IJIRR

International Journal of Information Research and Review
Vol. 04, Issue, 04, pp.3933-3936, April, 2017

Research Article

IMPLEMENTATION OF HANDS-FREE GESTURE AND VOICE CONTROL FOR SYSTEM INTERFACING

*1TanvishThakker, 1Vaibhav Chopda, 1Shrenik Ashar and 2Nasim Shah

1BE Student, Department of Information Technology,K.J. Somaiya Institute of Engineering and Information Technology, University of Mumbai, Maharashtra,India
2Assistant Professor, Department of Information Technology, K.J Somaiya Institute of Engineering and Information Technology, University of Mumbai, Maharashtra, India

ARTICLE INFO

Article History:

Received 13th January, 2017
Received in revised form
20th February, 2017
Accepted 05th March, 2017
Published online 30th April, 2017

Keywords:

Gesture, Voice, OpenCV,
NET, Recognition.

ABSTRACT

The project consists of two main components i.e. Gesture and Voice. Gesture Recognition is where we used two algorithms. Those are Lucas-Kanade and Haar Classifier. The second main component, which is the Voice component, consists of two sub-components, which are the voice recognizer and the voice synthesizer. The entire architectural idea is based on the basic floor concepts of OpenCV (Open Computer Vision). The architecture and component distribution looks very easy and systematic but gets complicated as the implementation proceeds. The belowdetailed description should make it easy for all us to understand this not so easy system.

INTRODUCTION

The currently deployed system is an efficient system with many advantages that overcome the problems the previously existing systems prevailed. We will look into these advantages a bit later but it is very important to understand what functions each component does in the entire system. The Haar Classifier will be used for face detection by the camera and Lucas-Kanade will be used to extract the features of the face so as to create a unique face every time as all faces are not the same. Similarly, for the voice recognition components, the voice recognizer will work efficiently with the microphone so as to detect the voice input of the system which will also remind it to save that input so that the synthesizer can process it. The voice synthesizer will sample the user's voice and break it into individual phrases so as to easily detect what the user wants to convey

Requirement Analysis

The system that we have developed is software oriented and has minimal hardware requirements. One can use these basic components and run this efficient system at a very minimal cost. Some of these requirements are:

Software Requirements

- Windows 7 or any equivalent
- Microsoft Visual Studio
- .NET Framework 4 and above
- JDK version 5.0 and above
- 512 MB RAM and above

Hardware Requirements

- Web Camera (30fps and above)
- Speaker
- Microphone
- A Computer with basic I/O devices

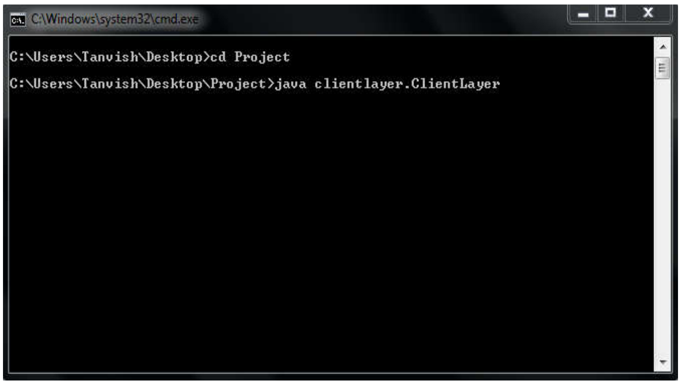
System Working

- Gesture Recognition Module:

The project's main source code is named ClientLayer and it is stored in a folder named clientlayer. So to extract the package from the folder and run it in java we use the above-

Corresponding author: Tanvish Thakker,
BE Student, Department of Information Technology,K.J. Somaiya Institute of Engineering and Information Technology, University of Mumbai, Maharashtra, India

depictedcommand. (Note: The java file is pre-compiled.) From figure 1 we can understand that in cmd we are just implementing the source code using JRE.



```
C:\Users\Tanvish\Desktop>cd Project

C:\Users\Tanvish\Desktop\Project>java clientlayer.ClientLayer
```

