

# Cursor Control Using Head and Hand Movement

Akshai P Sajeev, Arshin Salim, Unnikrishnan. R, Aby Abahai T

Computer Science, M. G. University, Kerala, India

## ABSTRACT

Head tracking systems can be used for controlling a mouse cursor on a computer system. Open-source projects which use such an approach require that the user holds his/her head still during and after a calibration procedure, as even the slightest head movements could affect the system function in an undesirable way. An Image processing system for mouse cursor control was developed, which uses a front camera to compensate head movements during eye tracking. An alternative is to include the usage of various sensors and hardware devices are very costly and not user friendly. So a new system was introduced in which the camera is used to capture the images of the users and compare it with a trained set of data and using the head movement a corresponding motion of the mouse is defined. This approach is not only user friendly but also cost effective. The software part of the system is integrated into the openCV platform as a plugin. The openCV platform is a project under development, with the goal to provide a new definition to computer vision. A machine learning technique called cascade classifier is used in the training. Though the system is easy and low cost, a large training is required and initial usage may find some difficulty.

**Keywords:** Face Detection, OpenCV, Mouse Control, Image Processing

## I. INTRODUCTION

Nowadays personal computer systems are playing a huge part in human's everyday lives as they are used in areas such as work, education and entertainment. What all these applications have in common is that the use of personal computers is mostly based on the input method via mouse and keyboard. Things will be very smooth if the input operations can be performed without the aid of any input devices and also situation will be very easy. While this is not a problem for a healthy individual, this may be an insurmountable barrier for people with limited freedom of movement of their limbs. In these cases it would be preferable to use input methods which are based on motor abilities of the head.

Today's all of the major laptops have an inbuilt webcam, not only for the laptops but also a webcam can be bought for a desktop with lesser cost. These devices capture users head and hand movements timely. Computer system is preoccupied with trained data set of head and hand images, so that each time the webcam captures the images of head and hand when it appears in front of the camera then it recognizes the image. Now the

system has the capability to identify whether it is a head or hand. After that the system maps the mouse cursor towards the x & y coordinate in which the head appears. These set of operations can be performed through the facilities of image processing. Image processing is nothing but it is an analysis and manipulation of a digitalized image, especially in order to improve its quality.

## II. METHODS AND MATERIAL

### A. Proposed System

The system proposed here must consist a personal computer connected with a quality web camera. Today's all of the major laptops have an inbuilt webcam, not only for the laptops but also a webcam can be bought for a desktop with lesser cost. These devices capture users head and hand movements timely. The system is almost entirely dependent on face detection, and hence the primary step towards the development involves the creation of a system to detect human face and its basic features. Face

detection is done using Cascade Classifier, which requires a very large set of samples for training. About 7000 images each of the face, eyes, and mouth were collected from various sources. An equal amount of negative images were also collected. While the positive image shows an object that must be detected, the negative image shows a similar picture with the object missing from it. The target objects thus extracted from the positive samples will be stored as a text file. The first field indicates the name of the image. The second field indicates the number of target objects detected, the third and fourth fields give the x and y coordinates, respectively of the top left corner of the rectangle denoting target object's location. A distinguishing attribute of head trackers are the Degree Of Freedom (DOF) which they are able to detect. The 6 relevant DOF can be assigned into two groups as follows:

- Traditional Movements
- Rotational Movements

The traditional movements include the moving up/down, moving left/right and moving forward/backwards whereas the rotational movements are defined as pitch, yaw and roll. Pitch is the movement defined as the tilting action forward and backward. Yaw is defined as the turning forward and backward. Roll is defined as the tilting action done side by side. One approach for head tracking is the visual face detection with a web camera. An image from a person's head is captured by the camera which is then processed to find distinguishing marks like eyes, ears, nose, mouth, chin, etc. Depending on how these marks are further processed up to 6 DOF can be achieved.

## **B. Implementation**

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover

methods. The system is designed as an independent unit, with a camera, a processor, and the trained data set. The camera should support high definition capture for optimum performance. Since the system has to be flexible enough to use any time of the day, lack of light must never be an issue. Therefore, a nonintrusive light, like the IR can be provided. The video thus captured, is given as the input to the processor. The processor must be capable of image processing, with at least 1 GB of memory, 300MHz speed and 2 GB RAM.

### **A. Sample Collection**

The system is almost entirely dependent on face detection, and hence the primary step towards the development involves the creation of a system to detect human face and its basic features. Face detection is done using Cascade Classifier, which requires a very large set of samples for training. Almost 7000 images each of the face, eyes, and mouth were collected from various sources. An equal amount of negative images were also collected. While the positive image shows an object that must be detected, the negative image shows a similar picture with the object missing from it.

### **B. OpenCV**

OpenCV (Open Source Computer Vision) is a cross-platform library of programming functions mainly aimed at real-time computer vision. It is free to use, and provides an array of functions for machine learning, neural networks, and other applications of artificial intelligence. To support machine learning, OpenCV provides a statistical machine learning library that contains boosting, decision tree learning, Artificial Neural Networks (ANN), Support Vector Machine (SVM), ensemble learning, etc. It is written in C++ and the primary interface is in C++ as well. There are bindings in Java, Python, and MATLAB/OCTAVE. To encourage adoption by a wider audience, wrappers in other languages such as C#, Ch, Perl, and Ruby have been developed. All of the new developments and algorithms in OpenCV are now developed in the C++ interface.

