Departamento de Eletrónica, Telecomunicações e Informática

Course 8309 - Integrated Masters Degree in Electronics and Telecommunications Eng.
Class Visão por Computador na Indústria
Year 2020/21

Report

Iteration 2 and 3

Authors:
89280 Guilherme Lourenço
89243 João Alegria
90246 Raúl Kevin Viana
Class: P1 Group 3

Date: 24/05/21
Professors: António Neves
Daniel Canedo

Summary: This report intends to describe the work done by our group during iteration 2 and 3 of our timeline.

Introduction

The project proposed by the Visão por Computador na Industria subject aims to identify legos in a real environment without controlling the lighting conditions. This course follows a Project Based Learning methodology in which students learn while actively participating in the implementation of a real project.

At the end of the project it is expected that students will be able to obtain information about the dimensions and color of a set of legos through the use of algorithms, a camera and python libraries (such as, for example, OpenCV).

This document will present the progress made by our group during iterations 2 and 3. During these iterations, in addition to continuing the work done in the previous iteration (legos detection through its color and preprocessing), was also intended to improve the solutions presented in the previous one and also proceed with the use of the camera and raspberry set provided by the professors.

Region Growing

Region growing is a simple region-based image segmentation method. The method starts with an initial seed (in this case a pixel chosen by a mouse click) that propagates by checking if the neighboring pixels satisfy a specific condition (in this case HSV values) related to the seed's reference value. In order to increase the algorithm's performance with pieces that featured multiple tones due to heterogeneous lighting we implemented a dynamic moving mean method that updates the HSV reference value as the region grows.

This method will be integrated in the calibration process to extract the minimum and maximum HSV values for each color in order to create the color masks for the image segmentation step.

Calibration

Types of Distortion

Some pinhole cameras can present significant distortion in the images. This distortion introduced in them can be of two types:

- **Radial**: this type of distortion makes straight lines appear to be curved.
- **Tangential**: in tangential, distortions arise due to the fact that the image was not taken exactly in parallel with the work plane. Resulting in an image with sections that appear to be larger than they really are.

In figure 1, it is possible to verify the presence of these two types of distortion. In the image we can see that the more we drift away from the center, geometric figures, which are usually composed of straight lines, start to have a curved appearance (more flattened). The more distant a part of the image is from its center, the greater the radial distortion in that part.

Iteration 2 and 3

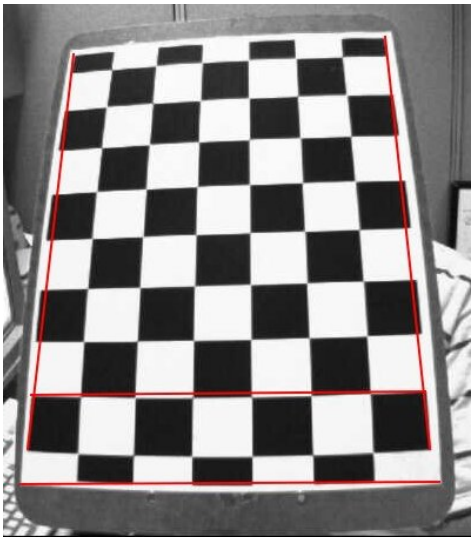


Fig. 1 - Radial Distortion

In Figure 2, taken with the raspberry, it appears that due to the way in which the photo was taken, it caused the upper part of the chessboard to look larger than it actually has.

