

# Projektarbeiten

## Rahmenbedingungen

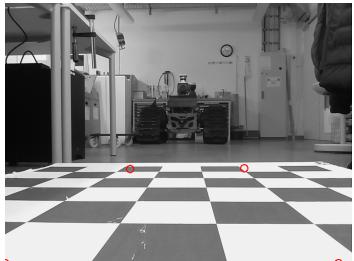
- Ein Projekt pro Gruppe zu maximal **3** Personen
- **8** Seiten Bericht pro Person (z. B. bei 3 Personen **24** Seiten exklusive Anhang)
- Schriftgröße: **11pt**; Zeilenabstand: **1.5**
- Abgabe des Berichtes am **24.2.2024**
- Praktische Vorführung des Projekts am: **6.3.2025**
- Programmiersprache: **C++**
- Umgebung & Compiler: ROS in Linux *Ubuntu LTS*, `catkin_make`

## Allgemeines

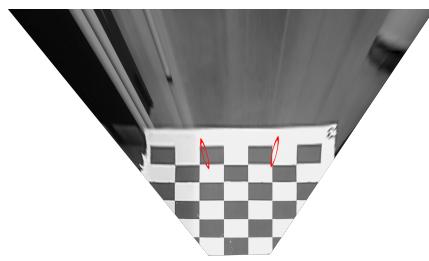
Obligatorisch ist das Erkennen eines roten Fadens. Orientieren Sie sich bei dem Aufbau des Berichtes an einer wissenschaftlichen Ausarbeitung, ähnlich einer Bachelorarbeit oder Masterarbeit. Recherchieren Sie umfangreich den aktuellen Stand von Wissenschaft und Technik.

# Projekte

**Aufgabe:** Stellen sie das Bild der RGB-Kamera als Draufsicht (*Birds-Eye-View*) dar. Schreiben Sie dazu einen Knoten welcher das Bild der RGB-Kamera abonniert, mit einem entsprechendem Algorithmus umrechnet und als Bild der Draufsicht ausgibt. Beachten Sie ein Kalibrieren der intrinsischen Kameraparameter. Stellen Sie die Laserscannerpunktewolke in der Draufsicht als Punkte dar. Darauf hinaus soll der durch die Kamera nicht erfassbaren Bereich ausgegraut werden. (100 Punkte)



(a) Original Bild



(b) Birds-Eye-View

1. Kalibrieren der **intrinsischen und extrinsischen Kameraparameter** der RGB-Kamera mithilfe eines Kalibriermusters.
2. Publizieren des rektifizierten (kalibrierten) Bildes auf einem **ROS-Topic** (z. B. `/camera/rectified_image`).
3. Bestimmen der **Transformationsmatrix** für die perspektivische Transformation (z. B. Homographie-Matrix).
4. Publizieren des **transformierten Bildes** in der Draufsicht (*Bird's-Eye-View*) auf einem **ROS-Topic** (z. B. `/camera/birdseye_image`).
5. Bestimmen der **Transformationsmatrix** zwischen der Draufsicht und der **Punktewolke des Laserscanners**.
6. Publizieren des Draufsicht-Bildes inklusive der eingezeichneten Laserpunkte als Overlay (z. B. `/camera/birdseye_with_lidar`).
7. Implementieren eines Algorithmus zum **Erkennen und Folgen** einer durch zwei Linien begrenzten Fahrbahn. Der Roboter soll dabei mittig der Linien fahren und beim ausbleiben einer der Liniensegmente interpolieren.

## Zusatzaufgaben:

1. Hinderniserkennung und -vermeidung (5 Sonderpunkte)
2. Darstellen von Trajektorie und Hindernissen in RViz (5 Sonderpunkte)
3. Bilddaten im *Stitching*-Verfahren als Karte darstellen (10 Sonderpunkte)

**Schlagwörter:** Birds Eye View, OpenCV, Scherung, intrinsische und extrinsische Kameraparameter, perspektivische Transformation