### **C. Cascade Classifier**

Here, `opencv_haartraining` has been used to train a cascade classifier. This process may take a number of days depending on the number of samples and stages of the cascade classifier. Once the application has finished execution, an XML file will be generated in the

specified folder. The XML file is the final trained cascade classifier. The OpenCV library provides a great number of functions for face detection, feature detection such as eyes, mouth, sunglasses, etc. Furthermore, it provides programs (or functions) that are used to train classifiers for their face detection system, called HaarTraining. We can create our own object classifiers using these functions. Object Detection using Haar feature-based cascade classifiers is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. The algorithm extracts images using a lot of positive (faces, eyes, etc.) and negative images (arbitrary images without faces). Each feature is a single value obtained by subtracting sum of pixels under various regions of the images. Features include edge features, line features, etc. For each feature, the pixels used for extraction differs. Now all possible sizes and locations of each image kernel are used to calculate plenty of features. But all the features extracted will not be useful for the purpose at hand. For example, in this paper only eyes, mouth, and the face are relevant. Other facial features like cheeks, nose, forehead, etc. are useless as far this paper is concerned. To extract only those features relevant to the project, we use a technique called Adaboost. For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Then we select the features that display the least error rate (best threshold). Initially each feature is given an equal weight. As the process continues, the weights are updated according to the results obtained so that accuracy is improved. The process is repeated until the required level of accuracy or the required number of features is found. Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. With around 6000 images, even 200 features can provide 95% accuracy.

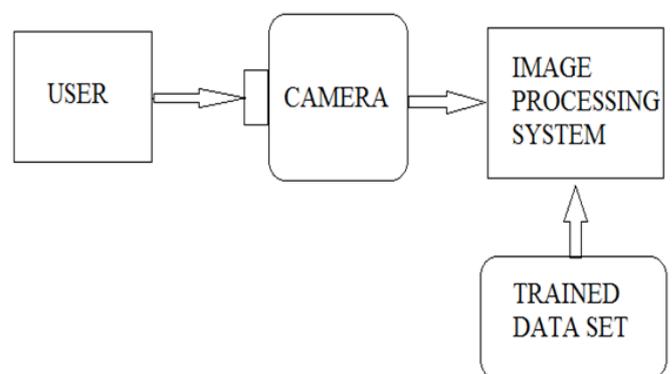
In an image, most of the image region is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot. Don't process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region.

For this they introduced the concept of Cascade of Classifiers. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region. This is how face detection and other feature detection is done using haartraining by means of cascade classifiers.

### III. RESULTS AND DISCUSSION

#### Implementation Procedure

After the training we will obtain an xml file as the trained cascade classifier. As OpenCv is a cross-platform library of programming functions, we should implement this functions in a suitable high level programming language. We chose python for implementation as it has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java. If you need a critical piece of code to run very fast, you can achieve this by writing that piece of code in C, and then combine that with your Python program. By importing cv2 class in python one can easily implement the method defined by OpenCV.



**Figure 1.** A simple system diagram

The next issue is to link the trained cascade classifier to the program. `cv2.CascadeClassifier(cascPath)` a build in function in the class cv2 easily link the trained xml file to the program where `cascPath` is the path where the

trained xml file is stored. The *root.warp\_pointer(x,y)* and *display.sync()* functions help to move the mouse pointer. The logic of implementing single click, double click and right click are given below

- Left-clicking is used for many common computer tasks, such as selecting objects, opening hyperlinks, and closing windows. Clicking and holding the left mouse button can be used to select text or perform drag and drop operations. In video games, left-clicking is typically used to perform the primary action, such as moving a character or firing a weapon. Left click can be implemented by creating a virtual timer and after the movement of mouse become idle for some time say 10s a left single click can be implemented
- The right mouse button is often used to open contextual menus, which are pop-up menus that change depending where you click. For example, if you right click on the desktop, you may see a menu pop up that includes "Change View Options" and "Change Desktop Background." If you right click on a folder, the menu might include options such as "Open" and "Properties." Certain programs, such as video games, may use the right click to perform other functions, such as firing a secondary weapon in a first-person shooter. Most programs, however, use the right click to open contextual menus. The right click can be implemented by the use of hand gesture. Whenever a hand is shown in front of camera we can implement the right click there.
- Double clicking involves clicking your mouse button quickly two times. To perform a double click, and not just two clicks, the mouse button must be pressed twice within a very short time, typically about half a second. Most operating systems allow you to lengthen or shorten the maximum time allowed for a double click, using the Mouse Control Panel or System Preference.

A double click is recognized by your computer as a specific command, just like pressing a key on your keyboard. Double clicking is used to to perform a variety of actions, such as opening a program, opening a folder, or selecting a word of text. In order to double click an object, just move the cursor over the item and press the left mouse button quickly two times. This is implemented by the motion of hand towards the camera.

The clicks (right, left and double) can be implemented by the *fake\_input()* method in python

#### IV. CONCLUSION

The day to day work of human being is mainly depending with the computer system especially through the so called laptop and desktops. The inputs can be given to the computer system via mouse and keyboard. Things will be very smooth if the input operations can be performed without the aid of any input devices and such situation will be very user-friendly. Such situation can be achieved through the use of input methods which are based on motor abilities of the head region. In this paper the mouse cursor and clicking control through head and hand gestures. This is done through the images sequentially taken by the webcam attached to the computer system. Further improvements on these topic will reduce the obstacles in the human computer interaction scenario and also increases the degree of freedom. A person can easily interfere with the computer without any predefined knowledge on the input devices such as mouse, digital pen etc. The application will be very much useful for the gaming, graphics design, etc. These will be an effective low cost approach for the disabled personalities.

#### V. REFERENCES

- [1] Ivan Culjak "A brief introduction to OpenCV" MIPRO, 2012 Proceedings of the 35th International Convention.
- [2] Anupam Agrawal, Rohit Raj and Shubha Porwal "Vision-based multimodal human-computer interaction using hand and head gestures" Information & Communication Technologies (ICT), 2013 IEEE Conference.