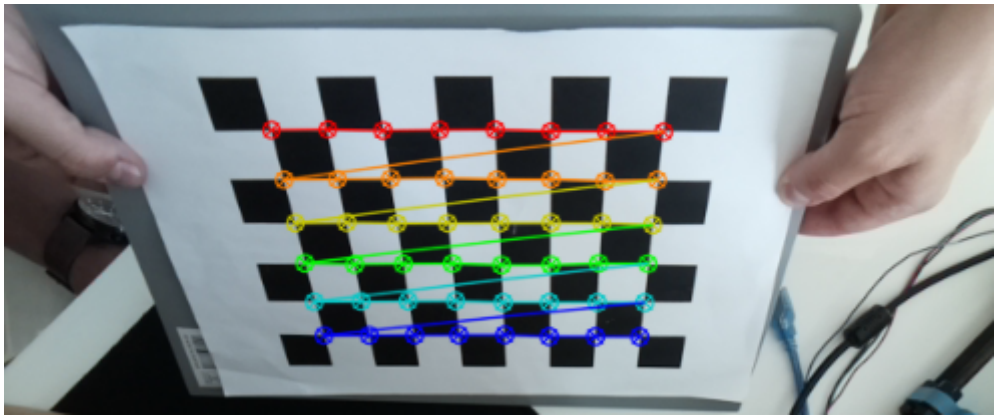


Fig. 2 - Tangential Distortion

The formulas below mathematically represent the distortions described above. In these, it is clear that with the increase of r, the distance to the center of the image, the greater the distortion in the image (both radial and tangential).

$$x_{radial} = x(1 + p_1 r^2 + p_2 r^4 + p_3 r^6)$$

$$x_{tangential} = x + [2p_4 xy + p_5(r^2 + 2x^2)]$$

$$y_{radial} = y(1 + p_1 r^2 + p_2 r^4 + p_3 r^6)$$

$$y_{tangential} = y + [p_4(r^2 + 2y^2) + 2p_5 xy]$$

Eq. 3 - Expressions for radial (left) and tangential (right) distortion

Still in the formulas above, it is concluded that there are 5 parameters associated with distortion (called distortion coefficients):

$$Parameters = p_1 p_2 p_3 p_4 p_5$$

Intrinsic and extrinsic parameters of the camera

To discover these 5 coefficients it is necessary to have some information about the camera, to be aware of its intrinsic and extrinsic parameters.
The intrinsic parameters are those that are exclusively associated with a given camera. These are the focal length and the optical center. With these it becomes possible to create a matrix that will serve to remove the noise caused by the lenses of a specific camera.
On the other hand, we have extrinsic parameters. These correspond to the translation and rotation vectors that make a match between the 3D coordinate systems with the image coordinates, i.e., coordinates of the "real world" are transformed into "virtual" coordinates.

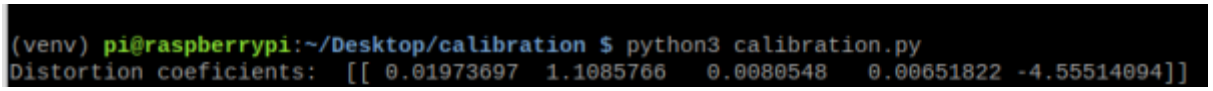


Fig. 3 - Distortion coefficients obtained

Having the intrinsic and extrinsic parameters, we can easily obtain the distortion coefficients p1, p2, p3, p4, p5. In figure 3, we have the coefficients obtained with the raspberry camera. To compute these values, the method suggested in the OpenCV documentation was used. This method consists of, using a chess board, taking a set of photos from the board (while the camera is in a fixed position) and using these photos to calculate the extrinsic parameters, taking into account the length of the chess square (24mm, in our case) and the relative position between the squares.

Re-projection Error

To confirm that the parameters obtained are accurate, we use the Re-projection error. By calculating this error we were able to obtain a good estimate of how accurate the calculated coefficients are. This calculation is done by comparing “real” and “virtual” coordinates (calculating the absolute norm between them). In figure 6, we can see that our coefficients were well calculated.

Fig. 4 - *Reprojection error*

Undistortion

After the calibration is done, we proceed to remove the distortion. In figure 4, it is possible to see the difference between a distorted image (the one on the right) and without distortion (the one on the left), although it is a bit difficult to discern. The image on the left shows that the sizes of the squares are more constant than in the image on the right. This figure shows the importance of calibration.

