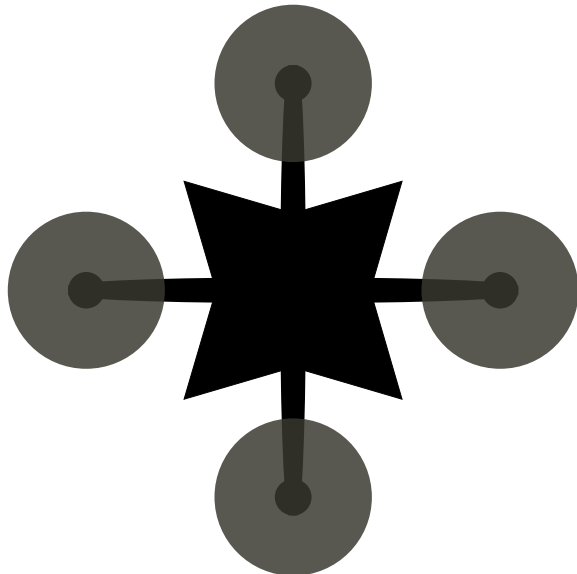


Abschlusspräsentation PSE

Kamerabasierte Steuerung eines Mikroquadropterschwarms

Daniela Grimm, Dennis Keck, Dominik Kiefer, Carina Kübler, Sebastian Schmidt, Lukas Werling



KIT
ROKOPTER

Aufgabenstellung

Implementierung eines Steuerungssystems
für einen oder mehrere gleichzeitig
fliegende Mikroquadrokopter
- Tracking und Positionsbestimmung der
Quadrokopter mit Hilfe von
Kinect-Kameras

Aufgabenstellung

- Implementierung eines Steuerungssystems für einen oder mehrere gleichzeitig fliegende Mikroquadrokopter
- Flugsteuerung einzelner Quadrokopter und des Schwarms
- Positionsdarstellung und Steuerung durch die GUI

Geplante Herausforderungen

Problem: Echtzeitsystem

- Lösung: Schnelle Rechner, Nutzung vorhandener schneller Softwarebibliotheken (OpenCV, cvBlob)

Geplante Herausforderungen

Problem: Drahtlose Kommunikation
- Lösung: Bereitgestellt durch cflib

Geplante Herausforderungen

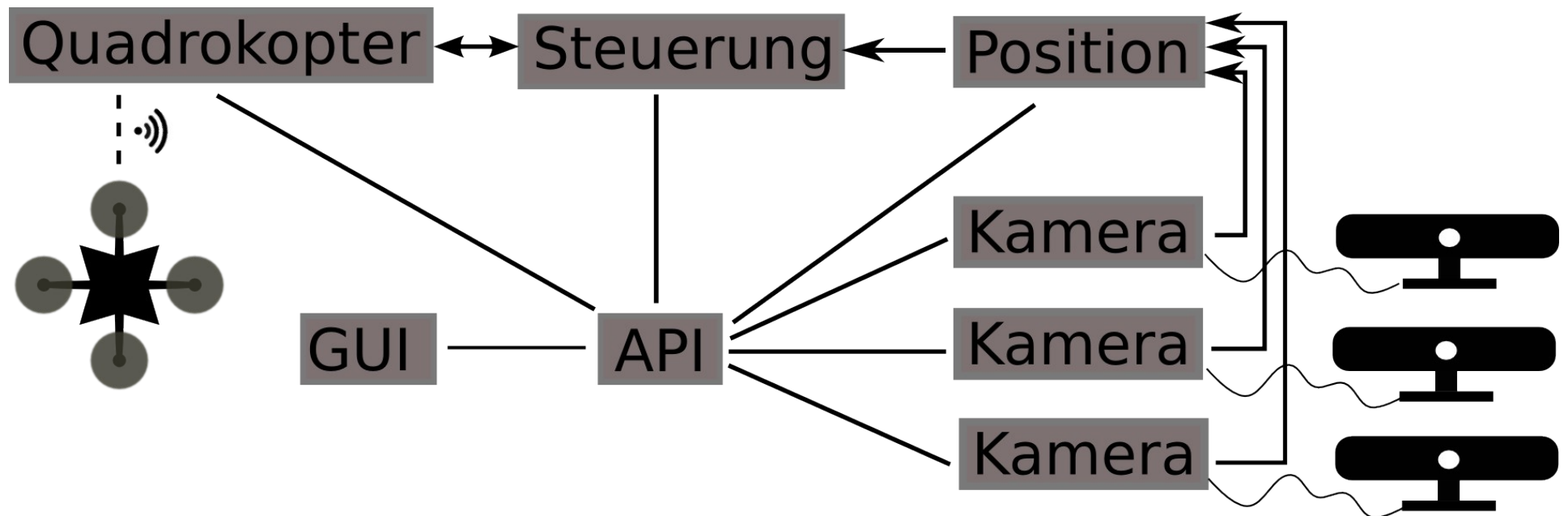
Problem: Unterscheidung einzelner
Quadrokopter

