Universidad Carlos III de Madrid

Systems of Perception

# GECKO

# Gesture Recognition

*Authors:*
David Estévez Fernández, 100282441
Irene Sanz Nieto, 100282826

*Teacher:*
ABDULLA HUSSEIN ABDULRAHM AL KAFF

December 12, 2013

# Contents

# 1 Introduction

In this document a gesture recognition software named GECKO is presented. This software will allow the user to interact with a computer using just hand gestures.

## 1.1 Motivation

The idea of implementing a gesture recognition software appeared ...

## 1.2 First Steps

The first approach was to make a color segmentation (using the HSV color space) and select the hand using the size of the contour. Once the hand was detected, it was possible (after pressing a key) to move the mouse with the hand. The control was not very accurate due to the different problems present in the system such as illumination changes or having parts of the palm and fingers with a color outside the selected skin color range. In this first approach, it was possible to select between the theoretical values and the custom values obtained by placing the hand inside a square and getting automatically the skin HSV range.

The theoretical skin range worked better than the custom range and as it was previously mentioned, the poor segmentation caused a fluctuant output image, that did not allow to obtain a good gesture recognition.

In order to better the hand's position and orientation estimation, two kalman filters were implemented. Those filters improved these features and allowed a moderatedly accurate mouse control, if the background did not interfere with the hand segmentation (i.e. the background had to be a plain color).

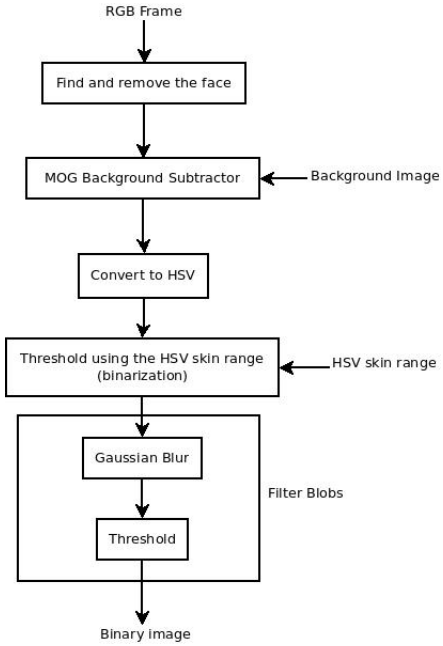
As it can be seen, the main problem this software had was the hand segmentation. Some research was made and different papers were found that allowed us to implement a better hand segmentation.

# 2 Software Explanation

The final version here presented is a software that can be divided in the following parts: Hand Segmentation, Hand Description, Gesture Recognition.

## 2.1 Hand Segmentation

The segmentation of the hand was one of the most important parts of the software since in it depends the correct functioning of the next parts. The flowchart of the segmentation of the code is as follows:



As it can be seen in the diagram, the first thing done is to detect the face using the "Face Detection using Haar Cascades" already implemented in OpenCV. After this detection, a binary mask is made in which appear a black square over the face to hide it.

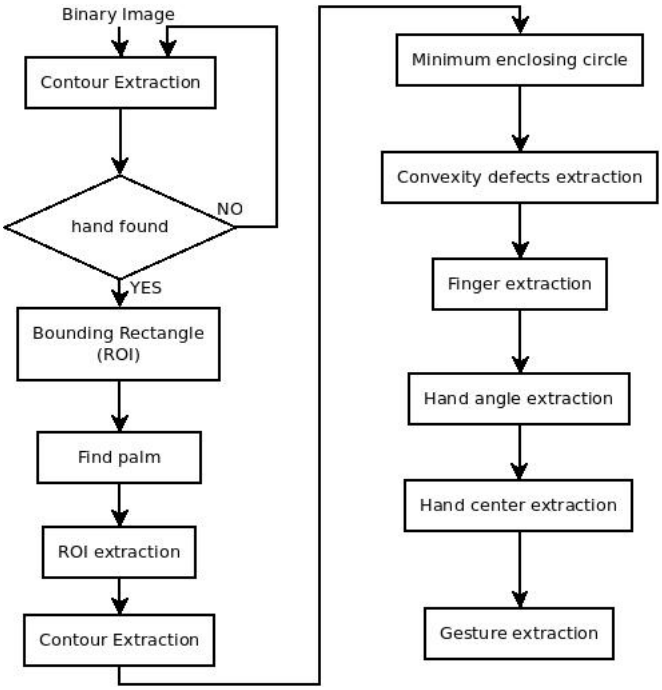
Next, a Mixture of Gaussians background subtractor is used to remove the background of the image and detect more easily the hand. From this step, another binary mask is created.

Afterwards, the resulting image with the latter masks being applied is converted to the HSV color space and is thresholded using the appropriate HSV skin range.

Finally, in order to better the output binary image, a gaussian blur and a thresholding is applied to eliminate blobs and unite different regions of the hand. It was chosen this configuration because it was much faster and gave better results than making an opening to the image.

## 2.2 Hand Description

The input of this section is the binary image that was the output of the previous one. This part of the software characterizes and extracts the hand's parameters. The flowchart is as follows:




As it can be seen in the diagram, the first step is to extract the contours of the hand and decide from its size if it is a hand or only a blob remaining from the hand segmentation.