Fig. 1

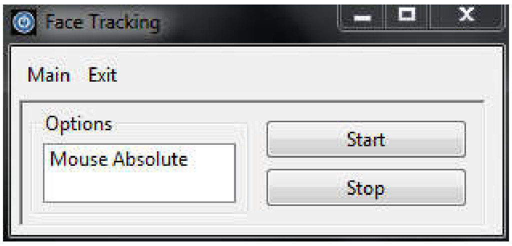
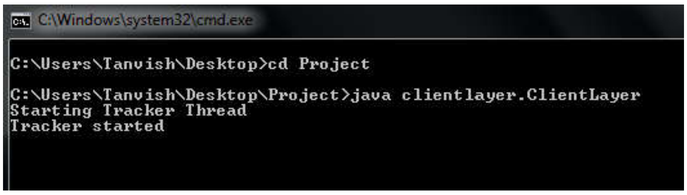


Fig. 2

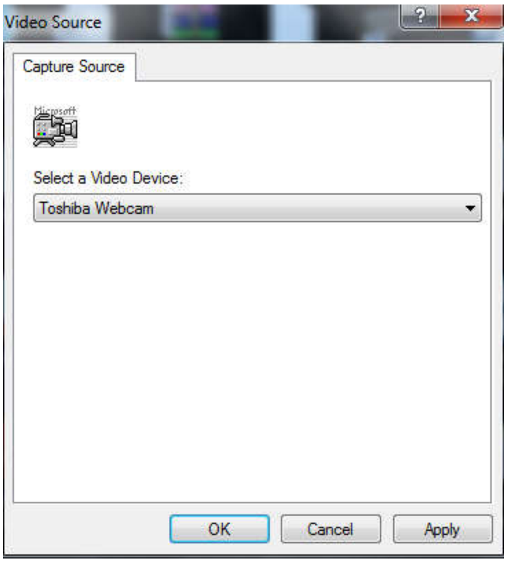


```
C:\Users\Tanvish\Desktop>cd Project

C:\Users\Tanvish\Desktop\Project>java clientlayer.ClientLayer
Starting Tracker Thread
Tracker started
```

Fig. 3

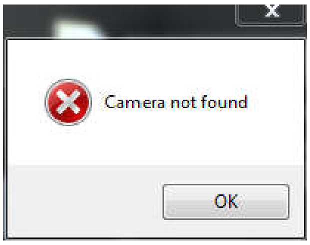
When we run the code using java, the thread from the code starts running. This thread runs the gesture UI i.e the front screen that is depicted in figure 2 and along with that, the thread running is depicted in figure 3 where the tracker for gesture recognition also starts.



Select a Video Device: Toshiba Webcam

Fig. 4

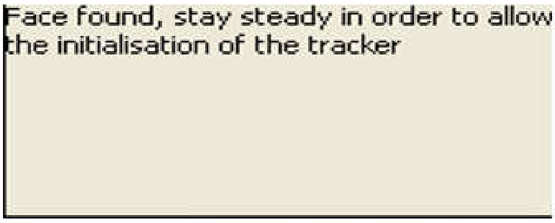
The tracker has one basic requirement i.e. the web camera and figure 4 is the initiation of the tracker which then enables the user to select a source of video input. All the source of video inputs will be shown in the drop-down menu. One should make sure that the input webcam device should have a minimum fps power of 30 or above because our eye blink is very fast and to process that one requires a good frame rate to capture each and every movement and give a precise mouse control action as anoutput.



Camera not found

Fig. 5

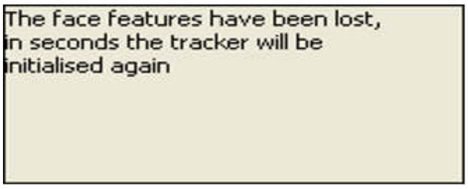
If no camera input device is found then it will show a box as depicted in figure 5. The only major requirement for gesture recognition is a web camera which cannot be avoided as it will not serve the purpose.



Face found, stay steady in order to allow the initialisation of the tracker

Fig. 6

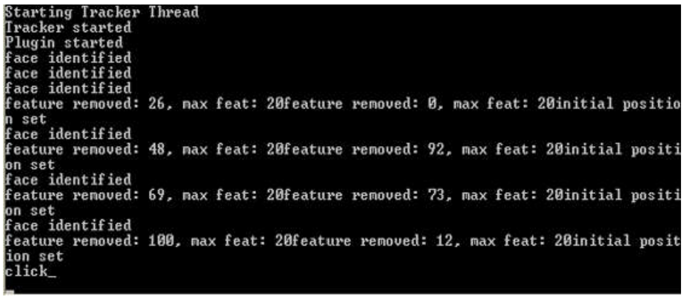
Once we select the input device and keep our face steady in front of the webcam we can see an information box which satisfies the user that one has fulfilled the system's requirement. The information box is depicted in the above figure 6.