- Lösung: Farbbasiertes Tracking

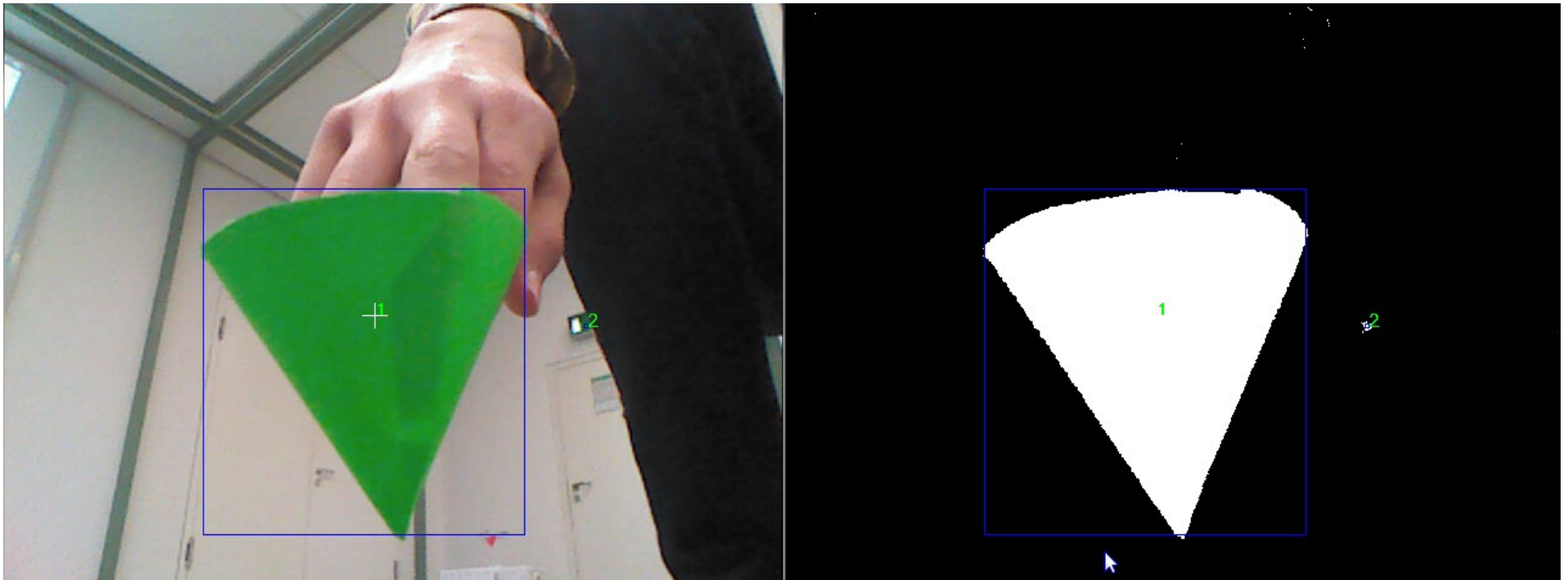
Herausforderungen

Problem: Verteiltes System
- Lösung: ROS

Modulübersicht



Kameramodul



Aufgabenstellung

Kamerakalibrierung

Aufgabenstellung

Farbbasiertes Tracking eines
Quadropters in Echtzeit

Herausforderungen

Tracking ist zeitkritisch

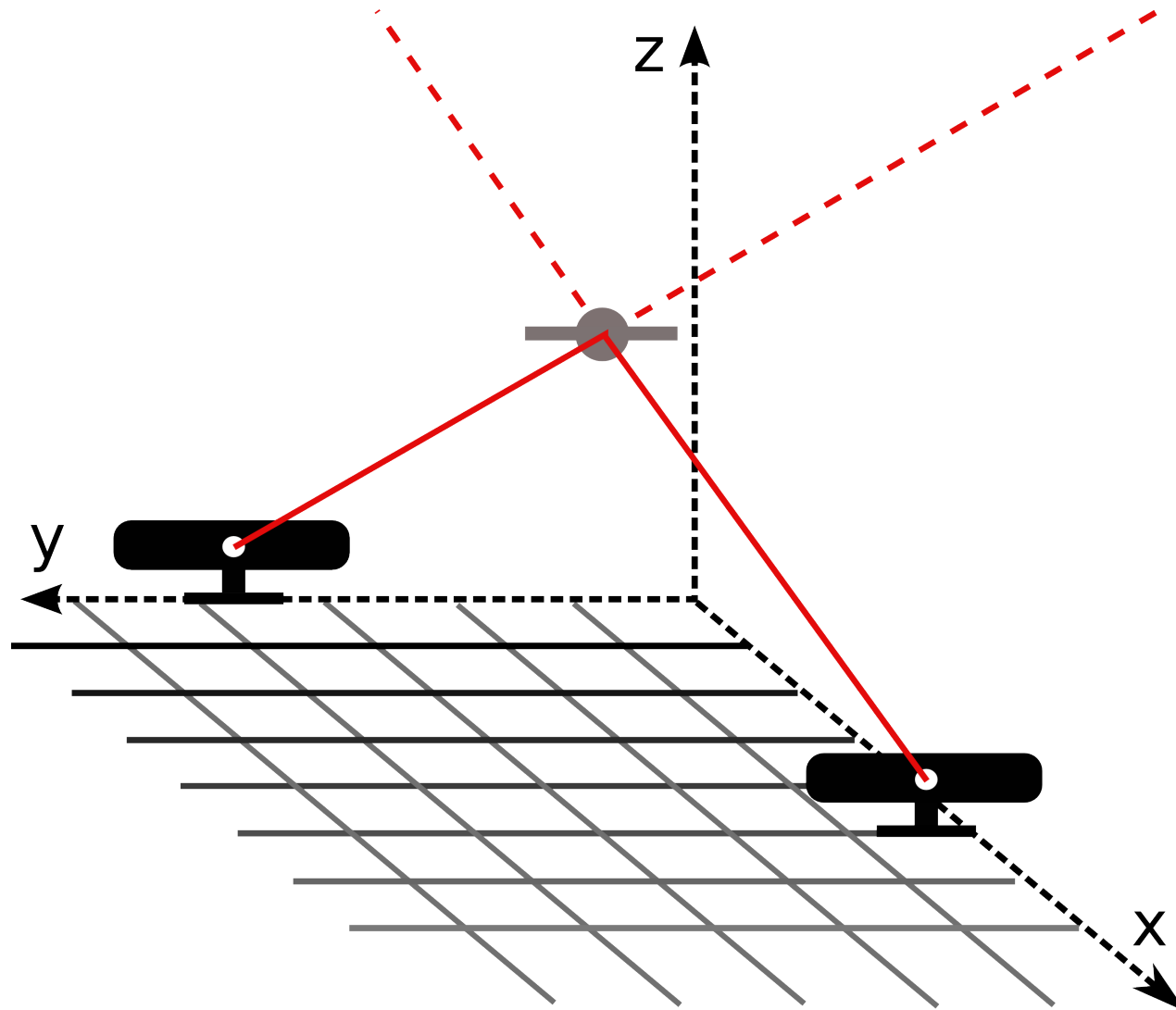
Herausforderungen

Komplexe Software zur Bildverarbeitung

Endergebnis

- Funktionierendes farbbasiertes Tracking mehrerer Quadrocopter
- Visualisierung des Trackings
- Funktionierende Kamerakalibrierung

Positionsmodul



Aufgabenstellung

Multikamerakalibrierung

Aufgabenstellung

Berechnung von Quadrokopterpositionen
aus Kameradaten

Aufgabenstellung

Berechnung der Geometrie des Trackingbereiches

Herausforderungen

Multikamerakalibrierung mit der
AMCC Toolbox

Lösung: enger Kontakt mit Entwickler

Herausforderungen

Transformation des
Kamerakoordinatensystems
parallel zum Boden

Herausforderungen

Geschwindigkeit der Positionsberechnung

Lösung: Berechnung ohne Matlab

Herausforderungen

Finden einer geeigneten Form des
Trackingbereiches

Lösung: Zwei aufeinanderstehende
Pyramiden

Herausforderungen

Gegenübergestellte Kameras liefern
ungenauere Daten

Lösung: Verwerten der Daten nur bei
 $\text{Winkel} > \pi/8$

Herausforderungen

Synchronisation der Kameradaten

Lösung: Zeitsynchronisation mit NTP

Endergebnis

- Tracking funktioniert schnell genug bei langsamen Bewegungen
- Geschwindigkeit der Crazyflies zu hoch
- Rauschen $< 1\text{mm}$ bei stehendem Quadrokopter
- Fehler ca. 5-10cm bei guter Kalibrierung