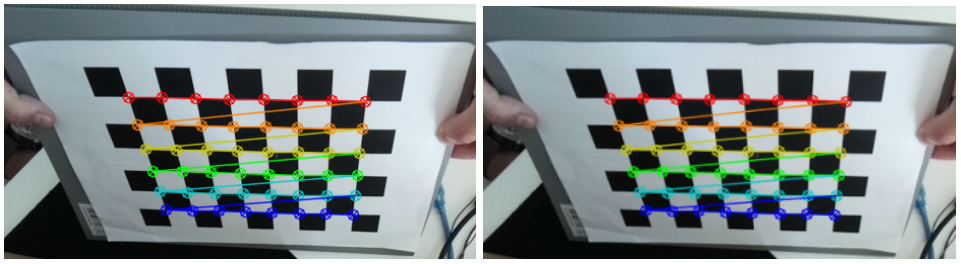


Fig. 5 - Calibration result (image on the left was undistorted and the one on the right is the original)

Feature extraction

One of the main objectives of the project is to obtain all the characteristics of a lego piece: ratio, color and the coordinates of its corners.

To obtain the color, since we are using color thresholding to detect the legos, when detecting we can immediately distinguish the color of the legos.

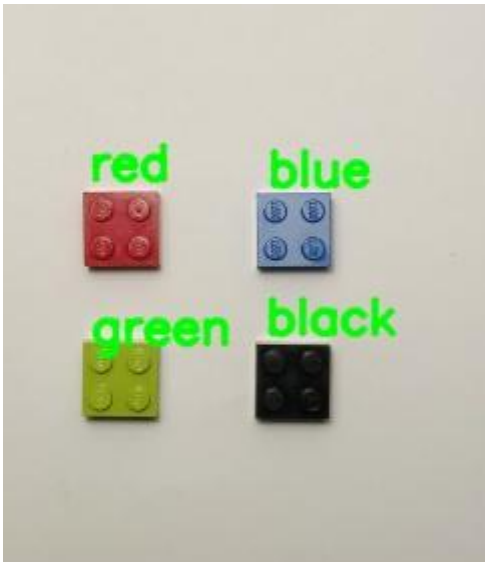


Fig. 6 - Colors of lego pieces identified

As for the coordinates of the legos of the legos corners, after detecting their location, we can use the findContours function of OpenCV to find the vertices of it.

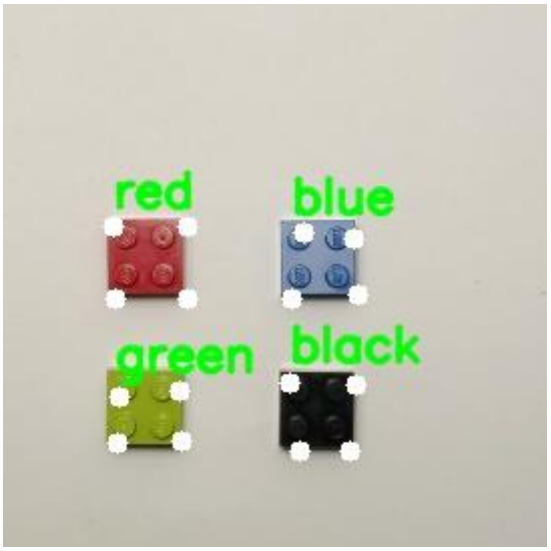
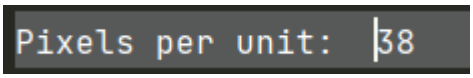


Fig. 7 - Corners of the lego pieces

Although the accuracy is not the best, it will be improved when we have a controlled lighting environment and well defined color ranges.

When it comes to the ratio, it is required to do two things:

- Find the reference value in pixels that corresponds to a unit of length, i.e., map the value of a unit of length to pixels.