The face features have been lost, in seconds the tracker will be initialised again

Fig. 7

If the user's face does not remain steady while the tracker is trying to capture the face features then it will update the information the box and will show a box just like Figure 7.



```
Starting Tracker Thread
Tracker started
Plugin started
face identified
face identified
face identified
feature removed: 26, max feat: 20feature removed: 0, max feat: 20initial positio
n set
face identified
feature removed: 48, max feat: 20feature removed: 92, max feat: 20initial positi
on set
face identified
feature removed: 69, max feat: 20feature removed: 73, max feat: 20initial positi
on set
face identified
feature removed: 100, max feat: 20feature removed: 12, max feat: 20initial posit
ion set
click_
```

Fig.8

When the tracker completes extraction of the users face features the control is passed to the mouse cursor and all the parameters are set. When an eye blink is detected it performs a function of a right click which is shown in figure 8.

Voice Control Module

When the .NET code for voice control is run using Microsoft Visual Studio we get a command control screen. The command control screen can be seen in figure 9.

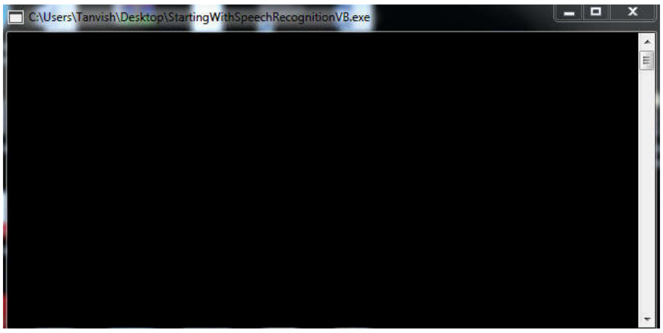
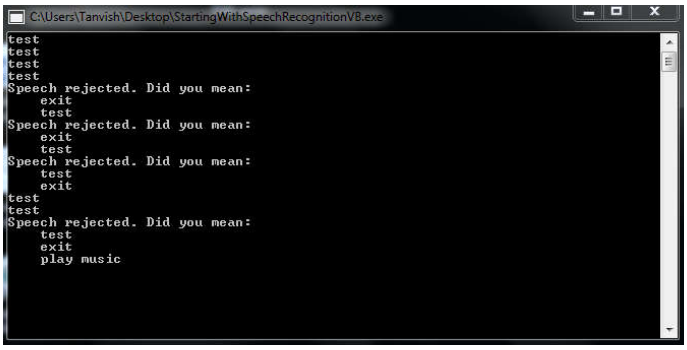

C:\Users\Tanvish\Desktop\StartingWithSpeechRecognitionVB.exe

Fig.9

The various pre-fed commands are depicted on the screen when it detects those phrases using the recognizer and performs that equivalent function when the synthesizer has done processing the command. When the command is not detected properly it shows a screen depicted in Figure 10 and will reject the speech i.e the voice input.


C:\Users\Tanvish\Desktop\StartingWithSpeechRecognitionVB.exe

```
test
test
test
test
Speech rejected. Did you mean:
    exit
    test
Speech rejected. Did you mean:
    exit
    test
Speech rejected. Did you mean:
    test
    exit
test
test
Speech rejected. Did you mean:
    test
    exit
    play music
```

Fig.10

Some of the pre-fed commands that are used in our project are:

Test → To check if the system is detecting the commands
Play music → To open Windows Media Player
Open C Drive → Opens C Drive in File Explorer
Open cmd → Opens Command Prompt
Exit → It exits the voice control module

But before directly exiting from the module it asks the user if he/she is sure to exit the system and one can answer back with a “Yes” or “No” and the system will act accordingly.