Steuerungsmodul

Ziel

- Quadrokopter Einzelsteuerung
 - Starten und Landen
 - Stabilisierung
 - Bewegung
- Formationssteuerung
 - Aufbau einer Formation
 - Rotation der Formation um einen Punkt
 - Bewegung der Formation

Herausforderungen

- Problem: Batterieabhängigkeit der Motorleistung
 - Lösung: Alle Thrust-Werte werden batterieabhängig berechnet
- Problem: Stabilisierung in z-Richtung
 - Lösung: PID-Regler für Thrust-Werte

Herausforderungen

- Problem: Stabilisierung in x- und y-Richtung
 - Lösung: P-Regler für Pitch- und Roll-Werte
- Problem: Orientierung der Quadrokopter
 - Lösung: Verwendung von Yawrate-Werten in einem weiteren P-Regler

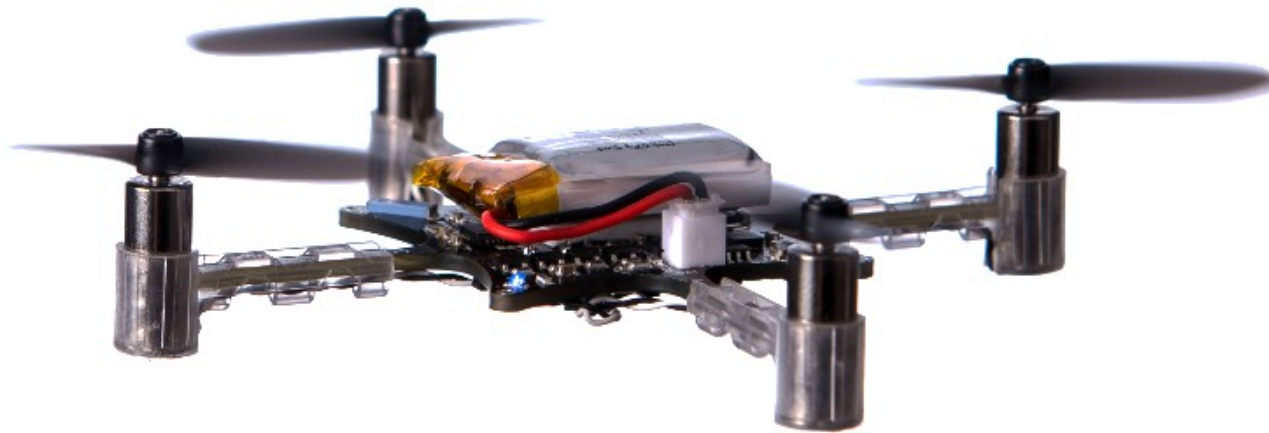
Herausforderungen

- Problem: Geeignete Koeffizienten zur Regelung
 - Lösung: Findung durch Tests
- Problem: Höhere Latenzen der Bilddaten bei schnelleren Bewegungen
 - Lösung: Notfallroutine um zurück in den Trackingbereich zu gelangen

Endergebnis

- Alle Funktionalitäten implementiert
- Steuerung der Quadrokopter (Thrust, Roll, Pitch und Yaw)
- Annäherung an gesetztes Ziel und Gegenlenken bei Abweichung
- Quadrokopter können sich größtenteils stabilisieren
- Landeprozess bei Verlassen des Trackingbereichs oder Shutdown-Befehl

