



Praktikum I – Homogene Kooordinaten; ImageJ, OpenCV

BV IM230 WS 2017/18, Prof. A. Siebert

Aufgabe 1. Homogene Koordinaten - Nicht-euklidische Geometrie

Diese Aufgabe können Sie auf Papier oder mit Ihrem Lieblingstool lösen.

- a. Gegeben seien die Geraden $x = 3$ und $x = 5$. Berechnen Sie deren Schnittpunkt S_1 .
- b. Gegeben seien die Geraden $y = 4x + 1$ und $y = 4x + 6$. Berechnen Sie deren Schnittpunkt S_2 .
- c. Berechnen Sie die Gerade l_∞ durch S_1, S_2 .
- d. Verifizieren Sie, dass der Punkt $\mathbf{P} = (10, 6, 1)^T$ nicht auf l_∞ liegt.

Aufgabe 2. 3-d Rotationsmatrix

Gegeben sei die extrinsische Matrix

$$\mathbf{M}_{\text{ext}} = \begin{pmatrix} 0.4534747 & -0.8528619 & 0.2588190 & 47.11 \\ 0.6606284 & 0.1267112 & -0.7399421 & 08.15 \\ 0.5982731 & 0.5065283 & 0.6208852 & 42.00 \end{pmatrix}$$

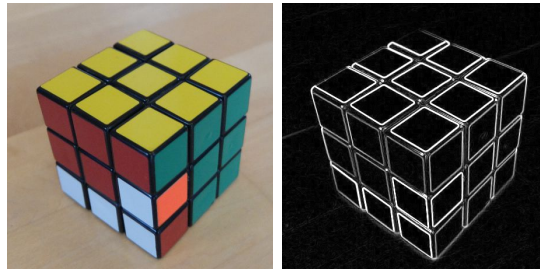
- a. Überzeugen Sie sich davon, dass die Rotationsmatrix von \mathbf{M}_{ext} orthogonal ist.
- b. Berechnen Sie die Rotationswinkel α, β, γ .

Die folgenden Aufgaben haben keinen theoretischen Anspruch, sondern sind eine Aufforderung an Sie, sich mit ImageJ und OpenCV bekannt zu machen.

Aufgabe 3. ImageJ: Sobel

Implementieren Sie ein Plugin, das ein gegebenes Farbbild in ein Grauwertbild umwandelt und auf diesem die Kanten mittels des Sobel-Operators findet.

ImageJ verfügt über die Methode `ImageProcessor.findEdges()`, welche den Sobel-Kantendetektor umsetzt. Diese Methode überschreibt das gegebene Bild. Nutzen Sie deshalb zunächst die Methode `ImageProcessor.duplicate()`, um eine Kopie des Eingabebildes zu erhalten.



Aufgabe 4. OpenCV: Canny

Implementieren Sie ein Programm, das ein gegebenes Farbbild in ein Grauwertbild umwandelt und auf diesem die Kanten mittels des Canny-Operators findet.

Wie schon im Skript-Beispiel gesehen, sollte dem Canny-Operator ein Gaußscher Mittelwertfilter (`GaussianBlur()`), mit Parameter σ , vorangehen.

Experimentieren Sie mit verschiedenen Werten von σ und den Hysterese-Parametern.

Verwenden Sie nach Möglichkeit die aktuelle Version 3.3.0 von OpenCV. Sie entscheiden selber, welche Sprache (C++, Python oder Java), welches Betriebssystem und welche IDE Sie verwenden wollen.

Wenn Sie jetzt noch Zeit und Lust haben, dann können Sie versuchen, die OpenCV-Funktionalität aus einem ImageJ-Plugin heraus aufzurufen. Vorgehen wie im Skript (Eingabebild in ImageJ laden, Umwandlung in OpenCV-Format, usw.). Sie können dazu die Methoden aus der Klasse `OCV2IJ.java` zur Umwandlung der Bildformate (nur Grauwertbilder) verwenden.