Project: Image-Goniometer

7A Team 2

Flores Esparza, María Yessenia
Biomedical Engineer
UAA
Aguascalientes, México
al212234@edu.uaa.mx

Herrera Avitia, María Fernanda
Biomedical Engineer
UAA
Aguascalientes, México
al254032@edu.uaa.mx

Laris Santos, Andrea
Biomedical Engineer
UAA
Aguascalientes, México
al2116634@edu.uaa.mx

Serna de la Torre, Verónica
Biomedical Engineer
UAA
Aguascalientes, México
al211806@edu.uaa.mx

***Goal*—. Use the classic hough transform to detect the lines in an extremity and measure the angle between them**

I. INTRODUCTION

A goniometer is a device that measures an angle or permits the rotation of an object to a definite position [1]. A medical goniometer is used for measures angles of elements joined by a joint. Today's goniometers are widely used tools, but they can be heavy, making the study more difficult. We seek to build a digital goniometer that allows us to detect the angle in images loaded in the Matlab program so that the program captures the angle formed between the limb, which in this case, will be angles formed by flexing the arm to 10 different positions.

For this, we will use a very important mathematical principle, the Hough Transform, the Hough Transform is a technique for detecting figures in digital images, with which it is possible to find all kinds of figures that can be expressed mathematically, such as lines, circles or ellipses. Our algorithm will use Hough to detect the most important lines in the image so that these serve us to calculate the angle between them.

The Hough transform is designed to detect lines, using the parametric representation of a line: The variable rho is the distance from the origin to the line along a vector perpendicular to the line and theta is the angle between the x-axis and this vector [2]. This document will be shown how can find peaks and lines in an image using Hough Transform, as well as the results obtained in the code tests.

II. METHODOLOGY

To start working 10 different pictures was captured using a web cam. The body extremity selected was an arm, so, to facility the edge detection, black tape was stuck along it.

Before applying Hough transform, the image was treated. First, it was loaded and passed to grayscale. Then a Gaussian filter was applied to eliminate noise and smoothed the image, sigma was estimated at trial an error. Just intensities less than 20 were considered because of the black tape. Next, to detect the edges the function canny was used, which return them in a binary image.

A theta vector was declared in the range of -90 to 90 degrees, as well as a rho vector, the last one from -d up to d, where d was the maximum distance between opposite corners in the image; to get d, Pythagoras theorem was used. The length of both vectors was used to give the dimensions of the voting matrix, H.

To implement Hough line transform it was necessary to fill H, an accumulator cell. Rho was calculated for each withe pixel, x_i, y_i, in the edge image using the normal representation of a line whit all the values in theta vector:

$$\varphi = x\cos\theta + y\sin\theta$$

Each time that a rounded rho calculated value was equal to a value in rho vector, a vote was accumulated in the position rho, theta of the matrix H.

Then, two peaks (lines) were located, which were the two most voted cells in the matrix H (two larger lines with different angle).

It was necessary to clear x or y from the equation to draw the two identified lines on the original image. In case of an angle of 90º, y values were calculated to avoid indeterminate. For the rest of the angles the variable x was clear.

MATLAB's documentation was very helpful to superimpose the two found lines in the original picture, since "lines" function was used, which only receives vertical arrays.

Finally, to estimate the angle between the arm, a simple algorithm was implemented, and the result was displayed and shown in the original picture.

III. RESULTS

In the following figures we can see the angle obtained in Matlab after going through different steps.

In figure 1 you can see that our original image went through a Gaussian filter and then through canny, which is an edges detector.

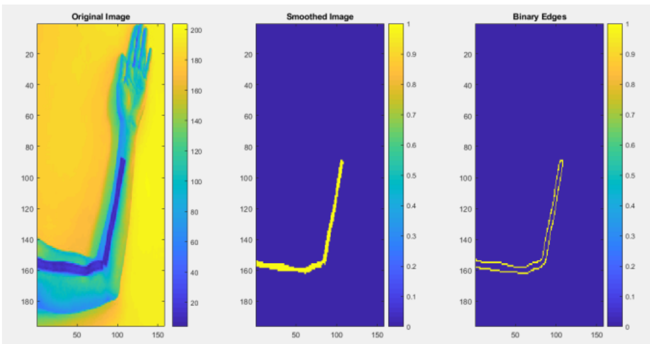

Fig. 1. Original image, smoothed image, and binary edges.

You can see the graph in figure 2, is our voting matrix, this already is the space of the Hough transform.

