Carnegie Mellon University Africa

18-799 Cognitive Robotics

Assignment 4
Determining the Pose of Bricks

Deadline: 18:00 Tuesday 30th March 2021

Problem Definition
Pose refers to the position and orientation of an object. The goal of this assignment is to develop an openCV computer vision application that determines the pose of coloured rectangular bricks placed on a flat surface. The pose of a brick is given by the location of the centroid and the angle ϕ between the major axis of the brick and the horizontal, where $0 \leq \phi < 180$, measuring a positive angle anticlockwise from horizontal; see Figures 1-5 and sample output.

You can assume that the camera views the brick from above, i.e. the principle ray of the camera is orthogonal to the plane of the supporting surface.

The application should accept image input from image files. The image file names are provided in an input file `assignment4Input.txt`.
The application should write the number of bricks and, for each brick, the coordinates of the centroid and the angle subtended by the major axis and the horizontal in degrees to the output file `assignment4Output.txt`.

The brick poses should be listed in increasing hue, starting with red. Remember that red has hue values of zero ± some small tolerance and that, in OpenCV the hue range is $0 \leq \text{hue} \leq 180$ (not 360, as you'd expect).

Input
The filenames of the images of the scenes containing coloured bricks.

Output
The output should begin with your student ID on a separate line.

This should be followed, for each test case, by the filename and the x and y coordinates in pixels and orientation in degrees of each brick, enclosed in parentheses. There should be one filename and set of pose data per line.

Sample Input
```
assignment4_0.png
assignment4_1.png
assignment4_2.png
assignment4_3.png
assignment4_14.png
```

Sample Output
```
studentid
assignment4_0.png: (288, 159,  57)
assignment4_1.png: (227, 151,  21)
assignment4_2.png: (208, 145, 142)
assignment4_3.png: (163, 153,  80) (259, 142, 105)
assignment4_14.png: (259, 190,  81) (165, 179, 177) (352, 193, 178)
```

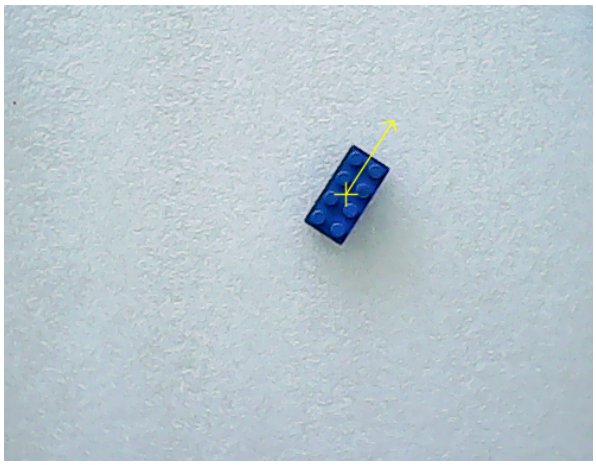

Figure 1.

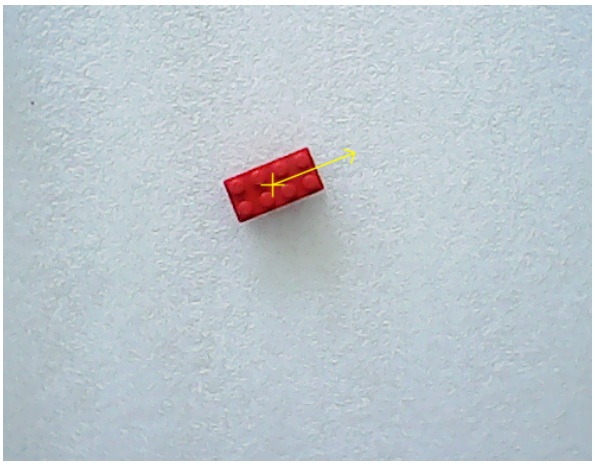

Figure 2.