If the size is appropriate, a bounding rectangle is found. Afterwards, the palm is extracted. In order to do that, a raw center of the palm and the radius of the inscribed circle is obtained from the points of the contour.

Then, the ROI (Region Of Interest), i.e. the rectangle that contains the maximum circle that represents the hand boundary.

Afterwards, another contour extraction inside the extracted ROI is made in order to eliminate once more the smaller contours that may appear.

Using the largest contour found in this last step, the minimum enclosing circle is obtained. This circle will be used as a hand's descriptor.

The following step is to compute the convexity defects, that will be used when extracting the number of fingers present in the image. In order to assure that a certain part of the image is a finger, three conditions must be fulfilled:

- The depth of the convexity defect must be in between the minimum inscribed radius and the maximum inscribed radius.
- The angle between the possible fingertips must be less than $90^{o}$.
- The fingertips must have a k-curvature inside a certain range.

### 2.3 Gesture Recognition

## 3 User Guide

This code was developed using the following libraries:

- OpenCV (v2.4.6.1).
- X11 (linux native libraries that allows the control of the windows and mouse).

The software was compiled using CMAKE (minimum version 2.8).

Therefore, those libraries are needed for the correct functioning of the software.

### 3.1 Compile the software [UBUNTU]

In order to compile the code, there are various options. Here it will be explained how to compile using qt and terminal.

To open the software as a Qt project, the only thing needed is to open the main CMakeLists.txt (gecko/CMakeLists.txt) with QtCreator. This will parse the whole project. Afterwards, press the "build" icon to build the project.

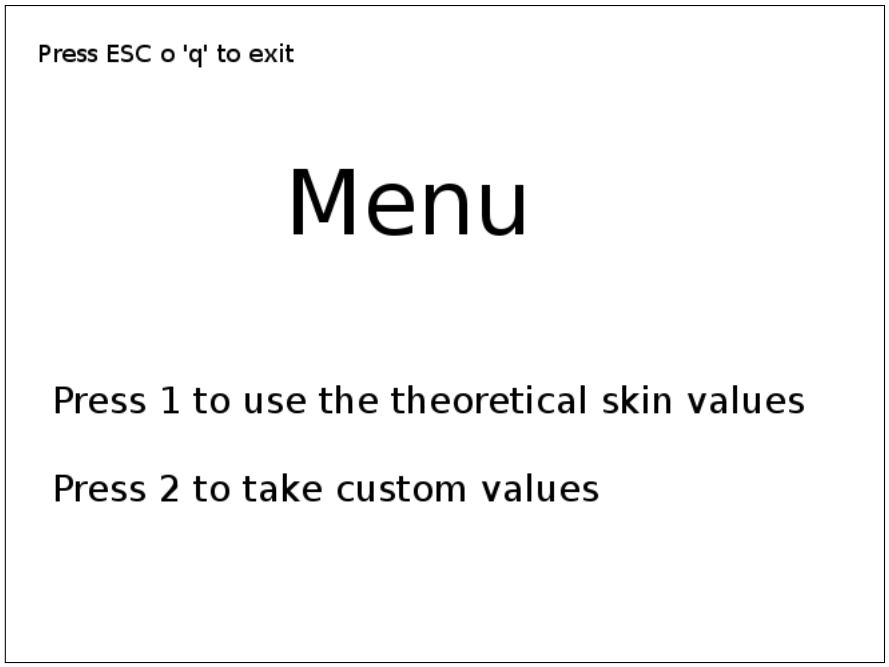
The folder structure used is the typical of a CMake project. In order to compile using the terminal follow this steps:

1. Open a terminal in the build directory (gecko/build).
2. Enter the command "cmake ..".
3. Enter the command "make".

### 3.2 Use the software

In order to execute the software, press the convinient button of the QtCreator IDE or open a terminal in the bin directory (gecko/bin) and enter the command "./gecko.cpp".

Once the code is compiled and started, a first window will appear with the information about the software. After pressing enter to continue, this menu will pop up:




The first option will take the theoretical HSV skin values proposed in **[CITE]**. Afterwards, the application will allow us to adjust those HSV ranges to adecuate them to the room's conditions. Once all the adjustments are done, pressing ENTER will start the program.

The second option will pop up a window with a green square in the middle. Place the hand so that the square is in the middle of your palm and press ENTER. The program will automatically calculate the HSV range of your skin. Afterwards you will be able once more to manually adjust those ranges. Pressing ENTER one last time will start the program.

The main program will track the hand and trace two circles, the first one around the palm and the second one surrounding the whole hand. The center of the palm is also shown, as well as the angle described by the hand.

There are different functionalities implemented in this software.

- Open hand(i.e. the program detects the five fingers): the mouse moves with the hand's movement.
- Closed hand: click.
- "Victory sign": the gedit text editor is opened.
- "Gun or L sign": a terminal is opened.

The two last functionalities may be changed using a configuration file named apps.config (gecko/data/apps.config). That configuration file contains the following:

```
$gedit$ 3
$gnome-terminal$ 4
```

The three refers to the victory sign, and the 4 to the gun or L sign. To change the functionality the only thing needed is to change the programs written between the two dollar signs with the wanted one.

# 4 Software Documentation

**[PASTE DOXYGEN OUTPUT]**