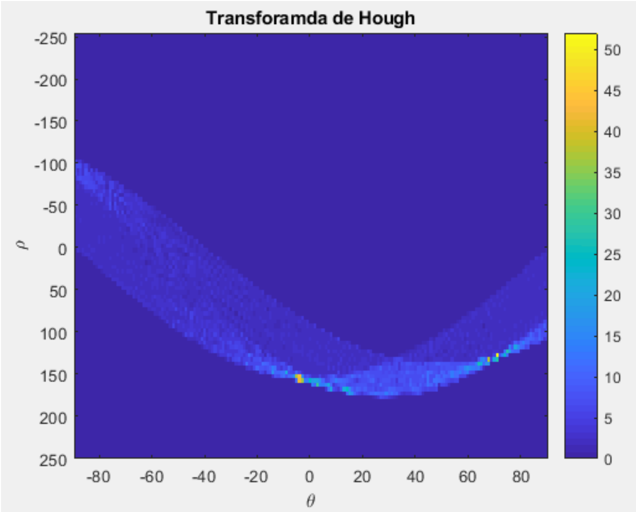

Fig. 2. Hough transform applied.

In figure 3 you can see the graph of the two most voted points obtained after making the calculation with the Hough transform.

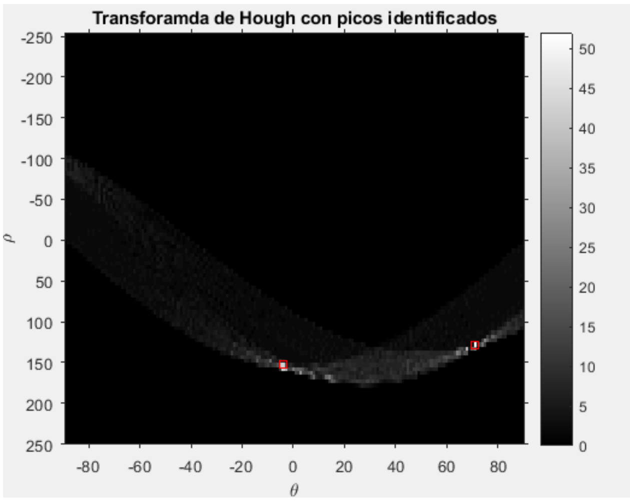

Fig. 3. Identified peaks.

Figure 4 shows our original image with the two lines identified in it, obtained whit Hough transform, and also the estimated angle between them.

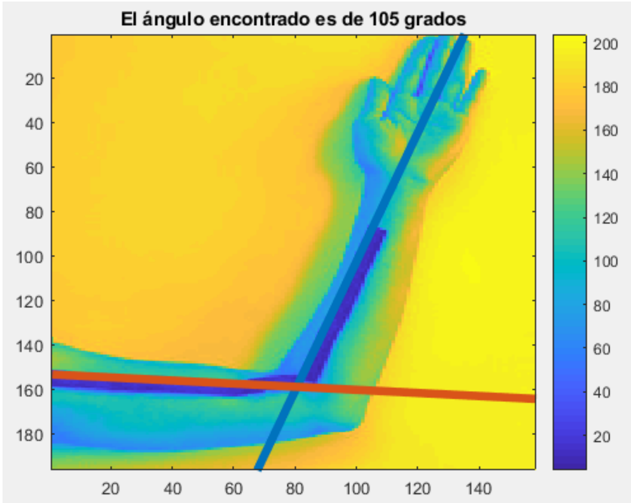

Fig. 4. Original image with identified lines and the found angle between them.

IV. CONCLUSIONS

We learned how to use all the topics that we'd seen in class, and we realized that even if we knew the basic stuffs or functions in Matlab, it helped us to create new functions as Hough transform, and we also will be able to create amazing functions implementing just our knowledge.

Of course, we had a few problems trying to solve our project like detecting our lines with the function "line" but at the end we solved this obstacle just knowing how the function line works. Another problem we had was trying not using for's, because we could do our project faster, but if we would have more knowledge about how the Matlab's functions works our code could be greater and faster.

Generally, nowadays the use of goniometers in medical terms is already used in therapeutic situations because it is used to measure the angle that each articulation opens. This helps directly to specialist to give the correct information when someone gets injured, because they check the proper way to help them to correctly open the angle of movement.

REFERENCES

[1] Goniometer - StatPearls - NCBI Bookshelf (nih.gov): https://www.ncbi.nlm.nih.gov/books/NBK558985/

[2] http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/HoughTrans_lines_09.pdf

[3] «MatWorks,» MATLAB, 2021. [En línea]. Available: https://www.mathworks.com/products/matlab.html. [Último acceso: 01 December 2021].