Hand tracking with OpenCV and C++

Myllena de Almeida Prado
myllena.pradoa@gmail.com

1 Introduction

The interaction with technology is present in the day by day of many people. One type of interaction happen throw the hand. In this way, identify key points hand is important to link the hand movement with commands in software programs, which improve the use of technology. On this scenery, this project of hand tracking was developed to understand the tracking hand better. The implementation of project was done with C++ and OpenCV[1]. The code of project is available on the GitHub repository: https://github.com/MyllenaAPrado/Hand-Tracking. The tracking hand implemented is not robust, but the program can identify hand's point. The requirement of program is a simple RGB camera, a clean background of image and small changes in the image without including the hand movement.

2 Hand model

The hand model is shaped by one point in each fingertips plus one point in the center of hand. This model was chosen due the limitation of point identify using only image processing. The tracking hand identify the contour of hand and the points which form the contour. From this points is choose the points of model hand. The hand model assume that the hand of user will be always up, one example of this is shown in the figure 1.

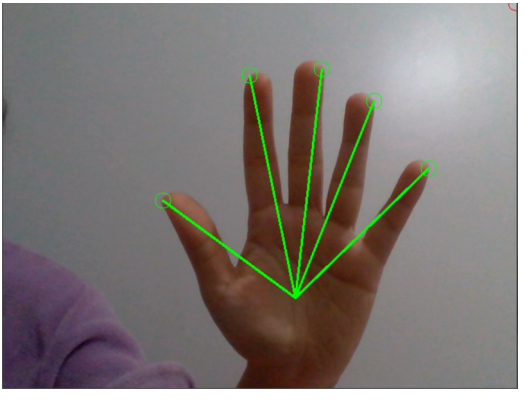

Figure 1: Hand model

3 Techniques to extract hand information

The implementation of hand tracking is divide in two parts: hand segmentation and identify key points. In the first part, the change of image pixels and the contour of hand are extracted. The output of this process is a image with a good hand segmentation of user. In the second part, The points of hand contour are process and the result is the final hand model established.

3.1 Segmentation

The hand segmentation has the goal of detach the format of hand from the rest of image. The process of segmentation is divided in four principal steps:

- Set a reference frame to be the background of image
- Identify the movement of hand based on reference image
- Improve the image of hand segmentation
- Identify the contour of hand

In the begin of program the first frame is capture and set to be a reference frame of background image.This reference image can be change any time enter *1*. In this initial it is important the hand not be present in the image for the tracking hand works well. A Gaussian filter is applied on this reference image and the color scale of frame pass to gray scale. This operation is done to reduce the noise on image and to improve the comparison between images in the future.

Figure 2: Frame of reference and example of position to tracking the hand.

The next part of segmentation capture the current frame to identify the hand. Thus, after the begin of program the hand can appear on the image. The current frame pass for the same process of reference frame with the Gaussian filter and gray scale. After the processing of both image, reference and current frame, they are compared with function *absdiff*. This function makes the basic principle of program to identify the hand that is consider the hand form by the pixels that changes in the comparison of images. After the detection of pixels variation a new process is done. At this time, a limit is established to filter the image from small noise with the function *threshold*. The output of function is dilate to enhance the form of generate image and to have a better contour.

Figure 3: Process of dilate in the image.

The follow step is identify the contour of hand. First, all the contours present in the image is captured, some of them are caused by interference of change in light or noise. The biggest contour is chosen based on the expect of the principal object moving in image is the hand. This contour will be used in the next part to identify points on hand.

3.2 Identify fingers points

The points of fingers are identified with two basics functions:*convehull* e *convexdefects* what are apply using the contour found before. The first function identify the smallest convex form which contains the hand contour. The result of *convexhull* can be seen in the left figure 4. In this image the function forms the white line around the hand.

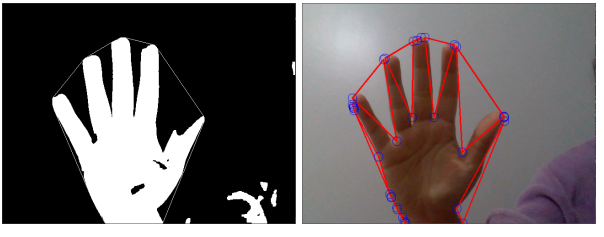

Figure 4: Convex hull and convex defects with points identified.