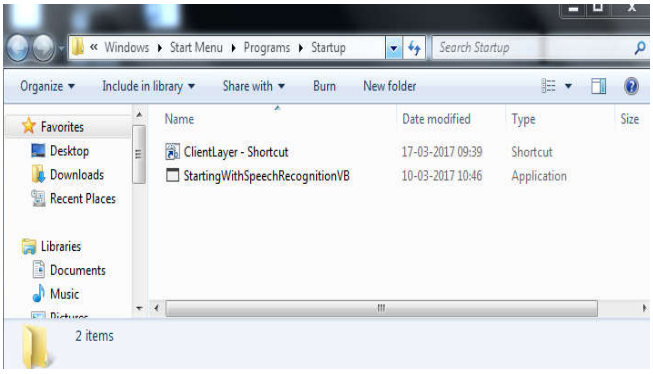
(Note: One can easily add new commands and provide a path for what process needs to be done by the system for that specific voice command.)

Advantage over Previous systems

Some of our system advantages over previously existing similar systems are:

Ease of use

This system is easy to use because the system will automatically start running as soon as the computer starts. This is done by including these programs in the shell startup menu of Windows as depicted in Figure 11.



Name	Date modified	Type	Size
ClientLayer - Shortcut	17-03-2017 09:39	Shortcut	
StartingWithSpeechRecognitionVB	10-03-2017 10:46	Application	

Fig.11

Fewer Hardware Requirements

The only hardware that is required to run the project is a basic computer, a webcam, and a microphone. It is very likely that these mechanisms are commonlyavailable in normal households nowadays due to technological advancements.

Low-Cost System

All the systems that exist till now probably depend on a lot of expensive hardware exclusively made for the system to run. This exclusive hardware is very expensive and this is a major drawback of all these systems when compared to our system’s optimal costing.

Conclusion

The project thus is a game changer in the field of image processing. There is a great future scope for this project and great extensions can be done. The gesture control can be used to design advanced games where the game can be completely controlled and played using eyes as cursor movements and voice to switch the functions within the game. This will be an ultimate hands-free experience, which will ease out the process and make it more interesting and fun to play. The voice dictation in voice control module can be extended so as to make additional use of it. This may also give an uplift for IOT (Internet of Things) and put voice into use for it. To conclude in short one can easily say that this project will bolster hands-free use of all devices in the very near future.

Acknowledgement

The project implementation would not have been possible without the constant support of our professors. We would like to especially thank Prof. Nasim Shah, our project guide at K.J. Somaiya Institute of Engineering and Information Technology, Sion, Mumbai for her constant support in the project. Her guidelines were the key to the implementation. Our special vote of thanks to our HOD (Head of Department) for IT, Prof. Uday Rote for motivating us at regular intervals to get our confidence levels back on. We would also like to thank our principal, Dr. Suresh Ukarande for that relieving support. All of this would not have been possible without the support and blessings of our parents whose constant push helped us reach this stage and showcase talent at such high platform.

REFERENCES

Alvarez, L. 2012. Progress in pattern recognition, image analysis, computer vision, and applications. 1st ed. Berlin: Springer.

Estellers, V., Gurban, M. and Thiran, J. 2012. On Dynamic Stream Weighting for Audio-Visual Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing, 20(4), pp.1145-1157.

Hussen Abdelaziz, A., Zeiler, S. and Kolossa, D. 2015. Learning Dynamic Stream Weights for Coupled-HMM-based Audio-visual Speech Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, pp.1-1.

International Conference on Image Processing, 2010. IEEE Transactions on Audio, Speech, and Language Processing, 18(1), pp.212-212.

Khachidze, V. 2012. Contemporary research on E-business technology and strategy. 1st ed. Berlin: Springer.

Kim, W., Suh, S., Hwang, W. and Han, J. (2014). SVD Face: Illumination-Invariant Face Representation. IEEE Signal Processing Letters, 21(11), pp.1336-1340.

Sarita, S. and Kumar Kaki, K. 2013. Mouse Cursor & Movements using Voice Controlled Mouse Pointer. *International Journal of Computer Applications*, 71(7), pp.27-34.

Yupapin, P., Hamedi, Salleh, Tan, Ismail, Ali, Dee-Uam, and Pavaganun, 2011. Human facial neural activities and gesture recognition for machine-interfacing applications. *International Journal of Nanomedicine,* pp. 3461.

****