# 2D Vision and Deep Learning: Ex. Sheet 3

Due on Monday, 12th November 2018 at 10:00

**Lukas Schulz, Michelle Borth, Julian Brummer**

# Task 4

Wir verwenden die OpenCV Funktionen antelle des ToolKits um die Kamera zu kalibrieren. Aus dem erste Schachbrett-Bild wird dann die Verzerrung herausgerechnet. Siehe calibratecam.py.

# Task 7

$$\begin{aligned} I^{F_2} \otimes (I^{F_1} \otimes I^E) &= \begin{bmatrix}-1 & 0 & 1\end{bmatrix} \otimes (\begin{bmatrix}-1 & 0 & 1\end{bmatrix} \otimes \begin{bmatrix}1 & 2 & 1\end{bmatrix}) = \begin{bmatrix}-1 & 0 & 1\end{bmatrix} \otimes \begin{bmatrix}2 & 0 & -2\end{bmatrix} = \begin{bmatrix}0 & 0 & 0\end{bmatrix} \\ &\neq \begin{bmatrix}2 & 4 & 2\end{bmatrix} = \begin{bmatrix}0 & 2 & 0\end{bmatrix} \otimes \begin{bmatrix}1 & 2 & 1\end{bmatrix} = (\begin{bmatrix}-1 & 0 & 1\end{bmatrix} \otimes \begin{bmatrix}-1 & 0 & 1\end{bmatrix}) \otimes \begin{bmatrix}1 & 2 & 1\end{bmatrix} \\ &= (I^{F_2} \otimes I^{F_1}) \otimes I^E \end{aligned}$$

# Task 8

$$I^A = I^F * I^E = \begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 0 & 0\end{bmatrix} * \begin{bmatrix}0 & 1 & 1 & 1\\ 1 & 1 & 0 & 1\\ 0 & 0 & 1 & 1\\ 1 & 1 & 0 & 0\end{bmatrix} = \begin{bmatrix}3 & 2 & 2 & 2\\ 2 & 1 & 3 & 2\\ 2 & 2 & 1 & 1\\ 1 & 2 & 2 & 2\end{bmatrix}$$