

# Industrielle Bildverarbeitung (SS 2022)

Sebastian Edelmann (Hochschule RheinMain)

Praktikum 4 am 13. Juni 2022 / Abgabe: 26. Juni 2022

## Aufgabe 1: Affine Transformationen (6 Punkte)

Schreiben Sie ein Skript 'transformMain.m', in dem Sie zunächst das Bild 'object.png' laden. Erzeugen Sie weiterhin eine schwarze Bildfläche der Größe 1024x1024 px, welche später als Hintergrund verwendet wird. Es soll nun eine Funktion 'transform.m' geschrieben werden, der sowohl das Objekt-Bild als auch der Hintergrund übergeben werden sollen. Weiterhin soll die Funktion die Parameter für die Skalierung sowie Verschiebung in beide Richtungen und einen Winkel für die Rotation übergeben bekommen.

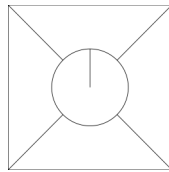


Abbildung 1: Objekt-Bild

In der Funktion sollen zwei *affine2d*-Objekte mit den entsprechenden Parametern für Skalierung und Rotation erzeugt werden. Mittels *imwarp* lassen sich diese nacheinander auf das Objekt-Bild anwenden. Daraufhin soll das Objekt-Bild auf den Hintergrund übertragen werden und um die angegebenen Werte verschoben werden. Die Größe des Hintergrundes soll sich nicht verändern lassen, sodass das Objekt-Bild sich auch aus dem Hintergrund schieben lässt. Zuletzt soll das Ausgangsbild angezeigt und somit die Anwendung der Funktion demonstriert werden.

## Aufgabe 2: Interpolation (7 Punkte)

Erstellen Sie eine Funktion 'intpol.m', welche mit einem x-Wert und einer Reihe von y-Werten verschiedene Interpolationen durchführen kann. Ein dritter Parameter soll die Art der Interpolation bestimmen, wobei

1. eine Nearest Neighbour Interpolation
2. eine lineare Interpolation

### 3. eine kubische Interpolation

ausgewählt werden können. Die Interpolation soll dabei ohne Zugriff auf die *interp*-Funktionen und somit selbst programmiert werden. Für die Interpolation sind weiterhin folgende Abtastwerte gegeben:

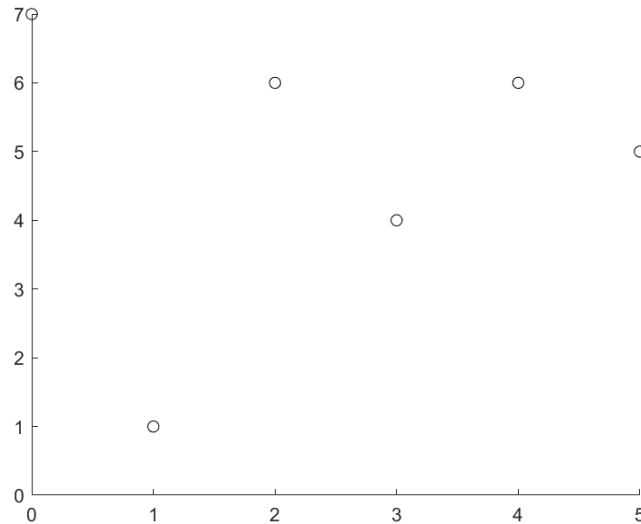


Abbildung 2: Abtastwerte

Führen Sie in einem Skript 'interpolationMain.m' mit der erstellten Funktion die drei verschiedenen Interpolationen der Abtastwerte im Bereich von zwei bis drei durch. Die Schrittweite der interpolierten Werte soll dabei 1/100 betragen. Stellen Sie daraufhin die Abtastwerte sowie die erstellten Interpolationen in einem gemeinsamen Graphen dar.

### Aufgabe 3: kNN-Klassifikation (7 Punkte)

Gegeben ist der Datensatz 'knnDataTrain.txt', der 20 klassifizierte zweidimensionale Trainingsdaten enthält. Die ersten 10 Daten gehören zur Klasse 1 und die letzten 10 zur Klasse 2. Daneben gibt es einen Datensatz 'knnDataTest.txt' mit 8 zweidimensionalen Testdaten ohne Label. Beide Datensätze sind in Abb. 3 dargestellt. Ihre Aufgabe besteht darin, die Testdaten mittels einer kNN-Klassifikation zu klassifizieren.

Schreiben Sie hierfür ein Skriptfile 'knnMain.m', in dem Sie:

1. Die beiden Datensätze zunächst laden und wie in Abb. 3 darstellen.
2. Die von Ihnen geschriebene Funktion 'knnClassifier' aufrufen, die als Parameter die beiden Datensätze sowie den k-Wert übergeben bekommt und die Klassenlabel der Testdaten zurückliefert. Um die k nächsten Nachbarn zu ermitteln, bietet sich die Matlab-Funktion '[sortA, Idx] = sort(A)' an, die neben dem sortieren Array auch die entsprechenden Indizes des unsortierten Arrays liefert. Für die Mehrheitsentscheidung können Sie die Matlab-Funktion 'mode' verwenden.

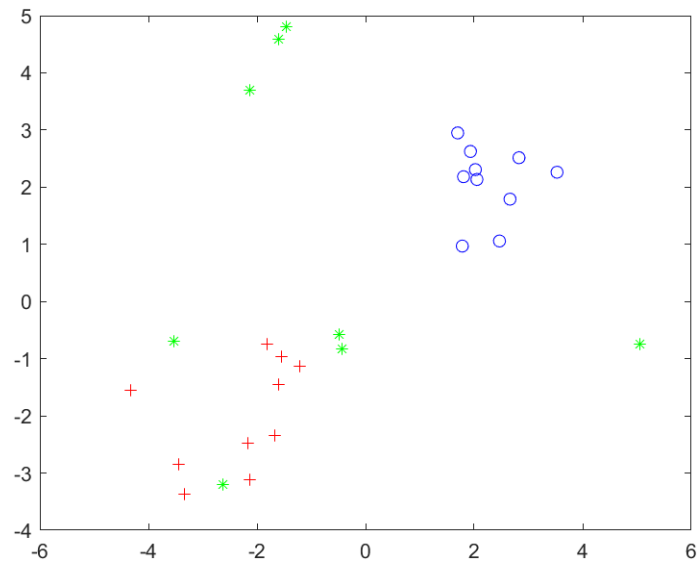


Abbildung 3: Trainingsdaten (+ und o) und Testdaten (\*)

3. Die klassifizierten Testdaten (Klasse 1 = rotes +, Klasse 2 = blaues o) zusammen mit den Trainingsdaten darstellen.