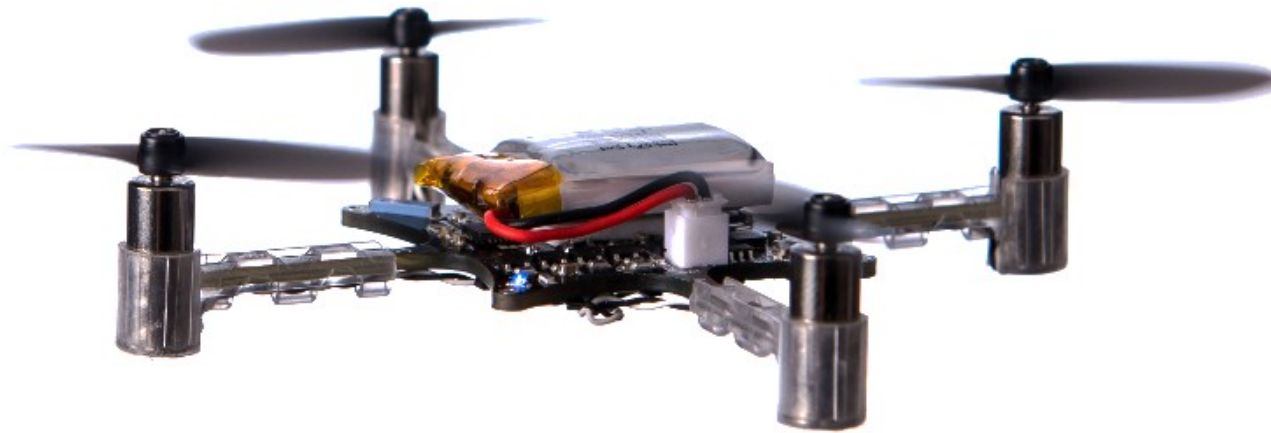
Quadrokoptermodul



Funktionalität

Auslesen der Sensorwerte
&
Steuerung der Quadrocopter

Crazyflie



“Open platform” Nano-Quadrocopter

Crazyflie Sensoren

- 3-Achsen Gyrometer & Accelerometer
 - 3-Achsen Magnetometer
 - Altimeter (Barometer)

