ROAD SIGN RECOGNITION

Antoine El-Hani and Coralie Kowalski
Outline

- Introduction
- Method and Algorithms
  - Code Structure
  - Languages and Platforms
  - Notes on E-Puck
- Experiments, Results and Statistics
- Conclusion
The goal is to get the E-puck out of a maze using Road Sign Recognition.

The IR sensors (0 and 7) are used in order to detect the obstacles (road signs).

The camera is used for the Road Sign processing.
Method and Algorithms (1)

Code structure

Infinity Loop

- Initialisation
- Move forward
- Detect obstacle (IR sensors)
- Stop
- Take a picture (Camera)
- Convert the picture to B&W
- Process the image
- Take a decision
- Turn according to D
- Repeat

Set speed on both wheels
Distance (obstacle, sensors) < Threshold
Set speed = 0 on both wheels
Discrete Fourier Transform
Based on a Decision Factor (D)
Set different speeds on wheels
Matlab

- Used for our signal processing strategy development
- Allowed us to verify the logic for the implementation of our algorithm
- Display of the signals and the results

The Math behind the Magic

\[ X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \]

Eq. 1. Discrete Fourier Transform

\[ M(k) = |X(k)| \]

Eq. 2. Magnitude of DFT

\[ D = \sum_{i=1}^{N} M^\text{Rows}_i - \sum_{j=1}^{N} M^\text{Columns}_j \]

Eq. 3. Decision factor \( D \)
Webots

- Allowed us to accurately simulate the e-puck (its physical characteristics) and its environment
- Easy debugging process (display messages throughout the code)

We were able to select the size of variables used, which saved us time and memory on real e-puck

- No memory issues
- Most important part: Translating the algorithm from Matlab to C
- We modified the code progressively while debugging it
The e-puck code is quite similar to the one implemented on Webots. The main differences are the name of libraries and variables, the limited memory, and the physical values!

- We perfected the code, then change the threshold values accordingly (distance, speed, rotation steps).
- Finally we tested our algorithm and changed the D threshold value until the robot could sense the difference between black and regular signs.
The format of the picture taken is 40x40 pixels.

A big library is provided! (i.e. computation of FFT magnitude available).

FFT magnitude vector contains 256 values. However we only used the first 128 values as these vectors are symmetric (with respect to the middle values).

Threshold value of $D$ determined as $D = 300$

$D < -300$ Turn Left

$-300 < D < 300$ Turn Back

$D > 300$ Turn Right

$D < -300$ Turn Left

$-300 < D < 300$ Turn Back

$D > 300$ Turn Right

Symmetric vectors!
Two Experiments
Three Lighting conditions

1. Detection of one sign at a time

2. Cardboard Maze including 3 “Left” signs, 3 “Right” signs and 1 black sign

Results

1. Perfect score with good lighting conditions for the two experiments

2. The success rate diminishes accordingly to lighting conditions

Notes

1. Results for Experiment 1 are better than for Experiment 2

2. The robot has to get 7 signs straight correctly in order to succeed in the maze!

<table>
<thead>
<tr>
<th>Sign 1. A. a.</th>
<th>Well illuminated</th>
<th>Mildly illuminated</th>
<th>Dark</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 %</td>
<td>97 %</td>
<td>80 %</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sign 1. A. b.</th>
<th>Well illuminated</th>
<th>Mildly illuminated</th>
<th>Dark</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 %</td>
<td>95 %</td>
<td>70 %</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sign 1. B. a.</th>
<th>Well illuminated</th>
<th>Mildly illuminated</th>
<th>Dark</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 %</td>
<td>97 %</td>
<td>75 %</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sign 1. B. b.</th>
<th>Well illuminated</th>
<th>Mildly illuminated</th>
<th>Dark</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 %</td>
<td>90 %</td>
<td>65 %</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sign 1. C.</th>
<th>Well illuminated</th>
<th>Mildly illuminated</th>
<th>Dark</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 %</td>
<td>96 %</td>
<td>95 %</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Success rate of our algorithm in different lighting conditions for single signs

<table>
<thead>
<tr>
<th>Maze Experiment</th>
<th>Well illuminated</th>
<th>Mildly illuminated</th>
<th>Dark</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 %</td>
<td>75 %</td>
<td>55 %</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Success rate of our algorithm in different lighting conditions for the maze
A quick look at our maze experiment!
Overall this project was very interesting! It allowed us to understand and apply the basics of programming a robot.

The results of our experiments are quite satisfying, even though we did not have enough time to find a solution for our algorithm to work perfectly in poor lighting conditions.

The steps to complete the project were well designed since they allowed us to progress logically in order to perfect each step before going on to the next one!

Any questions?