

Bildverarbeitung IM230
WS 2022/23
Prof. A. Siebert, PhD

Vorschläge Studienarbeiten

1. Rektifizierung von Stereo-Bildern

Quelle: Hartley/Zisserman, *Multiple View Geometry*, p.302-307.

Implementieren Sie die Stereo-Bild-Rektifizierung gemäß Hartley/Zisserman. Diese berechnet die Homographien für das Bildpaar aus der Fundamentalmatrix (und nicht wie Fusiello et al. aus bekannten Kameramatrizen). Wenden Sie die gefundenen Homographien auf die Stereo-Bilder an, um zu zeigen, dass die Epipolarlinien danach tatsächlich horizontal verlaufen.

Vergleichen Sie Ihre Ergebnisse mit denen von Fusiello (Praktikum IV) und von OpenCV `stereoRectifyUncalibrated()`.

2. Kamera-Rotation aus Fluchtpunkten

Quelle: Hartley/Zisserman, *Multiple View Geometry*, p.215

<https://cgcooke.github.io/Blog/computer%20vision/linear%20algebra/2020/04/12/R-From-Vanishing-Points.html>

Implementieren Sie ein Programm, dass aus zwei Bildern, die mit der gleichen Kamera (= identische intrinsische Matrix \mathbf{K}), aber aus unterschiedlichen Richtungen aufgenommen worden, die Rotation \mathbf{R} zwischen den beiden Kameras berechnet, gemäß dem Verfahren von HZ p.215.

Um an die Matrix \mathbf{K} zu gelangen, können Sie entweder eine volle Kalibrierung in OpenCV (mittels Kalibrierungsmuster, `Calib3d.calibrateCamera()` und $n > 10$ Bildern) durchführen oder drei orthogonale Fluchtpunkte im Bild identifizieren.

Vergleichen Sie Ihre Ergebnisse mit der *Ground Truth* und mit OpenCV `recoverPose()`.

3. Triangulierung / Tiefenwertberechnung

Quelle: Hartley/Zisserman, *Multiple View Geometry*, p.310-318

Implementieren Sie die Berechnung eines 3-d Punktes \mathbf{X} gemäß Algorithmus 12.1 in HZ p.318. Die erforderliche Fundamentalmatrix \mathbf{F} können Sie genau wie in (2) aus einer Kalibrierung extrahieren oder auch aus ausgewählten korrespondierenden Punkten berechnen (OpenCV: `Calib3d.findFundamentalMat()`).

Vergleichen Sie Ihre Ergebnisse mit der *Ground Truth*. Falls Sie eine volle Kalibrierung durchgeführt haben, können Sie Ihre Werte mit OpenCV `Calib3d.triangulatePoints()` vergleichen), ansonsten mit der Triangulierungsmethode aus Abschnitt 12.2 in HZ p.312.