Benjamin Scialom
A20430263

Computer Vision
CS-512
Homework 3 : Report

I. Problem statement

This assignment tackles the issue of ellipse fitting. What does that mean ? An ellipse fitting algorithm takes a large amount of data in input and select the parameters a, b, c, d, e and f which fit the best an ellipse.
Here the goal is to be able to draw ellipse where changes occur. Therefore, before drawing any ellipse, it is important to find the change in the image and between the frames of the camera.

It was asking to highlight the movement of the hand and to draw ellipses for each finger and the palm. This depends a lot on the background of the picture and the luminosity, so, I might not show a hand.

Of course, all the variables (almost) should be modifiable in order two see less or more on the image.

II. Proposed solution

The proposed solution is a program which gets the flow of the camera and then allow us to draw ellipses around the moving part of the picture.

In order to do that, I have first implement two functions which basically compete the **difference** between two frames in **canny edge**. These two functions let us see what is moving on the flow of the camera.
Then, two other functions are used to find the contours of the moving parts (fingers and palm) and to draw ellipses around them.

The only function I had to code from scratch was the one for the fitting of the ellipse. I started to do the method of the course (see AS3/src/ellipse_fitting_1) but I encountered an issue, so, I looked for another method of fitting. Indeed, the algorithm that was presented in class is very sensitive to « singular matrix ».

This other method is the **Improved Least Square Fitting Method**. You can find the publication associated to this method in AS3/doc. This method is closed from the one of the course but a little bit different. In that, way you can find a method fully operational and efficient in AS3/src/ellipse_fitting_2.

III. Implementation details

The two code I provide are very similar and you can see differences only for the function « fit ».

1. Issues and solutions

- I encountered the error « singular matrix cannot be treated » and I didn't know how to solve that. By looking on internet, I learned that it was a major draw back of the algorithm seen in class and that an improvement exist (see publication).

-> My issues can be applied to both of my codes

- Since openCV has two functions to draw ellipse it is hard for python to make the difference. So I struggled in choosing the right one. To solve this issue I asker the professor.

- One of my concerns was how to get the contours and the picture and being able to gather each group of points to draw the corresponding ellipse. I noticed afterward that the output of the function « findcontours » of open cv was group of points. I understood how find contours work by printing the result.

- My computer was lagging because of the complexity of the algorithm and the process in real time. To make it faster without using python, I just rescaled my camera. This is done thanks to the function « rescale_frame ».

- I had also an issue with the implementation of the difference for two reasons. First, I wasn't comparing the right flows of the camera. I mean it was not well actualized. Moreover, it was super difficult to see if it was working since if it change from the frame 1 to the frame 2, it will also change from the frame 2 to 3. In order two, to be sure of the implementation, I created a variable that you let you chose the separation time between the frames.

2. Instructions to run the program

Which code should your run :

First, you should run the file named « ellipse_fitting_1 » which correspond to the implementation of the method of ellipse fitting seen in class.
Then, you can run the file name « ellipse_fitting_2 »which correspond to the new method which resolve the « singular matrix error ».

How to use the algorithm :

Whether it be the file 1 or 2, the key to press are the same. The only difference is the outcome for the ellipse.

- If you press « g », you will obtain the canny edge detection of the current flux of the video.
- If you press « d », you will have the differences between each 1/24 frames. So if you put the value of the delay (track-bar) at 24 you will have one image per second.
- If you press « b », you will draw the ellipse corresponding to the contours of the moving parts observed.

How to use the track-bars ?

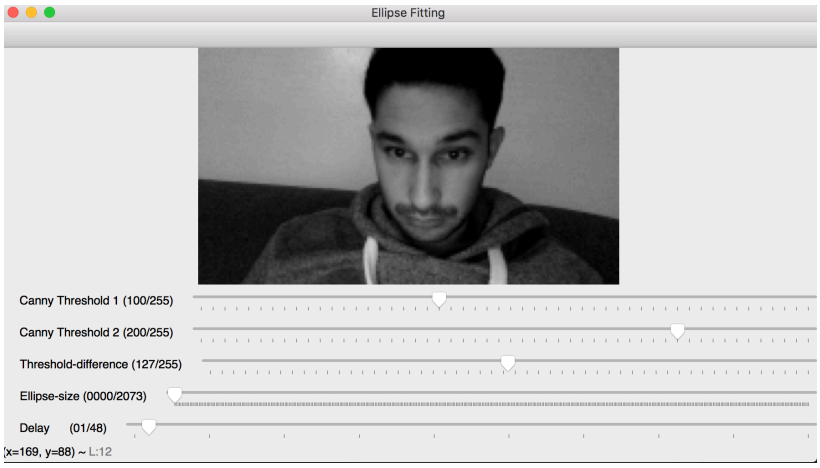
- The two first track-bars are call canny-edge threshold 1 and canny-edge threshold 2, It's only two parameters of the function to control the threshold in each dimension of the image.
- The track-bar named threshold-difference let you set the threshold of the difference between two image. Basically, if you wan to see only big movement it should be high but be careful to not putting it to high. A reasonable value is 5.
- The track-bar named « ellipse size » allows the user to select the minimum size of the ellipse he wants to see. That is to say, if he want to see only big object, the user would have to put it at a high value. Otherwise, the user will see every single object.

