichen Detektion eines Modellhubschraubers. Dies beinhaltet sowohl das Erkennen des Helikopters, als auch die Abstandsmessung von diesem.

Dies sollte mit Hilfe Bilderverarbeitungs- und Machine Learning-Techniken, sowie der Verwendung von zwei oder mehr Kameras umgesetzt werden.

Die Lernziele umfassten das Erlernen des Umgangs mit Kameras für die industrielle Bildverarbeitung, ein Verständnis für die Grundlagen industrieller Signalverarbeitung zu schaffen. Zudem sollten grundlegende KI-Verfahren erlernt werden.

1.2. Motivation

„Computer vision, or the ability of artificially intelligent systems to see like humans, has been a subject of increasing interest and rigorous research for decades now.“

— Naveen Joshi[2]

Das maschinelle Sehen gewinnt in den letzten Jahren immer mehr an Popularität. Sei es in der Forschung oder z.B. in der Spieleentwicklung mittels augmented reality.

1.3. Aufbau

Durch die steigende Relevanz in der Praxis wurde auch unser Interesse für dieses Themengebiet geweckt. Es ist spannend zu verstehen, wie komplex die Dinge, die für uns Menschen selbstverständlich erscheinen, eigentlich sind. Ist es nur das ermitteln der Tiefe eines Objekts im Raum.

Ein weiterer Anreiz für das Projekt waren die verschiedenen angewandten Technologien. Wir alle interessieren und sehr für das Programmieren. Viel Erfahrung in der Programmiersprache Python hatte aber anfangs keines der Teammitglieder. Somit war das Erlernen dieser Sprache eine weitere Motivation.

Auch die zum Großteil verwendete Bibliothek OpenCV hat das Interesse an das Projekt geweckt.

1.3. Aufbau

Die Ausarbeitung des Teamprojekts besteht aus drei Teilen.

Anfangs wird kurz auf die angewandten Technologien eingegangen. Anschließend wird die eigentliche Umsetzung und das Vorgehen erläutert. Zuletzt wird auf die aufgetretenen Probleme eingegangen und ein Fazit gezogen.

2

Technologien

2.1. Software

2.1.1. OpenCV

OpenCV ist eine Open-Source-Bibliothek, die über Algorithmen für maschinelles Sehen und Bildverarbeitung verfügt [4].



Abbildung 2.1: OpenCV

Quelle: https://de.wikipedia.org/wiki/OpenCV#/media/Datei:OpenCV_Logo_with_text.png

2.1.2. scikit-learn

Scikit-learn ist eine freie plattformunabhängige Python-Bibliothek, die für das maschinelle Lernen konzipiert ist. Die Software ist unter BSD lizenziert [5]. Von dieser Bibliothek wird lediglich die Implementierung des k-Means-Algorithmus verwendet.

2.2. Hardware



Abbildung 2.2: scikit

Quelle: https://de.wikipedia.org/wiki/Scikit-learn#/media/Datei:Scikit_learn_logo_small.svg

2.1.3. Open3D

Open3D ist eine Open-Source Bibliothek, die diverse Algorithmen für das Verarbeiten von 3D-Daten bereitstellt.

2.1.4. PyCapture

Mittels PyCapture werden die Kameras angesteuert. Diese Bibliothek liefert 15 Bilder pro Sekunde.

2.2. Hardware

2.2.1. Kamera

3

Bildverarbeitung und Umsetzung

3.1. Kalibrierung

Für eine Messung, bei der der Fehler minimiert werden soll, ist das Kalibrieren der Kameras unumgänglich. Durch die Linse einer Kamera entsteht eine tonnenförmige Verzeichnung [1]. Durch die Kalibrierung der Kamera kann diese Verzeichnung kompensiert werden.

VERZEICHNUNGSEIGENSCHAFTEN

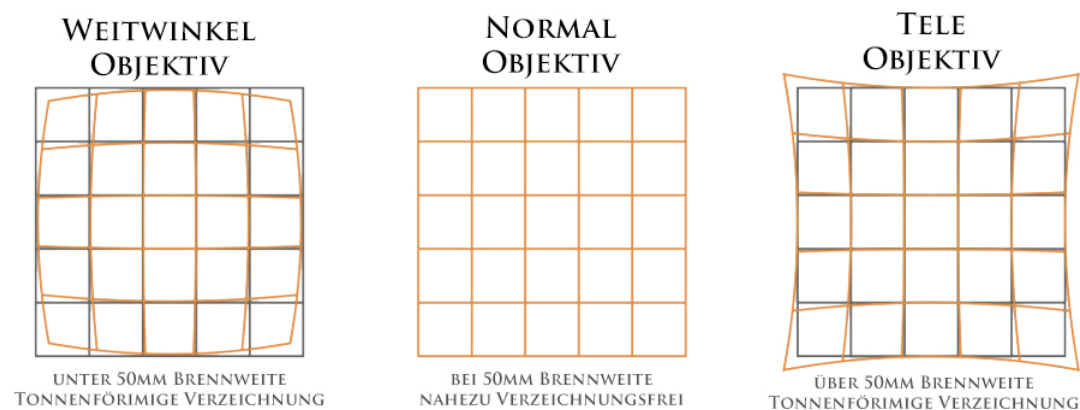


Abbildung 3.1: Verzeichnung

Quelle: <http://www.fotokurs-bremen.de/wp-content/uploads/2016/11/Objektiv-Verzeichnung.jpg>

