Comp2403 Computer Vision: Object Detection

C.B.Price October 2020

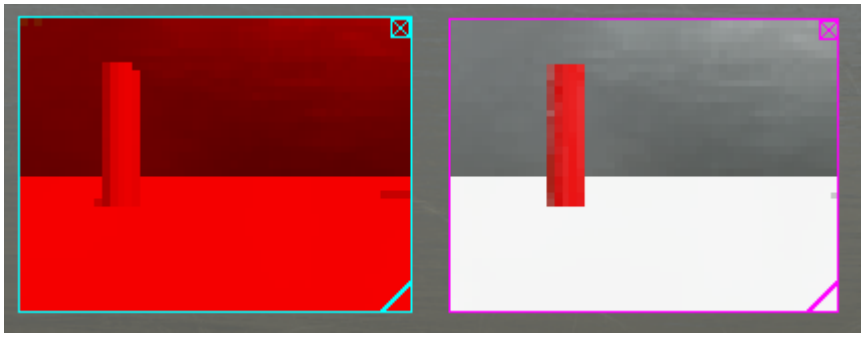
Purpose	To understand how to segment an image into objects of a single colour.
Files Required	Webots project folders on website. Use the world **CBP_ePuck_Camera_3.wbt**
ILO Contribution	1,2
Send to Me	If you are working online, please send movie-clip of your solution.
Homework	Look at CBP_Notes Chapter 4, sections 4.3 to 4.5

Activities

---

1 RGB and HSI Images

Open up the world **CBP_ePuck_Camera_3.wbt** and make sure that the controller **CBP_ePuck_Camera_3a.c** is selected and open in the editor. Expand the image display windows so you see something like this. The display on the left is giving you the Red component of the RGB image received by the camera



(a) Compile and run the script and you should see that tons of surfaces in the arena have a significant Red component. Therefore RGB images are not very useful to extract coloured objects.

(b) Look over the function **imageSegment_bad(…)** and note how the red pixels are extracted from the camera image, and copied directly into the **imageThresh** image.

(c) Now load up the controller **CBP_ePuck_Camera_3.c** in the same world. Look at the function **imageSegment(…)** which extracts the red hue from the RGB camera image and copies only red pixels into the **imageThresh** image. Consult CBP_Notes Chapter 4.

(d) Run the code and observe the correct extraction of objects with a red hue. You can copy-and-paste to add more coloured objects if you like

---

2 Detecting and moving to an object.

Continue with the world **CBP_ePuck_Camera_3.wbt** and the controller **CBP_ePuck_Camera_3.c**. There is a skeleton Finite State Machine.

(a) Complete the FSM code. You will need to do the following
(i) In STATE_SEARCH, add a line to transit to STATE_FOUND if the number of red pixels found **nr_red** is larger than some threshold, e.g., 10.
(ii) In STATE_FOUND add a call **int centre = getAvRedLocation(image1,width,height);** to return the horizontal centre of the object
(iii) Then add a test to see if the object is to the left of the centre of the image or to the right of centres and set **omegaL** and **omegaR** accordingly to make the robot turn correctly. Hint: you will need to use **width/2** plus a little offset.
(iv) Then transit to STATE_MOVETO_TARGET

(c) Check your code works for various starting places of the object (left or right of the image centre).

(d) Look at the HIS colour wheel, and try different ranges for red objects.

(e) You may want to introduce and detect other coloured objects.