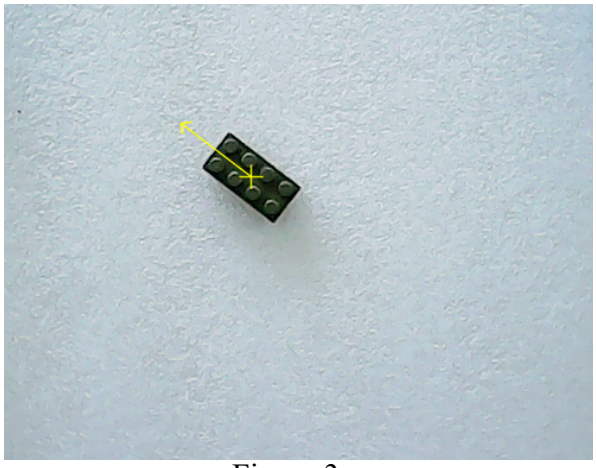

Figure 3.

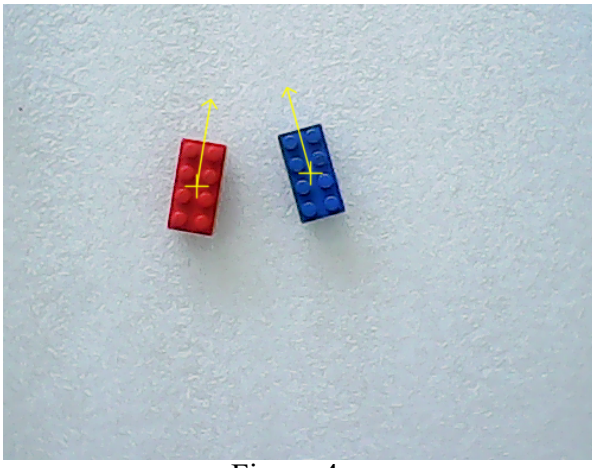

Figure 4.

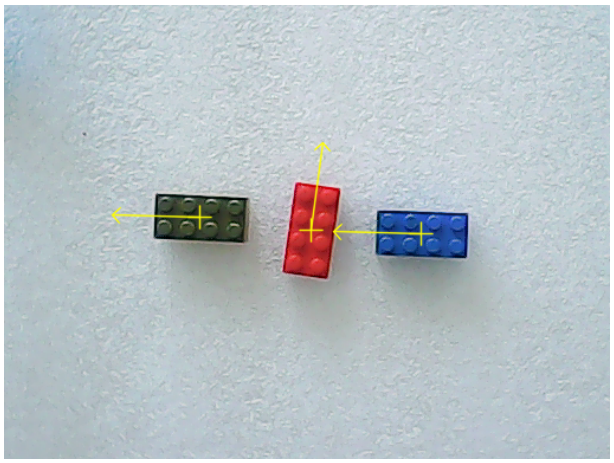

Figure 5.

Instructions
This is an individual assignment, not a group assignment. Collaboration is not allowed.

To help you get started, a skeleton program is provided. It provides the functionality for reading from the input file, reading and displaying and image, and writing to the output file. Specifically, six files are provided:

```
studentid.h
studentidApplication.cpp
studentidImplementation.cpp
CMakeLists.txt
package.xml
Assignment4Input.txt
```

In addition, the five image files listed in the input file are also provided.

Submit the three source code files named with your student ID, i.e.
```
mystudentidApplication.cpp
mystudentidImplementation.cpp
mystudentid.h
assignment3Input.txt
```
in a zip file named with your student ID by the deadline shown above.

The source code should contain adequate internal documentation in the form of comments. Internal documentation should include the following.

- A summary of the algorithm
- A summary of the manner in which the code was tested.

Do not forget to include your name in the internal documentation.

Marking Scheme
Marks will be awarded based on blind tests using images similar to those shown in Figs. 1-5, with the following content.

- One red brick:
  centroid: 5 marks
  orientation: 5 marks
- One blue brick:
  centroid: 5 marks
  orientation: 5 marks
- One green brick:
  centroid: 5 marks
  orientation: 5 marks
- One red and one blue brick:
  centroid and orientation x 2 20 marks
  correct order, sorted by hue: 10 marks
- One red, one blue, and one green brick:
  centroid and orientation x 3 30 marks
  correct order, sorted by hue: 10 marks