3.1. Kalibrierung

Durch die Kamerakalibrierung werden mehrere Parameter bestimmt:

Intrinsische Parameter Sind kameraspezifische Parameter. Hier bei handelt es sich um Hardwareeigenschaften der Kamera.

Das Berechnen der intrinsischen Werte ist essentiell für das Eliminieren der Verzerrung der Linse. Dadurch kann eine 3D-Punkteberechnung stattfinden.

Extrinsische Parameter Beschreiben die Kameraposition und Rotation in Weltkoordinaten.

Da es sich bei dem System um ein Stereokamera-System handelt, ist die Kalibrierung von diesem etwas komplizierter.

Zuerst müssen die Kameras gesondert kalibriert werden. Dies wird mit der Funktion *calibrateCamera* von OpenCV durchgeführt. Für die Kalibrierung wird ein Schachbrettmuster verwendet. Wichtig ist, dass bei der Kalibrierung beide Kameras dasselbe Bild verwenden. Für die Erkennung des Schachbretts wird die OpenCV-Funktion *findChessboardCorners* verwendet. Diese liefert die Objekt- und Bild-Punkte der Aufnahme. Bei den Objekt-Punkten handelt es sich um die 3D-Punkte des Bildes, bei den Bild-Punkten um die 2D-Punkte[3]. Für eine möglichst genaue Kalibrierung werden 50 Bilder verwendet. Anhand dieser Bilder wird jede Kamera mittels *calibrateCamera* kalibriert.

Die Ergebnisse dieses Vorgangs werden auf dem Computer gespeichert, sodass dieser nicht wiederholt werden muss.

Anschließend wird das Ergebnis dieser Kalibrierungen an die OpenCV-Funktion *stereoCalibrate* übergeben.

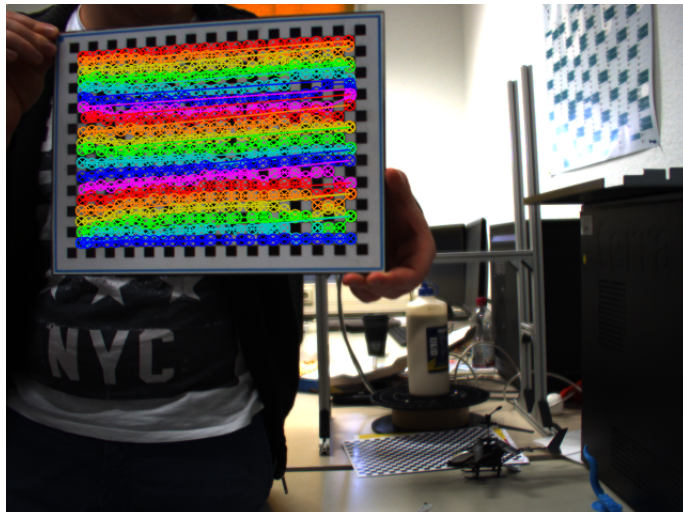


Abbildung 3.2: Stereo-Kalibrierung

3.1. *Kalibrierung*

3.1.1. Fehler Überprüfung

4

Probleme

Literatur

- [1] fotokurs bremen. *Verzeichnung*. URL: <http://www.fotokurs-bremen.de/objektive-brennweite-und-verzeichnung/> (besucht am 22. 02. 2020).
- [2] Naveen Joshi. *The Present And Future Of Computer Vision*. URL: <https://www.forbes.com/sites/cognitiveworld/2019/06/26/the-present-and-future-of-computer-vision/#490553c0517d> (besucht am 22. 02. 2020).
- [3] OpenCV. *OpenCV*. URL: https://docs.opencv.org/3.4/dc/dbb/tutorial_py_calibration.html (besucht am 22. 02. 2020).
- [4] Wikipedia. *OpenCV*. URL: <https://de.wikipedia.org/wiki/OpenCV> (besucht am 22. 02. 2020).
- [5] Wikipedia. *Scikit-learn*. URL: <https://de.wikipedia.org/wiki/Scikit-learn> (besucht am 22. 02. 2020).

Abbildungsverzeichnis

1	Kamera-System
2	Kamera-Kalibrierung und Punktwolke
2.1	OpenCV
2.2	scikit
3.1	Verzeichnung
3.2	Stereo-Kalibrierung