Road Sign Recognition

Deferne Sandra, Hertzog Marie, Hamada Mayumi
Goal

Find the way out of the maze
Goal

Find the way out of the maze

2 signs
- black
- wall
Over and over and over again

Movement in maze: Braitenberg

Detection of wall

Take a picture with camera

Picture analysis (fft)

Turn according to decision
Method: 3 steps

1. Choice of reliable recognition strategy in Matlab
   - Differentiate between horizontal and vertical orientation

2. Implementation of strategy in Webots
   - Simulation environment

3. Implementation of algorithm on e-puck
   - Real world conditions
Method: 3 steps

1. choice of reliable recognition strategy in matlab
differentiate between horizontal and vertical orientation

2. implementation of strategy in webots
simulation environment

3. implementation of algorithm on e-puck
real world conditions
Method: 3 steps

1. Choice of reliable recognition strategy in MATLAB
   - Differentiate between horizontal and vertical orientation

2. Implementation of strategy in Webots simulation environment

3. Implementation of algorithm on e-puck in real-world conditions
Matlab

amplitude of 2nd peak

fft

frequency of 2nd peak
fft on the horizontal stripes sign

sum on the columns

fft on sum of columns

sum on the rows

fft on sum of rows
fft on the broken stripes sign

sum on the columns

sum on the rows
Webots

- complete image processing
- black wall and white wall recognition
- increase the noise

- 💡 ambientIntensity [0,1], intensity [0,1]
- 🎨 colorNoise() [0,1]
## Video Webots

<table>
<thead>
<tr>
<th></th>
<th>ambientIntensity</th>
<th>intensity</th>
<th>colorNoise()</th>
</tr>
</thead>
<tbody>
<tr>
<td>initially</td>
<td>1.0</td>
<td>0.6</td>
<td>0.0</td>
</tr>
<tr>
<td>modified</td>
<td>0.8</td>
<td>0.1</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Webots
1. **static code**
   - image analysis part
     - Optimization
       - Distance to sign
       - Filter

2. **movement code**
   - Braitenberg to go forward
   - wall detection
   - move backward
   - rotation
Optimal distance
Filter

Threshold value
Dynamic threshold value
Black wall vs signs

Recognition of the picture as black based on its mean value

If black
-> turn around

If not black
-> filter
-> sum
-> fft
-> second peak
Movement code

- take into account the sensor distance of detection
- in Braitenberg, set the optimal time step
- be as precise as possible
- for rotation, time step will determine the angle of rotation
Video e-puck
## Performance: light conditions

<table>
<thead>
<tr>
<th></th>
<th>stripes</th>
<th>broken stripes</th>
<th>black wall</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>natural light</strong></td>
<td>16/20</td>
<td>15/20</td>
<td>9/10</td>
</tr>
<tr>
<td><strong>neon light</strong></td>
<td>15/20</td>
<td>16/20</td>
<td>10/10</td>
</tr>
<tr>
<td><strong>natural + neon light</strong></td>
<td>16/20</td>
<td>16/20</td>
<td>3/10</td>
</tr>
</tbody>
</table>
Performance E-Puck
Conclusion

- hard to deal with real world conditions
- necessity to come up with optimization ideas (filter, optimal distance)
- luminosity problems
- easier to distinguish the signs orientation than to differentiate them from the black wall
Thanks!