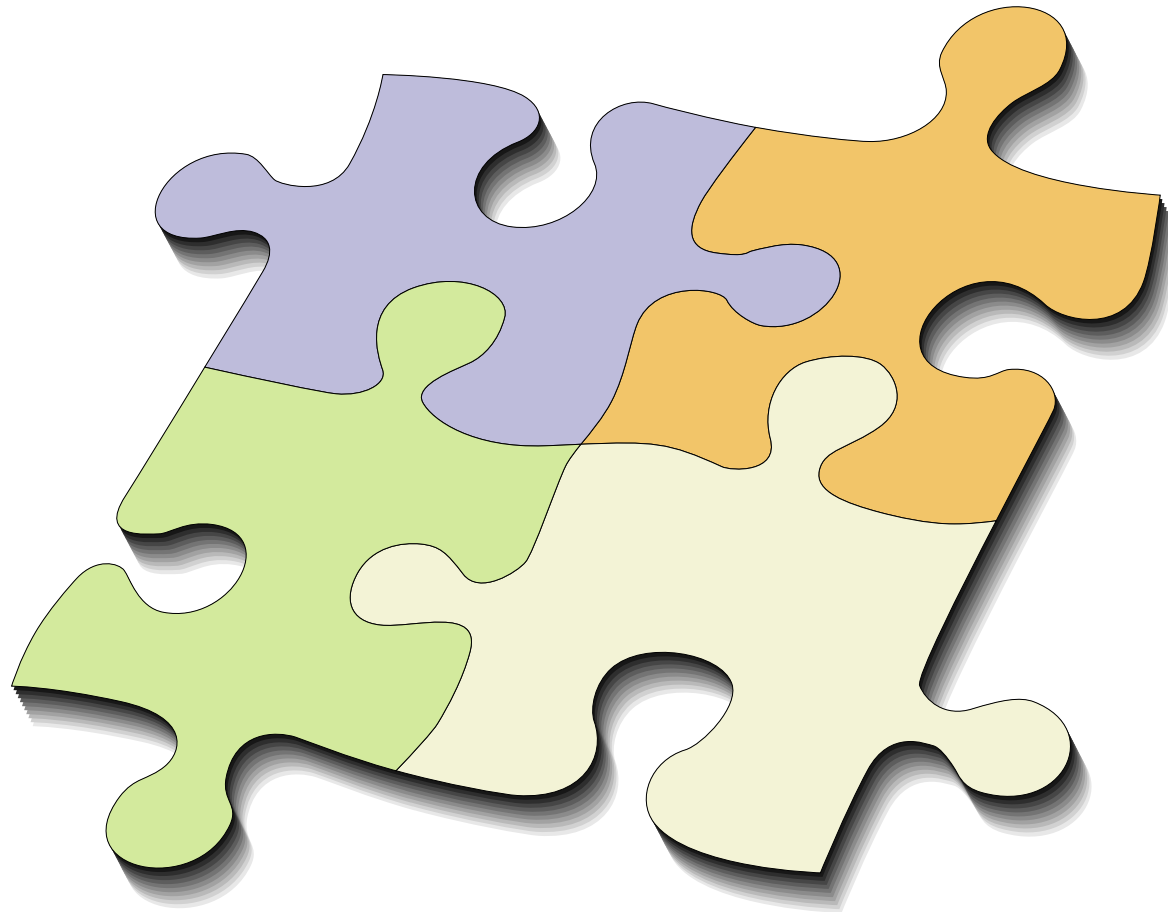
Herausforderungen

Mehrmals Änderungen an dem Crazyflie SDK, Umbenennung der Sensorvariablen

Herausforderungen

Teilweise nur schlechte und veraltete
Dokumentation

API



