

Comp3352 Computer Vision: Line Following

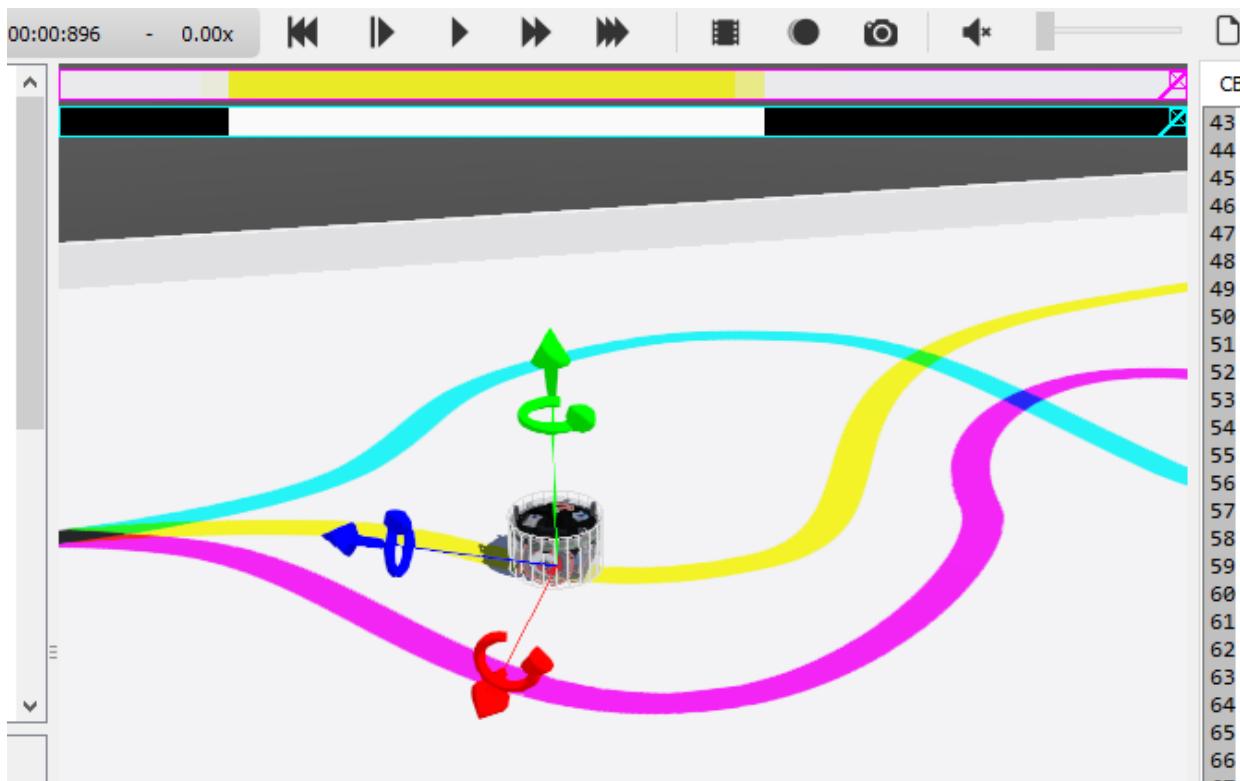
C.B.Price January 2021

Purpose	To use color information from a linear camera to follow a line.
Files Required	Webots project folders on website. Use the world CBP_3352_LinearCamera_1.wbt Controller is CBP_3352_LinearCamera_1.c
ILO Contribution	1
Send to Me	If you are working online, please send movie-clip of your solution.
Homework	

