Protokoll: Parameterabstimmung der Bildverarbeitung für die Knoten- und Kantenerkennung

Lukas Schmid

25. Mai 2025

Einleitung

Ziel dieses Protokolls ist es, die Vorgehensweise zur Parametereinstellung bei diesen zwei Bildverarbeitungsaufgaben zu dokumentieren:

1. Farbkorrektur mittels HSV-Schwellenwertfilterung.
2. Kreiserkennung mit der Methode `cv2.HoughCircles`.

Wir verwenden diese zwei Verfahren für die Winkelerkennung von ausgehenden Kanten eines Knotens.

Beide Verfahren wurden mit einem selbst programmierten, interaktiven OpenCV-Tool realisiert, das eine Echtzeit-Anpassung der Parameter via Trackbars ermöglicht.

1 Farbkorrektur mit HSV-InRange

Ziel

Das Ziel ist die Extraktion eines bestimmten Farbbereichs (z.B. ein farbiger Marker) aus einem Kamerabild oder Foto. Dazu wird das Bild in den HSV-Farbraum umgewandelt und mit `cv2.inRange` ein Binärbild erstellt, das nur die gewünschten Farben enthält.

Verwendete Parameter

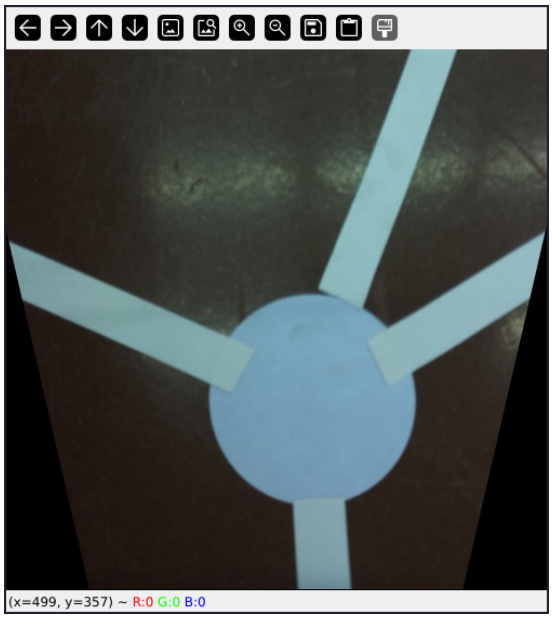
- **Low H / High H**: Farbtonbereich (Hue), Wertebereich: 0–179
- **Low S / High S**: Sättigung (Saturation), Wertebereich: 0–255
- **Low V / High V**: Helligkeit (Value), Wertebereich: 0–255

Vorgehensweise zur Parametereinstellung

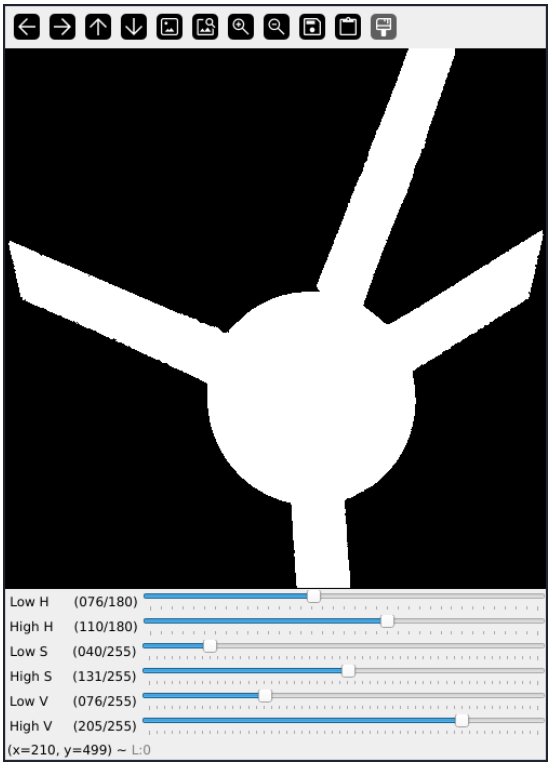
1. Die Kamera oder ein Beispielbild wird geladen.

```
python color_calibrator.py --image image.jpg
python color_calibrator.py --camera 0
```

2. Das Bild wird in den HSV-Farbraum konvertiert.
3. Mit Hilfe von sechs Trackbars werden die unteren und oberen Grenzen für H, S und V interaktiv angepasst.
4. Ziel ist es, dass das gefilterte Binärbild (Maske) nur die gewünschte Farbe enthält und möglichst wenig Störung zeigt.
5. Die Parameter werden notiert, sobald die Erkennung stabil und eindeutig ist.



