

HCI APPLICATION FOR PLAYING COMPUTER GAMES

Varun Ignatius. A, Vasanthaguru. R, Vengataramanan. B, Vishnu Shankar. B
Dept.of Computer Science and Engineering Saranathan College of Engineering
Trichy

Abstract— *This paper describes a command interface for games based on hand gestures and voice command defined by postures, movement and location. The system uses computer vision requiring no sensors or markers on the user or background. In voice command the speech recognizer, recognize the input from the user. It stores and the command pass through the game, action takes placed.*

Keywords— *computer vision; gesture recognition; voice command; human computer interaction;*

1. INTRODUCTION

Computer games are one of the most successful application domains in the history of interactive systems even with conventional input systems like mouse and keyboard. The existing system, mouse for instance seriously limits the way humans interact with computers. Introduction of HCI techniques to the gaming world would revolutionize the way humans play games. It is a command interface for games based on **hand gestures and voice command** defined by postures, movement and location. The proposed system uses a simple webcam and a PC for recognizing the input from the user and thus uses natural hand movements for playing games. This effectively reduces the cost of implementing HCI in conventional PCs. We propose a simple architecture for performing real time colour detection and motion tracking using a webcam. Since many colours are detected it is important to distinguish the specified colours distinctly. The next step is to track the motion of the specified colours and the resulting actions are given as input commands to the system.

Speech technology is a very popular term right now. And Speech Recognition is the process of automatically recognizing a certain word spoken by a particular speaker based on individual information included in speech waves. Using speech recognition, we can give commands to computer and the computer will perform the given task. The main objective of this project is to construct and develop a system to execute commands of operating system by using speech recognition system that is capable of recognizing and responding to speech inputs rather than using traditional means of input (e.g. computer keyboard, mouse), thus saving time and effort of the user. The proposed system is easier to increase the interaction between people and computers by using speech recognition; especially for those who suffer from health problems, for example, the proposed system helps physically challenged persons. This application will help in reduction in hardware

requirement and can be implemented in other electronic devices also. In this case, we are using TIC-TAC-TOE GAME.

2. RELATED WORK

A few works have been proposed recently to use free hand gestures in games using computer vision. A multimodal multiplayer gaming system combines a small number of postures, their location on a table-based interaction system and speech commands to interact with games and discusses results of using this platform to interact with popular games. In this study, a colour pointer has been used for object recognition and tracking. Instead of conventional finger tips a colour pointer has been used to make object detection easy and fast. Other tools facilitate the use of gesture recognition for applications in general, not only games.

Speech technology is a very popular term right now. Speech recognition is highly demanded and has many useful applications. Conventional system users use pervasive devices such a mouse and keyboard to interact with the system. Further more people with physical challenges find conventional system hard to use. A good system has minimal restrictions in interacting with its user. The speed of typing and hand writing is usually one word per second, so speaking may be the fastest communication form with a computer. The applications with voice recognition can also be a very helpful to for handicapped people who have difficulties with typing.

3. SPECIFIC REQUIREMENTS

For Hand Gestures, we are Using the Open CV. In this study, a colour pointer has been used for object recognition and tracking. A blue and green colours are used as colour pointers. Colour detection is performed using in build functions in Open CV. Open CV (Open Source Computer Vision) is a Library of programming functions mainly aimed at real-time computer vision. Originally developed by intel's research centre. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. Open CV's applications areas include: Facial recognition system, Gesture recognition, Human-computer interaction (HCI), Mobile robotics. Open CV is an open source computer vision and machine learning software library. Open CV was build to provide common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-Licensed products, Open CV makes it

Hci Application For Playing Computer Games

easy for business to utilize and modify the code. It has C,C++,Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. Open CV means mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA AND Open CL interfaces are being actively developed right now. There are over 500 algorithms and above 10 times as many functions that compose or support those algorithms. Open CV is written natively in C++ and has a template interface that works seamlessly with STL containers.

