

# Bilder gezeichnet als Traveling-Salesman-Problem

Christoph Hollizeck, Fabian Heck

18. Oktober 2020

# Inhaltsverzeichnis

- 1 Robotikpraktikum
  - Aufgabenstellung
  - Herangehensweise
  - Praktikumsverlauf

# Aufgabenstellung

Ein Bild wird in ein Traveling-Salesman-Problem umgewandelt und vom Scara-Roboter abgefahren

# Umwandeln des Bildes in TSP



Abbildung: Originalbild

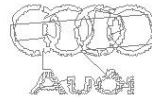


Abbildung: Umgewandelt

# Umwandlung erklärt

- Erzeugung eines Voroni Diagramms
- NN- Approximation um “nähste” Punkte zu finden

# Python Code

```
with open("./datapoints.txt", 'r') as f:
    content = f.readline()

convert = str(content).replace('(', '').replace(')', '')
convert = str(convert).replace('[', '').replace(']', '')

data = np.fromstring(convert, dtype=int, sep=',')
data = data.reshape(int(data.shape[0]/2), 2)
data = data.astype(str)

rcarriage = np.full((data.shape[0]), '\r', dtype="<U11")

x = np.full((data.shape[0]), "x", dtype="<U11")
y = np.full((data.shape[0]), "y", dtype="<U11")

data[:,0] = np.char.add(x, data[:,0])
data[:,0] = np.char.add(data[:,0], rcarriage)
data[:,1] = np.char.add(y, data[:,1])
data[:,1] = np.char.add(data[:,1], rcarriage)
```

- Strings werden bereinigt
- Punkte werden mit X und Y markiert

# Übergabe an Controller

```
def setX(self, line):  
    self.ser.write(data[line][0].encode())  
  
def setY(self, line):  
    self.ser.write(data[line][1].encode())  
  
def draw(self, data):  
    for i in range(data.shape[0]):  
        sleep(0.01)  
        self.setX(i)  
        sleep(0.01)  
        self.setY(i)
```

```
robo=ScaraRobo()
```

```
robo.draw(data)
```

- Die vorher generierten Strings werden encoded und mit einem kurzen delay an den Scara-Roboter übergeben

# Datenempfang

```
ISR(USART_RX_vect) {  
    buffer[bint] = UDR0;  
    if(buffer[bint] == '\r') {  
        buffer[bint] = '\0';  
        if(buffer[0] == 'x') {  
            memmove(buffer, buffer+1, strlen(buffer));  
            xPic[koordinatenrx] = atof(buffer);  
            koordinatenrx += 1;  
            uart_puts("\n\r x ");  
            uart_puti(atof(buffer));  
        } else if (buffer[0] == 'y') {  
            memmove(buffer, buffer+1, strlen(buffer));  
            yPic[koordinatenry] = atof(buffer);  
            koordinatenry += 1;  
            uart_puts("\n\r y ");  
            uart_puti(atof(buffer));  
        }  
        bint = 0;  
    } else {  
        bint++;  
    }  
}
```

- Mithilfe eines Interrupts wird die Übergabe des Strings abgefangen
- Das x bzw. y wird abgeschnitten und dann wird die Zahl als float in einem Array gespeichert



# Anfahren der Koordinaten

```
if(!lock){  
    gotoXY(xPic[currentstep], yPic[currentstep]);  
    lock = 1;  
    _delay_ms(10);  
    currentstep++;  
}
```

- Nun wird überprüft ob der Roboter gerade in Bewegung ist
- Falls nicht wird der nächste Punkt angefahren so lange bis das Bild fertig ist

# Zeitplanung

- Die ersten Aufgaben wurden sehr schnell erledigt (A-I) innerhalb von 4 Tagen
- Nach einer längeren Pause und ein paar Schwierigkeiten mit dem Scara-Roboter waren auch die anderen Aufgaben gelöst

# Ausblick/Reflexion

- Um Bilder aller Größen zu unterstützen wäre es notwendig die Bilder zu skalieren und zu shiften
- Die Ansprache der X,Y Koordinaten kann genauer und mit weniger Ruckler erfolgen (z.B. Zwischenpunkte definieren)

# Gibt es noch Fragen?