Outline

A. Project overview
B. Webots / MATLAB
C. Our algorithm
D. Real world implementation
E. Testing and optimisation
F. Conclusion
A. Project overview

The main goal of the project is the sign recognition.
A. Project overview

We have to code a program that recognizes the sign and tells the robot which direction to take.

The project have two main parts:

- Webots / Matlab
- Real implementation using the e-puck and C language
B. Webots / Matlab

- Simulation on computer
- No noise difficulties

- We developed our algorithm in webots
- We used MATLAB to visualise pictures and FFTs done

This was the easy part.
C. Our image processing algorithm

1. Take a picture
2. Chose a few rows and columns
3. Feed them to FFT
4. Take the mean of the sum the magnitude of each transform
5. Make a decision
1. Take a picture

- From 640x400 colored pixels to 40x40 greyscale

Original image

Turn left
(very theoretical picture)
2. Chose a few rows and columns

- Chose 3 columns and 3 rows
- First, last and middle ones
- We now deal with **6 vectors**
Input for FFT: vector
Output: vector!
C. Our image processing algorithm

4. Sum and mean the magnitudes

- It is easy to sum vectors
- We take the mean of these sums

Sum\text{Rows} = 102.865

Sum\text{Cols} = 26.667
5. Make a decision

\[
\text{\textbf{SumRows}} \quad < \quad \text{\textbf{SumCols}}
\]

\[\Rightarrow \text{TUR\,\,N\,\,L\,\,E\,\,F\,\,T}\]
C. Our image processing algorithm

Results of the Webots part

- Good results

- Relatively easy (simulation):
  - No noise
  - Always good light condition
  - No problem with the robot’s mood

=> We do not have the « real world » problems
D. Real implementation with e-puck and C

- Challenging

- Main things to implement:
  1. Move, avoid obstacles
  2. Stop in front of walls
  3. Take pictures
  4. Image processing algorithm
  5. Turn 90° angles
  6. Do everithing continuously ( while {1} )
D. Real implementation with e-pcuk and C

**Move and turn**

- We used Brainteberg coefficient to avoid getting stuck in corners.
- We tested odometry but it did not work, thought we tested it many times.
- We found another solution: make the wheels turn in opposite direction during a certain amount of time so that the robot makes a quarter of turn or half a turn.
D. Real implementation with e-puck and C

Take a picture

- Code provided

- But did the robot actually take the picture...?

- ...LED signals can help us!
  - The e-puck basically communicates with LEDs now.
  - Take a picture  -> blink
  - Process picture  -> all LEDs on
  - Turn left       -> LED n°4 blink
  - etc
Run our algorithm

Problem we had to solve:
- From picture string to ordered columns and rows
- Take only 3 rows and 3 columns and store them in another variable
- Take the FFT
- Sum the FFTs and make a mean of the columns and the rows, in order to get only two numbers that we could compare

D. Real implementation with e-pcuk and C
In function of the result of our algorithm, the robot must turn left, right or turn back.

The real problems are in this part:
- Is the robot making the good decision?
- What are the results of the algorithm?
- What about the noises?

D. Real implementation with e-pcuk and C
E. Testing and optimization

- Once the code seemed to work, we tested it under different conditions
E. Testing and optimization

- Some problems remain *unsolved*...
  - The obstacle threshold only applies for the computer room conditions
  - Black sign sometimes not detected
  - Wake up problems: first process fails all the time
  - Process multiple times
  - Sign size and camera resolution
E. Testing and optimization

- We aimed for good results, but not enough time. The efficiency reached is more or less 90%.

<table>
<thead>
<tr>
<th>Table 1: Measurement of the success rate for each sign, normal conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Left</td>
</tr>
<tr>
<td>Right</td>
</tr>
<tr>
<td>Around</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Measurement of the success rate for each sign, normal conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>left</td>
</tr>
<tr>
<td>right</td>
</tr>
<tr>
<td>around</td>
</tr>
</tbody>
</table>
E. Testing and optimization : Demo

(https://goo.gl/photos/rfXghb7c9btttdjQ19)
E. Testing and optimization

- The optimization was long but it was like a little victory when it finally worked.

- We often worked next to the other group and we always wanted to be the best! It was a good motivation.

- Testing in all possible ways was pretty fun.
F. Conclusion

- Very interesting project with lots of challenges
- We learned a lot about signal processing and the tools we can use to manage these signals
- We learned about the challenges that the real world brings to us. The simulation are easier but less interesting and satisfying
- The things we learned will be useful for our future as engineers!
- E-pucks are grumpy
Thank you!

- Questions?