For Voice Input, we are Using the Sphinx package. **CMU Sphinx**, also called Sphinx in short, is the general term to describe a group of speech recognition systems. **SPIHNX** the first large-vocabulary speaker-independent continuous-speech recognizer using multiple code books of various LPC-derived features. Two types of HMMs are used in **SPHINX** : context-independent phone model and function-word-dependent phone model. On a task using bigram grammar, **SPHINX** achieved a word accuracy. This demonstrates the feasibility of speaker-independent continuous-speech recognition, and the feasibility of appropriateness of hidden Markov models for such a task. These include a series of speech recognizers. The speech decoders come with acoustic models and sample applications. The available resources include in addition software for acoustic model training. These two specific requirements are used in the TIC-TAC-TOE GAME.

4.DESCRPTION HAND GESTURES

In the TicTacToe Game, the Open CV works :

The user use one color for the movement and another color for click options in their fingers. The user move the hands into the webcam and the images are captured by open CV tool. The captured images go to the segment analysis by BG Removal, and the segmented image is recognizes by the recognizer. The recognized movement is mapped to the mouse tracker, and if the another color will identified the action takes placed.

Both the use of gestures and having games as an application bring specific requirements to an interface and analyzing these requirements was one of the most important steps in designing Gestures. Gestures are most often used to relay singular commands or actions to the system, instead of tasks that may require continuous control, such as navigation. Therefore, it is recommended that gestures be part of a multimodal interface. This also brings other advantages, such as decoupling different tasks in different interaction modalities, which may reduce the user's cognitive load. So, while gestures have been used for other interaction tasks in the past, including navigation. All this leads to the requirement that the vocabulary of gestures in each context of the interface, while small, must be as simply and quickly modifiable as possible. Systems that require retraining for each set of

possible gestures, for instance, could prove problematic in this case, unless such training could be easily automated. The interface should also accept small variations for each gesture. Demanding that postures and movements be precise, while possibly making the recognition task easier, makes the interaction considerably harder to use and learn, demanding not only that the user remember the gestures and their meanings but also train how to do them precisely, greatly reducing usability. It could be argued that, for particular games, reducing the usability could actually be part of the challenge presented to the player (the challenge could be remembering a large number of gestures, or learning how to execute them precisely, for instance). While the discussion of whether that is a good game design practice or not is beyond the scope of this paper, Gestures opts for the more general goal of increasing usability as much as possible. This agrees with the principle that , for home and entertainment applications, ease of learning, reducing user errors, satisfaction and low cost are among the most important design goals. The system should also allow playing at home with minimal setup time required. Players prefer games where they can be introduced to the action as soon as possible, even while still learning the game and the interface. Therefore, the system should not require specific background or lighting conditions, complex calibration or repeated training. Allowing the use of the gesture-based interface with conventional games is also advantageous to the user, providing new options to enjoy a larger number of games. From the developer point of view, the system should be as easy as possible to integrate within a game, without requiring specific knowledge of areas such as computer vision or machine learning.

5. THE ABSTRACT FRAMEWORK

Figure 1 shows a UML Activity Diagram representing Gesture object flow model. *It* is responsible for the gesture model, while *Gesture Analysis* and *Gesture Recognition* define the interfaces for the classes that will implement gesture analysis and recognition. To these activities are added image capture and segmentation. *Gesture Capture* provides an interface for capturing 2D images from one or multiple cameras or prerecorded video streams (mostly for testing). The images must have the same size, but not necessarily the same color depth. A device could provide, for instance, one or more color images and a gray scale image to represent a dense depth map. *Gesture Segmentation* should usually find in the original image(s) one or both hands and possibly the head (to determine relative hand position).