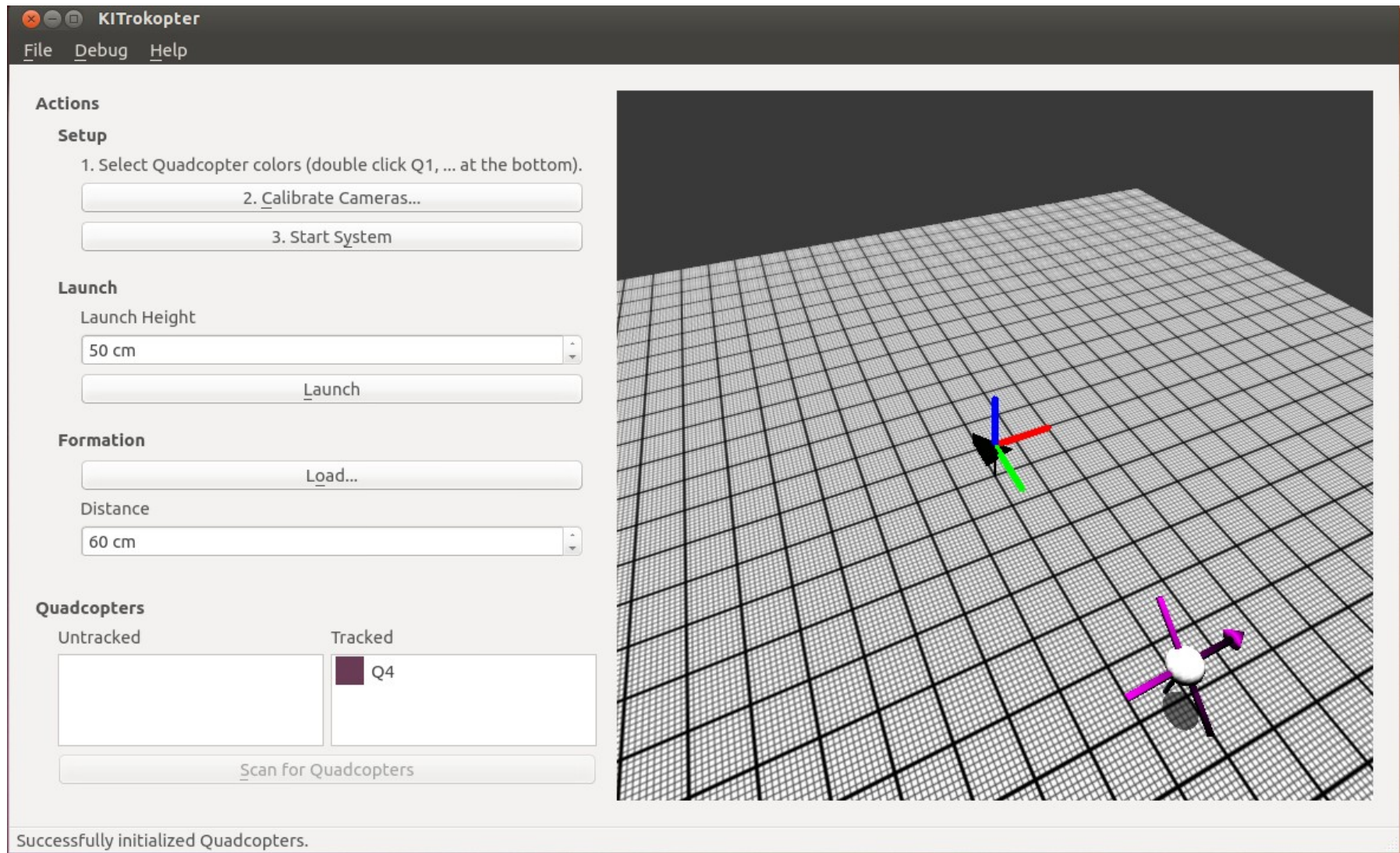
Funktionalität

Einheitliche Schnittstelle zum
verteilten Gesamtsystem

Funktionalität

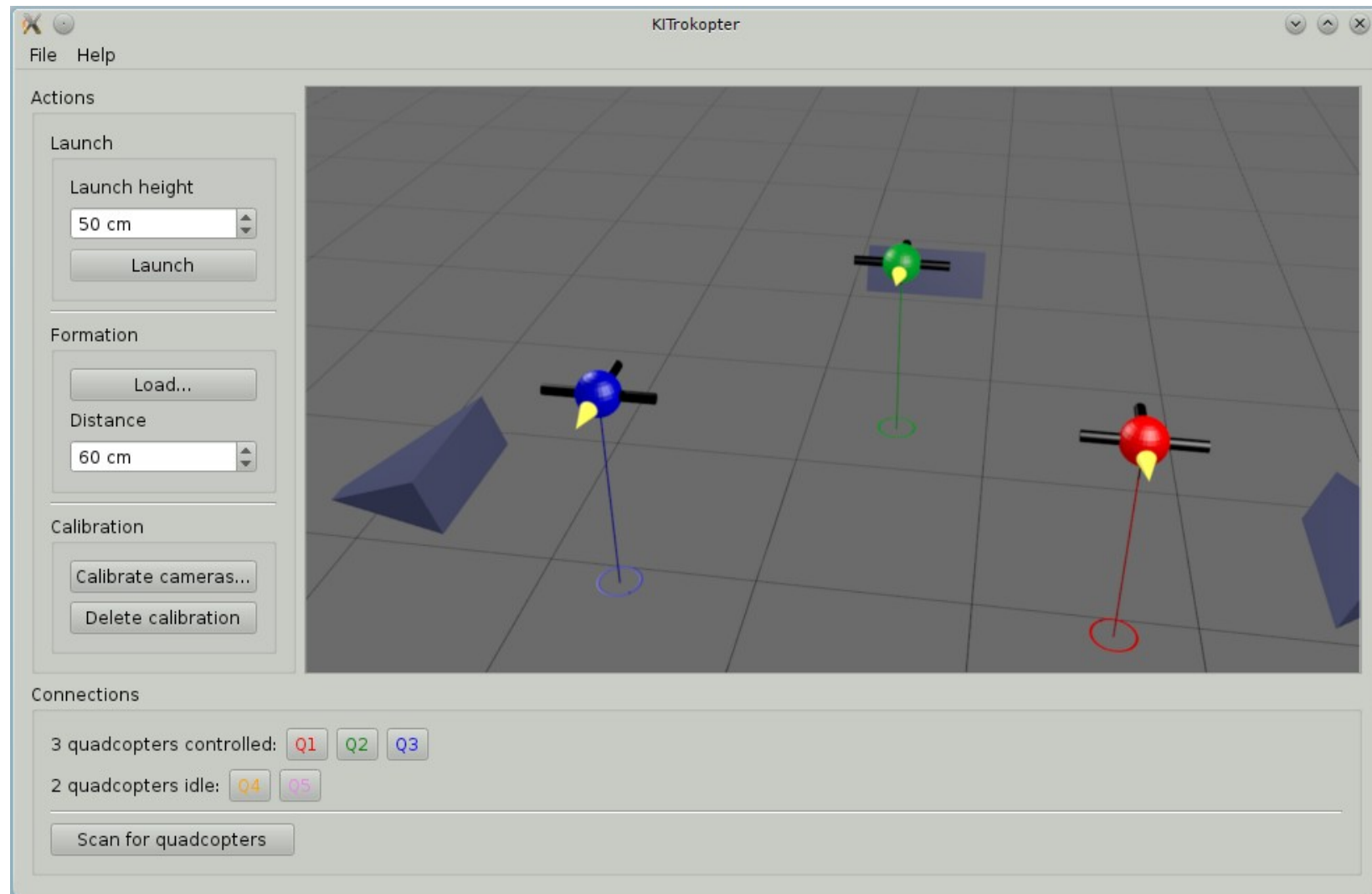
Controller Funktionalität für die GUI nach
dem MVC Modell

