Line following using a camera with e-puck

Anaïs Ladoy  |  Noemi Romano  |  Lou Curchod

Supervisor: Zeynab Talebpour  |  Course: Signals, Instruments and Systems

Module 1: Follow the line
- Tracking IR sensor through black path
- Controlling speed to follow the line

Module 2: Avoid the line
- After tracking, move away to avoid the next line

Module 3: Camera implementation
- Understand the working principle of a camera
- Image processing for line detection

Conclusion
- Next steps: improve the line following algorithm

Prezi
Line following using a camera with e-puck

Anaïs Ladoy | Noemi Romano | Lou Curchod

Supervisor: Zeynab Talebpour | Course: Signals, Instruments and Systems

Mode 1: Follow the line
- Tracking black lines on white paper
- Following a new path given by the user

Mode 2: End the line
- Stop after turning and dropping the background
- Continues end of line

Mode 3: Move inside the room

Conclusion
- Any problems encountered during implementation?
Introduction

Goals of the project:
- Perform line following
- Avoid obstacles on the top of line following
- Simulation on Webots
- Test simple line following in real world
Camera configuration and image analyzes

- Picture of 40x40 pixels
- Conversion to gray-scale picture
- Analyze based on the counting of the pixels

On Webots
- Array of size 40x40
- Picture taken every 64 ms
- Iteration in function of x (image height) and y (image width)
- Count the black pixels

Real e-puck
- Size of acquisition limited by memory size of the microcontroller
- Picture taken every 125 ms
- Inversion of the coordinates system
- Select only the bottom picture
- Stored in one dimension table
On Webots

- Array of size 40x40
- Picture taken every 64 ms
- Iteration in function of x (image height) and y (image width)
- Count the black pixels
Real e-puck

- Size of acquisition limited by memory size of the microcontroller
- Picture taken every 125 ms
- Inversion of the coordinates system
- Select only the bottom picture
- Stored in one dimension table

e_po3030k_config_cam(sensor_x1, sensor_y1, sensor_width, sensor_height, zoom_fact_width, zoom_fact_height, color_mode).
Visualising

Example of image captured by the camera and read by Matlab

Example of the array analyze
Line following using a camera with e-puck

Anaïs Ladoy  |  Noemi Romano  |  Lou Curchod

Supervisor: Zeynab Talebpour  
Course: Signals, Instruments and Systems

Module 1: Follow the line
- Tracking a line through black path
- Controlling speed based on error value

Module 2: Find the line
- Apply the same concept but to detect the right boundary
- Continues adjusting value

Module 3: Obstacle avoidance
- The robot is able to detect an obstacle and stop
- Continues steering and moving

Conclusion
- The project is now complete
- Future improvements: use color tracking and obstacle avoidance

Prezi
Mode 1: follow the line

- Counting if there are enough black pixels.
- Checking if there is black pixels in the last line.
On Webots

Count the black pixels in the last line, right and left separately
- left > right : turn left
- right > left : turn right
- left = right : go forward
On the real e-puck

- Same method but adapted to work with real interferences
- Analyze of the three last lines
Results

- Bumps due to systematic counting of the pixels
- Augmentation of speed wheel in the curve
- Delay between capture and response

Best response from the e-puck for this part
Mode 2 : find the line

- When the e-puck sees a line in the background.
- Go forward and turn to join it.
On Webots

Analyze both halves of the image and turn in the direction where the count of pixels is higher.

U-turn: when the number of pixels is under 10
  • Go forward till the end of the line.
  • Rotate of 180°.
  • Meanwhile, the camera stops taking pictures.
On real e-puck

Same analyze but without the implementation of U-turn.
Results

- Uncertain response for the U-turn.
- Difficulties when the e-puck arrives perpendicularly to the line.
- Shadows in the world on Webots.
Mode 3: avoid obstacles

- The e-puck should first avoid obstacles.
- If an obstacle is on the line, it should surround it and go back on the line.
On Webots

Analyze of the proximity sensors:
- When sensor values in front (IR 6, 7, 0 or 1) are high: e-puck turns left.
- As long as the e-puck goes along the obstacle (IR 2 or 5 high): goes forward.
- When it reaches the corner: IR 2 or 5 is very high, it turns right.
- Sees the line again and goes back on it.
On real e-puck

- The real e-puck does not surround the obstacle but avoids them using Braitenberg coefficients.

\[
\text{speed}_{left} = \sum_{i=0}^{n} \alpha_{left,i} \left(1 - \frac{ps\_value_i}{ps\_range}\right)
\]

\[
\text{speed}_{right} = \sum_{i=0}^{n} \alpha_{right,i} \left(1 - \frac{ps\_value_i}{ps\_range}\right)
\]
Results

- Difficulties to distinguish wall and obstacle
- We didn't have the same response on Webots and for the real e-puck.
Conclusion

- Challenging to use C language and to translate our ideas in this language.
- Difficulties to adapt the code to the real world.
- A lot of parameters influence the performance of the e-puck (battery, camera sensitivity, luminosity, shadows).
- Using FFT could have been more efficient.

Project could be improved but it still works well!
Line following using a camera with e-puck

Anaïs Ladoy | Noemi Romano | Lou Curchod

Supervisor: Zeynab Talebpour
Course: Signals, Instruments and Systems

**Mode 1: Follow the line**
- Following a blue or red line
- Initializing

**Mode 2: Find the line**
- After finding the line, change the background and continue to the next mode

**Mode 3: Avoid obstacles**
- The robot should avoid obstacles and continue following the line

**Results**
- The robot successfully follows the line and avoids obstacles

Conclude
- The project successfully demonstrates the use of a camera and image processing for line following and obstacle avoidance in the e-puck environment.