Hci Application For Playing Computer Games

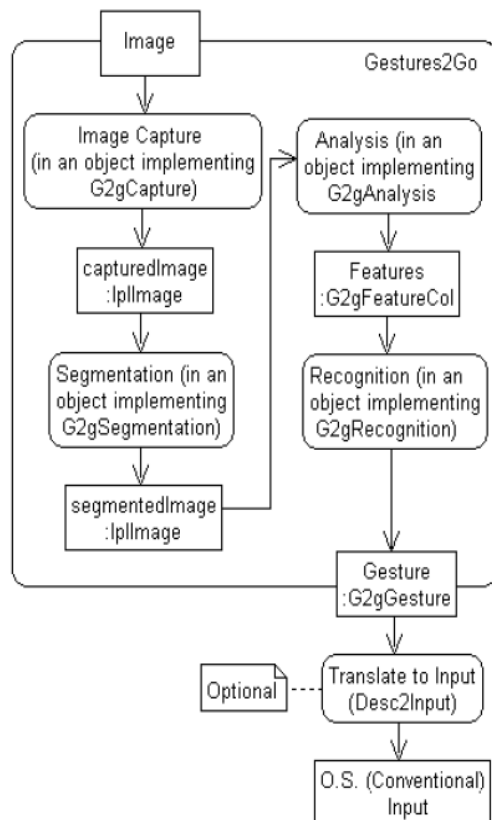
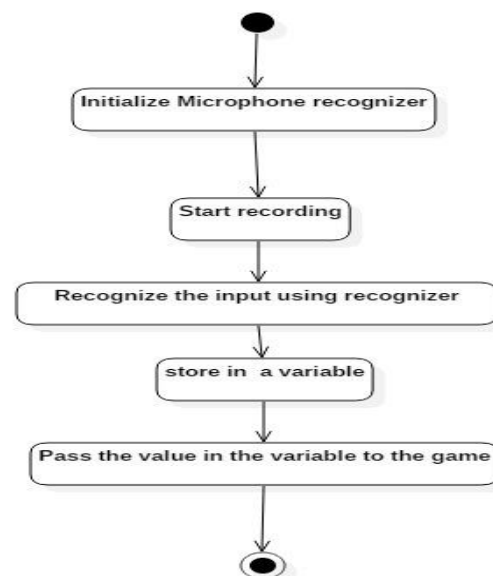


Figure 1 shows that the usual flow of information in Gestures in each time step is as follows: one or more images serve as input to the image capture model, which makes these images available as an OpenCV's Ipl Image object. The segmentation uses this image and provides a segmented image as an object of the same class (and same image size, but not necessarily color depth). Based on the segmented image, the analysis provides a collection of features as a *Gesture Feature Col* object which are in turn used by the recognition to output a gesture.

Gestre Feature Col is a collection of *Gesture Feature* objects. *Gesture Feature* contains a identifier string to describe the feature and either a scalar and an array of values (more often used) or an image (useful, for instance, for features in the frequency domain). *Gesture Feature* already defines several identifiers, for those features most often found in the gesture recognition literature, to facilitate the interface between analysis and recognition, but user-created identifiers may also be used. Input is an optional module that accompanies but is actually separate from Gestures2Go. It is responsible for facilitating, in a very simple way, both multimodal input and integration with games or engines not necessarily aware of Gestures. It simply translates its input, which is a description (a numerical or string ID or a XML description, for instance) that may be supplied either by Gestures2Go or any other system (and here lies the possibility of multimodal interaction), into another type of input, such as a system input (like a key down event) or input data to a particular game engine. In one of the tests, for instance, gestures are used for commands and a dancing mat is used for

navigation. Because this architecture consists mostly of interfaces, it is possible to create a single class that, through multiple inheritance, implements the entire system functionality. This is usually considered a bad practice in object orientation (should be avoided) and is actually one of the reasons why aggregation is preferred to inheritance. There are design patterns that could have been used to force the use of aggregation and avoid multiple inheritance, but Gestures opts for allowing it for a reason. Gesture recognition may be a very costly task in terms of processing, and must be done in real time for the purpose of interaction. Many algorithms may be better optimized for speed when performing more than one task (such as segmentation and analysis) together. Furthermore, analysis and recognition are very tightly coupled in some algorithms and forcing their separation could be difficult. So, while it is usually recommended to avoid using multiple inheritance and to implement each task in a different class, making it much easier to exchange one module for the other or to develop modules in parallel and in teams, the option to do otherwise exists, and for good reason.