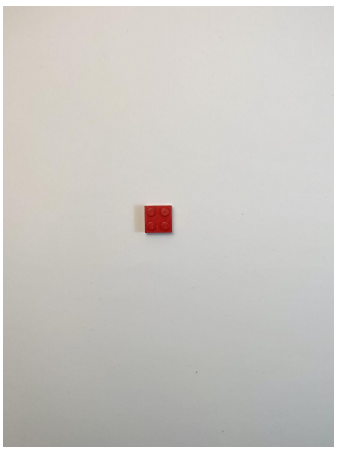


Fig. 8 - Pixels per unit obtained (left) and image used to obtain the reference value (right)

- Then, with the result above, it is possible to divide this standard value by the edges obtained after detecting the lego, in order to find out how many times these edges are greater than the reference value. The result corresponds to the value of the length of one side of the lego.

Iteration 2 and 3

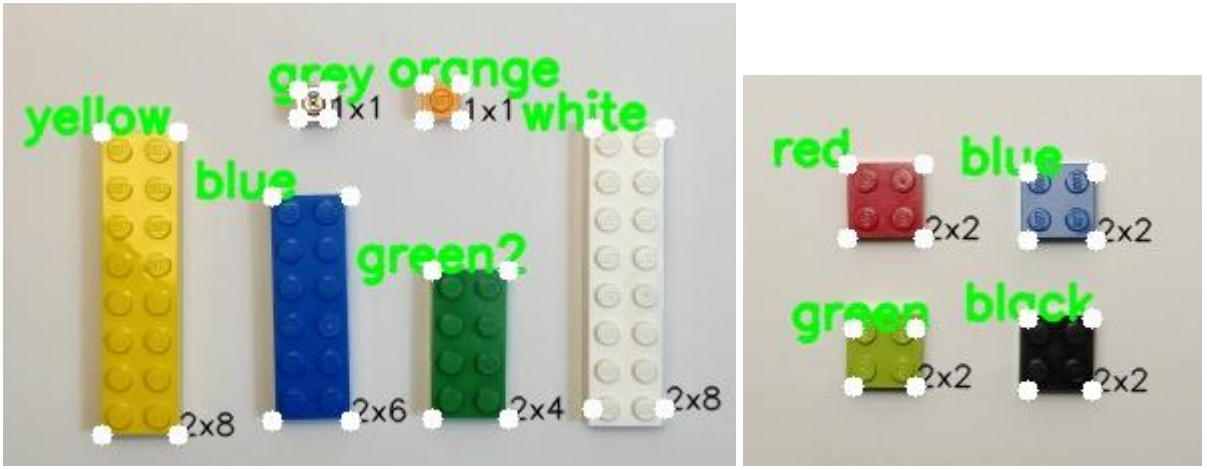


Fig. 9 - Images with ratios, corner positions and colors of the legos

Box development

In order to deal with real environments, we need to control it the best we can.

Dealing with luminosity

Since the main obstacle in the project is luminosity, our aim is to have the maximum control over it. By doing that we create a stable environment luminosity wise, where we can adjust the parameters, maximizing the performance of the algorithm.

In order to accomplish that, we used a box and 24 white LEDs. Inside the box, we lined the interior with white paper sheets and installed the LEDs on top of the box along with a hole for the Raspberry's camera. Using this setup we hope to reduce the effects of external luminosity and reflections, creating a stable and homogenous lighting environment for the system.

We used the raspberry's PWM with the intention of varying the duty-cycle and therefore adjusting the brightness of the LEDs.

Circuit Build

In order to drive these many LEDs, we used a transistor to act as a switch. We used 24 LEDs mounted in parallel each with a 390Ω current limiting resistor. With the current setup, the circuit is drawing approx. 100mA. As for the power supply we used a regular 5V USB smartphone charger.

We used a P channel MOSFET (IRF540N) with a very low $RDS_{on} = 0.044\,\Omega$ and a low $V_{th} = 2\,V$ in order to work with the raspberry's 3.3V logic. The use of the PMOS brings the possibility in the future to upgrade the circuit and draw extra current since the MOSFET allows a maximum 33A and 100V, which is more than plenty for our application.