GUI-Applikation





Ziel



Ziel

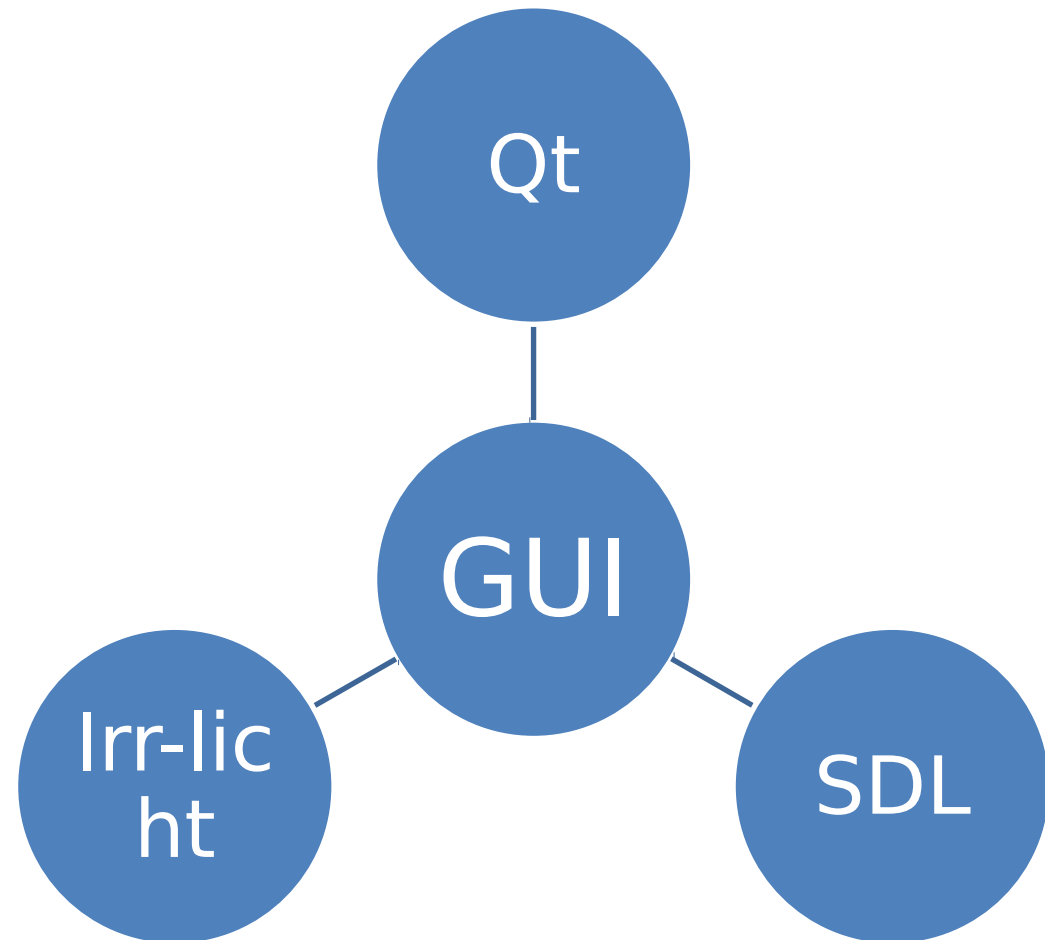
- Steuerzentrale
 - Multikamerakalibrierung
 - Systemstart
 - Verwaltung der Quadrocopter
- 3D-Ansicht

Herausforderungen

- Parallele Entwicklung der API
 - Blockierte Entwicklung der GUI
 - Testen und Debugging gleichzeitig für beide Module

Herausforderungen

- Integration verschiedener Bibliotheken



Fazit

Herausforderungen

- Aufwendiger Entwicklungszyklus
 - Immer Neustart des Gesamtsystems
- Notwendig
 - Zugriff via VNC
 - Bedienung der GUI nicht automatisierbar
- Kaum automatisiertes Testen möglich

Herausforderungen

- Häufige Reparaturen von kaputten Quadrocoptern
- Abhängigkeit aller Module
 - Entwicklung
 - Problemerkennung
 - Testen

Herausforderungen

- Stabilisierung der Quadrocopter
- Einschränkungen des selbstentwickelten Tracking mit Kinects:
 - Zu hohe Latenz bei hoher Fluggeschwindigkeit
 - Kleiner Trackingbereich
- Wahl der Koeffizienten der Regler

Endergebnis

- Sprachen: C++, Python
- Tools:
 - cflib
 - libfreenect, OpenCV, cvBlob
 - ROS
 - Matlab, AMCC Toolbox
 - Irrlicht, SDL, Qt

Endergebnis

- Tracking der Quadrokopter mit Hilfe von Kinect-Kameras
- Positionsbestimmung der Quadrokopter mit Hilfe von Matlab
- Kommunikation durch ROS
- Flugsteuerung und Steuerung durch die GUI
- API

Ausblick

- Andere Trackingansätze
 - Infrarotmarker
 - Tiefensensor
 - Hardwaresynchronisierte Kameras
- Andere Steuerungsansätze
 - Vorberechnung der Position
 - Komplexere Regler