6. SPEECH RECOGNITION



For Voice Input ,we are Using the Sphinx Library. Sphinx consists of in-build functions which are used to detect the speech input given by a particular user. In this case, we are giving the position number as the input for Tic Tac Toe Game through voice command. The Speech Recognizer, recognize the user input and store it, so that it can be used as a input in Tic Tac Toe game For Voice Input ,human voice which is sampled at rate of 16,000per second. It should be given in live mode .It should be noted that the system would have difficulty in recognizing accented English, hence it is recommended to give input as native English speakers would do. In Speech Recognizer, Platform speed directly affected our choice of a speech recognition system for our work.

Hci Application For Playing Computer Games

It is used to recognize the voice commands as spoken by user. The voice input is supposed to be given from its microphone. The speech recognition process decodes these input files, identifies the command and generates output accordingly. The Dictionary file is a separate file that describes the phonics of each word. It is a collection of pre-defined words that are relevant to various activities. Each word is separated into syllables. The output from the recognizer (i.e.) Recognition result is cross referenced with the dictionary file and correct word is returned as the result. The Decoder module then parses this word and converts into equivalent text. This text is then used by the command executor to execute the required functions. The decoder module is in build the SPHINX System.

The class Microphone from the sphinx for microphone control using java provides solve the essential functions for working with microphone connected to the system. The Start Recording function in the microphone class can be used to capture the audio from the microphone connected to the system. It returns an object of the result class which can be converted to text command using the recognize function in Recognizer class.

Speech Recognition is the process of identifying the speech in the recorded audio by using the phonetics dictionary. The phonetics for each word is stored in the dictionary file. The recognize function in the recognizer class uses the grammar file to recognize speech if any in the audio recorded by the microphone.

7. CONCLUSION

The System architecture proposed will completely revolutionize the way gaming applications are performed. The system makes use of web camera and which are an integral part of any standard system, eliminating the necessity of additional peripheral devices. Our endeavor for object detection and image processing in Open CV for the implementation of the gaming console provide to be practically successful. Here a person's motions traced and interpreted as commands. Most gaming application required additional hardware which is often very costly. The motive was to create this technology in the cheapest possible way under a standardized operating system using a set of wearable gesture able interfaces. This technique can be further extended to other services also. In Speech Recognition, the proposed system provides a better performance and experience in user interaction. It reduces potential time involved in conventional interactive methods. Voice recognition is irrefutably the feature of human computer interaction and our study proposes a new approach to utilize this approach for enhanced user experience and extend in ability of computer interaction to motor impaired users.

8. REFERENCES

- [1] D. Bowman, E. Kruijff, J. LaViola, I. Poupyrev, 3D User Interface: Theory and Practice, Addison-Wesley, 2005.
- [2] T. Stamer, B. Leibe, B. Singletary, J. Pair, "Mind-warping: towards creating a compelling collaborative augmented reality game", *Proc. of the 5th international conference on Intelligent user interfaces (IUI '00)*, pp. 256-259, 2000.
- [3] M. Snider, "Microsoft unveils hands-free gaming", *USA Today*, June 2009, [online] Available: http://www.usatoday.com/tech/gaming/2009-06-01-hands-free-microsoft_N.htm.
- [4] V. Dobnik, *Surgeons may err less by playing video games*, 2004, [online] Available: <http://www.msnbc.msn.com/id/4685909>.
- [5] S. LAAKSO, M. LAAKSO, "Design of a Body-Driven Multiplayer Game System", *ACM Comput. Entertain.*, vol. 4, no. 4, 2006.
- [6] A. Camurri, B. Mazzarino, G. Volpe, "Analysis of Expressive Gesture: The EyesWeb Expressive Gesture Processing Library" in *Gesture-Based Communication in Human-Computer Interaction*, Springer, pp. 469-470, 2004.
- B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, 1998.