Activities

1 Coding the Controller

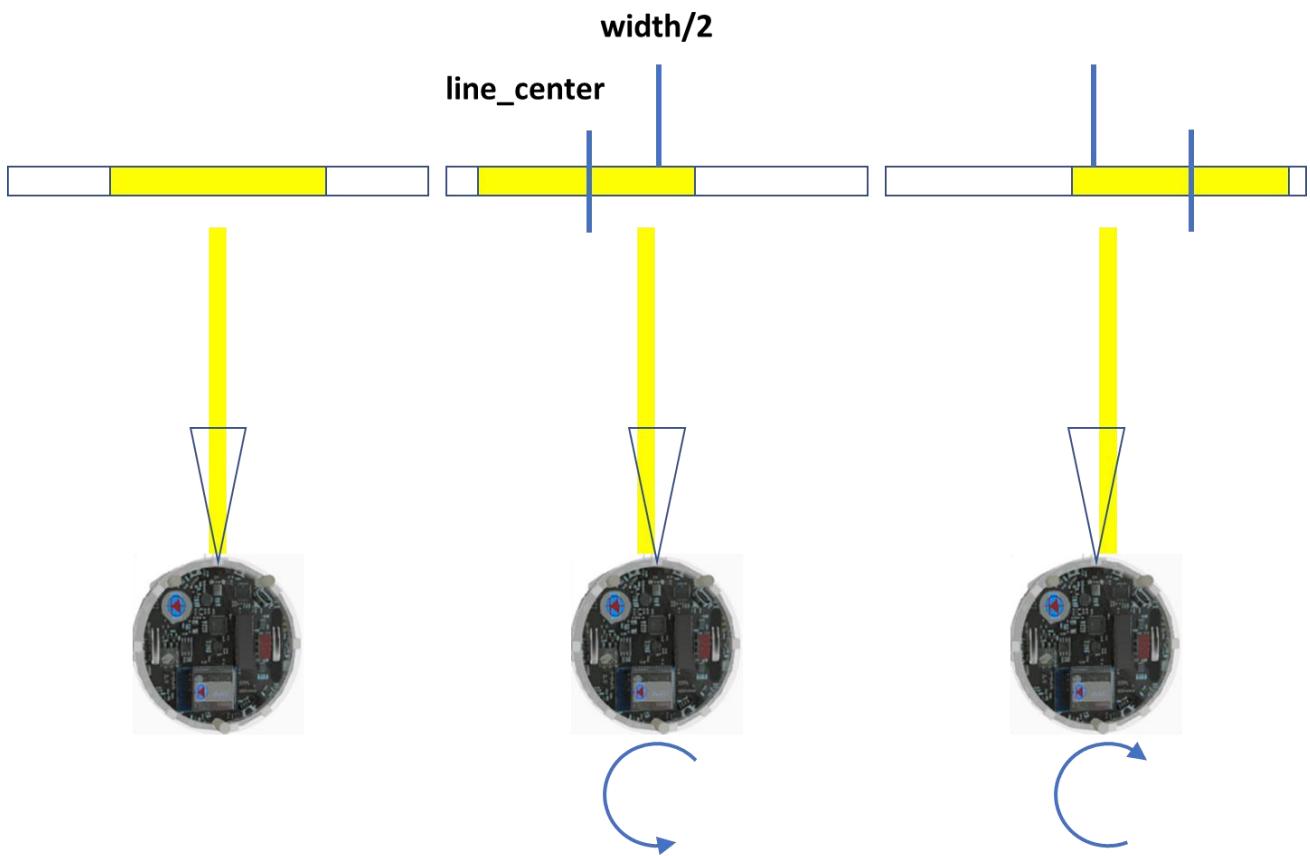
Let's start by getting the robot to follow the yellow line. So it's a good idea for it to start on the line like this



The camera has width 40 pixels and height 1 pixel, so is a “linear camera” and is pointing down as the camera frustum shows above. The image seen by the camera is shown at the top of the screen, and the image segmented into yellow pixels is shown underneath, represented as white pixels.

The basic image-processing code is supplied for you. This is located in the file **CBP_Helper_2.c** which is in the controller folder. The main function is **imageSegment(...)**. This takes in the camera image (called **image**) and outputs the segmented image **lineImage** which is also 40 pixels by 1 pixel. It also tells you the number of yellow pixels found.

The second function is **getAvLocation(...)** which returns the centre of the line in the segmented image and assigns it to the variable **line_center**. You will use this to generate the control signal to move the robot onto the line. Let's see how this works.



In the above diagrams, the white is the **lineImage** where the position of the yellow line is shown for three cases.

On the left, the yellow line is in the middle of the image, because the line is in the middle of the camera image. In the middle, the yellow line is to the left of the camera image, so the robot needs to turn anti-clockwise to get the yellow line to come back into the centre of the camera image. But of course, both motors will need some forward drive to make it move forwards.

- Write some code to get the **error** between the centre of the line and the image width. Use variables **line_center** and **width/2**. Divide your expression by **width** (if you have not done this) to normalize the error value.
- Now complete the expressions to generate the motor drive velocities. Here you must fill in the ??'s. Just think (and perhaps experiment with your code to gain understanding of the situation)

```
double fwdFactor = ?;  
omegaL = (fwdFactor + ??)*MAX_SPEED;  
omegaR = (fwdFactor + ??)*MAX_SPEED;
```

- Make sure you have uncommented lines as instructed in the code.

2 Possible Investigations – just for fun!

- Try to get the robot to jump across an intersection.

-
- (b) Create your own yellow track. This needs to be in the directory
...\\Desktop\\Webots_Release\\worlds\\textures
 - (c) Modify the controller so the robot can jump across intersecting colored lines
 - (d) Modify the function **imageSegment(...)** in the source file **CBP_Helper_2.c** so the robot will follow a line of a different color. You should refer to the color wheel discussed in the Mini-Lecture.
-