3. Possible improvements

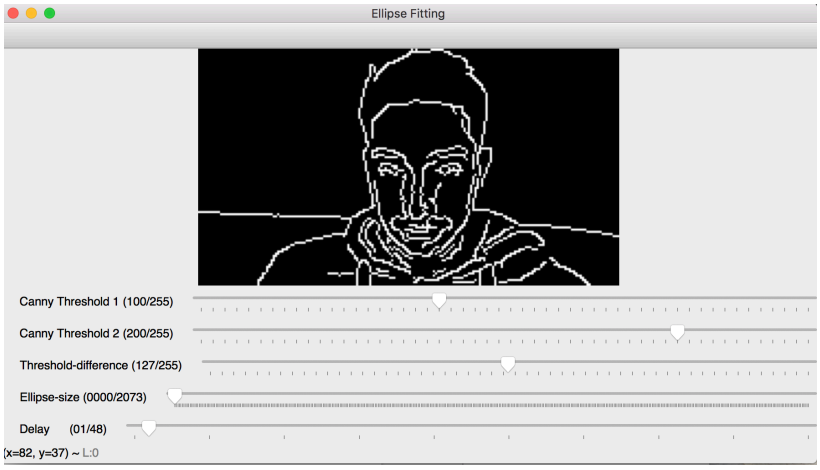
I implemented one improvement and I don't see any other one except make it faster. Maybe I should have cythonize the code.

IV. Results and discussion

First of all, the algorithm can display the current flux of the camera :



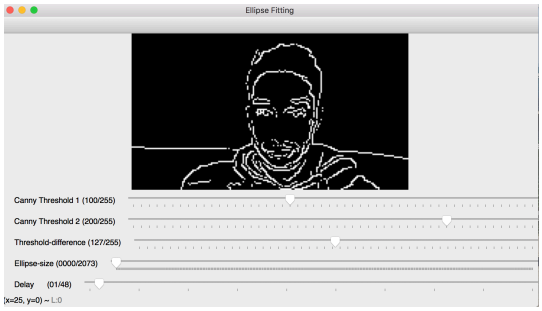
Then, we can apply the canny edge detection on the current flux of the camera : (press « g »)

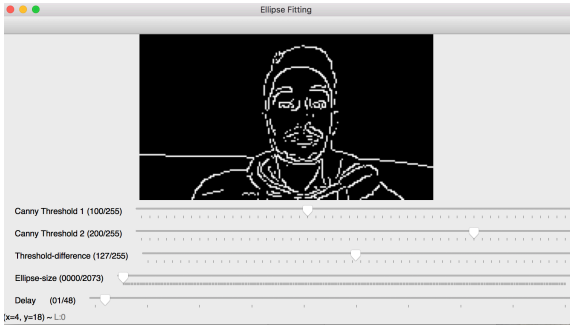


Another tool implemented is the difference between two frames : (press « d »)

On this picture, you can see that my hand has moved a little bit as white color on the picture proved it.

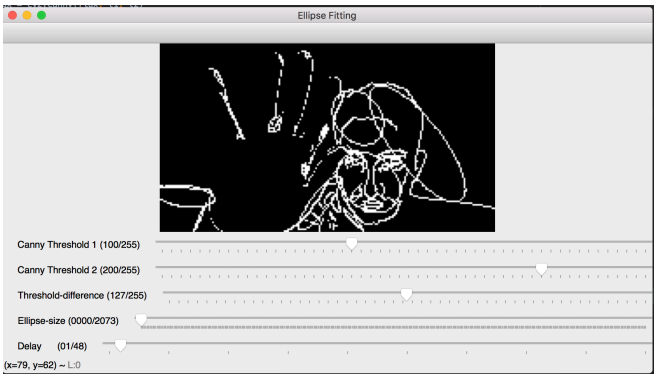
We can mix difference and canny edge but the result is less visible.





