

# Übungsblatt 5

---

## Aufgabe 1: Bildregistrierung

Implementieren sie den Algorithmus zur Registrierung von Bildern. Nutzen sie dabei Marker, die sie per Hand gesetzt haben (mindestens 4). Zeigen sie die Funktionalität ihres Algorithmus auf Basis eines selbst gewählten Beispiels.

## Aufgabe 2: Zhangs Algorithmus

Implementieren sie den Algorithmus von Zhang, wie er im Skript beschrieben ist. Nutzen sie dazu gerne die Beispielbilder aus Moodle und setzen sie die Marker per Hand.

Sie können auch alternativ die OpenCV Methode `cv2.findChessboardCorners()` nutzen um die Marker zu finden.

Ihr finales Ergebnis sollte eine Rotation/Translation Matrix wie in der Vorlesung sein. Optional können Sie sich die Datei `Maximum_Likelihood.py` aus Moodle herunterladen und diese nutzen, um die Ergebnisse zu optimieren. Nutzen Sie hierfür die Methode `optimize()` aus der Datei.

Ihr Ergebniss sollte sowohl eine Rotations/Translations Matrix, als auch die Matritzen A, B und H (Homography) aus der Vorlesung sein. Mit Hilfe der `optimize()` Methode erhalten Sie die optimierten Matritzen A und D (Distortion Coefficients), mit deren Sie die Bilder mit der Methode `cv2.undistort(image, A, D)` entzerren können.

**Wichtig:** Die Methode `optimize()` benötigt eine Mindestanzahl an Bildern, nutzen Sie deshalb am besten alle Bilder aus der Zip-Datei `chess_images.zip`.

**Hinweis:** Die Schach-Bilder aus der Datei `chess-images.zip` haben eine Größe von [6,9]. Sie sollten also mit `cv2.findChessboardCorners()` 54 Marker finden. Die Weltkoordinaten sollten ebenso 54 an der Zahl sein. Die Länge eines Schachbrett-Quadrats beträgt `12.5mm`, dies benötigen Sie zur Berechnung der Weltkoordinaten.