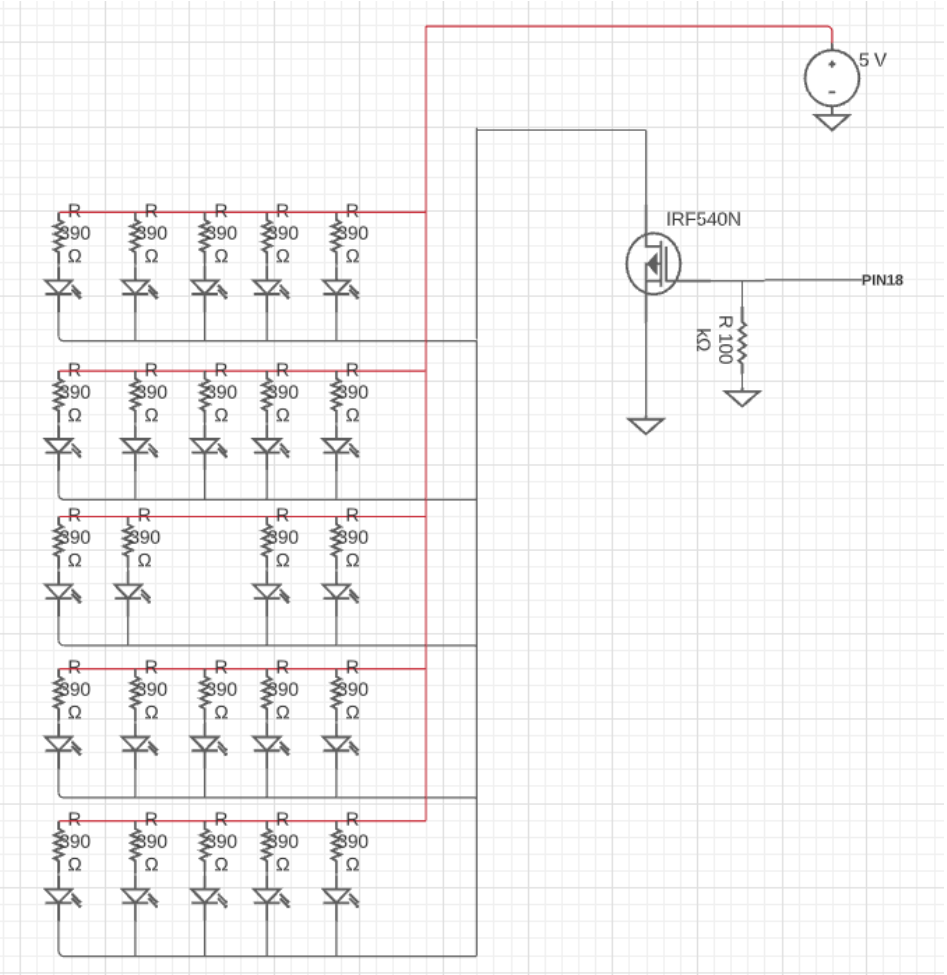


Fig. 10 - LED circuit

Conclusions

With the work done in theses iterations it its possible to conclude that:

- As for the calibration done, it is possible to conclude that it will be extremely important when we obtain the ratio of a lego or other measurements. With this it is possible to obtain lego widths and lengths with greater precision, and consequently get a more precise ratio of the legos.

- Regarding the extraction of features, we can say that the extraction of the color and the coordinates of the corners are done through methods with little margin of failure. However, obtaining the ratio is done in a not very elegant way. This depends on the distance from the camera to the legos. Implying that whenever the distance changes it is necessary to change the reference value that corresponds to a unit of length.

- Concerning the implementation of the Region Growing method we successfully developed a menu like program capable of extracting the HSV color gamuts and saving it to a JSON file.

- With the box development, we are now able to take photos of the LEGOS in a light controlled environment. For now, the current structure isn't very solid, so we expect to update the setup along the project development.