Originalbild



gefiltertes Binärbild mit Trackbars

HSV-Schwellenwertfilterung

Final gewählte Parameter

Dieses Verfahren wurde mit verschiedenen Lichtverhältnissen mit verschiedenen Knoten durchgeführt. Daraus wurden die nachfolgende Durchschnittswerte ausgewählt:

```
HSV values: 76/110 40/131 76/205
```

Diese Parameter filtern einen weissen Farbbereich mit ausreichender Sättigung und Helligkeit.

2 Kreiserkennung mit HoughCircles

Ziel

Ein Kreis (Knoten) soll robust und eindeutig erkannt werden.

Verwendete Parameter

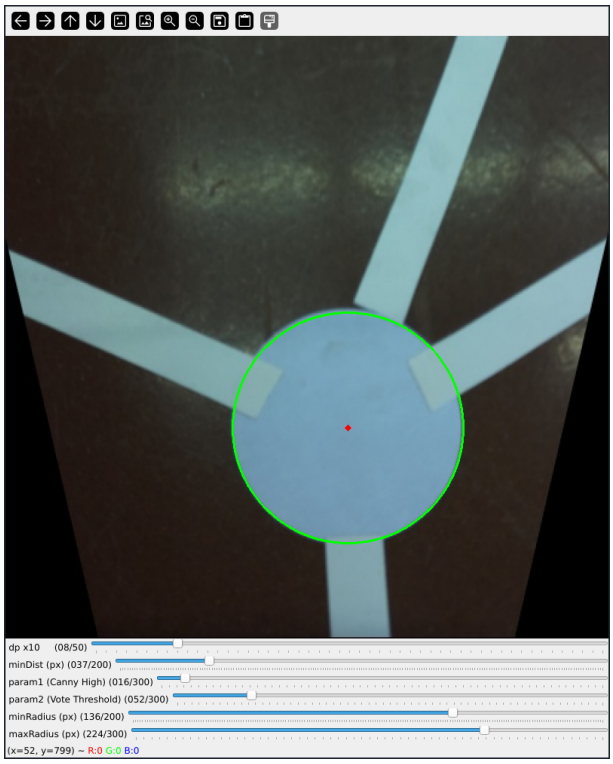
- **dp**: Inverser Verkleinerungsfaktor für den Akkumulator (`dp = 1.2` entspricht etwa 1/1.2 der Originalgrösse).
- **minDist**: Minimaler Abstand zwischen Kreiszentren, um Duplikate zu vermeiden.
- **param1**: Hoher Schwellenwert für den internen Canny-Kantendetektor.
- **param2**: Schwelle für die Zentren-Akkumulation (Empfindlichkeit der Kreiserkennung).
- **minRadius / maxRadius**: Erlaubter Radiusbereich für die zu erkennenden Kreise.

Vorgehensweise zur Parametereinstellung

1. Die Kamera oder ein Beispielbild wird geladen.

   ```
   python node_detection_calibrator.py --image image.jpg
   python node_detection_calibrator.py --camera 0
   ```

2. Das Bild wird in Graustufen konvertiert und mit einem Gauss-Filter (`cv2.GaussianBlur`) geglättet.
3. Der Benutzer kann interaktiv alle sechs Parameter über Trackbars verändern.
4. Zunächst wird der Radiusbereich so eingeschränkt, dass nur der gewünschte Kreis grössentechnisch erfasst wird.
5. Danach wird `dp` feinjustiert, um die Auflösung des Akkumulators zu optimieren.
6. Anschliessend werden `param1` (Kanten) und `param2` (Zentrumserkennung) angepasst, bis nur der relevante Kreis erkannt wird.
7. Die Parameter werden notiert, sobald die Erkennung stabil und eindeutig ist.



Kreis/Knoten Erkennung

Final gewählte Parameter

Dieses Verfahren wurde mit verschiedenen Lichtverhältnissen mit verschiedenen Knoten durchgeführt. Daraus wurden die nachfolgende Durchschnittswerte ausgewählt:

```
dp = 1.2
minDist = 30
param1 = 100
param2 = 80
minRadius = 20
maxRadius = 50
```

Diese Konfiguration führt zu einer stabilen Detektion eines einzelnen Kreises im relevanten Bildbereich.

Fazit

Durch die interaktive Abstimmung der Parameter mit Hilfe von OpenCV-Trackbars konnten sowohl der gewünschte Farbbereich als auch der kreisförmige Knoten schnell und präzise erfasst werden. Beide Tools bieten eine effiziente Grundlage für eine robuste Knoten- und Winkelerkennung