The second function is responsible to identify convex areas between the contour and the line generated by the function *convexhull* These convex areas are triangles around the hand how is shown in the right figure 4. The points which form this triangles will be used to determine the hand model. In the image have some points localized between the fingers which are used in the project handy [7]. Although in this project this points are not used.

The points obtained from function *convexdefects* are filtered in follow three steps for do the hand model:

- Established the center of hand
- Calculate the angles between hand to get which are smaller than 90 °
- Choose the points above a value of y

The center of hand is established using the function *boundingRect* which draw a rectangle around the contour of hand. The center of this rectangle plus a value of 90 in the y axis determines the center of hand. The axis x in the image represent the width and the axis y

represent the height. The center of hand can vary because the center of rectangle is variable. Therefore sometimes in the hand model the center can be in the middle of hand or more close to wrist. The next step is establish which point are probable to belong to the hand model according to the angle of triangles of *convexdefects*. These angles are calculated with two auxiliary triangles shown in figure 5.

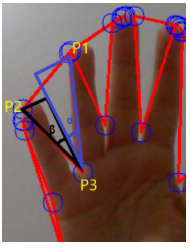

Figure 5: Angle calculated.

The equations 1, 2 e 3 determine this angle.

$$\beta = \arctan\left(\frac{P3_x - P1_x}{P3_y - P1_y}\right) \tag{1}$$

$$\alpha = \arctan\left(\frac{P2_x - P3_x}{P2_y - P3_y}\right) \tag{2}$$

$$\theta = \alpha + \beta \tag{3}$$

All the points P1 of the triangles which have angle smaller than 90° is probable to belong to the hand model. After that, the points are filtered again and the points P1 which have y axis value above of 50 plus rectangle center are chosen. And the last requirement is the point P2 of triangle has to be above of point P3 of the same triangle in y axis. Finally, the tracking process finish and one example of output is shown on figure 6.

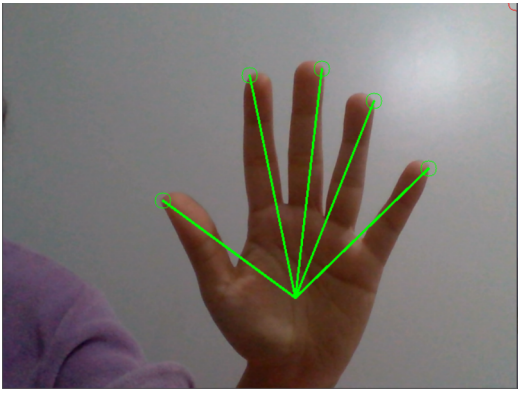

Figure 6: Hand tracked.

The hand tracking process is not robust so some times the tracking can not identify fingers even that present on image that is seen in figure 7. However, the tracking identifies all the points proposed for the hand model.

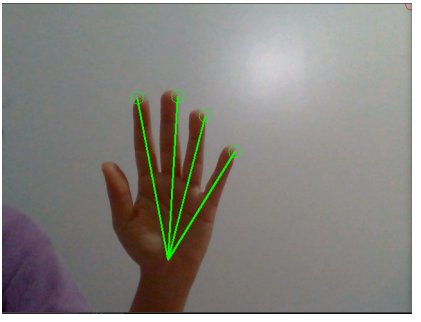

Figure 7: Hand tracked.

4 Potential improvements

The hand tracking identifies all the proposed points of hand model, but sometimes the tracking has a problem even with a clean background and all fingers present on image. Furthermore, the use of tracking for other applications is limited due to the need to clean background and little movement of objects beyond hand. Therefore, the tracking can be improved in the parts of segmentation and identification points. One option for this is use neural networks, which identify hands more precise and have less problem with movement around hand.

References

[1] *OpenCV Open Source Computer Vision 4.5.1*. URL https://docs.opencv.org/3.4.4/index.html.

[2] Pierfrancesco Soffritti. Handy, hand detection with opencv, 2018. URL https://pierfrancesco-soffritti.medium.com/handy-hands-detection-with-opencv-ac6e9fb3cec1.