



# 3D Computer Vision

Sommersemester 2020 - Übungsblatt 3

WSI / Visual Computing

Dozent: Prof. A. Schilling

Tutor: Dipl.-Inf. M. Lange

## Stereo Vision

Im ersten Teil dieses Aufgabenblattes soll eine Disparity-Map anhand eines bereits rektifizierten Beispielbildpaares berechnet werden.

### a) Korrespondenzen (2 Punkte)

Schreiben Sie die Namen aller Ihrer Teammitglieder in den Dateinamen der Datei `Team_...`.

Als Rahmenprogramm ist eine GUI gegeben, welche zwei Bilder laden und nebeneinander anzeigen kann. Wird mit der Maus in das linke Bild geklickt, soll mittels einer Korrelationsfunktion (SSD) ein drittes Bild erstellt werden, welches für jeden Punkt des zweiten Bildes die Stärke der Korrelation zum markierten Punkt des ersten Bildes darstellt. Außerdem wird die Position der stärksten Korrelation bestimmt und im zweiten Bild dargestellt.

Anhand der Variablen `wx` und `wy` wird die Fenstergröße für die Korrelation bestimmt. Über die Option *Grayscale* werden die Bilder erst in Graustufenbilder konvertiert, bevor sie weiter verarbeitet werden. Die anderen Parameter sind für die erste Teilaufgabe nicht relevant.

Implementieren Sie die im Rahmenprogramm vorgegebenen Funktionen `correlation` und `responseImage`. Achten Sie darauf, dass die Funktionen RGB- und Graustufenbilder verarbeiten können.

### b) Rektifizierte Bildpaare (2 Punkt)

Zu dem Aufgabenblatt sind rektifizierte Bildpaare gegeben. Bei rektifizierten Bildpaaren liegen korrespondierende Punkte auf der selben Bildzeile.

Für diese Bildpaare soll Ihr Programm nun zu jedem Punkt des ersten Bildes die entsprechende Korrespondenz im zweiten Bild ermitteln. Optimieren Sie Ihr Programm so, dass es nicht mehr das gesamte Bild sondern nur noch die entsprechenden Bildzeilen nach Korrespondenzen absucht. Implementieren Sie dafür die Funktion `responseImageRect`.

Über die Option *Rectified* können Sie die Optimierung aktivieren und über den Parameter *Search Range* die Suche über eine komplette Bildzeile auf einen kleinen Bereich begrenzen.

### c) Disparity-Map (2 Punkte)

Berechnen Sie den Abstand zwischen den Punkten im ersten Bild und deren Korrespondenz im zweiten Bild. Dieser Abstand wird Disparität genannt. Implementieren Sie dafür die Funktion `computeDisparity`.

Über die Schaltfläche *Compute Disparity-Map* können Sie die Funktion ausführen.

**Achtung:** Die Berechnung kann sehr lange dauern.

#### d) Left-Right-Check (2 Punkte)

Zum Unterdrücken falscher Korrespondenzen soll ein Left-Right-Check implementiert werden: Nur diejenigen Korrespondenzen, deren Korrespondenz ihre Korrespondenz auch wieder sie selbst sind, sollen übernommen werden. Implementieren Sie dafür die Funktion `computeLR`.

Über die Schaltfläche *Left-Right-Check* können Sie die Funktion ausführen.

**Achtung:** Die Berechnung kann sehr lange dauern.

## Camera Calibration

Im zweiten Teil des Aufgabenblattes soll eine einfache Kamerakalibrierung mit Matlab durchgeführt werden.

#### Kalibrierung (2 Punkte)

Drucken Sie ein Checkerboard (Schachbrettmuster) mit quadratischen Kacheln aus. Sie können es selbst erstellen oder eines aus dem Internet nutzen. Für die Kalibrierung mit Matlab müssen die Rechtecke quadratisch sein, überprüfen Sie nach dem Druck dass diese nicht verzerrt sondern korrekt quadratisch sind und messen Sie die Größe, Matlab benötigt diese.

Wählen Sie eine Kamera und machen Sie viele Fotos von dem Muster. Einige bei denen das Muster nur in der Mitte des Bildes von etwas weiter weg (so dass es nicht den Großteil des Bildes ausfüllt) aus verschiedenen Perspektiven zu sehen ist und dann noch weitere Bilder in denen es in allen weiteren Bereichen des Bildes liegt auch aus unterschiedlichen Entfernungen.

Starten Sie den Camera Calibrator in Matlab: Apps → Image Processing and Computer Vision → Camera Calibrator. (Sie benötigen die *Camera Calibration Toolbox*).

Laden Sie zuerst nur die Bilder rein, auf denen das Muster nur klein (max.  $\frac{1}{4}$  der Bildebene) und in der Mitte des Bildes zu sehen ist. Machen Sie mehr Bilder, falls Matlab das Muster nicht in mindestens zehn davon erkennt. Wählen Sie 3 Koeffizienten für die radiale Verzerrung und lassen Sie Matlab die Kalibrierung durchführen.

Machen Sie einen Screenshot der sowohl eines der Bilder entzerrt zeigt als auch die pattern-centric view der Extrinsic und den Bar Graph der Rückprojektionsfehler.

Nun laden Sie alle Bilder, lassen Matlab kalibrieren und machen Sie wie davor einen Screenshot.

Gibt es Unterschiede in den beiden Bildern, wenn ja - welche und woher könnten diese kommen? Warum sind Bilder verzerrt (allgemein)? Sind Ihre Bilder verzerrt - wenn ja, wieso? Falls nein, wieso nicht? Hat Entzerrung einen Nachteil für den Informationsgehalt? Begründen Sie.

Exportieren Sie die Kameraparameter, geben Sie den Namen der Kamera und die Intrinsics inklusive der RadialDistortion Parameter in Ihrer Abgabe an.

**Abgabe:** Siehe ILIAS Übung 3