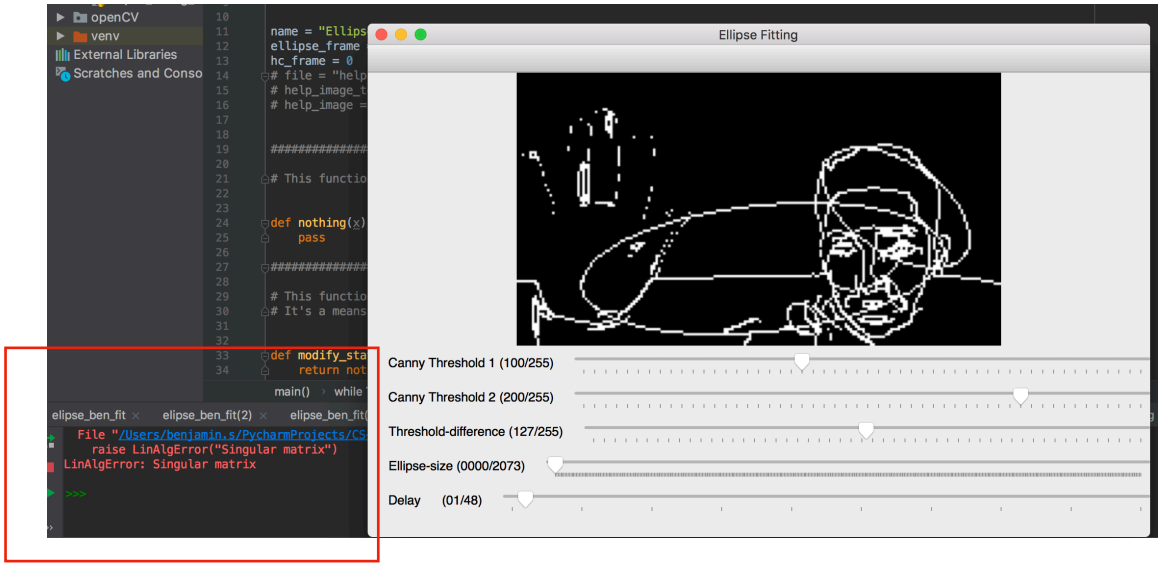
I try to move during the two successive picture. We can see that lines are drawn around my nose because I moved. It is harder to see the difference when we are using at the same time canny edge since too few pixels are used and it only curves, lines … .

Finally, we can draw ellipses thanks to the implementation:



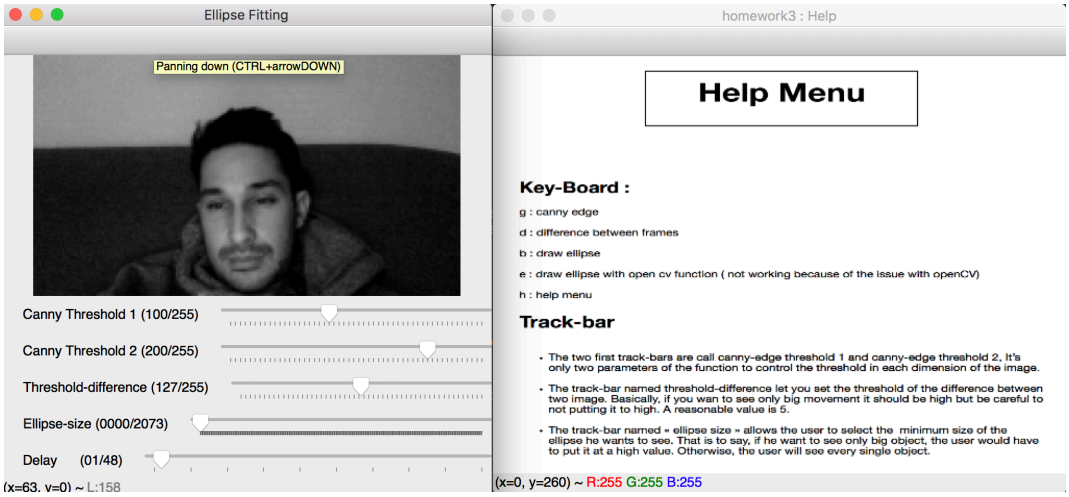
We can see some ellipse. I am sorry there are no clear ellipse on my end but it depends a lot on the slope of the camera and the light but if you try you will see that it works.

I want to show you the result with the algorithm seen in class which crash when he has to compute a singular matrix.



Even if we have the error, we can see that ellipses are drawn. It's just a particular case in which it doesn't work.

If you want to know anything about he key board or the track bar you can press « h » .



V. References

- https://stackoverflow.com/questions/18595099/python-opencv-how-i-can-use-cv2-ellipse
- https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=minarearect#minarearect
- The publication I put in the the doc folder : NUMERICALLY STABLE DIRECT LEAST SQUARES FITTING OF ELLIPSES by Radim Halír and Jan Flusser